优良布尔函数的混合禁忌搜索算法

王维琼，许豪杰，崔萌，谢琼
（长安大学理学院，陕西 西安 710064）

摘 要：为保障对称密码算法的安全性，其构成算法中所使用的布尔函数必须具有优良的密码学性质。结合禁忌搜索算法和爬山算法的优点，提出了一种新的优良布尔函数启发式生成算法——混合禁忌搜索算法。应用该算法，可以快速得到大量具有高非线性度、低自相关性、一阶弹性、最优代数次数、最优代数免疫度、最优（次优）抵抗快速代数攻击能力等的布尔函数。仿真结果表明，所提算法搜索能力强，运行速度快，且搜索出的布尔函数的密码学性质优于已知的优化算法的结果，也弥补了采用构造法构造布尔函数的一些缺陷。

关键词：布尔函数；禁忌搜索算法；弹性；非线性度
中图分类号：TP301.6
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022096

Hybrid tabu search algorithm for excellent Boolean function

WANG Weiqiong, XU Haojie, CUI Meng, XIE Qiong
School of Sciences, Chang’an University, Xi’an 710064, China

Abstract: Boolean function in symmetric cryptographic algorithm must satisfy excellent cryptographic criteria to ensure the security of the algorithm. By combining the advantages of tabu search algorithm and hill climbing algorithm, a new heuristic generation algorithm called hybrid tabu search algorithm for excellent Boolean function was proposed. A large number of Boolean function with high nonlinearity, low autocorrelation, one-resilient, optimal algebraic degree, optimal algebraic immunity, optimal (suboptimal) resistance to fast algebraic attacks could be obtained quickly by applying the proposed algorithm. Simulation results demonstrate that the cryptographic properties of the Boolean function obtained by the proposed algorithm with strong search ability and fast running speed are better than the results of known optimization algorithm. Moreover, the algorithm also provides good Boolean function that cannot be obtained by using construction method.

Keywords: Boolean function, tabu search algorithm, resiliency, nonlinearity

0 引言

对称加密算法为保障数据存储安全和通信网络信息传输安全提供了有力的理论基础和技术支撑。布尔函数是对称加密算法的核心部件，其密码学性质的好坏直接决定了对称密码算法的安全性。为了抵抗各种攻击，一个安全的密码系统中所使用的布尔函数需满足很多性质，例如平衡性、高非线性度、低自相关性、高代数次数、高相关免疫阶、高代数免疫度以及高抵抗快速代数攻击能力等。但这些密码学性质之间存在着复杂的相互制约关系，因此，如何找到各方面密码学性质都较为优良的布尔函数是密码学领域的一个研究重点和难点。

目前，解决该问题的方法主要有两类，一类是

收稿日期：2022−02−09；**修回日期**：2022−04−02
基金项目：国家自然科学基金资助项目（No.11901049）；陕西省自然科学基础研究计划基金资助项目（No.2020JQ-343）；陕西省高校科协青年人才托举计划基金资助项目（No.20200505）
Foundation Items: The National Natural Science Foundation of China (No.11901049), The Natural Science Basic Research Program of Shaanxi (No.2020JQ-343), Young Talent Fund of University Association for Science and Technology in Shaanxi (No.20200505)

借助代数和组合的理论进行构造，另一类是利用启发式算法进行搜索。学者们在布尔函数构造方面得到了许多结果[1-5]。Carlet 等[6]构造了一类非线性度下界为 $2^{n-1}-\frac{n2^{\frac{n}{2}}(2\ln 2)}{\pi}$ 且满足最优代数免疫度和高抵抗快速代数攻击能力的 n 元平衡布尔函数，遗憾的是该类函数不具有一阶弹性。Tu 等[7]基于 PS 类函数构造了一类具有最优代数免疫度、最优代数次数和高非线性度的平衡布尔函数，又称 Tu-Deng 函数，而后 Tu 等[8]对 Tu-Deng 函数进行了修改，得到了一类非线性度下界为 $2^{n-1}-2^{k-1}-3k2^{\frac{k}{2}}\ln 2-7(k=\frac{n}{2})$ 且满足一阶弹性、最优代数免疫度和最优代数次数的布尔函数，但该类函数未考虑抵抗快速代数攻击能力这一指标。Zhang 等[9]通过对 PS⁻类 bent 函数进行修改，构造了一类变元个数为偶数、非线性度为 $2^{n-1}-2^{\frac{n}{2}}-2^{\left\lceil\frac{n}{4}\right\rceil}$ 且满足一阶弹性的布尔函数，该结果是目前已知的偶变元一阶弹性函数非线性度的最优结果，遗憾的是该构造方法仅适用于偶数变元的布尔函数。已知的构造方法普遍存在的问题是难以在构造时兼顾所有的密码学指标，一般是针对少数几个指标来构造函数，然后通过对某个特例进行修改来满足其他性质。结合一些理论结果的启发式搜索算法可以弥补构造法的这一缺陷，得到代数构造法无法得到的布尔函数，例如 Saber 等[10]利用启发式搜索算法得到了一些非线性度为 240、弹性阶为 3、代数次数为 5 的 9 元布尔函数；Liu 等[11]利用模拟退火算法得到了一些非线性度为 488、弹性阶为 2、代数次数为 7 的 10 元布尔函数。遗憾的是，文献[10-11]中搜索到的布尔函数虽然足够优良，但是只针对 9 元或 10 元的布尔函数，不具有普适性。Yang 等[12]基于模拟退火算法和爬山算法对 8~14 元的布尔函数进行搜索，贾少帅等[13]基于引力搜索算法对 8 元和 9 元的布尔函数进行搜索，都搜索到了具有一阶弹性、较高非线性度、最优代数次数、最优（次优）代数免疫度以及次优抵抗快速代数攻击能力的布尔函数。利用启发式算法研究布尔函数的成果还有很多[14-18]，在此不再赘述。

本文期望在保证高非线性度和一阶弹性这 2 个最重要的密码学指标的前提下，获得其他密码学性质也足够优良的布尔函数。以高非线性度、低自相关性、一阶弹性为优化目标，以最优代数次数、最优代数免疫度、最优（次优）抵抗快速代数攻击能力为约束条件，本文设计了新的代价函数，并在禁忌搜索（TS, tabu search）算法和爬山（HC, hill climbing）算法的基础上进行改进，提出了一种混合禁忌搜索（HTS, hybrid tabu search）算法。应用该算法对 8~14 元的布尔函数进行搜索，得到了若干满足上述所有密码学指标的布尔函数。以随机生成的平衡布尔函数作为初始输入，除了 10 元布尔函数需要尝试多个输入之外，本文算法可以搜索到满足上述密码学性质的全部布尔函数，弥补了一些构造方法受限于变元个数的奇偶性，且构造出的函数只能满足部分密码学性质等缺陷。相较于已有的启发式搜索算法，本文算法搜索到的布尔函数的密码学性质更为优良，且数量更多。

1 预备知识

定义 1 设 $\mathbb{F}_2$ 为二元域 $\{0,1\}$，$\mathbb{F}_2^n$ 为 $\mathbb{F}_2$ 上的 n 维向量空间，n 元布尔函数就是从 $\mathbb{F}_2^n$ 到 $\mathbb{F}_2$ 上的一个映射。

记 B_n 为所有 n 元布尔函数的集合，对任意的 $f\in B_n$，可以用多种方法来表示，例如代数正规型、真值表等。f 的代数正规型为

$$f(x_1,x_2,\cdots,x_n)=\sum_{b\in\mathbb{F}_2^n}\lambda_b\left(\prod_{i=1}^{n}x_i^{b_i}\right) \tag{1}$$

其中，$\lambda_b\in\mathbb{F}_2$，$b=(b_1,b_2,\cdots,b_n)\in\mathbb{F}_2^n$。$f$ 的代数次数定义为

$$\deg(f)=\max\left\{\sum_{i=1}^{n}b_i\mid\lambda_b\neq 0\right\} \tag{2}$$

当 $\deg(f)=1$ 时，称 f 为仿射函数；当 $\deg(f)=1$ 且 f 的常数项为 0 时，称 f 为线性函数；当 $\deg(f)>1$ 时，称 f 为非线性函数。

f 的真值表为

$$\boldsymbol{f}=(f(\boldsymbol{X}_0),f(\boldsymbol{X}_1),\cdots,f(\boldsymbol{X}_{2^n-1})) \tag{3}$$

其中，$\boldsymbol{X}_0=(0,0,\cdots,0)$，$\boldsymbol{X}_1=(0,0,\cdots,1)$，…，$\boldsymbol{X}_{2^n-1}=(1,1,\cdots,1)$。$f$ 的真值表是一个长度为 2^n 的 $\{0,1\}$ 向量，因此集合 B_n 的基数为 2^{2^n}，这使当 $n\geqslant 8$ 时，穷举法搜索布尔函数几乎不可能实现。f 的汉明重量 $\mathrm{wt}(f)$ 为 f 的真值表中非零分量的个数。若 f 的真值表中 0 和 1 的个数相等，则称 f 为平衡布尔函数，此时 f 的汉明重量 $\mathrm{wt}(f)=2^{n-1}$。

定义 2 对于任意的 n 元布尔函数 f，其在 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$ 处的 Walsh 变换定义为

$$W_f(\boldsymbol{\alpha})=\sum_{\boldsymbol{x}\in\mathbb{F}_2^n}(-1)^{f(\boldsymbol{x})+\boldsymbol{x}\boldsymbol{\alpha}} \tag{4}$$

容易得出，平衡布尔函数在 $\mathbf{0}\in\mathbb{F}_2^n$ 处的 Walsh 谱值为 0。

定义 3　对于任意的 n 元布尔函数 f，其非线性度定义为

$$N_f=\min_{l\in A_n}d_{\mathrm{H}}(f,l) \tag{5}$$

其中，A_n 表示所有 n 元仿射函数的集合，$d_{\mathrm{H}}(f,l)=\mathrm{wt}(f\oplus l)$ 表示 f 和 l 之间的汉明距离。由非线性度 N_f 与 Walsh 变换的定义不难得出

$$N_f=2^{n-1}-\frac{1}{2}\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right| \tag{6}$$

由 Parseval 恒等式[19]

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f^2(\boldsymbol{\alpha})\right|=2^{2n} \tag{7}$$

可知

$$N_f\leqslant 2^{n-1}-2^{\frac{n}{2}-1} \tag{8}$$

当 n 为偶数时，存在 $f\in B_n$ 的非线性度可达到这一上界，即对任意的 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$，都有 $\left|W_f(\boldsymbol{\alpha})\right|=2^{\frac{n}{2}}$，称这样的布尔函数为 Bent 函数。

定理 1[20]　Xiao-Massey 定理。一个 n 元布尔函数 f 是 m 阶相关免疫的，当且仅当对任意的 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$，$1\leqslant\mathrm{wt}(\boldsymbol{\alpha})\leqslant m$，有 $W_f(\boldsymbol{\alpha})=0$。

若 f 满足 m 阶相关免疫的同时还是平衡的，则称 f 为 m 阶弹性布尔函数。

定理 2[21]　对任意的 m 阶弹性函数 $f\in B_n$，其代数次数 $\deg(f)$ 与弹性阶 m 之间满足如下关系

$$\deg(f)\leqslant n-m-1 \tag{9}$$

定义 4　对于任意的 n 元布尔函数 f，其在 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$ 处的自相关函数定义为

$$C_f(\boldsymbol{\alpha})=\sum_{\boldsymbol{x}\in\mathbb{F}_2^n}(-1)^{f(\boldsymbol{x})+f(\boldsymbol{x}+\boldsymbol{\alpha})} \tag{10}$$

度量布尔函数 $f\in B_n$ 自相关性的常用指标为绝对值与平方和[22]，计算式分别为

$$\Delta_f=\max_{\boldsymbol{\alpha}\neq 0,\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|C_f(\boldsymbol{\alpha})\right| \tag{11}$$

$$\sigma_f=\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}C_f^2(\boldsymbol{\alpha}) \tag{12}$$

不难证明，自相关平方和指标与 Walsh 变换之间有如下关系[23]

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}W_f^4(\boldsymbol{\alpha})=2^n\sigma_f \tag{13}$$

定义 5　设 $f,h\in B_n$，若 $fh=0$，则称 h 为 f 的零化子[24]。f 的所有零化子的集合为

$$\mathrm{AN}(f)=\{h\in B_n\mid fh=0\} \tag{14}$$

定义 6　设 $f\in B_n$，其代数免疫度定义为[25]

$$\mathrm{AI}(f)=\min\{\deg(h)\mid h\in\mathrm{AN}(f)\bigcup\mathrm{AN}(f\oplus 1),h\neq 0,h\in B_n\} \tag{15}$$

对任意的 $f\in B_n$，都有 $\mathrm{AI}(f)\leqslant\left\lceil\frac{n}{2}\right\rceil$，称达到这一上界的函数 f 具有最优代数免疫度。当 $\mathrm{AI}(f)=\left\lceil\frac{n}{2}\right\rceil-1$ 时，称 f 的代数免疫度是次优的。

定义 7　设 $f\in B_n$，f 抵抗快速代数攻击能力定义为[26]

$$\mathrm{FAA}(f)=\min\left\{\deg(g)+\deg(h)\mid fg=h,\deg(g)<\frac{n}{2},g,h\neq 0\in B_n\right\} \tag{16}$$

$\mathrm{FAA}(f)$ 越大，f 抵抗快速代数攻击能力越强。当 $\mathrm{FAA}(f)=n$ 时，称 f 抵抗快速代数攻击能力为最优的；当 $\mathrm{FAA}(f)=n-1$ 时，称 f 抵抗快速代数攻击能力为次优的。

2　混合禁忌搜索算法

禁忌搜索算法由 Glover[27]于 1986 年提出，并于 1989 年和 1990 年进行进一步的完善[28-30]。禁忌搜索算法基于对人类记忆功能的模仿，是一种迭代型的全局邻域优化算法。如何通过修改真值表来得到密码学性质更好的布尔函数本质上属于一个组合优化问题，而禁忌搜索算法在组合优化等领域的成功应用为这一问题的解决提供了一种新思路。

禁忌搜索算法有许多优点，例如选取优良解的概率远大于选取劣解的概率；通过“禁忌”来避免算法陷入局部最优；所有候选解的代价函数值都弱于当前最优解时，会接受“弱解”等。但同时禁忌搜索算法也有一定的缺点：一是对初始解有较强的依赖性；二是迭代过程是串行的，因此收敛速度较慢。爬山算法可以很好地解决这 2 个问题，它是一种基于贪心思想的算法，每次移动都只接受比当前

解状态更优的解，到达某个“峰顶”就停止移动。其优点是实现过程较为简单且收敛速度快，缺点是很容易陷入局部最优解，因为所到达的“峰顶”可能并不是全局最优解。

结合 TS 算法和 HC 算法的优点，本文提出了一种搜索能力更强、执行速度更快的优良布尔函数搜索算法——HTS 算法。一方面，由 HC 算法生成初始解可以解决 TS 算法对初始解的依赖性；另一方面，在 TS 算法每次迭代结束后对当前最优解执行 HC 算法，可以提高 HTS 算法的收敛速度。HTS 算法流程如图 1 所示。

2.1 HTS 算法细节设计

2.1.1 邻域解的选取

满足平衡性是一阶弹性布尔函数的必要条件。为了保证所生成的邻域解的平衡性，本文中 HTS 算法的邻域移动方式采用 $2-\text{opt}$，即交换布尔函数真值表中 2 个不同值的位置。对于一个 n 元布尔函数 f，其 $2-\text{opt}$ 邻域的大小为 $2^{n-1}\times 2^{n-1}=2^{2n-2}$。若选取整个邻域作为候选解，则计算量相当庞大，算法迭代时间也会过长，因此随机性地从 $2-\text{opt}$ 邻域中选取一部分解作为候选解是更好的选择。

2.1.2 禁忌对象及禁忌表长度的选取

禁忌对象通常可以选取解本身或者代价函数值等，由于选取代价函数值作为禁忌对象可能会错失优良解，而且布尔函数真值表本身就是二进制编码形式，因此本文选取布尔函数本身作为禁忌对象。禁忌表的更新方式采用先入先出规则，即禁忌表每次更新时，将要禁忌的对象放到禁忌表的末端，而最早进入禁忌表的禁忌对象将从禁忌表中释放。禁忌长度就是禁忌表的长度，即处于禁忌表末端的禁忌对象在不被特赦的情况下被释放时所经历的禁忌表更新次数。若禁忌长度过大，则容易造成算法收敛速度过慢以及存储量过大；若禁忌长度过小，则容易使算法在寻优时陷入“迂回”搜索。本文中 HTS 算法的禁忌长度为 $\text{len\_tabu}=\lceil 0.015\text{num\_cs}\rceil$，其中 num_cs 为候选解的个数。

2.1.3 特赦准则

禁忌搜索算法中的“禁忌”不是绝对的禁止，当存在一个优于当前最优解的禁忌候选解时，特赦准则的作用就是将这个禁忌候选解解禁，从而实现更优的性能。本文中 HTS 算法的特赦准则是基于代价函数值的准则，即若某个禁忌候选解的代价函数值优于当前最优解，就从禁忌表中解禁此候选解并更新该候选解为当前最优解。

2.1.4 代价函数

代价函数决定了算法优化的方向，对算法优化的结果起着至关重要的作用。以高非线性度、低自相关性、一阶弹性为优化目标，以最优代数次数、最优代数免疫度、最优（次优）抵抗快速代数攻击能力为约束条件，本文提出了 2 个代价函数。

由式(6)可知，非线性度越大，$\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|$ 越小，Walsh 谱值的分布越均匀；由式(13)可知，自相关平方和指标与 Walsh 谱值的四次方和成正比。因此，为了提升布尔函数的非线性度以及自相关性，本文在代

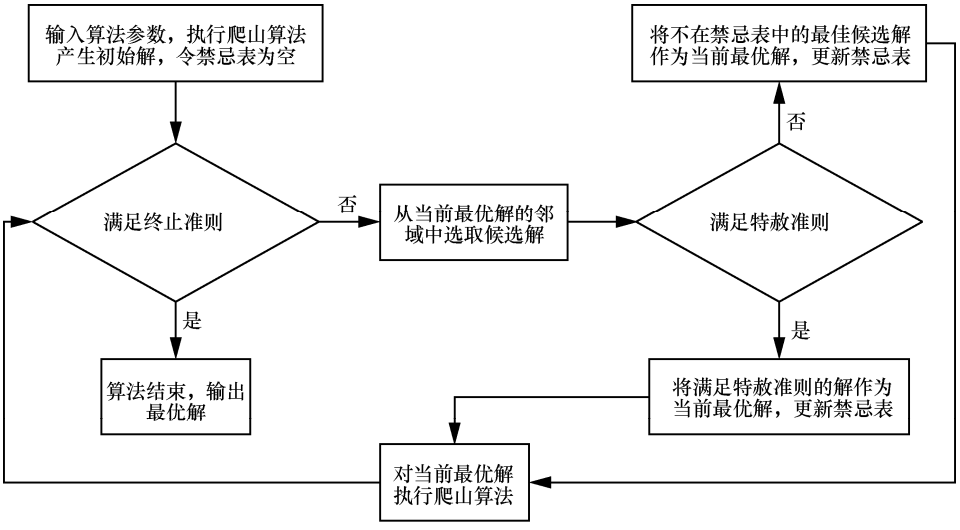

图 1 HTS 算法流程

价函数 $\mathrm{cost}_1(f)$ 中引入了 $\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|^4$。由定理 1 可知，相关免疫阶为 1 的布尔函数在 $\mathrm{wt}(\boldsymbol{\alpha})=1$ 处的 Walsh 谱值为 0。因此，除 $\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|^4$ 之外，还需额外对 $\mathrm{wt}(\boldsymbol{\alpha})=1$ 处的 Walsh 谱值施加一个惩罚项 $\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n,\mathrm{wt}(\boldsymbol{\alpha})=1}\left|W_f(\boldsymbol{\alpha})\right|^K$，从而得到 HTS 算法中的第一个代价函数为

$$\mathrm{cost}_1(f)=\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|^4+\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n,\mathrm{wt}(\boldsymbol{\alpha})=1}\left|W_f(\boldsymbol{\alpha})\right|^K \quad (17)$$

惩罚项中 K 的值不是定值，K 值过大会导致除一阶弹性外的其他密码学性质变差，K 值过小会导致一阶弹性缺失，因此 K 值的选取应当适中。对任意的 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$，满足 $\mathrm{wt}(\boldsymbol{\alpha})=1$ 的 $\boldsymbol{\alpha}$ 共有 n 个，在整个 $\mathbb{F}_2^n$ 空间中的占比为 $\frac{n}{2^n}$。由 $\frac{n+1}{2^{n+1}}\frac{2^n}{n}=\frac{n+1}{2n}$ 可知，当 $n>1$ 时，这个比值会随着 n 的增大而减小，因此 K 值的大小应随着 n 的增大而增大。经过反复实验，K 值的选取如表 1 所示。

表 1　　K 值的选取

n	K
8	8.6
9	9.8
10	10.7
11	11.7
12	12.6
13	13.7
14	14.6

由于在 HTS 算法中使用代价函数 $\mathrm{cost}_1(f)$ 所得到的一阶弹性布尔函数的非线性度未达到期望值，因此需要对代价函数进行修正。在满足一定条件的情况下，线性变换[31]可以在保持非线性度以及自相关绝对值指标等密码学性质不变的同时，将布尔函数的弹性阶由 0 变为 1。因此，第二个代价函数先不考虑一阶弹性这一优化目标。由于 Bent 函数具有最大的非线性度以及最小的自相关绝对值，且对任意的 $\boldsymbol{\alpha}\in\mathbb{F}_2^n$，都有 $\left|W_f(\boldsymbol{\alpha})\right|=2^{\frac{n}{2}}$；平衡布尔函数的非线性度及自相关性虽然无法达到 Bent 函数的结果，但通过使布尔函数的 Walsh 谱值的绝对值逼近 $2^{\frac{n}{2}}$，可以使布尔函数逐渐逼近 Bent 函数这两方面的性质。因此在 HTS 算法中引入第二个代价函数，即

$$\mathrm{cost}_2(f)=\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left(\left|W_f(\boldsymbol{\alpha})\right|^3-2^{\frac{3n}{2}}\right)^2 \quad (18)$$

2.1.5　终止准则

为了避免算法在搜索的过程中错失符合条件的布尔函数或者进行多余的迭代，除了设置最大迭代次数 max_iter 这一终止准则外。针对不同的代价函数设置了不同的提前终止准则。本文算法使用代价函数 $\mathrm{cost}_1(f)$ 时所对应的提前终止准则为

$$\mathrm{term}_1(f)=\left(N_f\geqslant\exp\_N_f\right)\wedge\left(r(f)=0\right) \quad (19)$$

其中，$\exp\_N_f$ 为期望的非线性度的最小值；$r(f)=\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n,\mathrm{wt}(\boldsymbol{\alpha})=1}\left|W_f(\boldsymbol{\alpha})\right|$，由定理 1 可知，当 $r(f)=0$ 时，f 具有一阶相关免疫性。设 $f\in B_n$，定义集合 $\mathrm{WZ}(f)=\left\{\boldsymbol{\alpha}\in\mathbb{F}_2^n\mid W_f(\boldsymbol{\alpha})=0\right\}$ 为 f 的 Walsh 谱值为 0 的点的集合。若在该集合中存在 n 个线性无关的向量，则由这些向量可以构成一个 $n\times n$ 的非奇异矩阵 $\boldsymbol{B}_f$。令 $\boldsymbol{C}_f=\boldsymbol{B}_f^{-1}$，就可以通过线性变换[31]构造一个新的函数 $f'(\boldsymbol{X})=f(\boldsymbol{C}_f\boldsymbol{X})$。$f'$ 不仅具有一阶相关免疫性，而且与 f 一样具有相同的非线性度、自相关绝对值指标、代数次数、代数免疫度、抵抗快速代数攻击能力等。结合定理 2 的结论可知，线性变换前的布尔函数 f 的代数次数也必须小于或等于 $n-2$，且集合 $\mathrm{WZ}(f)$ 的基数必须大于或等于 n，从而得到使用代价函数 $\mathrm{cost}_2(f)$ 所对应的提前终止准则为

$$\mathrm{term}_2(f)=\left(N_f\geqslant\exp\_N_f\right)\wedge\Big(\left(r(f)=0\right)\vee\left(\left(\deg(f)\leqslant n-2\right)\wedge\left(\left|\mathrm{WZ}(f)\right|\geqslant n\right)\right)\Big) \quad (20)$$

2.1.6　HTS 算法中的爬山算法

除了减少 HTS 算法对初始解的依赖性以及提高算法的收敛速度之外，还需 HC 算法在优化过程中进一步提高布尔函数的非线性度这一优化目标。由式(6)可知，对任意的 $f\in B_n$，$\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|$ 越小，f 的非线性度就越大。因此，与 2.1.4 节中的代价函数不同，HC 算法所采用的代价函数为 $\mathrm{cost}_{\mathrm{hc}}(f)=\max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|W_f(\boldsymbol{\alpha})\right|$。在 HC 算法的具体步骤中，本文还引入了一种深度优先搜索的思想，具体步骤如算法 1 所示。

算法 1 HC 算法

步骤 1 输入 n 元平衡布尔函数 f 的真值表，计算其代价函数值 $\mathrm{cost}_{\mathrm{hc}}(f)$。

步骤 2 令 $i=1$，并记 f 的单点优化集合 C_0 和 C_1 都为空。

2.1) 对 f 真值表第 i 个位置的值进行取反，并将其记为 g，计算 g 的代价函数值 $\mathrm{cost}_{\mathrm{hc}}(g)$。若 $\mathrm{cost}_{\mathrm{hc}}(g)<\mathrm{cost}_{\mathrm{hc}}(f)$，转到步骤 2.2)；否则，转到步骤 2.3)。

2.2) 若 $g(i)=1$，将 i 加入 C_0；否则，将 i 加入 C_1。

2.3) 令 $i=i+1$，若 $i\leqslant 2^n$，转到步骤 2.1)；否则，执行步骤 3。

步骤 3 判断 C_0 或 C_1 是否为空集，若是，则算法结束，输出 f 为最优解；否则，执行步骤 4。

步骤 4 令 $C=C_0\times C_1$（集合的笛卡儿积），记集合 C 中位置对的个数为 num_C，令 $j=1$，$\mathrm{flag}=0$。

4.1) 记集合 C 中第 j 个位置对为 $C(j)$，对 f 的真值表的 $C(j)$ 位置处的值进行取反，并将其记为 h，计算 h 的代价函数值 $\mathrm{cost}_{\mathrm{hc}}(h)$。若 $\mathrm{cost}_{\mathrm{hc}}(h)<\mathrm{cost}_{\mathrm{hc}}(f)$，那么令 $f=h$，$\mathrm{flag}=1$，并再对 f 执行爬山算法；否则，执行步骤 5。

4.2) 令 $j=j+1$，若 $j\leqslant$ num_C，转到步骤 4.1)；否则，执行步骤 5。

步骤 5 若 $\mathrm{flag}=0$，则算法结束，输出最优解 f。

下面，给出算法 1 执行的一个实例。

例 1 随机生成一个 5 元平衡布尔函数 $f=(1100\ 0010\ 0000\ 0000\ 1100\ 1111\ 1110\ 1111)$，$f$ 的非线性度为 6、自相关绝对值指标为 24、代数次数为 4、代数免疫度为 2、抵抗快速代数攻击能力为 3、弹性阶为 0。以 f 作为输入，计算 $\mathrm{cost}_{\mathrm{hc}}(f)=20$，执行步骤 2~步骤 4，当 $h=(1110\ 0010\ 0000\ 0000\ \mathbf{0}100\ 1111\ 1110\ 1111)$ 时，有 $\mathrm{cost}_{\mathrm{hc}}(h)=16<\mathrm{cost}_{\mathrm{hc}}(f)$。令 $f=h$，$\mathrm{cost}_{\mathrm{hc}}(f)=\mathrm{cost}_{\mathrm{hc}}(h)=16$，再次执行步骤 2~步骤 4，当 $h=(11\mathbf{11}\ 0010\ 0000\ 0000\ \mathbf{0}100\ 1111\ 1110\ \mathbf{0}111)$ 时，有 $\mathrm{cost}_{\mathrm{hc}}(h)=12<\mathrm{cost}_{\mathrm{hc}}(f)$，令 $f=h$，$\mathrm{cost}_{\mathrm{hc}}(f)=12$，再次执行步骤 2~步骤 4，此时不存在 h 使 $\mathrm{cost}_{\mathrm{hc}}(h)<\mathrm{cost}_{\mathrm{hc}}(f)$，因此算法 1 结束，输出 f。最终输出 f 的真值表为 (11**11** 0010 0000 0000 **0**100 1111 1110 **0**111)，相比算法最初输入的布尔函数，其非线性度由 6 提高至 10，自相关绝对值指标由 24 降低至 16，抵抗快速代数攻击的能力由 3 提高至 4，代数次数、代数免疫度及弹性阶保持不变。

2.2 HTS 算法步骤设计

2.2.1 算法改进

由于传统的禁忌搜索算法在每次迭代过程中是串行的，即首先逐一计算每个候选解的代价函数值，然后将每次计算的结果与当前最优代价函数值进行比较，以此确定是否要更新当前最优代价函数值和禁忌表。因此，当前最优代价函数值和禁忌表在迭代过程中是动态变化的，从而导致传统禁忌搜索算法无法使用并行计算，当候选解的个数较多时，算法运行速度会很慢。对此，除了引入 HC 算法来提高算法的收敛速度之外，本文还做了如下改进来提高算法的运行速度。在每次迭代时，将计算候选解的代价函数值与更新当前最优代价函数值和禁忌表分开进行，即首先使用并行计算得到所有候选解的代价函数值，并选取代价函数值最优的候选解作为当前最优解，然后将那些顺序在当前最优解之后且代价函数值劣于迭代前最优代价函数值的候选解全部舍弃，此时更新当前最优代价函数值和禁忌表只需要按先后顺序对所保留的候选解进行判断。

令 $\max\_\mathrm{iter}=10$，$\exp\_N_f=2^{n-1}-2^{\left\lfloor\frac{n}{2}-1\right\rfloor}$，代价函数为 $\mathrm{cost}_2(f)$，实验环境及其他参数的选取在第 3 节中给出。对不同变元数分别执行 4 次算法，记录每次的运行时间并计算其平均值。表 2 给出了算法改进前后运行时间对比。从表 2 可以看出，改进后算法的运行速度大大提高了。

表 2 算法改进前后运行时间对比

n	运行时间/s	
	改进前	改进后
8	2.41	1.66
9	3.51	2.52
10	10.28	6.14
11	28.39	12.58
12	176.42	40.36
13	556.82	126.62
14	1 864.96	415.16

2.2.2 算法步骤

将改进后的 TS 算法与 HC 算法相结合，就得到了如算法 2 所示的 HTS 算法。

算法 2　HTS 算法

步骤 1　输入候选解个数 num_cs、禁忌长度 len_tabu、最大迭代次数 max_iter 以及非线性度期望值 exp_N_f，令禁忌表 tabu 为空表。

步骤 2　随机生成一个平衡布尔函数 f_{rand}，对其执行 HC 算法，将 HC 算法的执行结果 f 作为初始解，判断 f 是否满足终止条件 term(f)，若满足，则算法结束，输出 f 为最优解；否则，计算 f 的代价函数值 cost(f)，令当前最优解 best_f = f 以及当前最优代价函数值 best_cost = cost(f)，记迭代次数 iter 为 0。

步骤 3　在 best_f 的 2-opt 邻域中随机选取 num_cs 个候选解，并计算所有候选解的代价函数值。

步骤 4　判断是否存在某个候选解 cf 满足终止条件 term(cf)，若存在，则算法结束，输出 cf 为最优解；否则，执行步骤 5。

步骤 5　计算所有候选解中的最优代价函数值 min_cost，并与 best_cost 进行比较，若 min_cost < best_cost，则更新当前最优解 best_f 为 min_cost 所对应的候选解，执行步骤 6；否则，执行步骤 7。

步骤 6　对候选解进行筛选，只保留那些顺序在 best_f 之前（包括 best_f）并且代价函数值小于 best_cost 的候选解。记筛选后的候选解的个数为 num_cs_rem，令 $i=1$，

6.1)　将第 i 个保留的候选解的代价函数值 cost(f_i) 与 best_cost 进行比较，若 cost(f_i) < best_cost，则更新保留的候选解中最优代价函数值 best_cost = cost(f_i)，转到步骤 6.2)；否则，转到步骤 6.3)。

6.2)　判断 f_i 是否在禁忌表 tabu 中，若是，则 f_i 满足特赦准则，此时更新禁忌表，先将 f_i 从禁忌表 tabu 中解禁，再将 f_i 加入禁忌表中；否则，将 f_i 加入禁忌表 tabu 中，更新禁忌表，执行步骤 7。

6.3)　令 $i=i+1$，若 $i \leqslant$ num_cs_rem，则返回步骤 6.1)；否则，执行步骤 8。

步骤 7　此时要接受弱解，即从所有不属于禁忌表的候选解中选取一个代价函数值最优的解作为 best_f，并更新最优代价函数值 best_cost 为 best_f 的代价函数值，将 best_f 加入禁忌表 tabu 中，更新禁忌表，执行步骤 8。

步骤 8　对 best_f 执行 HC 算法，将优化后的函数记为 hc_f，判断 hc_f 是否满足终止条件 term(hc_f)，若满足，则算法结束，输出 hc_f 为最优解；否则，执行步骤 9。

步骤 9　令 iter = iter + 1，若 iter < max_iter，转到步骤 3；否则，对 best_f 执行爬山算法并将优化后的函数 hc_f 输出为最终结果。

下面，给出算法 2 执行的一个实例。

例 2　输入 num_cs = 50、len_tabu = 1、max_iter = 200、exp_N$_f$ = 12，代价函数为 $\text{cost}_2(f)$，以例 1 最终的输出 f = (11**11** 0010 0000 0000 **0**100 1111 1110 **0**111) 作为初始解，且 f 不满足终止条件，令 best_f = f，best_cost = $\text{cost}_2(f) = 1.51\times10^7$。执行步骤 3 和步骤 4，不存在候选解满足终止条件。执行步骤 5，有 min_cost = 6.12×10^6。令 best_f = (11**11** 0010 0000 00**1**0 **0**100 11**0**1 1110 **0**111)，执行步骤 6，更新禁忌表及 best_cost = min_cost = 6.12×10^6。执行步骤 8，对 best_f 执行算法 1，执行后的输出为 hc_f = (**0**111 0010 **1**000 00**1**0 **0**100 11**0**1 1110 **0**111)，hc_f 满足终止条件，算法 2 结束，输出 hc_f。

对 hc_f 进行线性变换得 hc_f' = (0101 1000 1100 0111 0111 1001 0010 1010)，至此，就得到了一个非线性度为 12、自相关绝对值为 8、代数次数和代数免疫度为 3、抵抗快速代数攻击能力为 4 的 5 元一阶弹性布尔函数。对 5 元一阶弹性布尔函数而言，除抵抗快速代数攻击能力为次优之外，hc_f' 的其他密码学指标均达到了最优。

3　实验结果及分析

本文的实验环境是主频为 2.5 GHz 的 Intel core i5-7300HQ 处理器，内存为 8 GB，编程软件采用 MATLAB 2019a。对 8~14 元的布尔函数进行搜索时，一些参数的选取如表 3 所示。

表 3　**参数的选取**

n	num_cs	len_tabu
8	500	8
9	1 000	15
10	2 000	30
11	3 100	47
12	4 300	65
13	5 600	84
14	7 000	105

由于所得到的结果都是具有一阶弹性的布尔函数，为了方便表示，将布尔函数的密码学性质用 $(N_f,\Delta_f,\deg,\mathrm{AI},\mathrm{FAA})$ 表示，其中，N_f 为非线性度，Δ_f 为自相关绝对值，deg 为代数次数，AI 为代数免疫度，FAA 为抵抗快速代数攻击能力。在 HTS 算法中，代价函数 $\mathrm{cost}_1(f)$（$\mathrm{HTS}_{\mathrm{cost}_1}$ 算法）与 $\mathrm{cost}_2(f)$（$\mathrm{HTS}_{\mathrm{cost}_2}$ 算法）的结果对比如表 4 所示。

表 4 $\mathbf{HTS}_{\mathbf{cost}_1}$ 算法与 $\mathbf{HTS}_{\mathbf{cost}_2}$ 算法的结果对比

n	$\mathrm{HTS}_{\mathrm{cost}_1}$	$\mathrm{HTS}_{\mathrm{cost}_2}$
8	(116,24,6,4,7)	(116,24,6,4,7)
9	(236,48,7,5,8)	(236,48,7,5,8)
10	(484,80,8,5,10)	(488,72,8,5,9)
11	(984,112,9,6,10)	(988,104,9,6,10)
12	(1 988,144,10,6,11)	(1 996,152,10,6,11)
13	(4 012,216,11,7,12)	(4 020,232,11,7,12)
14	(8 072,320,12,7,13)	(8 080,320,12,7,13)

值得注意的是，$\mathrm{HTS}_{\mathrm{cost}_1}$ 搜索得到了抵抗快速代数攻击能力达到最优的 10 元布尔函数实例。整体来看，除非线性度未达到期望值外，$\mathrm{HTS}_{\mathrm{cost}_1}$ 算法得到的布尔函数的各方面性质都较优。当 $n\geqslant 10$ 时，$\mathrm{HTS}_{\mathrm{cost}_2}$ 算法所得的结果相较 $\mathrm{HTS}_{\mathrm{cost}_1}$ 算法更优。附录给出了 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法搜索得到的 8~12 元布尔函数的一些实例。

将 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法得到的结果与文献[8]（代数构造）、文献[9]（代数构造）、文献[12]（模拟退火算法）以及文献[13]（引力搜索算法）进行对比，结果如表 5 所示。其中，—表示使用对应参考文献中的方法未能给出相应的结果；–表示对应文献未能给出相应的性质。

总体而言，本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法的结果优于文献[12-13]的结果。文献[8-9]采用构造法所得的 12 或 14 元布尔函数的非线性度稍优于本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法的结果，但本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法所得函数的代数次数和抵抗快速代数攻击能力稍优于文献[8-9]的结果；另一方面，文献[8-9]的构造法对奇变元布尔函数不适用，而本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法并不受变元个数的限制，且能够搜索得到大量满足所需密码学指标的布尔函数。

下面，具体分析表 5 中各个密码学指标的对比结果。

1) 代数次数。由定理 2 中代数次数与弹性阶的关系可知，除文献[9]中 12 和 14 元布尔函数的代数次数之外，表 5 中其他布尔函数的代数次数都达到了最优。

2) 代数免疫度。当 n 为偶数时，表 5 中布尔函数的代数免疫度全部达到最优。当 n 为奇数时，本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法搜索得到的布尔函数的代数免疫度也全部达到最优；文献[12-13]结果中的布尔函数的代数免疫度是次优的；而文献[8-9]的结果是基于 Bent 函数构造的，未能给出奇数元的结果。

3) 抵抗快速代数攻击的能力。文献[12-13]以及本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法得到的布尔函数的抵抗快速代数攻击能力都达到了 $n-1$（次优）；Carlet[32]证明了文献[8]中所构造的布尔函数的抵抗快速代数攻击能力较弱；文献[9]的结果中，除 8 元布尔函数之外，其他布尔函数的抵抗快速代数攻击能力皆未达到次优。

4) 非线性度。当 $n\leqslant 9$ 时，本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法得到的布尔函数的非线性度与文献[12]的结果相同，都优于文献[13]的结果；当 $n\geqslant 10$ 时，本文 $\mathrm{HTS}_{\mathrm{cost}_2}$ 算法的结果优于文献[13]的结果。对于变元个数只为偶

表 5 一阶弹性布尔函数的密码学性质对比

n	文献[8]	文献[9]	文献[12]	文献[13]	$\mathrm{HTS}_{\mathrm{cost}_2}$
8	(112,–,6,4,–)	(116,32,6,4,7)	(116,48,6,4,7)	(112,48,6,4,7)	(116,24,6,4,7)
9	—	—	(236,48,7,4,8)	(232,80,7,4,8)	(236,48,7,5,8)
10	(484,–,8,5,–)	(488,–,8,5,8)	(484,96,8,5,9)	—	(488,72,8,5,9)
11	—	—	(984,136,9,5,10)	—	(988,104,9,6,10)
12	(1 996,–,10,6,–)	(2 008,96,9,6,9)	(1 988,184,10,6,11)	—	(1 996,152,10,6,11)
13	—	—	(4 012,288,11,6,12)	—	(4 020,232,11,7,12)
14	(8 100,–,12,7,–)	(8 112,–,11,–,–)	(8 072,368,12,7,13)	—	(8 080,320,12,7,13)

数的情形，当$n \leqslant 10$时，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法得到的布尔函数的非线性度与文献[9]的结果相同，皆优于文献[8]的结果；当$n = 12,14$时，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法与文献[8]中得到的布尔函数的非线性度皆弱于文献[9]的结果。

5) 自相关性。与上述 4 个密码学性质不同，对一个布尔函数而言，其自相关绝对值越小，该布尔函数的自相关性就越好。本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法搜索得到的布尔函数的自相关绝对值优于文献[12-13]的结果；当$n = 12$时，文献[9]中布尔函数的自相关绝对值优于本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法的结果；而文献[8]中未考虑这一性质。

4　结束语

本文以高非线性度、低自相关性、一阶弹性为优化目标，以最优代数次数、最优代数免疫度、最优（次优）抵抗快速代数攻击能力为约束条件，提出了 2 个代价函数。本文在 HC 算法和 TS 算法上进行改进，得到了一种新的布尔函数快速生成算法——HTS 算法。与传统的 TS 算法相比，HTS 算法不仅搜索能力更强，而且运行速度也更快。利用本文算法对 8~14 元的布尔函数进行搜索，得到了满足几乎所有密码学性质的布尔函数。当$n = 8,10$时，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法得到的一阶弹性布尔函数的非线性度都达到了目前已知的最大值。与其他启发式算法相比，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法的搜索能力更强，所得的结果也更好。不同于文献[8-9]中的代数构造法所存在的一些局限性，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法不仅对变元个数为奇数时的情形也同样适用，而且以随机生成的平衡布尔函数作为初始输入（除了 10 元布尔函数需要尝试多个输入之外），可以搜索到满足上述密码学性质的全部布尔函数。

当$n = 12,14$时，本文$\mathrm{HTS}_{\mathrm{cost}_2}$算法生成的布尔函数的非线性度与目前已知的最大值还存在着一些差距，除了B_n的基数会随着n的增大爆炸性增长而导致计算机搜索更加困难之外，HTS 算法的性能以及代价函数的选取也可能是导致这一差距的原因。因此，下一步的研究方向是在保持其他优良密码学性质的同时，提高布尔函数的非线性度以及降低其自相关绝对值，而将一些代数构造的方法加入启发式搜索算法中无疑是一种解决问题的新思路。

附录

$n = 8,(116,24,6,4,7)$

3639 7756 304C C72F D09F 425A F13A 1E49 0972 ED40 EAA7 C1F8 BE95 AAC9 D843 462F

$n = 9,(236,48,7,5,8)$

3EA4 427D EF09 C463 2F64 E3B0 B950 9D14 4A2A CD57 42BD ACFC 5882 179A 2F9A CEAE A11F B352 E489 2775 36EB E145 F17E 02DC 8C86 26ED 61F5 2E60 1F57 7D32 781C A071

$n = 10,(488,72,8,5,9)$

CAAC 4954 3C59 39D1 C724 B6FB B2E7 89F4 AD63 4BFA 8250 FA13 2D4E FA02 2247 9A0A 7968 3E1B 9983 3146 4CE4 53A7 6B42 5C92 0FE3 B3F4 668F 78E4 2071 79DC 7BD3 267A F70B 5EBB E015 4EE7 5C40 BC7C 17A0 19AE 794D 0121 775F 2A2F FA1C EC6A 621C 59CC CCB1 C0A4 1C6F 07C5 DFC0 AEE2 81EF 7852 3205 4339 DF9E 8835 059F 10D8 79EC 3DB6

$n = 11,(988,104,9,6,10)$

0D0D 65C0 7F85 16AF 82DF 0126 C7CF 47DD 9734 985F 71F2 3A65 2DA9 B6F7 2985 5AF3 118C A003 7568 9D9F 20D4 97BC 7920 3681 0163 F04C 3CC9 C9A9 6D36 43CB 0FC3 8674 B5A0 EF5E A548 D121 9E1B 9A44 C578 6291 3EB2 0174 F0BA 93B3 8D73 C8D7 3488 BFAC 8F18 C399 59E0 4FC8 E455 963B C7FB 354D D3A5 E86F 7837 1A1B B191 E8BE BE44 629F FFE5 0D2D 1FF3 9534 C876 EEA9 11A5 74D2 7EE0 1468 0730 400F E963 ECED 0C76 3B62 2B76 D5D7 3FCA C9CA AB98 EA20 9B7F CAB9 08CD 2E02 F847 9B23 53E6 F440 8BA3 6D95 5612 EAF6 4AAB CC73 510F 6DA3 E598 56B2 0810 B4F9 736B 40A0 91BD 5ABE 9480 82CF 155D 7A66 BBCA 2643 4984 1DA6 88C9 88C0 FD5E 5DEB 7DB3 2BAD 6957 E082 73A4 CF01

$n = 12,(1996,152,10,6,11)$

A8C7 C7A5 ABB9 6349 3704 C774 EF9F C0DA CB92 BFA8 AB54 4315 077B A0B3 984D DA89 B55B 2CAA 4D28 A082 7F63 5CC4 E3F7 2F7C A0CB 178D F559 AFA6 456B 92F5 0EA3 6324 D781 8044 94DC 66CC 3119 01E7 06E8 990D BE22 D706 6439 5954 DDB3 B66D D607 1016 5C13 2DFC A8E1 A572 CB94 0940 1DF4 11DA F1BD F071 B540 3E58 CC02 56CC 47E2 6B8D 3648 A16B B451 F017 0564

9BAA D9BF B702 3CB5 6D2C 438D FFAA C016 6B62 3BDF 6EA1 F08C 7084 AD57 00E0 1D1E ADCF 2067 942B 28CC 679B 853B 131D 629A EF1E F65C 78BF EA42 1A46 39E8 93E0 729F 5B28 AEA4 6B26 989D 13A6 5971 FB74 0CFD 7028 231A 10CB 178F 89B1 BD9D A8DD 3AB8 9BD2 1797 4B3B 0311 DA7B 7F77 939D CD35 E39D CF21 D626 BE91 DEE2 9024 05A6 F036 AFB5 FAD4 A4E5 0A3C 4B46 F109 9BE6 29B1 C055 00C1 7B75 BEB5 A052 4DCF CB16 B154 6A16 DFBE C322 ECE8 9952 7791 E7C3 CE24 05B1 2A42 5B3F 07F7 B29A D326 89C7 0602 F6FD A2A6 1E72 DCE5 977A 6EDF 195D 27E4 AE57 10C3 5A1E 4ED1 39A5 AD45 8BFA 786F 0D93 6518 CC4E 4CEF F2BE C827 C60C 90D2 16FC 198F 799E B7E4 23FE BF35 72B4 67C1 7EB2 C1FB 9D99 838C C2C3 FFE0 6048 547E C715 E0D4 0269 4E3C BB79 8248 7A90 0C9E 38C2 195D 3146 507F B24A D934 1B21 67A8 0EA9 00AE C29D D0FC BE54 0C96 CB84 AD3F 39B3 7B71 53D5 812A 8EE2 205A 4DB7 F4A2 B483 294D D76F 600F FDF5 DA9C 5CB0 7029 E038 2A80 D6C8 45C7 8CE0 4B82 41F8 7E1E 21A1 FD87 4D6F

参考文献：

[1] 张卫国, 肖国镇. 具有偶数个变元的高非线性度平衡布尔函数的构造[J]. 电子学报, 2011, 39(3): 727-728.
ZHANG W G, XIAO G Z. Construction of balanced Boolean functions with high nonlinearity on even number variables[J]. Acta Electronica Sinica, 2011, 39(3): 727-728.

[2] 欧智慧, 赵亚群, 李旭. 一类密码函数的构造与分析[J]. 通信学报, 2013, 34(4): 106-113.
OU Z H, ZHAO Y Q, LI X. Construction and analysis of one class of cryptographic functions[J]. Journal on Communications, 2013, 34(4): 106-113.

[3] 孙天锋, 胡斌, 杨阳. Plateaued 函数的构造方法研究[J]. 电子与信息学报, 2018, 40(10): 2352-2357.
SUN T F, HU B, YANG Y. Research on the construction of Plateaued functions[J]. Journal of Electronics & Information Technology, 2018, 40(10): 2352-2357.

[4] 杜蛟, 刘春红, 庞善起. 4t-1 元旋转对称 2-弹性函数的构造[J]. 通信学报, 2020, 41(11): 169-175.
DU J, LIU C H, PANG S Q. Constructions of rotation symmetric 2-resilient functions with 4t-1 number of variables[J]. Journal on Communications, 2020, 41(11): 169-175.

[5] ZHANG F R, PASALIC E, WEI Y Z. Constructions of balanced Boolean functions on even number of variables with maximum absolute value in autocorrelation spectra $< 2^{\frac{n}{2}}$ [J]. Information Sciences, 2021, 575: 437-453.

[6] CARLET C, FENG K Q. An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity[C]//2008 International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2008: 425-440.

[7] TU Z R, DENG Y P. A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity[J]. Designs, Codes and Cryptography, 2011, 60(1): 1-14.

[8] TU Z R, DENG Y P. Boolean functions optimizing most of the cryptographic criteria[J]. Discrete Applied Mathematics, 2012, 160(4/5): 427-435.

[9] ZHANG W G, PASALIC E. Improving the lower bound on the maximum nonlinearity of 1-resilient Boolean functions and designing functions satisfying all cryptographic criteria[J]. Information Sciences, 2017, 376: 21-30.

[10] SABER Z, UDDIN M F, YOUSSEF A. On the existence of (9, 3, 5, 240) resilient functions[J]. IEEE Transactions on Information Theory, 2006, 52(5): 2269-2270.

[11] LIU W M, YOUSSEF A. On the existence of (10, 2, 7, 488) resilient functions[J]. IEEE Transactions on Information Theory, 2009, 55(1): 411-412.

[12] YANG J P, ZHANG W G. Generating highly nonlinear resilient Boolean functions resistance against algebraic and fast algebraic attacks[J]. Security and Communication Networks, 2015, 8(7): 1256-1264.

[13] 贾少帅, 张凤荣. 基于引力搜索的布尔函数生成算法[J]. 计算机应用研究, 2021, 38(2): 430-434.
JIA S S, ZHANG F R. Boolean function generation algorithm based on gravitational search algorithm[J]. Application Research of Computers, 2021, 38(2): 430-434.

[14] CLARK J A, JACOB J L. Two-stage optimisation in the design of Boolean functions[C]//2000 Australasian Conference on Information Security and Privacy. Berlin: Springer, 2000: 242-254.

[15] 李超, 胡朋松, 海昕. 布尔函数设计中的爬山算法及其改进[J]. 通信学报, 2007, 28(3): 130-133.
LI C, HU P S, HAI X. Improved hill-climbing methods in the design of Boolean function[J]. Journal on Communications, 2007, 28(3): 130-133.

[16] BEHERA P K, GANGOPADHYAY S. An improved hybrid genetic algorithm to construct balanced Boolean function with optimal cryptographic properties[J]. Evolutionary Intelligence, 2022, 15(1): 639-653.

[17] JAKOBOVIC D, PICEK S, MARTINS M S R, et al. Toward more efficient heuristic construction of Boolean functions[J]. Applied Soft Computing, 2021, 107: 107327.

[18] CARLET C, JAKOBOVIC D, PICEK S. Evolutionary algorithms-assisted construction of cryptographic Boolean functions[C]//Proceedings of the Genetic and Evolutionary Computation Conference. [S.l.:s.n.], 2021: 565-573.

[19] MACWILLIAMS F J, SLOANE N J A. The theory of error correcting codes[M]. Amsterdam: Elsevier, 1977.

[20] XIAO G Z, MASSEY J L. A spectral characterization of correlation-immune combining functions[J]. IEEE Transactions on Informa-

tion Theory, 1988, 34(3): 569-571.

[21] SIEGENTHALER T. Correlation-immunity of nonlinear combining functions for cryptographic applications [J]. IEEE Transactions on Information Theory, 1984, 30(5): 776-780.

[22] ZHANG X M, ZHENG Y L. GAC-the criterion for global avalanche characteristics of cryptographic functions[J]. Journal of Universal Computer Science, 1995, 1(5): 316-333.

[23] CARLET C. Partially-bent functions[J]. Designs, Codes and Cryptography, 1993, 3(2): 135-145.

[24] COURTOIS N T. Algebraic attacks on combiners with memory and several outputs[C]//2004 International Conference on Information Security and Cryptology. Berlin: Springer, 2004: 3-20.

[25] MEIER W, PASALIC E, CARLET C. Algebraic attacks and decomposition of Boolean functions[C]//2004 International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004: 474-491.

[26] COURTOIS N T. Fast algebraic attacks on stream ciphers with linear feedback[C]//2003 Annual International Cryptology Conference. Berlin: Springer, 2003: 176-194.

[27] GLOVER F. Future paths for integer programming and links to artificial intelligence[J]. Computers & Operations Research, 1986, 13(5): 533-549.

[28] GLOVER F. Tabu search—part I[J]. ORSA Journal on Computing, 1989, 1(3): 190-206.

[29] GLOVER F. Tabu search—part II[J]. ORSA Journal on Computing, 1990, 2(1): 4-32.

[30] GLOVER F. Tabu search: a tutorial[J]. Interfaces, 1990, 20(4): 74-94.

[31] MAITRA S, PASALIC E. Further constructions of resilient Boolean functions with very high nonlinearity[C]//Proceedings of Sequences and Their Applications. Berlin: Springer, 2002: 265-280.

[32] CARLET C. On a weakness of the Tu-Deng function and its repair[R]. Cryptology ePrint Archive, 2009.

[作者简介]

王维琼（1979− ），女，重庆人，博士，长安大学教授，主要研究方向为编码理论与密码学。

许豪杰（1998− ），男，山西文水人，长安大学硕士生，主要研究方向为密码学。

崔萌（1997− ），女，河南信阳人，长安大学硕士生，主要研究方向为密码学。

谢琼（1998− ），女，河南永城人，长安大学硕士生，主要研究方向为密码学。