

一种基于混沌优化的混合粒子群算法

邹 毅, 朱晓萍, 王秀平

(沈阳工程学院 电气工程系, 辽宁 沈阳 110136)

摘 要: 粒子群算法是一类基于群智能的优化搜索算法。该算法初期收敛很快, 但后期易陷入局部最优。为了提高粒子群算法的性能, 将粒子群算法全局搜索的快速性和混沌算法的一定范围内的遍历性二者结合, 提出一种基于混沌优化的混合粒子群算法。该算法首先用粒子群算法进行快速搜索, 当出现早熟收敛时, 对局部较优的部分粒子和全局极值采用混沌优化策略。对两个典型的测试函数进行仿真表明, 该算法能够摆脱局部极值, 得到全局最优。将其用于(N+M)系统费用模型求解, 得到最优解。同样验证了该算法搜索效率、精度优于一般的粒子群算法, 同时具有较好的收敛稳定性。

关键词: 粒子群算法; 混沌; 优化; 混合; (N+M)容错

中图分类号: TP306.1

文献标识码: A

文章编号: 1673-629X(2009)11-0018-05

A Hybrid PSO Algorithm Based on Chaos Optimization

ZOU Yi, ZHU Xiao-ping, WANG Xiu-ping

(Department of Electrical Engineering, Shenyang Institute of Engineering, Shenyang 110136, China)

Abstract: Particle Swarm Optimization(PSO) is a kind of optimizations based on swarm intelligence. The algorithm weakens quickly in initial stage, but falls into local extreme value easily in the latter. With PSO algorithm's rapid global searching and chaos' ergodicity in certain range, a hybrid PSO algorithm based on chaos is presented. The algorithm fast search with PSO algorithm first, then the chaos optimization is adopted for the better part of the particles and global extreme value when the optimization is in premature and convergence. The test of the two functions and solving the optimization of (N+M) fault-tolerant system show that search efficiency, accuracy of hybrid PSO algorithm are better than general PSO algorithm, while with better convergence stability.

Key words: PSO algorithm; chaos; optimization; hybrid (N+M) fault-tolerant

0 引言

粒子群优化(Particle Swarm Optimization, PSO)是一类基于群智能的优化搜索算法, 是由 Kennedy 和 Eberhart 通过对鸟群飞行行为研究, 于 1995 年提出的仿生进化算法^[1~3]。粒子群算法既简单又有深刻的智能背景。近年研究表明, PSO 在多维空间函数寻优、动态目标寻优等方面具有收敛速度快、解质量高、鲁棒性好等优点, 因此在函数优化、神经网络训练、模式分类、模糊系统控制等领域得到广泛的应用^[4~6]。

一般 PSO 算法具有全局的搜索能力, 搜索的初期收敛很快, 效率较高, 但后期容易陷入局部最优。混沌优化算法^[7,8]具有随机性、小范围内的遍历性等特点。将二者结合, 提出了一种基于混沌优化的混合粒子群算法(Chaos Hybrid PSO, CHPSO)。该算法首先

采用 PSO 算法进行快速搜索, 当出现早熟收敛时, 对局部较优的部分粒子和全局极值采用混沌优化策略。CHPSO 算法可避免单一搜索机制下的停滞现象, 加强搜索能力, 提高搜索效率。基于典型的高维函数的实验和对(N+M)系统费用模型求解结果验证了该算法的有效性和高效性。

1 基本粒子群优化算法

PSO 算法^[1~3]将种群中每个个体看成多维搜索空间中一个没有质量和体积的粒子, 这些粒子在搜索空间有位置和速度二重属性, 粒子根据自身的飞行经验和同伴的飞行经验对自己的速度进行动态调整, 即每个粒子通过自身的最优值和群体的最优值来不断调整自己的前进方向和速度大小, 这样粒子将根据其适应度不断飞向较优区域, 并最终找到问题的最优解。

设一般优化问题为:

$$\min f(x)$$
$$x = (x_1, x_2, \dots, x_n) \quad a_j \leq x_j \leq b_j, j = 1, 2, \dots,$$

收稿日期: 2009-03-12; 修回日期: 2009-05-29
基金项目: 辽宁省教育科研计划项目(2004F012)
作者简介: 邹毅(1971-), 男, 辽宁沈阳人, 副教授, 研究方向为优化算法及智能控制理论。

在 PSO 算法中, 用粒子的位置表示所求问题的解。设在 n 维的目标搜索空间中, 由 N 个粒子组成粒子群, 在第 k 次迭代时, 第 i 个粒子的位置表示为 $X_i(k) = (x_{i1}(k), x_{i2}(k), \dots, x_{in}(k))$, 飞行速度表示为 $V_i(k) = (v_{i1}(k), v_{i2}(k), \dots, v_{in}(k))$ 。粒子每一维的速度 V_i 都被一个最大速度 V_{\max} 所限制, V_{\max} 决定当前位置与最优位置之间区域的分辨率。由 PSO 算法执行时, 首先随机初始化 N 个粒子的位置和速度, 然后进行迭代寻优过程。在每一次迭代过程中, 粒子 i 通过跟踪两个极值来更新自己的位置和速度, 一个极值是整个粒子群体到目前为止找到的最优解, 称为全局极值, 表示为:

$$M_g(k) = (m_{g1}(k), m_{g2}(k), \dots, m_{gn}(k))$$

一个极值为粒子 i 目前搜索到的最优解, 称为个体极值, 表示为:

$$M_i(k) = (m_{i1}(k), m_{i2}(k), \dots, m_{in}(k))$$

粒子 i 在第 $(k+1)$ 次迭代时, 根据下列规则更新自己的速度和位置:

$$v_{ij}(k+1) = wv_{ij}(k) + c_1r_1(0,1)(m_{ij}(k) - x_{ij}(k)) + c_2r_2(0,1)(m_{gj}(k) - x_{ij}(k)) \tag{1}$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1) \tag{2}$$

式中: w 为惯性权重; c_1, c_2 为两个学习因子; $r_1(0,1)$ 和 $r_2(0,1)$ 是两个均匀分布在 $(0,1)$ 之间的随机数; $i = 1, 2, \dots, N$ 。

2 基于混沌优化的混合粒子群算法

分析上式可知, 如果在多次迭代过程中全局最优粒子位置 M_g 不发生变化, 则其速度更新主要取决于个体极值的变化, 于是速度将越来越小, 粒子群趋同, 所以粒子群算法从优化性能上收敛较快, 但易陷入局部极值。

混沌优化算法充分利用混沌特性, 即随机性、遍历性、规律性, 让变量在一定范围内遍历所有状态, 得到最优解。混沌优化算法在小空间搜索效果较好, 在粒子群算法中引入混沌算法, 不但能增强其对全局最优解的精细搜索, 而且可以使其更易跳出局部极值, 从而使算法的性能得到提高。

2.1 混沌优化算法

针对一般优化问题, 通常选取常用的混沌模型, 即一维的 Logistic 映射, 其模型为:

$$t(k+1) = ct(k)(1 - t(k)), k = 0, 1, 2, \dots, 0 \leq t(k) < 1 \tag{3}$$

当控制变量 $c \in (3.56, 4.0)$ 时, 式(3) 进入混沌状态, 具有混沌的一般特性, 当 $c = 4.0$ 时, 则处于完

全混沌状态。

混沌优化算法的步骤如下^[7,8]:

Step1: 对式(3) 分别赋予 n 个微小差别的初值, 得到 n 个轨迹不同的混沌变量 $t_j(k), k = 1, f^* = f(x^*), x^*$ 为当前最优解, f^* 为当前最优适应值。

Step2: 用混沌变量进行搜索。

$$x_j(k) = x_j^* + p_jt_j(k) - q_j \quad p_j, q_j > 0 \tag{4}$$

式中 p_j, q_j 为常数, 通过调整其值, 把混沌变量的取值范围变换到相应的优化变量的取值范围。

计算性能指标 $f(k) = f(x(k)), x(k) = (x_1(k), x_2(k), \dots, x_n(k))$

Step3: 如果 $f(k) < f^*$, 则 $x^* = x(k)$, 如果达到给定的迭代次数则结束; 否则转到 Step2。

2.2 基于混沌优化的粒子群混合算法 (CHPSO)

由于粒子群算法有全局收敛快的特点, 为了加快搜索过程, 首先执行粒子群算法, 当粒子群出现早熟收敛现象时, 说明优化陷入局部极值, 这时在指定的混沌迭代次数内对部分较优的粒子和全局极值进行混沌优化运算, 更新这些粒子位置后继续执行粒子群算法。粒子群算法和混沌算法交替进行, 当算法收敛条件满足后, 优化过程结束, 得到最优解。具体步骤如下:

Step1: 随机初始化粒子群的位置和速度, 其速度应在限制的最大速度 V_{\max} 之内。

Step2: 将个体极值设为当前位置值, 全局极值设为初始群体的最佳极值。

Step3: 判断算法收敛是否满足, 如满足则转向 Step7。

Step4: 执行粒子群算法。具体方法如下:

- 1) 按(1), (2) 式更新粒子位置和速度。
- 2) 如果粒子适应度优于个体极值的适应度, 则更新个体极值。
- 3) 如果有个体极值适应度优于全局极值的适应度, 则更新全局极值。

Step5: 判断粒子群是否为早熟收敛, 如果早熟收敛则执行混沌优化算法, 否则继续执行粒子群算法。早熟收敛主要有两个标志:

- ① 粒子群严重聚集;
- ② 最优粒子在多次迭代后无变化。

具体执行如下步骤:

1) 计算群体适应度方差 σ^2 , 预设最小群体适应度方差 σ_{\min}^2 。

$$\sigma^2 = \sum [(f_i - \bar{f}) / \bar{f}]^2 \tag{5}$$

其中 f_i 为第 i 个粒子的适应度, \bar{f} 为当前所有粒子的平均适应度, f 为归一化因子, 其取值要保证 $|f_i -$

$f)/f \leq 1$ 。

当 $\sigma^2 < \sigma_{\min}^2$ ，粒子群严重聚集，过早收敛，执行 Step6；否则执行下一步。

2) 计算全局最优粒子位置 M_g 连续不变化的迭代次数。当其达到预设的阈值时，说明粒子群进化缓慢，陷入停滞，执行 Step6；否则转到 Step3。

Step6: 对部分较优的粒子和全局极值按混沌优化算法进行优化运算。

对部分较优的粒子混沌优化，可以达到两个目的：一是这些粒子适应度较高，对其进行混沌搜索，较易得到新的最优粒子；二是混沌优化完成后，这些粒子离开原来位置，粒子群的多样性增加，这样有助于摆脱局部极小，增强全局搜优能力。由于全局极值附近出现最优解的概率较高，因此加强对全局极值的混沌搜索，且指定比前者更多的混沌迭代次数。当完成指定混沌迭代次数后，转到 Step3。

Step7: 输出结果，算法运行结束。

3 性能分析

为验证基于混沌优化的粒子群混合算法 (CHPSO) 的性能，选取 Rosenbrock 函数和 Rastrigin 函数进行测试，并与基本粒子群算法进行比较。

Rosenbrock 函数: $f(x) = \sum_{j=1}^n (100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2) - 100 < x_j < 100$ (6)

其全局极值点在 $x^* = (1, 1, \dots, 1)$ ，全局极值为 $f(x^*) = 0$ 。

Rastrigin 函数: $f(x) = \sum_{j=1}^n (x_j^2 - 10\cos(2\pi x_j) + 10) - 100 < x_j < 100$ (7)

其全局极值点在 $x^* = (0, 0, \dots, 0)$ ，全局极值为 $f(x^*) = 0$ 。

在测试中上述两个函数均取维数 $n = 10$ ；搜索空间 $x_j \in [-100, 100]$ ；设定初值范围 $x_j \in [40, 50]$ 。CHPSO 算法的粒子群规模 $N = 40$ ；惯性权重 w 较大有利于跳出局部最小点，而较小的 w 有利于算法收敛，这里惯性权重 $w = 0.7$ ；学习因子 $c_1 = c_2 = 2.0$ ；全局极值连续不变化的迭代次数阈值为 10；预设最小群体方差 $\sigma_{\min}^2 = 3$ 。在混沌优化环节中， $c = 4.0$ ；选取较优的个体粒子数为 10，其混沌迭代次数为 10；全局最优粒子的混沌迭代次数为 20。基本 PSO 的粒子群规模、惯性权重和学习因子与 CHPSO 相同。

首先测试给定精度下的进化代数和寻优时间。给定精度 $\epsilon = 1.0$ ，即 $|f^* - f_{\min}| \leq \epsilon, f^*$ 为算法输出的最优适应值， f_{\min} 为函数极值；CHPSO 算法进化代数为

粒子群迭代次数和混沌迭代次数之和。两种算法分别进行 100 次随机实验，表 1 给出了两种算法进化代数的平均值和平均时间。其次测试给定进化代数下两种算法达到的精度和最优解方差，给定的进化代数为 1000，精度和最优解方差取 100 次实验的平均值，结果见表 2。

表 1 和表 2 的数据表明：在给定的精度下，CHPSO 算法达到全局最优所需的进化代数和寻优时间要比基本 PSO 都少很多，效率更高；在给定的进化代数下，CHPSO 算法精度更高，最优解方差更小，收敛稳定性更好。

表 1 给定精度下各算法测试结果比较

函数名称	算法	平均进化代数	平均寻优时间(s)
Rosenbrock	PSO	17253	769.3
	CHPSO	2682	146.5
Rastrigin	PSO	9432	486.4
	CHPSO	1856	98.1

表 2 给定进化代数下各算法测试结果比较

函数名称	算法	平均精度	最优解方差
Rosenbrock	PSO	15.1173	51.2314
	CHPSO	2.1036	2.4231
Rastrigin	PSO	4.2948	2.3432
	CHPSO	1.7890	1.6728

为了直观反映两种算法的寻优性能，给出了两种函数的平均最优适应值随进化代数变化曲线，如图 1 和图 2 所示，其中平均最优适应值为 100 次实验每一代最优适应值的平均值。由图可见，基本 PSO 算法易陷入局部最优点，这导致寻优过程时间长，搜索最优值的效率较低；CHPSO 算法采取的混沌优化策略不但加强了对最优解的局部搜索，而且算法更容易跳出局部极值，增强了全局搜索能力，因此搜索效率、精度和收敛稳定性都有较大提高。

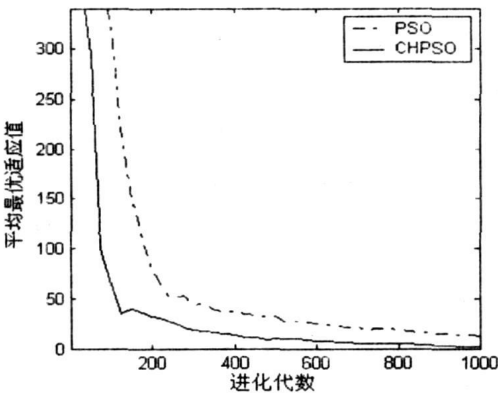


图 1 Rosenbrock 函数平均最优适应值曲线

4 (N+M)容错费用模型的优化求解

(N+M)容错系统的费用主要包括：购置控制器

的费用、运行费用、维护费用、由于控制器不可靠造成机组被迫停机而带来的直接经济损失和间接经济损失。 $(N+M)$ 系统优化的数学模型^[9-10]为:

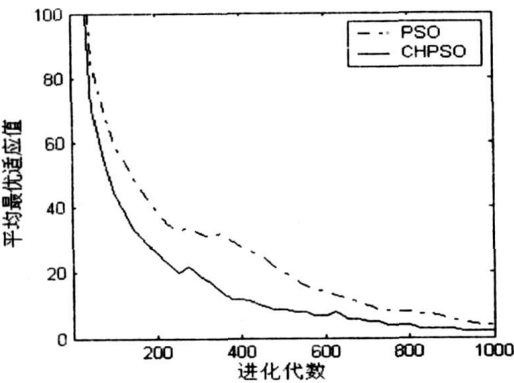


图 2 Rastrigin 函数平均最优适应值曲线

$$\min C = \frac{K_0(\lambda_0/\mu_0)^j}{(\lambda/\mu)^j}(n+m) + K_1 \sum_{i=1}^n iP_{m+i} \quad (8)$$

约束条件为 $A \geq GA$

$$P_{m+i} = \frac{C_{n+m}^{m+i}(\lambda/\mu)^{m+i}}{(1+\lambda/\mu)^{n+m}}$$

其中

$$A = \frac{\sum_{i=0}^m C_{n+m}^i(\lambda/\mu)^i}{(1+\lambda/\mu)^{n+m}}$$

其中 C 为 $(N+M)$ 容错系统的费用; K_0 为单台控制器 λ_0/μ_0 时对应的成本; K_1 表示单台机组停机所付出的代价系数; A 为系统可用度; GA (Guaranteed Availability) 为保证可用度。

上述费用模型是一个同时具有离散和连续变量的混合优化问题, 其中决策变量为 $n, m, \lambda/\mu$, n 和 m 是整型变量, λ/μ 为连续变量。当 n 为一定值, 上述问题即求解 m 和 λ/μ , 使 $(N+M)$ 容错系统的费用最小。

在该模型中, 取 $K_0 = 15$; $\lambda_0/\mu_0 = 0.01$, $j = 0.1$; $K_1 = 120000$; 保证可用度 $GA = 0.9995$, 取 $n = 1, 2, \dots, 26$ 。

CHPSO 算法中, 对整型变量 m , 算法在粒子群体的产生、迭代过程中采用 Matlab 中的 round 函数取整。由于该模型约束条件复杂, 不能转化为无约束极值问题。因此在确定粒子的过程中, 首先确定包含最优解的

区域, 在此最优解区域为 $0 \leq m \leq n, 0 \leq \lambda/\mu \leq 1$ 。在此区间产生随机粒子, 验证其是否满足约束条件, 即 $A \geq GA$ 。在迭代过程中, 每个粒子要验证是否满足约束条件, 如不满足, 则需调整, 直到满足条件为止。CHPSO 算法的预设最小群体方差 $\sigma_{\min}^2 = 0.1$, 计算 m 和 λ/μ , 结果见表 3。当 n 一定时, 得到最优的 m 和 λ/μ 组合。

从表 3 可以看出: 最佳备件量在随着控制器总台数 n 的增加而增加, 但备件 m 增加幅度较 N 少得多, 因此与 $2N$ 容错系统比较, $(N+M)$ 可靠性和经济性明显占优。在采用相同数量的备件的情况下, 随着 n 的增加 λ/μ 相应减小。这是因为在控制器台数增加的情况下, 只有减小 λ/μ 的值, 才能保证可用度要求。

为了与基本粒子群算法进行比较, 对 $n = 20$ 的 $(N+M)$ 容错优化模型进行求解, 对比结果如表 4 所示。从表中可以看出, 在 50 次迭代内, 混沌粒子群混合算法达到全局最优成功率比基本粒子群算法高出很多, 搜索成功的平均时间也较短。

表 4 各种算法对比实验结果比较 ($n = 20$)

算法	PSO	CHPSO
达到全局最优成功率(%)	49.5	99.5
(迭代 50 次)		
搜索成功的平均时间(s)	20.2	4.2

图 3 给出平均系统总费用与进化代数变化曲线, 平均系统总费用取 100 次实验的每一代的平均值。由图 3 可知, 在求解带有约束条件和混合变量的非线性规划问题上, CHPSO 算法的搜优效率和精度同样高于基本粒子群算法。

5 结束语

粒子群算法从生物群体的社会行为中得到启发, 是一种新颖的智能优化搜索算法, 其实现简单, 但易陷入局部最优, 有待于进一步改进和完善。文中提出基于混沌优化的混合粒子群优化算法(CHPSO), 即在搜索的后期对部分较优粒子和全局极值采用混沌优化策略, 以此有效地调节算法的全局搜索和局部搜索能力, 并改善种群的多样性。用测试函数进行了性能分析,

表 3 $(N+M)$ 容错系统优化模型粒子群算法结果

n	1	2	3	4	5	6	7	8	9
m	1	1	1	2	2	2	2	3	3
λ/μ	0.00371	0.00262	0.00220	0.01094	0.00962	0.00864	0.00784	0.01852	0.01713
n	10	11	12	13	14	15	16	17	18
m	3	3	4	4	4	4	4	5	5
λ/μ	0.01595	0.01495	0.02614	0.02467	0.02341	0.02227	0.02126	0.03207	0.03044
n	19	20	21	22	23	24	25	26	
m	5	5	5	6	6	6	6	6	
λ/μ	0.02925	0.02813	0.02718	0.03697	0.03578	0.03458	0.03353	0.03252	

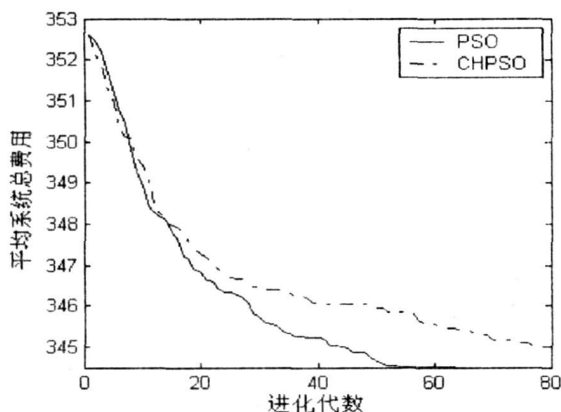


图 3 平均系统总费用与进化代数变化曲线

验证了该算法能够摆脱局部极值, 得到全局最优。将 CHPSO 算法用于 $(N+M)$ 系统费用模型求解, 得到最优解。

新算法搜索效率、精度等方面均优于基本粒子群算法, 同时具有较好的收敛稳定性。

参考文献:

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995: 1942—1948.
- [2] Eberhart R C, Kennedy J. A new optimizer using particle

swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE Press, 1995: 39—43.

- [3] Eberhart R C, Simpson P K, Dobbins R W. Computational intelligence PC tools[M]. Boston, MA: Academic Press Professional, 1996.
- [4] 邹毅, 朱晓萍, 霍龙. 一种改进粒子群算法及其应用[J]. 沈阳工程学院学报, 2006, 3(2): 283—286.
- [5] 张雪东, 赵传信, 季一木. 一种混合粒子群算法及其在 Job Shop 问题中的应用[J]. 计算机技术与发展, 2006, 16(9): 109—111.
- [6] 曾万里, 危韧勇, 陈红玲. 基于改进 PSO 算法的 BP 神经网络的应用研究[J]. 计算机技术与发展, 2008, 18(4): 49—51.
- [7] 雷德明. 利用混沌搜索全局最优解的混合遗传算法[J]. 系统工程与电子技术, 1999, 21(12): 81—83.
- [8] 张春慨, 王亚英. 混沌在实数编码遗传算法中的应用[J]. 上海交通大学学报, 2000, 12(12): 1658—1660.
- [9] Ye Luqing, Wang Shengtie. Control Maintenance Strategy for Fault Tolerant Mode and Reliability Analysis of Hydro Power Stations[J]. IEEE Transactions on Power Engineering, 2001, 16(3): 340—345.
- [10] Wang Shengtie, Ye Luqing, Malik O P. Intelligent Networked $(N+M)$ Fault Tolerant Systems for Hydro Power Stations[J]. Electric Power Systems Research, 2001, 8(59): 39—45.

(上接第 17 页)

操作, 避免遭受 DoS 攻击。

根据以上的分析不难看出, ESP 所提供的功能涵盖了 AH 的功能, 只需要对 ESP 的认证范围稍加修改, 使它在原有安全服务的基础上提供对整个 IP 数据包的认证, 就可以完全实现 AH 的功能, 从而取代 AH 协议。为此, 基于简化 IPSec 体系结构, 提高安全协议运行效率的思想, 笔者建议不妨考虑通过取消 AH 协议来实现对 IPSec 体系结构的简化。

5 结束语

AH 和 ESP 作为 IPSec 中的两个非常重要的安全协议, 已经在 IPv4 和 IPv6 的网络中得到了重要的应用, 但是从开始设计 IPSec 协议的那一天起, IPSec 体系结构就走向了日趋复杂的境地, 变得越来越庞大而不合理^[9]。目前, 人们对 IPSec 的研究, 不论是理论研究还是应用研究都还在不断地进行中, 特别是在 IPv6 环境中的研究还有许多工作要做。简化、改进 IPSec 现有的体系结构, 是 IPSec 研究的重点和难点, 也是基础性的工作。文中仅对 AH 和 ESP 两者的区别和联系进行了分析, 并提出了不妨取消 AH 协议的建议, 这是

一个值得进一步探讨的问题。

参考文献:

- [1] 关慧, 刘俊, 曹连刚. 基于 IPv6—IPSec 的网络安全访问的实现[J]. 微计算机信息, 2008, 24(2—3): 95—96.
- [2] IPv4 到 IPv6: 互联网的发展趋势[EB/OL]. 2007—03—02. <http://www.51cto.com/art/200703/41354.htm>.
- [3] 蓝集明, 张海燕. 对 IPv4/IPv6 过渡技术的分析与研究[J]. 电脑知识与技术, 2008, 4(7): 1885—1886.
- [4] Kent S. RFC 4302: IP Authentication Header[S]. 2005.
- [5] Kent S, Seo K. RFC 4301: Security Architecture for the Internet Protocol[S]. 2005.
- [6] Kent S. RFC 4303: IP Encapsulating Security Payload (ESP)[S]. 2005.
- [7] 李振强, 赵晓宇, 马严. IPv6 技术揭秘[M]. 北京: 人民邮电出版社, 2006: 225—280.
- [8] 江伟, 苏本跃, 周健. IPSec 在基于 IPv6 的校园网安全中的应用研究[J]. 计算机技术与发展, 2007, 17(2): 229—230.
- [9] 黄智, 龚向阳, 阙喜戎, 等. IPSec 协议的研究和分析[J]. 计算机工程与应用, 2002, 39(11): 160—162.