

用于多峰值函数优化的对数自适应排挤遗传算法

刘文涛^{1*}, 胡家宝²

(1. 武汉轻工大学 数学与计算机学院, 武汉 430023; 2. 武汉理工大学 计算机科学与技术学院, 武汉 430070)

(*通信作者电子邮箱 xkwk@qq.com)

摘要:排挤遗传算法能够比较稳定地获取多个峰值,但其求解效率不高,在有限的遗传代数下无法获得较高的求解精度,需要较多的迭代次数。为了快速求出多峰函数的所有最优解,提出了一种基于对数自适应的排挤遗传算法。该算法结合小生境排挤遗传和爬山算子,根据遗传代数对爬山算子的距离值进行对数自适应计算,使种群在遗传过程中保持多样性。通过对多个一维和二维多峰函数的实验和比较分析,测试结果表明,该算法在有限的遗传代数下既能保证求解精度又能提高收敛速度,能够比较稳定地求得所有最优解,是求解多峰函数问题的有效算法。

关键词:排挤遗传算法;多峰函数优化;爬山算子;对数自适应

中图分类号: TP301.6 **文献标志码:** A

Logarithmic adaption crowding genetic algorithm for multimodal function optimization

LIU Wentao^{1*}, HU Jiabao²

(1. School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan Hubei 430023, China;

2. College of Computer Science and Technology, Wuhan University of Technology, Wuhan Hubei 430070, China)

Abstract: Crowding genetic algorithm can obtain multiple optima of multimodal functions, but it has low efficiency, and cannot get a higher precision in limited iterations. In order to obtain all optima of the multimodal function quickly, the crowding genetic algorithm based on logarithmic adaption was presented combined with niche crowding genetic and climbing operators. The algorithm computed the distance values of climbing operators by logarithmic adaption according to the iterations, which made the population maintain genetic diversity in the process. According to the experiments and comparative analysis of several one-dimensional and two-dimensional multimodal functions, the test results show that the algorithm can ensure both the solution accuracy rate and the convergence speed in the limited iterations, and obtain all optimal solutions more stably. It is proved to be an effective algorithm for the multimodal function problems.

Key words: crowding genetic algorithm; multimodal function optimization; climbing operation; logarithmic adaption

0 引言

多峰值函数优化是很多实际问题的数学模型,例如最优控制设计、复杂系统和结构、模糊系统参数优化等,如何高效精确获取全部极值点具有重要的理论意义和实用价值。在对多峰值函数优化时,很多算法容易陷入局部最优,或者只能获取部分最优,求解效率和精度不高。常用的函数优化方法有模拟退火法^[1]、禁忌搜索^[2]、粒子群算法^[3]、人工鱼群算法^[4]、蚁群算法^[5]、遗传算法^[6]、免疫算法^[7]、人工蜂群算法^[8]等。其中遗传算法是求解函数优化的有效算法,但基本遗传算法只能用于求解单个极值点,无法获取所有极值点。于是提出了多种基于改进遗传算法的多峰值函数优化方法,其中小生境遗传算法具有代表性。小生境遗传算法利用生物特性,产生多个小生境,每个小生境都能够遗传进化,在多峰函数优化中一个峰值可以有一个小生境,从而能达到获取多个峰值的优化结果^[9]。小生境遗传算法^[10-11]包括共享机制、排挤算法、聚类算法等,其中排挤遗传算法是一种通用的求解多峰值函数优化的算法,其优点是能够比较稳定地获取多个峰值,但其缺点是求解效率不高,在有限的遗传代数下无

法获得较高的求解精度,需要较多的迭代次数。本文把爬山算子加入排挤遗传算法中,并引入一种对数自适应的距离值,对多个测试函数的实验中,通过对几个典型算法的比较,可以发现本文算法能够在保证精度的前提下,又能提高求解的效率,可有效应用于多峰函数的优化。

1 自适应排挤遗传算法

1.1 爬山算子过程

设置爬山过程中的距离 d ,对于基本的爬山算子, d 为固定值,可以设置为一个较小的浮点数。基本爬山算子的步骤如下:

Step1 设置 d 为爬山距离, Ind 为要操作的个体。

Step2 设置爬山的方向,对于一维函数只有两个方向表示左右,对于二维函数可以设置多个方向,例如8个,表示左右前后以及对角线方向。设置 $direc(d1, d2)$ 为 $\{(d, d), (-d, -d), (d, -d), (-d, d), (d, 0), (-d, 0), (0, d), (0, -d)\}$ 。

Step3 对于每个方向,分别计算新的坐标值, $(x1, x2) \leftarrow (Ind.x1 + one.d1, Ind.x2 + one.d2)$ 。

收稿日期:2013-12-16;修回日期:2014-02-21。

作者简介:刘文涛(1977-)男,湖北广水人,副教授,主要研究方向:智能计算、系统与网络安全;胡家宝(1950-)男,湖北武汉人,副教授,主要研究方向:计算机网络。

Step4 判断新的 $(x1, x2)$ 是否在坐标的定义域内,如果不是就计算下一个方向;如果在定义域内,就判断新的适应度 $fitness(x1, x2)$ 是否大于旧的适应度 $fitness(Ind)$ 。如果满足,就生成新的个体来替换旧的个体 $(Ind.x1, Ind.x2) \leftarrow (x1, x2)$,并计算出新个体染色体。

在基本的爬山算子中, d 为常量,但由于多峰函数的图形走势有的地方平缓,有的地方陡峭,所以如果用固定的值,无法适应不同的坡度。对于坡度陡峭的地方, d 不能太大,否则会跳过极值点而导致峰值被忽略;但如果为了精度,而把 d 设置得过小,则可能导致算法运行缓慢,效率低下。所以针对此情况,可以根据求解的过程对 d 进行自适应设置适当的值。在遗传代数的初期可以设置 d 为较大的值,这样保证算法的速度;在遗传代数的后期,设置 d 为较小的值,可以保证求解的精度。根据遗传代数对 d 进行自适应设置,可以定义线性自适应和对数自适应。对于线性自适应,其公式如下:

$$d = \min_d - (m - \text{gen}) / (m - 1) * (\min_d - \max_d) \quad (1)$$

其中; m 表示最大遗传代数, \min_d 表示最小的 d 值, \max_d 表示最大的 d 值。例如 \max_d 取值0.01, \min_d 取值为0.00001,中间遗传代数的 d 根据上述公式计算获得。开始和结束时 d 的取值不同,会影响算法的性能。线性自适应对于坡度变化比较均匀的多峰函数比较适合,但对于变化复杂的多峰函数可能无法取得较好效果。根据多峰函数的特点,一般峰值曲线都有一个渐变的过程,所以考虑使用对数自适应来计算 d 可以获得较好的效果。对数自适应公式如下:

$$d = \min_d - (x_{\max} - x) / (x_{\max} - 1) * (\min_d - \max_d) \quad (2)$$

其中:

$$x = \lg(\text{gen}) / \lg(g_value) + 1$$

$$x_{\max} = \lg(m) / \lg(g_value) + 1$$

g_value 为对数的底(例如2), gen 为当前的遗传代数,其他参数同上。例如,假设遗传代数为200, \max_d 取值0.01, \min_d 取值为0.00001,其变化曲线如图1所示。

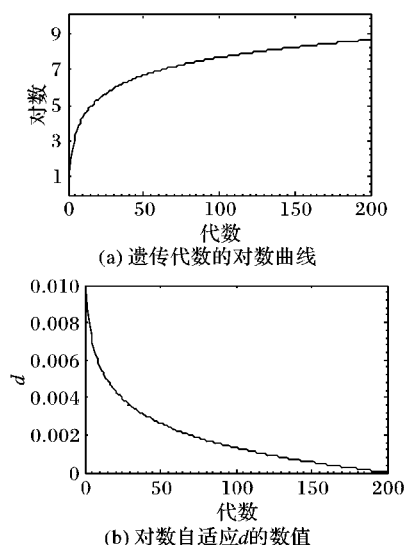


图1 遗传代数和对数自适应 d 的变化曲线

基于对数自适应,使 d 在刚开始变化较快,但越到后面则

d 的变化越缓慢,既保持了初期的计算速度,又保证了后期求解峰值的精度。因为排挤遗传算法越到后面遗传个体越向峰值靠拢,这时主要考虑精度,对于特别容易被忽略的峰值,需要较小的 d 来满足搜索精度。

1.2 算法步骤

Step1 产生初始种群,计算每个个体的适应度,种群个数为 n , n 为偶数。

Step2 定义遗传代数,进入遗传代数循环,遗传代数为 m 。

Step3 在种群中随机选择两个个体作为父代个体,记为 $(f1, f2)$,父代个体不能重复选择;令两个父代的适应度为 $fitness(f1)$ 、 $fitness(f2)$ 。

Step4 对两个父代个体进行交叉运算,产生两个子代个体,记为 $(s1, s2) \leftarrow crossover(f1, f2)$ 。

Step5 再对两个子代个体进行变异运算, $(s1, s2) \leftarrow mutation(s1, s2)$ 。

Step6 再对两个新的子代个体进行爬山算子计算, $(s1, s2) \leftarrow climb(s1, s2)$ 。

Step7 重新计算两个子代个体的适应度值,记为 $fitness(s1)$ 、 $fitness(s2)$ 。

Step8 设置两个父代个体和两个子代个体之间的欧氏距离,令 $d1 = distance(f1, s1)$; $d2 = distance(f1, s2)$; $d3 = distance(f2, s1)$; $d4 = distance(f2, s2)$ 。对父代和子代进行排挤操作,其过程如下:

Begin

if($d1 + d4 \leq d2 + d3$) {

if($fitness(s1) > fitness(f1)$) { $f1 \leftarrow s1$;

if($fitness(s2) > fitness(f2)$) { $f2 \leftarrow s2$;

else {

if($fitness(s2) > fitness(f1)$) { $f1 \leftarrow s2$;

if($fitness(s1) > fitness(f2)$) { $f2 \leftarrow s1$;

End

Step9 重复执行 Step3 ~ Step8 步骤 $n/2$ 次。

Step10 计算新种群的适应度,进入下一代遗传,直到满足退出条件终止。退出条件可以为遗传达到了最大遗传代数或者求解结果满足了求解精度。

1.3 算法度量

为了比较不同算法的计算结果,可以利用每代最优解比率(Global Optimum Ratio, GOR)和平均最优解比率(Average Optimum Ratio, AOR)来衡量。GOR是指算法的每次迭代过程中代表最优解的个体数目跟所有个体总数的比值,反映算法的求解质量,GOR的值为0~1,1表示最好的算法性能。AOR是用每代的最优解比率之和除以遗传代数,既能反映算法的质量,也能反映算法的求解效率,同样,AOR如果为1表示理想值,但一般不可能为1。

另外,多样性也是度量此算法的一个很好的指标^[12-13]。根据遗传算法的特性,生物种群进化离不开多样性,在多峰函数优化中,多样性尤为重要。它能保证算法搜索到多个不同的全局最优解,而不局限于一个,防止算法过早进入某个局部最优解。选择算子抑制多样性,交叉算子促进多样性,变异算子可提高多样性。但多样性过大,会阻碍算法效率,这就需要

设置合适的算子参数,以便保持多样性的前提下,又不能降低运算效率。

设群体大小为 n ,解空间映射为二进制编码,其长度为 L ,种群个体集合为 $P = \{p_1, p_2, \dots, p_n\}$,其中 p_j 个体定义为 $(p_{1j}, p_{2j}, \dots, p_{Lj})$, $j = 1, 2, \dots, n$,则多样性的定义^[14]如下:

$$m(P) = 1 - \frac{1}{L * n} \left[\sum_{i=1}^L \max \left(\sum_{j=1}^n p_{ij}, \sum_{j=1}^n (1 - p_{ij}) \right) - \sum_{i=1}^L \min \left(\sum_{j=1}^n p_{ij}, \sum_{j=1}^n (1 - p_{ij}) \right) \right] \quad (3)$$

可知 $m(P) \in [0, 1]$, 当为 0 时,种群多样性消失;当为 1 时,多样性最大。在算法测试中,用平均多样性 (Average Diversity, AD) 来描述整个遗传过程中种群的分布,用最后一代的种群多样性 (Population Diversity, PD) 来描述算法最后的染色体分布情况,以便了解个体是否分布在各个不同的峰值附近。

2 实验测试

2.1 函数选择

为了测试算法对于多峰函数峰值点布局不同的求解性能,采取四种不同高度和不同距离的一维函数,它们具有代表性。

1) F1, 等高等距函数:

$$f(x) = \sin^6(5\pi x); x \in [0, 1]$$

此函数有 5 个等值且分布均匀的全局极大值点。

2) F2, 等距不等高函数:

$$f(x) = \exp\left(-2\log(2) * \left(\frac{x - 0.1}{0.8}\right)^2\right) \sin^6(5\pi x); x \in [0, 1]$$

该函数有 1 个全局极大值点, 4 个局部极大值点, 这些极值点分布均匀。

3) F3, 等高不等距函数:

$$f(x) = \sin^6(5\pi(x^{3/4} - 0.05)); x \in [0, 1]$$

该函数有 5 个等值的极大值点, 但分布不均匀。

4) F4, 不等高不等距函数:

$$f(x) = \exp\left(-2\log(2) * \left(\frac{x - 0.08}{0.854}\right)^2\right) \sin^6(5\pi(x^{3/4} - 0.05)); x \in [0, 1]$$

该函数有 1 个全局极大值点, 4 个局部极大值点, 这些极值点分布不均匀。

二维函数也选择了两种典型的函数。

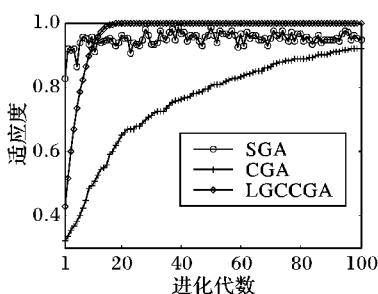


图2 函数 F1 的平均适应度变化曲线

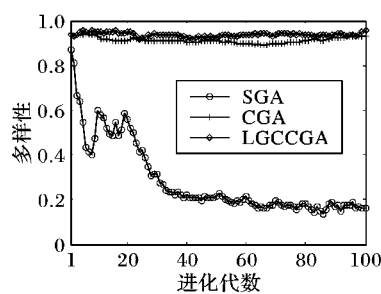


图3 函数 F1 的多样性变化曲线

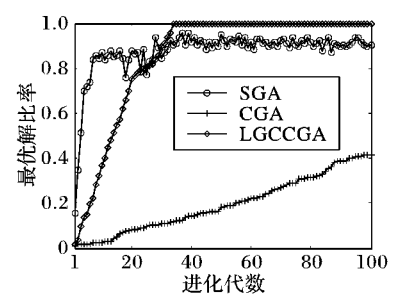


图4 函数 F1 的最优解比率变化曲线

5) F5, Branin 函数:

$$f(x, y) = \left(-1.275 \frac{x^2}{\pi} + 5 \frac{x}{\pi} + y - 6\right)^2 +$$

$$\left(10 - \frac{5}{4\pi}\right) \cos(x) + 10;$$

$$x \in [-5, 10], y \in [0, 15]$$

该函数有 3 个全局极小值, 分别在 $(-3.14159, 12.275)$, $(3.14159, 2.2750)$, $(9.42478, 2.475)$ 处有最小值 0.39788736。

6) F6, CrossInTray 函数:

$$f(x, y) = -0.0001(|\sin(x)\sin(y)| \cdot$$

$$\exp(|100 - \frac{\sqrt{x^2 + y^2}}{\pi}| + 1)^{0.1};$$

$$x \in [-10, 10], y \in [-10, 10]$$

该函数分别在 $(1.34941, -1.34941)$, $(1.34941, 1.34941)$, $(-1.34941, 1.34941)$, $(-1.34941, -1.34941)$ 处有 4 个全局极小值为 -2.06261 。

2.2 测试结果

为了在同一水平衡量算法的性能,一维函数的所有算法参数设置为种群大小 200, 遗传代数 100, 交叉率 0.6, 变异率 0.01, 选择算子使用赌轮盘方法, 交叉算子使用单点交叉方法, 变异使用均匀变异方法, 实验环境为 Matlab 和 VC++。其中最优解精度设置为 0.001, 分别用简单遗传算法 (Simple Genetic Algorithm, SGA)、排挤遗传算法 (Crowding Genetic Algorithm, CGA) 以及对数自适应爬山排挤遗传算法 (Logarithmic Climbing CGA, LGCCGA) 对函数 F1 至 F4 进行实验, 图 2~4 分别为函数 F1 的平均适应度分布曲线、多样性变化曲线和最优解比率曲线。

图 2 表示 SGA 适应度上下摆动, 无法把所有个体稳定聚集在一个峰值附近, 而 CGA 的适应度能稳步上升, 但在有限遗传代数内无法快速聚集, 而 LGCCGA 能够快速达到最优解, 而且在最优解能保持适应度稳定。图 3 说明 SGA 的种群慢慢聚集到一个最优解附近, 种群多样性渐渐丧失, 无法获取多个最优解, 而 CGA 和 LGCCGA 在整个的遗传代数中都能够保持较高多样性, 使其具有一次性获取多个最优解的能力。图 4 表明 SGA 最优解比率比 CGA 高, 但其所有个体都只能聚集在一个峰值附近, CGA 的最优解比率较低, 它需要更多的遗传代数才能获得较高最优解比率, 而 LGCCGA 能够快速达到最大的最优解比率, 在 40 代左右就把所有个体都聚集在多个峰值附近, 而且精度最优。

表1列出了对各个一维函数的求解结果。其中给出了算法种类 (Types of Algorithm, TA)、最终种群的多样性值 PD、整个遗传过程中的平均多样性值 AD、最终种群的最优解比率 GOR 和平均最优解比率 AOR, 以及求解的最优解个数 (Number of Optimal, NO); 另外, 为了测试爬山算子的距离 d 对算法的影响, 又给出两个 d 固定的爬山排挤算法 (Climbing CGA, CCGA), 其中 d 分别取值为 0.01 和 0.001; 还给出线性自适应爬山排挤遗传算法 (Linear Climbing CGA, LCCGA), 分别对 $F4$ 函数进行测试。

表1 一维函数的求解结果

$f(x)$	TA	PD	AD	GOR	AOR	NO
F1	SGA	0.163 182	0.186 982	0.905 000	0.834 800	1
	CGA	0.930 909	0.932 477	0.415 000	0.235 400	5
	LGCCGA	0.955 000	0.946 177	1.0	0.859 800	5
F2	SGA	0.104 545	0.193 423	0.935 000	0.842 600	1
	CGA	0.788 182	0.911 391	0.420 000	0.206 000	5
	LGCCGA	0.784 091	0.833 659	0.980 000	0.884 750	5
F3	SGA	0.208 636	0.224 641	0.905 000	0.874 300	1
	CGA	0.844 545	0.851 641	0.455 000	0.256 850	5
	LGCCGA	0.720 455	0.774 250	1.0	0.876 400	5
F4	SGA	0.235 455	0.079 868	0.860 000	0.822 200	1
	CGA	0.805 455	0.873 136	0.390 000	0.240 300	5
	CCGA ($d=0.001$)	0.740 455	0.844 405	0.995 000	0.724 150	5
	CCGA ($d=0.01$)	0.771 364	0.821 659	0.965 000	0.750 550	5
	LCCGA	0.742 727	0.797 873	0.990 000	0.807 750	5
	LGCCGA	0.729 545	0.806 682	0.990 000	0.873 550	5

表1表明, 对于所有函数, 无论是最终最优解比率还是平均最优解比率, LGCCGA 的最优解比率都比 CGA 要高, 表明在有限的遗传代数下 LGCCGA 能够获得更高的求解精度和速度; 而 CGA 和 LGCCGA 的多样性相差无几, 但都比 SGA 具有较高的多样性, 这表明它们都能够在遗传过程中保持较高的多样性而获取较多的峰值; 对函数 $F4$ 测试表明, CCGA、LCCGA 和 LGCCGA 三个算法都比 CGA 具有较高的最优解比率, 而且 LGCCGA 比 CCGA 和 LCCGA 要高, 表明基于对数自适应的爬山算子更好。

对于二维函数优化, 参数设置为种群大小 300, 遗传代数 200, 交叉率 0.7, 变异率 0.02, 最优解精度为 0.01, 其他参数设置与一维函数优化相同。二维函数的优化结果如表2所示。

表2 二维函数的求解结果

$f(x, y)$	TA	PD	AD	GOR	AOR	NO
F5	SGA	0.000 758	0.000 938	0.000 000	0.000 000	0
	CGA	0.703 030	0.803 076	0.380 000	0.124 650	3
	LGCCGA	0.773 788	0.853 861	0.523 333	0.252 417	3
F6	SGA	0.000 303	0.070 396	0.993 333	0.923 550	1
	CGA	0.479 697	0.650 833	0.910 000	0.701 683	4
	LGCCGA	0.733 636	0.809 601	0.943 333	0.738 650	4

从表2可看出, 由于二维函数的复杂性, SGA 可能无法收敛于最优解, 例如对于 $F5$, 其最优解比率为 0, 而 LGCCGA 算法的最优解比率都比 CGA 要高, 并且 CGA 和 LGCCGA 算法

的多样性都比 SGA 要高很多。这说明在进化的遗传代数中, 它们都能保持较高的种群多样性, 并能把个体聚集到多个最优解峰值附近。

3 结语

为了获取多峰函数中的多个最优解, 需要增加个体多样性, 保证遗传算法能够同时收敛于多个峰值, 利用排挤遗传算法可以满足要求, 但其收敛速度较慢。本文引入爬山算子, 提高了收敛速度, 并能同时获得较高的精度。对爬山算子距离参数研究, 通过距离固定、线性自适应和对数自适应的几种算法, 对多个函数优化结果可知, 对数自适应 LGCCGA 算法具有较快的寻优速度, 在有限的遗传代数下就能够获取较高的最优解比率, 并且能够在每次遗传过程中保持较大的多样性, 有效保证一次性获取多个峰值最优解。

参考文献:

- [1] WANG H, TANG G. An asynchronous parallel simulated annealing algorithm for function optimization problems[J]. Control and Decision, 2005, 20(5): 579-582. (王华, 唐国金. 函数优化问题的一种异步并行模拟退火算法[J]. 控制与决策, 2005, 20(5): 579-582.)
- [2] XU P, MIAO Q, LI W, et al. Adaptive simulated annealing algorithm and tabu search algorithm based on the function complexity[J]. Acta Electronica Sinica, 2012, 40(6): 1218-1222. (许鹏飞, 苗启广, 李伟生, 等. 基于函数复杂度的自适应模拟退火和禁忌搜索新算法[J]. 电子学报, 2012, 40(6): 1218-1222.)
- [3] CHEN D, ZHAO C. Ladder particle swarm optimization and application in function optimization[J]. Journal of System Simulation, 2007, 19(24): 5659-5662. (陈得宝, 赵春霞. 阶梯型粒子群算法及在函数优化中的应用[J]. 系统仿真学报, 2007, 19(24): 5659-5662.)
- [4] DENG T, YAO H, DU J. Improved artificial fish swarm mixed algorithm for multimodal function optimization[J]. Journal of Computer Applications, 2012, 32(10): 2904-2906. (邓涛, 姚宏, 杜军. 多峰函数优化的改进人工鱼群混合算法[J]. 计算机应用, 2012, 32(10): 2904-2906.)
- [5] HE Z, WANG X. Adaptive bee-ant colony optimization[J]. Application Research of Computers, 2012, 29(1): 130-134. (何宗耀, 王翔. 蜂群-蚁群自适应优化算法[J]. 计算机应用研究, 2012, 29(1): 130-134.)
- [6] WANG X, CAO L. Genetic algorithms — theory, application and software[M]. Xi'an: Xi'an Jiaotong University Press, 2002. (王小平, 曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002.)
- [7] YE H, LUO F, XU Y. Improve dimmune algorithm and its applications to function optimization[J]. Systems Engineering and Electronics, 2011, 33(2): 464-467. (叶洪涛, 罗飞, 许玉格. 改进的免疫算法及其在函数优化中的应用[J]. 系统工程与电子技术, 2011, 33(2): 464-467.)
- [8] BABAYIGIT B, OZDEMIR R. A modified artificial bee colony algorithm for numerical function optimization[C]// Proceedings of the 2012 IEEE Symposium on Computers and Communications. Piscataway: IEEE Press, 2012: 245-249.

(下转第 1675 页)

- scheme for mobile application based on H. 264[J]. The Journal of China Universities of Posts and Telecommunications, 2008, 15: 73 - 78.
- [4] WANG Y, DENG H. The video encryption scheme based in H. 264 on perceptual encryption algorithm standards [J]. Application of Electronic Technique, 2012(1): 56. (王亚民, 邓虎超. H. 264 标准中基于感知加密算法的视频加密方案[J]. 电子技术应用, 2012(1): 56.)
- [5] AU YEUNG S K, ZENG B. A new design of multiple transforms for perceptual video encryption [C]// Proceedings of the 2012 19th IEEE International Conference on Image Processing. Piscataway: IEEE Press, 2012: 2637 - 2640.
- [6] AU YEUNG S K, ZENG B. Improved perceptual video encryption using multiple 8×8 transforms in MPEG-4 [C]// Proceedings of the 2012 7th International ICST Conference on Communications and Networking in China. Piscataway: IEEE Press, 2012: 185 - 188.
- [7] TONG L, DAI F, ZHANG Y. Compressive sensing based video scrambling for privacy protection [C]// Proceedings of the 2011 IEEE Visual Communications and Image Processing. Piscataway: IEEE Press, 2011: 1 - 4.
- [8] FENG J. ZUC algorithm: 3GPP LTE international encryption standard [J]. China Information Security, 2011(12): 45 - 46. (冯秀涛. 3GPP LTE 国际加密标准 ZUC 算法[J]. 信息安全与通信保密, 2011(12): 45 - 46.)
- [9] GUAN J, DING L, LIU S. Guess and determine attack on SNOW3G and ZUC [J]. Journal of Software, 2013, 24(6): 1324 - 1333. (关杰, 丁林, 刘树凯. SNOW3G 与 ZUC 流密码的猜测决定攻击[J]. 软件学报, 2013, 24(6): 1324 - 1333.)
- [10] REN G, QIAO S, HEI Y. The application and implementation of ZUC stream cipher in the digital image encryption [J]. Science Technology and Engineering, 2013, 13(3): 766 - 770. (任高峰, 乔树山, 黑勇. 祖冲之算法在数字图像加密中的应用与实现[J]. 科学技术与工程, 2013, 13(3): 766 - 770.)
- [11] MA J, XU J, BAO Y, *et al.* Compressive sensing and its application: from sparse to low-rank regularized optimization [J]. Signal Processing, 2012, 28(5): 609 - 623. (马坚伟, 徐杰, 鲍跃全, 等. 压缩感知及其应用: 从稀疏约束到低秩约束优化[J]. 信号处理, 2012, 28(5): 609 - 623.)
- [12] ATHIRA V, GEORGE S N, DEEPTHI P P. A novel encryption method based on compressive sensing [C]// Proceedings of the 2013 International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing. Piscataway: IEEE Press, 2013: 271 - 276.
- [13] HUANG R, SAKURAI K. A robust and compression - combined digital image encryption method based on compressive sensing [C]// Proceedings of the 2011 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Washington, DC: IEEE Computer Society, 2011: 105 - 108.
- [14] BERINDE R, INDYK P. Sparse recovery using sparse random matrices [EB/OL]. [2013-10-10]. <http://people.csail.mit.edu/indyk/report.pdf>.
- [15] LIN X, LU G, YAN J, *et al.* Measurement matrix of compressive sensing based on Gram-Schmidt orthogonalization [C]// Proceedings of the 2011 6th International Conference on Image and Graphics. Piscataway: IEEE Press, 2011: 205 - 210.
- [16] FANG H, ZHANG Q, WEI H. Image reconstruction based on improved backward optimized orthogonal matching pursuit algorithm [J]. Journal of South China University of Technology: Natural Science, 2008, 36(8): 23 - 27. (方红, 章权兵, 韦穗. 改进的后退型最优正交匹配追踪图像重建方法[J]. 华南理工大学学报: 自然科学版, 2008, 36(8): 23 - 27.)
- [17] LIU F, KOENIG H. A survey of video encryption algorithms [J]. Computers and Security, 2010, 29(1): 3 - 15.
- [18] YAO Y, XU Z. Visual security assessment for video encryption based on structural distortion [J]. Computer Engineering, 2008, 34(23): 23 - 25. (姚晔, 徐正全. 基于结构失真度的视频加密视觉保密性评价[J]. 计算机工程, 2008, 34(23): 23 - 25.)
- [19] WANG Z, LU L, BOVIK A C. Video quality assessment based on structural distortion measurement [J]. Signal Processing: Image Communication, 2004, 19(2): 121 - 132.
- [20] CHEN J, LIU J, CAO X. In-depth understanding of video codec technology: H. 264 standard and reference model [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2012. (陈靖, 刘京, 曹喜信. 深入理解视频编解码技术: 基于 H. 264 标准及参考模型 [M]. 北京: 北京航空航天大学出版社, 2012.)
- [21] CHEN G, LIAO X. Image encryption based on discrete chaotic maps [J]. Journal of Computer Applications, 2005, 25(S1): 121 - 123. (陈果, 廖晓峰. 一种基于混沌映射的图像加密算法[J]. 计算机应用, 2005, 25(S1): 121 - 123.)
- [22] RACHLIN Y, BARON D. The secrecy of compressed sensing measurements [C]// Proceedings of the 2008 46th Annual Allerton Conference on Communication, Control, and Computing. Piscataway: IEEE Press, 2008: 813 - 817.

(上接第 1648 页)

- [9] ZHOU C, XIE A. Dynamic niche-based self-organizing learning algorithm [J]. Journal of Software, 2011, 22(8): 1738 - 1748. (周传华, 谢安世. 一种基于动态小生境的自组织学习算法[J]. 软件学报, 2011, 22(8): 1738 - 1748.)
- [10] QIAO P, ZHENG L, MA L. Research on a niche genetic algorithm [J]. Journal of Harbin University of Science and Technology, 2011, 16(1): 90 - 93. (乔佩利, 郑林, 马丽丽. 一种小生境遗传算法研究[J]. 哈尔滨理工大学学报, 2011, 16(1): 90 - 93.)
- [11] LU Q, XIE P, SUN B. Sharing scheme-based adaptive hybrid genetic algorithm [J]. Computer Simulation, 2012, 29(12): 274 - 278. (陆青, 谢品杰, 孙波. 基于共享机制的自适应混合遗传算法[J]. 计算机仿真, 2012, 29(12): 274 - 278.)
- [12] ZHANG X, DAI G, XU N. Study on diversity of population in genetic algorithms [J]. Control Theory and Applications, 1998, 15(1): 17 - 23. (张晓绩, 戴冠中, 徐乃平. 遗传算法种群多样性的分析研究[J]. 控制理论与应用, 1998, 15(1): 17 - 23.)
- [13] XING X, YAN J. A novel genetic algorithm based on diversity maintaining and its simulation [J]. Computer Simulation, 2010, 27(4): 206 - 209. (邢小军, 闫建国. 一种基于多样性保持的遗传算法及其仿真[J]. 计算机仿真, 2010, 27(4): 206 - 209.)
- [14] LI M. The basic theory and application of genetic algorithms [M]. Beijing: Science Press, 2002: 164 - 165. (李敏强. 遗传算法的基本理论与应用 [M]. 北京: 科学出版社, 2002: 164 - 165.)