UCLV
Universidad Central
"Marta Abreu" de Las Villas

MFC
Facultad de Matemática
Física y Computación

Departamento de Matemática

# TRABAJO DE DIPLOMA

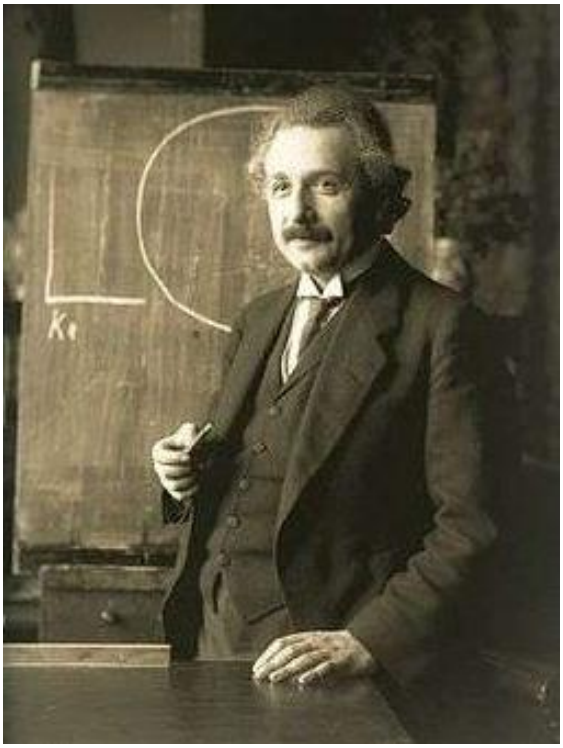
***Diseño de un nuevo algoritmo heurístico para obtener funciones booleanas con fuertes propiedades criptográficas.***

---

*Autora: Lilian Bárbara Pérez Sosa.*
*Tutor: Msc. Gonzalo Palencia Fernández.*

Santa Clara 2018


Para cualquier información contacte con:
Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830
Teléfonos.: +53 01 42281503-1419



***"La mente intuitiva es un sagrado regalo y la mente racional es un fiel sirviente. Nosotros hemos creado una sociedad que honra al sirviente y se ha olvidado del regalo."***

*Albert Einstein*

## Dedicatoria

Este trabajo va dedicado a mis padres, por guiarme por el camino correcto y por siempre brindarme todo su apoyo.

## Agradecimientos

A mis padres y demás familiares, con los que he podido contar siempre. A mi tutor Gonzalo J. Palencia Fernández por su orientación y confianza. En especial a mi tío Jesús Alberto Sosa por su ayuda incondicional y a Ignacio González. A mis compañeros de aula, por ser el mejor grupo que hubiera podido tener y en especial a Pedro Leonardo por toda su ayuda en los estudios.

A todos, mi más sincero agradecimiento.

## Resumen

En la presente tesis se expone el diseño de un algoritmo heurístico híbrido entre Recocido Simulado y Búsqueda Tabú, capaz de encontrar funciones booleanas $m$-resistentes ($m = 1$ o $m = 2$) con alta no linealidad, un alto grado algebraico y que satisfacen un alto orden de Criterio de Propagación. Este algoritmo usa como solución candidata inicial una función construida algebraicamente que posee las propiedades de ser $\mathrm{m}$-resistente, tener alta no linealidad y alto grado algebraico; además, el algoritmo contribuye a incluir la propiedad de un alto criterio de propagación; dirigiendo así la búsqueda a regiones del espacio con buenas propiedades. El algoritmo se mueve en el espacio de búsqueda impidiendo que las propiedades de la función booleana bajen de un umbral pre-especificado, mientras que a la vez se incluyen propiedades mediante el uso de una función de costo.

## Abstract

In the present thesis the design of a hybrid heuristic algorithm between Simulated Annealing and Tabu Search capable of finding m- resilient Boolean functions (m = 1 or m = 2) with high non-linearity, a high algebraic degree and satisfying high order of Propagation Criteria. This algorithm uses as an initial candidate solution an algebraically constructed function that has the properties of being m- resilient, having high non-linearity and high algebraic degree; also, the algorithm helps to include the property of a high propagation criterion; directing the search to regions of space with good properties. The algorithm moves in the search space preventing the properties of the Boolean function from falling below a pre-specified threshold, while at the same time including properties by using a cost function.

# Índice

## Introducción

Históricamente la Criptografía ha sido vista como un arte más que una ciencia, pero existiendo siempre dos grupos bien diferenciados: los criptógrafos, cuyo trabajo es diseñar sistemas criptográficos; y los criptoanalistas, cuyo trabajo es tratar de infringir estos sistemas criptográficos. La humanidad vive en estos momentos la bien conocida “Era de la Información”, época histórica caracterizada por la propiedad intelectual y el conocimiento privado, consideradas ambas de extremo valor. La información es usada de muchas formas: financieramente (mediante transacciones), legalmente (en documentos), militarmente (planes y estrategias) y políticamente. La protección de esta información, durante su almacenamiento, tránsito y uso diario, es vital; además, poner dicha información en compromiso puede resultar en pérdidas financieras, exposición de secretos comerciales y militares, así como la pérdida de cuantiosas vidas humanas. Los cifrados simétricos tienen una clave común secreta compartida por las partes que se comunican, mientras que los cifrados asimétricos usan diferentes llaves para el proceso de cifrado y descifrado. Los primeros pueden ser categorizados en cifrado en bloque o en flujo, donde los datos de entrada al cifrado toman bloques de datos o un flujo continuo de bits, respectivamente. Otro tipo de sistema criptográfico de cifrado que comprime datos para formar una marca con el objetivo de proveer integridad y/o autenticidad, es la bien conocida función hash. Las funciones booleanas desempeñan un papel importante en la Criptografía moderna por su capacidad para satisfacer, con mayor seguridad, la demanda continua de las comunicaciones. Aunque desde finales de la década de 1980 ha podido apreciarse un creciente interés por investigar en esta área, existen todavía muchos problemas en relación con el diseño y análisis de funciones booleanas para la Criptografía. El nivel de seguridad alcanzado en las aplicaciones basadas en estas funciones se mide por la calidad de las propiedades combinatorias dentro de las mismas. La selección de funciones booleanas con fuertes propiedades criptográficas reduce la eficacia de los ataques de criptoanálisis avanzados, incluyendo el criptoanálisis lineal y el criptoanálisis diferencial; así como el avance de la tecnología computacional, que funciona

tanto a favor como en contra de la seguridad criptográfica puede atentar contra dicha seguridad.

El diseño y análisis de las funciones booleanas para las aplicaciones criptográficas normalmente implican una cantidad considerable de procesamiento computacional. En particular, debido al gran número de variables de entrada que este análisis supone, se requiere una elevada demanda de recursos informáticos. La construcción de funciones criptográficamente útiles es también una tarea difícil. Una gama de técnicas algebraicas están disponibles actualmente para la construcción de tales funciones, sin embargo, estos métodos pueden ser complejos, y no siempre producen una variedad suficiente de funciones. Los métodos heurísticos son un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad. Se usa el calificativo heurístico en contraposición a exacto. Es conocido que a medida que aumenta el tamaño del parámetro de entrada, se hace rápidamente poco factible hacer una búsqueda exhaustiva en la totalidad del espacio. Debido a esto, para investigar sobre estas funciones, particularmente en un espacio de búsqueda grande, es necesario emplear técnicas que dirijan la búsqueda a ciertas regiones de interés en el espacio, regiones que exhiben una o más características deseadas que influyen a que la función sea fuerte en términos criptográficos; por lo que se emplean los métodos metaheurísticos.

**Problema científico.**

¿Cómo obtener funciones booleanas con fuertes propiedades criptográficas mediante el uso de las Metaheurísticas?

**Hipótesis del problema.**

Si se implementan algoritmos basados en métodos metaheurísticos a partir de construcciones algebraicas, entonces se obtendrán funciones booleanas con fuertes propiedades criptográficas.

**Objetivo general.**

Diseñar un nuevo algoritmo heurístico con construcciones algebraicas para obtener funciones booleanas con fuertes propiedades criptográficas.

**Objetivos específicos.**

- Analizar las características y propiedades criptográficas de las funciones booleanas, así como las relaciones entre las propiedades que poseen.
- Estudiar los principales algoritmos heurísticos e híbridos relacionados con la investigación.
- Aplicar algoritmos heurísticos en conjunto con construcciones algebraicas previamente desarrolladas, para optimizar propiedades criptográficas.

**Novedad Científica.**

Se plantea una nueva estrategia de búsqueda de funciones booleanas con propiedades criptográficas fuertes usando métodos heurísticos a partir de funciones iniciales construidas algebraicamente.

La presente tesis contiene una introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas. El capítulo 1 presenta una exposición de la Teoría de Funciones Booleanas que será relevante para comprender el trabajo realizado en esta investigación. Se incluyen definiciones y teoremas ampliamente detallados, relacionados con el tema de funciones booleanas, así como explicaciones que proveen la estructura necesaria para reconocer y analizar las características y limitaciones en esta área de trabajo. Se definen y analizan propiedades criptográficas de las funciones booleanas, así como las relaciones entre pares particulares de estas y la medida en que pueden coexistir en la misma función. En el capítulo 2 se presentan métodos heurísticos relacionados con la obtención de funciones booleanas con fuertes propiedades criptográficas, así como otro método que no ha sido utilizado directamente en la obtención de funciones booleanas; pero sí en otras áreas de la criptografía. Este capítulo incluye una descripción de sus algoritmos, así como una discusión sobre el funcionamiento del proceso implicado para

desarrollar dicho algoritmo. Básicamente se realiza una exposición sobre cuatro métodos heurísticos: Búsqueda Tabú, Hill Climbing, Algoritmos Genéticos y Recocido Simulado, así como métodos híbridos entre pares de estos y comparaciones que se han hecho entre algunos de ellos en otros tipos de problemas. El capítulo 3 constituye un nuevo trabajo en esta área de investigación; contiene la descripción y el algoritmo de un nuevo método heurístico que ha sido desarrollado para esta tesis. Este nuevo algoritmo propone la idea de aplicar métodos heurísticos en la optimización de funciones booleanas obtenidas algebraicamente, de forma tal que dicha construcción inicial brinde de por sí una función suficientemente buena que satisfaga cierto grupo de propiedades criptográficas, y los métodos heurísticos contribuyan a mantener estas propiedades, así como incorporar otras nuevas; con este fin se utiliza un método heurístico híbrido de Recocido Simulado y Búsqueda Tabú, aprovechando las fortalezas de cada una de estas técnicas y minimizando sus debilidades. La función inicial posee las propiedades de ser $m$-resistente, tener alta no linealidad y alto grado algebraico y el algoritmo contribuye a incluir la propiedad de un alto criterio de propagación. Este capítulo provee las demostraciones rigurosas de la construcción algebraica utilizada en las funciones booleanas iniciales.

Finalmente se presentan las conclusiones sobre la investigación realizada. Además se recomiendan algunas direcciones para futuras investigaciones en esta área.

# Capítulo 1: Teoría de Funciones Booleanas

En este capítulo se realiza un análisis de las funciones booleanas a partir de sus características, propiedades, relaciones entre algunas de sus propiedades y resultados necesarios para que el lector pueda comprender a cabalidad la investigación llevada a cabo.

## 1.1 Características de las funciones booleanas

Ahora se expondrán las definiciones más importantes referentes a las características criptográficas de las funciones booleanas para tener un conocimiento básico previo para su análisis.

Sea $V_n$ el espacio vectorial de dimensión $n$ sobre el campo de los elementos de $\mathbb{F}_2$. Para dos vectores de $V_n$, $a = (a_1, \dots, a_n)$ y $b = (b_1, \dots, b_n)$, se define el producto escalar como $a \cdot b = a_1b_1 \oplus a_2b_2 \oplus \dots \oplus a_nb_n$ donde la multiplicación y adición $\oplus$ se hacen sobre $\mathbb{F}_2$.

**Definición 1:** Una función booleana $f$ en $n$ variables es una correspondencia de $V_n$ a $\mathbb{F}_2$. La secuencia de ceros y unos definida por $(f(\mathbf{v_0}), f(\mathbf{v_1}), \dots, f(\mathbf{v_{2^n-1}}))$ es la llamada tabla de verdad, donde $\mathbf{v_0}$=(0, . . . , 0, 0),$\mathbf{v_1}$=(0, . . . , 0, 1), . . ., $\mathbf{v_{2^n-1}}$ =(1, . . . , 1,1), ordenados por orden lexicográfico. El álgebra de todas las funciones booleanas sobre $V_n$ se denotará por $\mathfrak{B}_n$.

El total de funciones booleanas de $n$- variables es $2^{2^n}$. Claramente, cuando el número de entradas $n$, crece, entonces el espacio de las funciones booleanas de salida crece drásticamente.

**Definición 2:** A cada función booleana se asocia su forma polar, denotada por $\hat{f}: V_n \longrightarrow \mathbb{R}^*$ y definida como:

$$\hat{f}(x) = (-1)^{f(x)} = 1 - 2f(x)$$

O sea, es un vector de $2^n$ elementos, donde cada elemento $\epsilon\{-1,1\}$.

**Definición 3:** Una función booleana sobre $V_n$ se puede expresar como un polinomio en

$\mathbb{F}_2[x_1, x_2, \dots, x_n]/({x_1}^2 - x_1, {x_2}^2 - x_2, \dots, {x_n}^2 - x_n)$, conocida como forma algebraica normal (ANF, siglas en inglés), que es:

$$f(x) = \sum_{\boldsymbol{a} \in V_n} c_{\boldsymbol{a}} {x_1}^{\propto_1} \dots {x_n}^{\propto_n}$$

donde $c_{\boldsymbol{a}} \in \mathbb{F}_2$ y $\boldsymbol{a} = (a_1, \dots, a_n)$.

La ANF (también conocida como polinomio de Zhegalkin o expresiones Reed-Müller de polaridad positiva) puede ser calculada en forma práctica usando un algoritmo de mariposa Divide y Vencerás llamado *Transformada Rápida de Möbius* (Elhosary, Hamdy, Farag, & Rohiem, 2013), este algoritmo posee complejidad computacional $O(n2^n)$.

**Definición 4:** Se conoce como grado algebraico al número de variables en el monomio de mayor orden con coeficiente diferente de cero. (Denotado como $\deg(\mathrm{f})$)

**Teorema 1:** Sea una función booleana de $n$ variables. Entonces $deg(f) < n$ si la función booleana tiene un peso hamming par.

**Definición 5:** Una función afín $\ell_{\mathrm{a,c}}$ en $\mathrm{V_n}$ es una función de la forma

$$\ell_{\mathrm{a,c}}(\mathrm{x}) = \mathrm{a} \cdot \mathrm{x} \oplus \mathrm{c} = \mathrm{a_1 x_1} \oplus \mathrm{a_2 x_2} \oplus \dots \oplus \mathrm{a_n x_n} \oplus \mathrm{c},$$

donde $\mathrm{a} = (\mathrm{a_1}, \dots, \mathrm{a_2}) \in \mathrm{V_n}$, $\mathrm{c} \in \mathbb{F}_2$. Si $\mathrm{c} = 0$, entonces $\ell_{\mathrm{a,c}} (= \ell_{\mathrm{a}})$ es una función lineal.

**Definición 6:** Una transformación afín en la entrada de una función de n variables, $\mathrm{f(x)}$, es definida como la función resultante, $\mathrm{g(x)}$ producida por el reemplazamiento del vector de entrada $\mathrm{x}$ con el producto de $\mathrm{x}$ con la matriz de transformación $\mathrm{A}$, y posteriormente aplicar una traslación, $\mathrm{b}$ . Note que $\mathrm{x} \epsilon \mathrm{V_2^n}$ , $\mathrm{A}$ es una matriz binaria no singular de $\mathrm{nxn}$ y $\mathrm{b} \epsilon \mathrm{V_2^n}$. Puede ser expresado $\mathrm{g(x)} = \mathrm{f(Ax \oplus b)}$. Si $\mathrm{b} = 0$, se refiere a una transformación lineal. También una transformación afín no es invalidada por una traslación de la entrada de la forma $\mathrm{g(x)} = \mathrm{f(Ax \oplus b) \oplus d}$, $\mathrm{d} \epsilon \mathrm{V_2}$.

**Definición 7:** El peso de Hamming de un vector $\mathrm{x} \in \mathrm{V_n}$, denotado por $\mathrm{wt(x)}$, es el número de 1´s en el vector $\mathrm{x}$. La distancia de Hamming entre dos funciones $\mathrm{f, g: V_n} \longrightarrow \mathbb{F}_2$, denotada por $\mathrm{d(f, g)}$, es el número de elementos diferentes en correspondencia con las posiciones de la tabla de verdad, se define como*:*

$$d(f,g) = wt(f \oplus g) = \sum_{\boldsymbol{x} \in V_n} f(\boldsymbol{x}) \oplus g(\boldsymbol{x})$$

La correlación entre dos funciones puede ser vista como el grado de similitud entre dos funciones. Esta relación puede ser expresada en términos de la distancia hamming entre dos funciones y es numéricamente representada como un número real entre -1 y 1, conocido como el coeficiente de correlación.

**Definición 8:** La función de autocorrelación $\hat{r}_{\hat{f}}(a)$, $\alpha \in V_n$, se define como:

$$\hat{r}_{\hat{f}}(a) = \sum_{x \in V_n} \hat{f}(x) \cdot \hat{f}(x \oplus a)$$

Note que $\hat{r}_{\hat{f}}(0) = 2^n$. El valor de correlación entre dos funciones booleanas $f$ y $h$ es definido como:

$$c(f,h) = 2\Pr\big(f(x) = g(x)\big) - 1 = 1 - \frac{d(f,h)}{2^{n-1}}$$

Note que un coeficiente de correlación 0 entre $f$ y $g$ indica que estas funciones están completamente no correlacionadas, o sea, información sobre una no aporta nada al conocimiento de la otra, mientras que por otro lado un coeficiente de correlación 1 o -1 significa que están positivamente correlacionadas $(f = g)$ o negativamente correlacionadas $(f = -g)$.

**Definición 9:** La transformada de Walsh-Hadamard de una función $f$ sobre $V_n$ es la correspondencia $W(\hat{f}): V_n \longrightarrow \mathbb{R}$, definida como:

$$W(\hat{f})(w) = \sum_{x \in V_n} \hat{f}(x)(-1)^{w \cdot x} = \sum_{x \in V_n} (-1)^{f(x)}(-1)^{w \cdot x}$$

$f$ puede ser recuperada por la transformada inversa de Walsh-Hadamard:

$$\hat{f}(x) = 2^{-n} \sum_{w \in V_n} (-1)^{W(\hat{f})(w)} (-1)^{w \cdot x}$$

El espectro de Walsh-Hadamard de la función $f$ es la lista de los $2^n$ coeficientes de la transformada de Walsh-Hadamard cuando $\mathbf{w}$ varía, coeficientes que como se puede notar, toman valores en $[-2^n, 2^n]$.

A lo largo de esta tesis será utilizado tanto la notación $W(\hat{f})(\boldsymbol{w})$ como $\hat{F}(\boldsymbol{\omega})$ para designar la Transformada de Walsh-Hadamard de la función $f$ en el vector $\boldsymbol{w}$.

**Teorema 2 (Ecuación de Parseval):** Para toda función booleana de n variables se cumple que:

$$\sum_{\mathrm{u}\in \mathrm{V_n}} (\mathrm{W}(\hat{\mathrm{f}})(\mathrm{u}))^2 = 2^{2\mathrm{n}}$$

Demostración:

$$\sum_{\boldsymbol{u}\in V_n} (W(\hat{f})(\boldsymbol{u}))^2 = \sum_{\boldsymbol{u}\in V_n} \left( \sum_{\boldsymbol{x}\in V_n} (-1)^{f(\boldsymbol{x})} (-1)^{l_{\boldsymbol{u}}(\boldsymbol{x})} \sum_{\boldsymbol{x}\in V_n} (-1)^{f(\boldsymbol{x})} (-1)^{l_{\boldsymbol{u}}(\boldsymbol{x})} \right)$$

$$= \sum_{\boldsymbol{x}\in V_n} (-1)^{f(\boldsymbol{x})} (-1)^{f(\boldsymbol{x})} \sum_{\boldsymbol{u}\in V_n} (-1)^{l_{\boldsymbol{u}}(\boldsymbol{x})} (-1)^{l_{\boldsymbol{u}}(\boldsymbol{x})}$$

$$= 2^n 2^n$$

$$= 2^{2n}$$

Este valor es constante para todas las funciones booleanas de n variables. Si una función es booleana, los valores en el espectro de Walsh deben satisfacer la ecuación de Parseval. Aún si la función satisface la ecuación de Parseval no es necesariamente booleana.

**Definición 10:** Se denota la concatenación de dos funciones booleanas de $(n-1)$ variables, $f(\boldsymbol{x})$ y $g(\boldsymbol{x})$, para formar una función booleana de $n$ variables, $h(\boldsymbol{x})$, por $h(\boldsymbol{x}) = f(\boldsymbol{x})\|g(\boldsymbol{x})$. En términos de la forma algebraica normal, la concatenación se puede expresar como:

$$h(x_1, \dots, x_n) = f(\boldsymbol{x}) \oplus x_n (f(\boldsymbol{x}) \oplus g(\boldsymbol{x}))$$

La Transformada de Walsh-Hadamard de $h(\boldsymbol{x})$, $W(\hat{h})(\boldsymbol{w})$, se puede encontrar mediante el siguiente proceso:

$$W(\hat{h})(\boldsymbol{w}) = W(\hat{f})(\boldsymbol{w}) + W(\hat{g})(\boldsymbol{w}) \; para \; \boldsymbol{w} \in \{0, \dots, 2^n - 1\}$$

$$W(\hat{h})(\boldsymbol{w} + 2^{n-1}) = W(\hat{f})(\boldsymbol{w}) - W(\hat{g})(\boldsymbol{w}) \; para \; \boldsymbol{w} \in \{0, \dots, 2^n - 1\}$$

## 1.2 Propiedades criptográficas de las funciones booleanas

Las propiedades criptográficas más relevantes de esta tesis son definidas en esta sección.

**Definición 11:** Una función booleana se llama balanceada si su peso de Hamming es exactamente $2^{n-1}$. Similarmente se define una función no balanceada como aquella cuyo peso de Hamming es diferente de $2^{n-1}$.

La importancia de la propiedad de balance radica en el hecho de que mientras mayor sea la desviación del valor $2^{n-1}$, mayor es la posibilidad de obtener una aproximación lineal de la función, lo cual representa una debilidad en términos de criptoanálisis.

**Teorema 3:** Una función booleana $f$ es balanceada si y solo si $W(\hat{f})(0) = 0$.

Demostración: La demostración de este teorema sale directamente de la definición de Transformada de Walsh-Hadamard (Linda Burnett, 2005)

Note que el grado algebraico de una función booleana balanceada no excede $n - 1$, como evidencia el teorema 1.

**Definición 12:** La no linealidad de una función $f$, denotada por $\mathcal{N}_f$, se define como:

$$\mathcal{N}_f = \min_{\phi \in \mathcal{A}_n} d(f, \phi)$$

donde $\mathcal{A}_n$ es la clase de todas las funciones afines sobre $V_n$.

Una de las propiedades criptográficas más importantes de una función booleana es la no linealidad. Como las funciones afines son conocidas como criptográficamente débiles, mientras más grande sea la distancia de $f$ con las funciones afines, más resistente será la función a los ataques criptoanalíticos (Linda Burnett, 2005)

**Teorema 4:** La no linealidad de $f$ se determina por la Transformada de Walsh-Hadamard de $f$, y es:

$$\mathcal{N}_f = 2^{n-1} - \frac{1}{2} \max_{u \in V_n} \left| W(\hat{f})(u) \right|$$

Demostración: Se obtiene de la observación sobre la Transformada de Walsh-Hadamard (Linda Burnett, 2005).

**Lema 1:** Sean $f(x)$ y $g(x)$ funciones booleanas de $n$ variables con los vectores de la Transformada de Walsh-Hadamard $W(\hat{f})(w)$ y $W(\hat{g})(w)$ respectivamente. Sea:

$$g(x) = f(i) \oplus 1 \text{ para algún } i \in \{0,1,\dots,2^n - 1\}$$

$$g(x) = f(x)\ \{\forall x | x = (0,1,\dots,2^n - 1)\} - \{i\}$$

Entonces $W(\hat{g})(w) = W(\hat{f})(w) + \Delta_{W(\hat{f})}$ donde $\Delta_{W(\hat{f})} \in \{-2, +2\}$.

Demostración:

$$W(\hat{g})(w) = \sum_{x \in \{0,2,..,2^n - 1\}} (-1)^{g(x)} (-1)^{l_w(x)}$$

$$= \sum_{x \neq i} (-1)^{f(x)} (-1)^{l_w(x)} + (-1)^{f(i)} (-1)^{l_w(i)}$$

$$= \sum_{x \in \{0,2,\dots,2^n - 1\}} (-1)^{f(x)} (-1)^{l_w(x)} + \Delta_{W(\hat{f})}$$

$$= W(\hat{f})(w) \pm 2$$

Donde $\Delta_{W(\hat{f})} = 2(-1)^{f(i)} (-1)^{l_w(i)}$ (Linda Burnett, 2005)

**Corolario 1:** Sean $f(x)$ y $g(x)$ funciones booleanas de $n$ variables con no linealidad $\mathcal{N}_f$ y $\mathcal{N}_g$ respectivamente. Sea:

$$g(x) = f(i) \oplus 1 \text{ para algún } i \in \{0,1,\dots,2^n - 1\}$$

$$g(x) = f(x)\ \{\forall x | x = (0,1,\dots,2^n - 1)\} - \{i\}$$

Entonces

$$\mathcal{N}_g = \mathcal{N}_f \pm 1$$

Demostración: La demostración de este teorema sale directamente de la definición de no linealidad y la aplicación del Lema 1.

La no linealidad de una función booleana de $n$ -variables es invariante sobre las transformaciones afines (Linda Burnett, 2005).

El concepto de *Criterio de Avalancha Global* (GAC, siglas en inglés) fue propuesto inicialmente por Zhang y Zheng en (Zheng & Zhang, 2000), donde las características generales de avalancha de una función booleana son evaluadas con el uso de su indicador absoluto y el indicador de suma de cuadrados.

**Definición 13:** La función de autocorrelación $\hat{r}_{\hat{f}}(a)$, $\alpha \in V_n$, se define como:

$$\hat{r}_{\hat{f}}(\boldsymbol{a}) = \sum_{\boldsymbol{x} \in V_n} \hat{f}(\boldsymbol{x}) \cdot \hat{f}(\boldsymbol{x} \oplus \boldsymbol{a})$$

**Corolario 2:** Sea $AC_{max}$ el indicador absoluto (o máximo valor absoluto de autocorrelación) derivado de la función de autocorrelación $\hat{r}(\alpha)$. Entonces:

$$AC_{max} = \max_{\alpha \in V_n} |\hat{r}(\alpha)|$$

(Linda Burnett, 2005)

**Corolario 3:** Sea $\sigma$ el indicador de suma de cuadrados, también derivado de la función de autocorrelación $\hat{r}(\boldsymbol{\alpha})$. Entonces:

$$\sigma = \sum_{\boldsymbol{\alpha}} \hat{r}^2(\boldsymbol{\alpha})$$

con $\boldsymbol{\alpha} \in V_n$ (Linda Burnett, 2005).

**Definición 14:** Una función booleana $f(x)$ en $n$ variables satisface el Criterio Estricto de Avalancha (SAC, siglas en inglés) si al cambiar uno de los $n$ bits en la entrada $x$, la función nueva cambia su salida para exactamente la mitad de los vectores de entrada con respecto a la función original, o sea, si para todo $s \in V_n$ con $wt(s) = 1$ se cumple que:

$$\sum_{x \in V_n} f(x) \oplus f(x \oplus s) = 2^{n-1}$$

Alternativamente, una función booleana satisface SAC si la función de autocorrelación de $\hat{f}(\boldsymbol{x}), \hat{r}_{\hat{f}}(\boldsymbol{a})$ contiene todos los valores de cero en las posiciones $s$ donde $hw(s) = 1$.

El hecho de que una función satisfaga el SAC significa que pequeños cambios en la entrada de la función conllevan a grandes cambios en el vector salida (efecto avalancha), de hecho contribuye a un gran cambio de forma uniforme, lo cual constituye una propiedad criptográfica muy útil, ya que hace difícil inferir la entrada a partir de la salida (Cusick & Stănică, 2009).

**Teorema 5:** Una función booleana $\mathrm{f}: \mathrm{V_n} \longrightarrow \mathbb{F}_2$ satisface el SAC sí y solo sí la función $\mathrm{f}(\mathbf{x}) \oplus \mathrm{f}(\mathbf{x} \oplus \mathbf{a})$ es balanceada $\forall\, \mathbf{a} \in \mathrm{V_n}$ con $\mathrm{wt}(\mathbf{a}) = 1$.

Demostración*:* Este teorema se desprende directamente de la definición de Criterio Estricto de Avalancha (Cusick & Stănică, 2009).

**Definición 15:** Una función booleana $\mathrm{f}(\mathrm{x})$ en $\mathrm{n}$ variables satisface el Criterio de Propagación de grado $\mathrm{k}$ ($\mathrm{PC}(\mathrm{k})$, siglas en inglés) si cambiando cualesquiera $\mathrm{i}$ ($1 \leq \mathrm{i} \leq \mathrm{k}$) de los $\mathrm{n}$ bits en la entrada de $\mathrm{x}$, la función nueva cambia su salida para exactamente la mitad de los vectores de entrada con respecto a la función original, o sea, si para todo $\mathrm{s} \in \mathrm{V_n}$ con $1 \leq \mathrm{wt}(\mathrm{s}) \leq \mathrm{k}$ se cumple que:

$$\sum_{\mathrm{x} \in \mathrm{V_n}} \mathrm{f}(\mathrm{x}) \oplus \mathrm{f}(\mathrm{x} \oplus \mathrm{s}) = 2^{\mathrm{n}-1}$$

El Criterio de Propagación es una propiedad de las funciones booleanas que permite a una función obtener buena difusión, mediante el aseguramiento de la uniformidad de la salida bajo cambios en la entrada de la función, debido a la fuerte dependencia de la salida de todos los bits en la entrada. Sistemas de cifrado que emplean funciones booleanas con buenas características de avalancha disminuyen la vulnerabilidad a ataques diferenciales (Linda Burnett, 2005).

Nótese que el Criterio de Propagación de grado 1 ($PC(k)$) es equivalente al Criterio Estricto de Avalancha (SAC), (Linda Burnett, 2005).

Como se expresó anteriormente el Criterio de Avalancha Global se puede evaluar usando el indicador absoluto, $AC_{max}$, y el indicador de suma de

cuadrados, $\sigma$. Pequeños valores de cada una de las dos características de GAC indica mejores propiedades de avalancha de la función. Mientras que por otro lado la función de autocorrelación constituye un medio útil para medir las dos propiedades, indicador absoluto e indicador de suma de cuadrados, con respecto a las que el proceso de optimización está dirigido con el objetivo de mejorar las propiedades de avalancha de las funciones (Linda Burnett, 2005).

**Definición 16:** Una función booleana $f(x)$ en $n$ variables, es Inmune de Correlación de orden $m$, si es estadísticamente independiente del subconjunto $m$ variables de entrada, donde $1 \leq m \leq n$.

Alternativamente, el orden de Inmunidad de Correlación de una función se puede determinar usando la relación existente entre la Transformada de Walsh-Hadamard y el peso de Hamming de las entradas (Cusick & Stănică, 2009).

**Definición 17:** Una función booleana $f(x)$ en $n$ variables, es Inmune de Correlación de orden $m$, denotado por $CI(m)$, si para todo $\omega$ tal que $1 \leq wt(\omega) \leq m$, se cumple que $\hat{F}(\omega) = 0$.

**Teorema 6 (Desigualdad de Siegenthaler):** Sea una función booleana $f(x)$ de $n$ variables con grado algebraico $deg(f)$ y orden de Inmunidad de Correlación $m$, entonces:

$$n \geq m + deg(f) + \epsilon \text{ donde } \epsilon = \begin{cases} 0 \text{ si la función es balanceada} \\ 1 \text{ si la función no es balanceada} \end{cases}$$

Demostración: (Thomas Siegenthaler, 1984) .

El orden de Inmunidad de Correlación de una función puede ser alterado mediante la complementación de bits en la tabla de verdad. Esta operación implica que los valores de la Transformada de Walsh-Hadamard pueden cambiar de valores diferentes de cero a cero y viceversa en posiciones donde el peso de Hamming de $\boldsymbol{\omega}$ dictan un cambio en el orden de Inmunidad de Correlación (Linda Burnett, 2005).

**Definición 18:** Una función que es balanceada e Inmune de Correlación de orden $m$ se dice que es $m$-resistente (del inglés “resilient”).

## 1.3 Funciones de Bent

Un conjunto especial de funciones booleanas exhibiendo características únicas fue descubierto por Rothaus en (Rothaus, 1976), a las cuales este se refirió como funciones bent; aunque en realidad ya para ese momento existían algunos trabajos sobre dichas funciones. Rothaus fue quien identificó colectivamente estas funciones y acuñó el término "bent".

**Definición 19:** Una función booleana $\mathrm{f}$ en $\mathrm{n}$ variables se dice que es bent si y solo si la Transformada de Walsh-Hadamard de los coeficientes de $\hat{\mathrm{f}}$ son todos $\pm 2^{\mathrm{n}/2}$, o sea, $\mathrm{W}(\hat{\mathrm{f}})^2$ es constante.

A continuación se exponen las características principales de las funciones bent:

- Las funciones bent solo existen en los espacios de dimensión par, hecho que se desprende de la definición (Cusick & Stănică, 2009).
- La no linealidad de una función bent será $\frac{2^{\mathrm{n}}-2^{\frac{\mathrm{n}}{2}}}{2}$ , y como la Ecuación de Parseval se debe cumplir, queda claro que este es la máxima posible (Cusick & Stănică, 2009). Esto indica que las funciones bent se encuentran a máxima distancia del conjunto de las funciones afines por lo que a menudo son referidas como "perfectamente lineales" (Linda Burnett, 2005).
- Por otro lado, como no existen valores cero en el espectro de la Transformada de Walsh-Hadamard, las funciones bent no exhiben ningún orden de Inmunidad de Correlación (Linda Burnett, 2005).
- El vector de autocorrelación de una función bent de $n$ variables ($n$ par) es de la forma $\hat{r}(\boldsymbol{a}) = (2^n, 0, \dots, 0)$. Entonces, las funciones bent satisfacen criterio de propagación de grado $n$ (Linda Burnett, 2005).
- Aunque las funciones bent exhiben propiedades criptográficas óptimas en términos de máxima no linealidad y autocorrelación (mínima) perfecta, estas funciones tienen un peso de Hamming de $2^{n-1} \pm 2^{\frac{n}{2}-1}$, o sea, nunca son balanceadas (Linda Burnett, 2005).
- Además, estas funciones poseen grado algebraico no superior a $\frac{n}{2}$ para $n > 2$ (Linda Burnett, 2005).

- Si f es una función bent entonces $f(\boldsymbol{x}) \oplus f(\boldsymbol{x} \oplus \boldsymbol{\gamma})$ es balanceada para todo $\boldsymbol{\gamma} \in V_n$ diferente de cero (Cusick & Stănică, 2009).

Las características de no ser balanceada y tener bajo grado algebraico no son deseadas en cuanto a aplicaciones criptográficas se refiere, de aquí que las funciones bent no sirvan para uso directo en aplicaciones criptográficas (Linda Burnett, 2005).

## 1.4 Relación entre las propiedades criptográficas de las funciones booleanas

Una situación ideal sería una combinación de un gran número de propiedades criptográficas deseables, tales como las discutidas anteriormente, todas con medidas adecuadas. En realidad, tales funciones no pueden existir debido a las interrelaciones entre ciertas propiedades criptográficas y reglas estrictas que limitan las características de la función booleana. En este epígrafe se comentan las interrelaciones específicas entre pares de propiedades y discutimos cómo estas propiedades se afectan entre sí, prestando especial atención a las propiedades de no linealidad, avalancha e Inmunidad de Correlación.

### 1.4.1 No linealidad y avalancha

De lo anterior expuesto podemos decir que la propiedad de no linealidad es una expresión de la diferencia entre una función y la función afín más cercana en el conjunto. La no linealidad a menudo se calcula aplicando la transformada walsh hadamard de una función booleana. La característica de avalancha es la cualidad que describe qué tan bien se propaga la entrada a lo largo de un proceso y afecta de manera uniforme. Los siguientes teoremas reflejan la relación existente entre las propiedades de no linealidad y avalancha en términos de cuanto afecta una a la otra cuando están presentes en una misma función booleana.

**Teorema 7 (Wiener-Khintchine):** Sea $\mathrm{f}(\mathrm{x})$ una función booleana de $\mathrm{n}$ variables con Transformada de Walsh-Hadamard de la tabla de verdad $\mathrm{W}(\hat{\mathrm{f}})(\mathrm{w})$ y función de autocorrelación $\hat{\mathrm{r}}_{\hat{\mathrm{f}}}(\alpha)$. Entonces:

$$\sum_{\alpha} \hat{\mathrm{r}}_{\hat{\mathrm{f}}}(\alpha)\hat{\mathrm{I}}_{\alpha}(\mathrm{w}) = (\mathrm{W}(\hat{\mathrm{f}})(\mathrm{w}))^2$$

donde $\hat{I}_{\alpha}(w)$ es la forma polar de la función lineal de $w$ definida por $\alpha$.

**Demostración**: (Linda Burnett, 2005), para la demostración.

Se sabe que si el indicador de suma de cuadrados, $\sigma$, es grande, entonces $\hat{r}(\boldsymbol{\alpha})$ tendrá valores de gran magnitud. Por tanto, de este teorema se puede deducir que si $\sigma$ es grande, es muy probable que la no linealidad de la función sea baja. Por otro lado si $\sigma$ es pequeño, $\mathcal{N}_f$ puede ser alto (Linda Burnett, 2005).

**Teorema 8 (Zheng,Zhang):** Sea $f(x)$ una función booleana de $n$ variables que satisface PC(k). Entonces:

i) $\mathcal{N}_f \geq 2^{n-1} - 2^{n-1-\frac{k}{2}}$

ii) $\mathcal{N}_f = 2^{n-1} - 2^{n-1-\frac{k}{2}}$ sí y solo sí:

a) $n$ es impar, $k = n - 1$, y $f(x)$ es de la forma:

$$f(x) = g(x_1 \oplus x_n, \dots, x_{n-1} \oplus x_n) \oplus h(x_1, \dots, x_n)$$

Donde g es una función bent de $n - 1$ variables y h es una función afín de $n$ variables; o

b) $n$ es par, $k = n$ y $f(x)$ es una función bent.

Demostración: (Zheng & Zhang, 2003).

Este teorema demuestra que mientras más grande sea el grado de Criterio de Propagación, $k$, más pequeño será la no linealidad mínima de la función, y por lo tanto alta no linealidad es posible (Linda Burnett, 2005).

**Teorema 9:** Sea $f(x)$ una función booleana de $n$ variables con función de autocorrelación $\hat{r}_{\hat{f}}(\alpha)$ y no linealidad $\mathcal{N}_f$. Entonces $\mathcal{N}_f$ satisface que:

$$\mathcal{N}_f \leq 2^{n-1} - \frac{1}{2}\sqrt{2^n + AC_{max}} \qquad \text{donde } AC_{max} = \max_{\alpha}\left|\hat{r}_{\hat{f}}(\alpha)\right|.$$

**Demostración**: (Zhang & Zheng, 1996) para la demostración.

De este teorema se tiene que para $n$ grande, la no linealidad de una función se puede estimar usando información de la función de autocorrelación (Linda Burnett, 2005).

**Teorema 10:** Sea $\mathrm{f(x)}$ una función booleana de $\mathrm{n}$ variables con indicador de suma de cuadrados, $\sigma(\hat{\mathrm{f}})$, y máximo valor absoluto de la Transformada de Walsh-Hadamard $\mathrm{W}(\hat{\mathrm{f}})_{\max}$. Entonces:

$$\sigma(\hat{\mathrm{f}}) \leq 2^{\mathrm{n}}(\mathrm{W}(\hat{\mathrm{f}})_{\max})^2$$

Demostración: (Canteaut, Carlet, Charpin, & Fontaine, 2000) para la demostración.

Se observa que mientras mayor sea la no linealidad de la función (o sea, menor $W(\hat{f})_{max}$), menor será la cota superior del indicador de la suma de cuadrados de la función (Linda Burnett, 2005).

Estos cuatro teoremas brindan fuerte evidencia de que existe una relación entre las propiedades criptográficas de no linealidad y avalancha. No linealidad y avalancha se complementan la una a la otra en una función, o sea, optimizando una se mejora la otra.

### 1.4.2 No linealidad e inmunidad de correlación.

La relación entre las propiedades criptográficas de no linealidad e Inmunidad de Correlación se puede estudiar a través de la Ecuación de Parseval. Para obtener una Inmunidad de Correlación $m$, se tiene que cumplir que $\forall \boldsymbol{\omega} \in V_n$ con $1 \leq wt(\boldsymbol{\omega}) \leq m,\ W(\hat{f})(\boldsymbol{\omega}) = 0$. Entonces, mientras mayor sea el orden de Inmunidad de Correlación, más posiciones habrá en el espectro de Walsh-Hadamard iguales a cero y para que se cumpla la Ecuación de Parseval, será necesario que los valores diferentes de cero en el espectro sean grandes, lo cual resulta en baja no linealidad. En sentido opuesto, mientras más alta sea la no linealidad, más pequeña debe ser la magnitud de los valores diferentes de cero en el espectro y por lo tanto debe haber mayor cantidad de elementos diferentes de cero en el mismo, esto último implica que es necesario que el orden de Inmunidad de Correlación sea pequeño (Linda Burnett, 2005).

Mediante la desigualdad de Siegenthaler se puede hacer un análisis de la relación existente entre el grado algebraico de la función y su orden de Inmunidad de Correlación. Dado que la dimensión $n$ del espacio de definición de la función booleana permanece fija, entonces un alto orden de Inmunidad de Correlación implica un bajo grado algebraico. Inversamente, un alto grado

algebraico implica un bajo orden de Inmunidad de Correlación (Linda Burnett, 2005).

Una cota superior para la no linealidad de una función $m$-resistente fue propuesto independientemente en (Zheng & Zhang, 2001), (Y. V. Tarannikov, 2000), (Y. Tarannikov & Kirienko, 2001) y (Sarkar & Maitra, 2000).

**Teorema 11:** Sea $f(x)$ una función $m$-resilente de $n$ variables con $0 \leq m \leq n-2$ y sea $deg(f)$ el grado algebraico de la función $f(x)$. Entonces para cada función afin de $n$- variables, $\alpha$ , $d(f,\alpha)$ es divisible por $2^{m+1+\frac{n-m-2}{deg(f)}}$

***Demostración*:** (Carlet, 2002).

De este teorema obtenemos información acerca de la divisibilidad de los valores posibles de no linealidad de $f(x)$, como un valor dependiente del número de variables de la función, el orden de inmunidad de correlación y el grado algebraico.

**Teorema 12:** Sea $f(x)$ una función booleana $m$-resistente de $n$ variables con $0 \leq m \leq n-2$, y no linealidad $\mathcal{N}_f$. Entonces $\mathcal{N}_f \leq 2^{n-1} - 2^{m+1}$.

De este teorema y los comentarios anteriores se puede observar que no linealidad e Inmunidad de Correlación son propiedades opuestas. Si se optimiza no linealidad se obtiene como consecuencia un bajo orden de Inmunidad de Correlación. Mientras que por otro lado si se asegura un alto grado de Inmunidad de Correlación disminuye el máximo grado de no linealidad alcanzable (Linda Burnett, 2005).

### 1.4.3. Inmunidad de Correlación y avalancha.

A continuación se expone brevemente la relación existente entre la propiedad de Inmunidad de Correlación y avalancha, específicamente en términos de Criterio de Propagación.

**Teorema 13:** Sea $f(x)$ una función booleana balanceada de $n$ variables y grado algebraico $r$ e Inmune de Correlación de orden $m$. Entonces $r \leq n-m-1$.

**Teorema 14:** Sea $f(x)$ una función booleana $m$-resistente de $n$ variables que satisface $PC(k)$. Entonces $m+k \leq n-2$.

Demostración*:* (Zheng & Zhang, 2003) para la demostración.

De estos teoremas se puede establecer que para un $n$ fijo, mientras más grande sea el orden de resistencia, menor será el grado de Criterio de Propagación de una función booleana. En sentido contrario, mientras mayor sea el grado del Criterio de Propagación, menor será el orden de resistencia (Linda Burnett, 2005).

La relación entre las propiedades criptográficas de Inmunidad de Correlación y avalancha se ha demostrado que son conflictivas, cuando se trata de que estas estén presentes en una misma función booleana, lo que constituye una optimización favorable para una, contribuye de forma negativa a la otra.

## 1.5 Conclusiones

De la teoría expuesta en este capítulo está clara la necesidad de crear métodos para la obtención de funciones booleanas fuertes satisfaciendo determinadas propiedades criptográficas con el objetivo de reducir la vulnerabilidad de los sistemas de cifrado. En este capítulo se ha introducido la teoría de funciones booleanas necesaria para el desarrollo de esta tesis. Particularmente se han presentado una serie de definiciones y teoremas necesarios para la comprensión del tema que se está desarrollando. Además, se ha estudiado las relaciones entre algunas de las propiedades criptográficas de las funciones booleanas.

# Capítulo 2: Métodos heurísticos

En este capítulo se presentan métodos heurísticos relacionados con la obtención de funciones booleanas con fuertes propiedades criptográficas, así como otro método que no ha sido utilizado directamente en la obtención de funciones booleanas pero sí en otras áreas de la criptografía. Se incluye un análisis sobre el proceso para desarrollar dichos algoritmos además de métodos híbridos entre pares de estos algoritmos y comparaciones que se han hecho entre algunos de ellos en otros tipos de problemas.

## 2.1 Búsqueda Tabú

La Búsqueda Tabú (TS) es un poderoso algoritmo que se ha aplicado con gran éxito a muchos problemas combinatorios difíciles. Esta heurística realiza búsquedas múltiples entre diferentes soluciones y almacena las mejores soluciones en una memoria adaptable.(Kaddouri & Omary, 2017)

Fue propuesta por F. Glover en 1986. El algoritmo se llama Tabú porque hay prohibición de recomenzar desde las soluciones visitadas recientemente. Desde entonces, el método se ha vuelto muy popular, gracias a sus éxitos para resolver diversos problemas. Este algoritmo introduce una noción de memoria en la estrategia de exploración del espacio de búsqueda. La Búsqueda Tabú usa procedimientos iterativos locales o de vecindario para pasar de la solución x a una solución x' (en las proximidades de x) hasta que se cumplan las condiciones de parada.

### Principio de Búsqueda Tabú

El principio de la Búsqueda Tabú se basa en un método para moverse en el espacio de las soluciones, buscando continuamente mejorar la mejor solución actual y almacenando en la memoria la lista de movimientos previos, guiando así la investigación fuera de las zonas previas visitadas.

La idea básica está inspirada en las técnicas de investigación utilizadas en inteligencia artificial. Esto es para mantener el seguimiento del camino pasado del proceso de investigación en uno o más recuerdos y para usar esta información para orientar el desarrollo futuro. En la práctica, no se memoriza todo el desplazamiento (es muy costoso en la memoria), pero se impide el acceso a algunas soluciones durante un cierto número de iteraciones.

La vecindad de una solución se define por una transformación elemental (movimiento) que permite el cambio de una solución a otra solución cercana con una ligera modificación de la estructura de la solución.

**La Búsqueda Tabú se basa en:**

- El uso de estructuras de memoria flexibles (a corto, mediano y largo plazo) que permiten la exploración completa de los criterios de evaluación y el historial de búsqueda.
- Un mecanismo de control basado en alternar entre las condiciones que restringen (tabú de restricción) y las que liberan (criterio de aspiración) el proceso de búsqueda.
- La incorporación de las estrategias de intensificación y diversificación de la búsqueda:
    - La estrategia de intensificación utiliza la memoria de mediano plazo y sirve para fortalecer la búsqueda en las regiones de las mejores soluciones encontradas recientemente.
    - La estrategia de diversificación utiliza la memoria a largo plazo y sirve para buscar en nuevas regiones (Kaddouri & Omary, 2017).

Algoritmo de Búsqueda Tabú

El algoritmo de búsqueda de tabú se presenta a continuación:

1. Entrada: $f$ (función booleana a optimizar).

2. Inicializar parámetros: el tamaño de la lista tabú STABU, el tamaño de la lista de posibilidades consideradas en cada iteración SPOSS y el número máximo de iteraciones MAX.

3. Inicialice la lista de tabú con las funciones tomadas aleatoriamente o definidas según el interés y calcule el costo de cada función en la lista tabú.

4. Para I = 1, ..., MAX hacer:

a. Encuentra la mejor función con el costo más bajo en la lista tabú actual, KBEST.

b. Para j = 1, ..., SPOSS hacer:

i. b.1 Seleccionar dos posiciones de bit, $i$ y $j$ , donde $f(i) \neq f(j)$.

b.2 $g = GenerarMovimientoDesde(f)$

b.3 $g = KNEW$.

ii. Compruebe si KNEW ya se encuentra en la lista de posibilidades generadas para esta iteración o la lista tabú. Si es así, regrese al paso 4(b) i.

iii. Agregue KNEW a la lista de posibilidades para esta iteración.

c. De la lista de posibilidades para esta iteración, encuentre la función con el costo más bajo, PBEST.

d. En la lista de tabú, busque la función con el costo más alto, TWORST.

e. mientras el costo de PBEST es menor que el costo de TWORST:

i. Reemplace TWORST con PBEST.

ii. Encuentra el nuevo PBEST.

iii. Encuentra el nuevo TWORST.

5. Obtenga la mejor solución de la lista de tabú, KBEST (la que tiene el menor costo). (Saveetha, Arumugam, & Kiruthikadevi, 2014)

## 2.2 Recocido Simulado

El Recocido Simulado (SA) se llama así debido a su analogía con el proceso físico recocido con sólidos, en el que un sólido cristalino se calienta y luego se deja enfriar muy lentamente hasta que logre su configuración de celosía cristalina más regular posible (es decir, su estado mínimo de energía reticular), y por lo tanto está libre de defectos cristalinos. Si el horario de enfriamiento es lo suficientemente lento, la configuración final da como resultado un sólido con tal estructura estructural superior integridad. El Recocido Simulado establece la conexión entre este tipo de comportamiento termodinámico y la búsqueda de mínimos globales para un problema de optimización discreto. Además, proporciona un medio algorítmico para explotar tal conexión (Henderson, Jacobson, & Johnson, 2003).

La técnica de Recocido Simulado fue introducida por primera vez en el año 1983 por Kirkpatrick, Gelatt y Vecchi. Esta técnica fue usada inicialmente con el fin de optimizar propiedades criptográficas de funciones booleanas en por Clark y Jacob, posteriormente los mismos autores junto a Stepney, Clark y Jacob realizaron en (John A. Clark, Jacob, Stepney, Maitra, & Millan, 2002a) un algoritmo híbrido de Hill Climbing con Recocido Simulado.

Recocido Simulado es una técnica de búsqueda local que permite escapar de los óptimos locales. Desde el estado actual un movimiento en su vecindad es generado y considerado; movimientos de mejora son siempre aceptados, mientras que movimientos de empeoramiento también pueden ser aceptados de forma probabilística dependiendo de una temperatura $T$ de la búsqueda y la medida en que la solución empeorara. A cada temperatura se considera un número nuevo de movimientos, inicialmente la temperatura es alta y virtualmente todo movimiento es posible. Gradualmente la temperatura va disminuyendo y se vuelve más difícil aceptar movimientos de empeoramiento. Eventualmente el proceso se "congela" y solo movimientos de mejoría son aceptados. Si ningún movimiento es aceptado después de un tiempo, la búsqueda se detiene.

En (John A. Clark & Jacob, 2000), los autores usan el Recocido Simulado junto con una función de aptitud motivada en la Ecuación de Parseval para dirigir la búsqueda hacia áreas donde las técnicas de búsqueda tradicionales como Hill Climbing, pueden encontrar excelentes resultados.

La técnica de Recocido Simulado tiene los siguientes parámetros principales:

- La temperatura $T$.
- La razón de enfriamiento $\alpha \in (0,1)$.
- El número de movimientos $N$ considerado en cada ciclo de temperatura.
- El número CiclosMáximosFallidos de ciclos fallados consecutivamente en una misma temperatura (donde ningún movimiento es aceptado) antes de que la búsqueda se detenga.
- El número máximo $NI_{max}$ de ciclos de temperatura considerados antes de que la búsqueda se detenga.

A continuación se expone el algoritmo de Recocido Simulado expuesto en (John A. Clark & Jacob, 2000):

Algoritmo de Recocido Simulado.

**1.** Datos de entrada: $n$ (dimensión de la función booleana requerida).

**2.** Sea $T_0$ la temperatura inicial. Aumentar la temperatura hasta que el porcentaje de movimientos aceptados dentro de un ciclo interior de $n$ ensayos haya superado cierto umbral (los autores sugieren $95\%$).

**3.** Hacer $NI = 0$ (número de iteraciones), $terminado = falso$ y $CIDUA = 0$ (ciclos interiores desde última aceptación) y generar aleatoriamente una solución inicial actual $\hat{f}_{act}$.

**4.** Mientras (no $terminado$), hacer desde 4.a hasta 4.d

a. Ciclo interior: repetir $n$ veces

i. $\hat{f}_{new} = generarMovimientoDesde(\hat{f}_{act})$

ii. Calcular cambio en el costo

$$\Delta_{cost} = cost(\hat{f}_{new}) - cost(\hat{f}_{act})$$

iii. Si $\Delta_{cost} < 0$ entonces aceptar el movimiento, o sea, $\hat{f}_{act} = \hat{f}_{new}$

iv. De otra forma generar un valor $u$ de una variable aleatoria distribuida uniformemente en el intervalo $(0,1)$. Si $e^{-\Delta_{cost}/t} > u$ entonces aceptar el movimiento, de otra forma rechazarlo.

b. Si ningún movimiento ha sido aceptado en el ciclo interior más reciente, entonces:

$$CIDUA = CIDUA + 1$$

c. $T = T * \alpha$, $NI = NI + 1$

d. Si $(CIDUA > CiclosMáximosFallidos)$ o $(NI > NI_{max})$ entonces $terminado = verdadero$.

**5.** Datos de salida: $\hat{f}_{act}$ (función booleana final del proceso)

### 2.3 Algoritmo Genético

Los principios básicos de los Algoritmos Genéticos (GA) fueron establecidos por Holland (1975), con el objetivo de estudiar autómatas celulares. Son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Se basan en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes (Darwin, 1859). Los GA usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema. Cuanto mayor

sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior.
Hoy en día son una de las líneas más prometedoras en Inteligencia Artificial, y han sido utilizados en un gran número de aplicaciones. A finales de la década del 90, también se comenzó a utilizar Algoritmos Genéticos con el propósito de optimizar propiedades criptográficas de funciones booleanas en (William Millan, Clark, & Dawson, 1997) y (William Millan, Clark, & Dawson, 1998a).

La idea básica de los Algoritmos Genéticos radica en la Teoría de la Evolución por Selección Natural de Charles Darwin. Durante el proceso de selección natural los miembros de una población se aparean para producir la descendencia. En el evento pueden ocurrir mutaciones, seguido de un proceso de selección donde solo los individuos más adaptados sobreviven para convertirse en la próxima generación. Los Algoritmos Genéticos clásicos usan selección, apareamiento y mutación. Por otro lado, debe existir un mecanismo para evaluar soluciones arbitrarias, conocido como la función de aptitud.

A continuación se define cierta terminología relacionada al proceso de selección natural:

- Población: Conjunto actual de soluciones candidatas.
- Ascendencia: Par de individuos de la población elegidos para procrear mediante cierto mecanismo de reproducción.
- Descendencia: Individuo(s) resultado del proceso de apareamiento entre dos (o más) individuos de la ascendencia.
- Apareamiento: Proceso en el cual dos (o más) individuos de la ascendencia son combinados o apareados para producir un descendiente (denotado por $\mathrm{apar}(f, g)$).
- Aptitud: Medida tomada para decidir cuáles individuos de una generación sobrevivirán a la siguiente (denotada por $\mathrm{apt}(f)$).

A continuación se representa el Algoritmo Genético básico para mejorar las propiedades criptográficas de funciones booleanas.

Algoritmo Genético.

1. Datos de entrada: $n$ (dimensión de la función booleana requerida).
2. Definir $apt(f)$ como la función de aceptación de una función booleana $f$
3. Definir la función de apareamiento de dos funciones booleanas de $n$ variables, $f$ y $g$, como $apar(f, g)$.
4. Generar un conjunto aleatorio de $T$ funciones booleanas de $n$ variables, representando la población inicial, $P_i$ donde $i = 1, \dots, T$.
5. Iterar:
    a. Generar un conjunto de $\frac{T(T-1)}{2}$ funciones booleanas (descendencia), $C_i = apar(P_j, P_k)$, donde $j = 1, \dots, T; k = 1, \dots, T;\ j \neq k$.
    b. Formar el conjunto combinado ordenado $S = P \cup C$, donde $S = \left\{S_1, \dots, S_{T+\frac{T(T-1)}{2}}\right\}$ y $apt(S_l) \geq apt(S_{l+1}), l = 1, \dots, T + \frac{T(T-1)}{2} - 1$.
    c. Mantener las T mejores funciones reemplazando $P_i = S_{i_j}$ donde $i = 1, \dots, T$ y $j = 1, \dots, T$.
    d. Resetear: Opcional.
    e. Iterar hasta que el criterio de parada especificado sea alcanzado.
6. Datos de salida: $h$ (mejor función booleana de las $T$ funciones finales del algoritmo).

(Linda Burnett, 2005)

### 2.4 Hill Climbing

Muchas tareas computacionales y cognitivas implican una búsqueda a través de un gran espacio de posibles soluciones. Escalada, o búsqueda local es una estrategia para buscar tal espacio de solución

Desde finales de 1990, Hill Climbing se ha mostrado como una técnica heurística efectiva en investigaciones de naturaleza criptográfica. Se ha realizado gran cantidad de investigación en la aplicación de Hill Climbing para la optimización de funciones booleanas. La técnica básica de Hill Climbing se basa en buscar, a cada iteración, qué elementos de la función modificar de

forma tal que resulte en una mejora con respecto a los resultados anteriormente obtenidos (Linda Burnett, 2005). Al final del proceso se supone que se obtiene la mejor solución alcanzable, teniendo en cuenta por supuesto que Hill Climbing devuelve un óptimo local, aunque puede ser utilizado en el algoritmo la aplicación de una generador aleatorio de funciones booleanas, de forma tal que se pueda aplicar Hill Climbing a cada una de éstas, y el algoritmo pueda moverse en un espacio de búsqueda más grande.

Con respecto a aplicaciones criptográficas, Hill Climbing es el proceso que se refiere a la complementación de uno o más valores de la tabla de verdad que resultarán en mejoras iterativas de la propiedad criptográfica en cuestión o la función de aptitud (o función de costo), donde esta es la medida de una o varias propiedades criptográficas exhibidas por la función (Linda Burnett, 2005). “Aptitud” es usado naturalmente para problemas de maximización y “costo” para problemas de minimización (John A. Clark & Jacob, 2000). Para utilizar esta función de aptitud como criterio para decidir si aceptar o no funciones booleanas con el objetivo de ser utilizadas en la próxima iteración, Hill Climbing necesita la creación de conjuntos de mejora. Los conjuntos de mejora se definen de acuerdo a la función de aptitud que se utiliza en el proceso de Hill Climbing (Linda Burnett, 2005). Si consideramos la no linealidad como la medida a optimizar, los conjuntos de mejora correspondientes son aquellas posiciones en el espectro de Walsh-Hadamard de una función booleana que toma valores que pueden afectar la propiedad a optimizar como resultado directo de la complementación de elementos en la tabla de verdad.

Como ya se mencionó anteriormente, en (Kocak, Kurt, & ztop, 2012) y (William Millan et al., 1998a)  Millan, Clark y Dawson usan Hill Climbing para optimizar funciones booleanas. En el primer artículo se optimiza con respecto a la propiedad de no linealidad usando dos algoritmos, en el primer caso el algoritmo usa complementación de un único bit, mientras que en el segundo usa complementación de dos bits. La idea básica de estos algoritmos radica en la identificación de aquellos conjuntos en los cuáles la complementación de uno o dos bits, respectivamente, constituye una mejora en la no linealidad de la función, caracterizada por el uso de la Transformada de Walsh-Hadamard. En

el segundo artículo solo se ofrece una regla más general para la implementación del caso de complementación de dos bits.

De resultados anteriores es conocido que el cambio de uno o dos bits en la tabla de verdad de una función booleana resultan en un cambio de la Transformada de Walsh-Hadamard de la función en un valor $\{-2,2\}$ o $\{-4,0,4\}$ respectivamente. Entonces, para un cambio de dos bits en la tabla de verdad de una función booleana, los seis conjuntos de mejora que se necesitan en el proceso de Hill Climbing son:

$$\mathcal{J}_1 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = \mathrm{W}(\hat{\mathrm{f}})_{\max}\right\}$$

$$\mathcal{J}_2 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = -\mathrm{W}(\hat{\mathrm{f}})_{\max}\right\}$$

$$\mathcal{J}_3 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = \mathrm{W}(\hat{\mathrm{f}})_{\max} - 2\right\}$$

$$\mathcal{J}_4 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = -(\mathrm{W}(\hat{\mathrm{f}})_{\max} - 2)\right\}$$

$$\mathcal{J}_5 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = \mathrm{W}(\hat{\mathrm{f}})_{\max} - 4\right\}$$

$$\mathcal{J}_6 = \left\{\emptyset: \mathrm{W}(\hat{\mathrm{f}})(\omega) = -(\mathrm{W}(\hat{\mathrm{f}})_{\max} - 4)\right\}$$

donde $\mathrm{W}(\hat{\mathrm{f}})_{\max}$ representa el valor máximo absoluto en el espectro de Walsh-Hadamard.

Estos representan los únicos valores que pueden afectar la no linealidad de una función booleana después de que ha tenido lugar un cambio de dos bits en la tabla de verdad de una función booleana. Sea $\mathcal{J}(\mathrm{f}) = \mathrm{U}_{\mathrm{i}=1}^{6}\mathcal{J}_{\mathrm{i}}$ la unión de los conjuntos de mejora de una función booleana $\mathrm{f}$.

A continuación se expone el algoritmo referido como Método de Hill Climbing.

Algoritmo de Hill Climbing.

1. Datos de entrada: $\mathrm{n}$ (dimensión de la función booleana requerida).
2. Definir $\mathrm{apt}(\mathrm{f})$ como la función de aceptación de una función booleana $\mathrm{f}$
3. Generar una función booleana aleatoria de $\mathrm{n}$ variables.
4. Iterar:

a. Construir conjuntos de mejora $\mathcal{J}(f)$ y calcular $\mathrm{apt}(f)$.
b. Seleccionar dos posiciones de bit, $i$ y $j$, donde $f(i) \neq f(j)$
c. Construir función candidata $g(x) = f(x)$, donde $x = 0, \dots, i-1, i+1, \dots, j-1, j+1, \dots, 2^n - 1$; y $g(i) = f(i) \oplus 1$, $g(j) = f(j) \oplus 1$.
d. Construir conjuntos de mejora $\mathcal{J}(g)$ y calcular $\mathrm{apt}(g)$. Si $\forall \emptyset \in \mathcal{J}(g)$, $\left|W(\hat{f})(\emptyset)\right| < W(\hat{f})_{max}$, entonces $f = g$.
e. Si el ciclo alcanza el número máximo de iteraciones sin hacer mejoras, terminar.

**5.** Datos de salida: $g$ (función booleana final), $\mathrm{apt}(g)$ óptimo alcanzado en la función de aptitud.

### 2.5 Métodos híbridos

La aplicación del Algoritmo Genético a un problema a menudo va seguida de una búsqueda local. Glover y Laguna (Reeves, 1993) indican varias maneras en que las ideas de Búsqueda Tabú podrían incorporarse en algoritmos genéticos. La noción de hibridación es poderosa y ha sido rápidamente adoptada para técnicas emergentes.

En GA aplicados a problemas de ingeniería (en este caso, la optimización de circuitos lógicos) la función de aptitud física suele ser compleja y la evaluación de la aptitud requiere mucho tiempo. El tiempo de ejecución por lo tanto, es una consideración importante cuando se diseña un GA para la optimización, por lo tanto, una tabla de búsqueda para la evaluación de la aptitud es deseable. En este artículo (Miller & Thomson, 1996) se sugiere un número de algoritmos híbridos que combinan una GA con una técnica de búsqueda de vecindario (TABU) para proporcionar este rendimiento y fiabilidad. Se introdujo una técnica de búsqueda de vecindario (Búsqueda de Tabú) que podría combinarse para reemplazar el GA cuando este punto de cercana convergencia ha sido alcanzado (Miller & Thomson, 1996).

La aplicación conjunta de Algoritmos Genéticos y el método de Hill Climbing con fines criptográficos fue realizada por primera vez en (William Millan et al., 1998a). Los autores de este artículo aplicaron un método combinado a varias poblaciones de funciones booleanas generadas aleatoriamente con el objetivo de mejorar ciertas propiedades criptográficas. En el mismo artículo se demostró

experimentalmente la ventaja de aplicar un método combinado de Algoritmos Genéticos y Hill Climbing para la optimización de funciones booleanas, en vez de aplicarlos independientemente. La combinación de estas dos técnicas se llevó a cabo mediante la aplicación del Algoritmo de Hill Climbing a cada una de las $\frac{T(T-1)}{2}$ funciones booleanas obtenidas en el paso 5. a. del Algoritmo Genético mediante el apareamiento de las $T$ funciones booleanas generadas inicialmente tomadas dos a dos. Esto fue realizado anterior a la medición de la aptitud de cada uno de los nuevos individuos y la formación del conjunto ordenado combinado.

Por otro lado la aplicación conjunta de Recocido Simulado junto a Hill Climbing se hizo por primera vez en (John A. Clark & Jacob, 2000). La idea del algoritmo expuesto por los autores en este artículo está en tomar la función final obtenida $\hat{f}_{act}$ al aplicar el algoritmo de Recocido Simulado expresado anteriormente, que de por sí debe tener un bajo valor de la función de costo, y aplicar a partir de esta el algoritmo de Hill Climbing para alcanzar la solución final del algoritmo híbrido $\hat{f}_{fin}$ que es localmente óptima con respecto a la no linealidad, ya que se ha minimizado el valor de $W(\hat{f})_{max}$(Keily Alejandro Vicente, 2016).

Es posible utilizar heurísticas específicas del problema para determinar una buena solución de partida para las ejecuciones de recocido. Entonces se adoptan temperaturas de inicio más bajas para evitar los beneficios de una buena solución inicial destruida por fluctuaciones aleatorias excesivas. Otro enfoque es incorporar el recocido en una heurística de construcción que funciona construyendo sobre una solución parcial previa. Finalmente, la incorporación de una heurística de búsqueda local (como escalador de colinas) después de aplicar el recocido es muy común.

En este artículo (Sigauke & Talukder, 2003), se simuló Recocido y algoritmos de Búsqueda Tabú para el problema de enrutamiento del vehículo. Se desarrollan métodos aproximados basados en descendencia, hibridación simulada híbrida / Búsqueda Tabú, y algoritmos de Búsqueda Tabú y se investigan diferentes estrategias de búsqueda. Se implementa una estructura de datos especial para el algoritmo de Búsqueda Tabú que ha reducido notablemente el tiempo de cálculo en más del 50%. Además, los elementos de

la Búsqueda Tabú se han incorporado en el Recocido Simulado para manipular la temperatura

En (Tao Zhang, 2012) se hicieron algunos experimentos con dos estrategias de optimización (GA, SA) de construcción de funciones booleanas y se hizo una comparación simple entre estos dos métodos. La estrategia basada en el Algoritmo Genético tiene una fácil implementación y menos restricciones que la basada en algoritmo de Recocido Simulado. Sin embargo, método basado en GA parece ser más sensible a los parámetros iniciales que el otro, que siempre puede obtener una solución óptima global teóricamente.

Estos algoritmos (TS, SA y GA) tienen muchas similitudes, pero también poseen características distintivas, principalmente en sus estrategias para buscar el espacio de solución. Las tres heurísticas se aplican en el mismo problema de optimización y se comparan. De las tres heurísticas experimentadas en este trabajo (Youssef, Sait, & Adiche, 2001), TS exhibió el mejor rendimiento con respecto a calidad de la solución, así como la calidad del espacio solución buscado. Además, con respecto a la complejidad de implementación y ajuste de los parámetros de algoritmos, GA requiere el mayor esfuerzo. SA y TS requirieron esfuerzos similares y menores que GA.

Técnicas como el Recocido Simulado, los Algoritmos Genéticos y la Búsqueda Tabú no necesitan ser utilizados de forma aislada. De hecho, estas y otras técnicas a menudo se usan en combinación, o bien las ideas que surgen en una técnica se toman prestadas y se adaptan para su uso en otra.

## 2.6 Conclusiones

En este capítulo se ha mostrado la idea de métodos heurísticos existentes en el tema de la obtención de funciones booleanas con propiedades criptográficas fuertes. Esencialmente se han expuesto cuatro de los métodos heurísticos existentes en el área de estudio vigente: Búsqueda Tabú, Recocido Simulado, Algoritmos Genéticos y Hill Climbing; así como métodos híbridos combinando algunos de estos, demostrando los magníficos resultados que se pueden llegar a encontrar cuando se explotan las virtudes de estas métodos heurísticos en conjunto. El método de Búsqueda Tabú no ha sido empleado anteriormente en la optimización de funciones booleanas criptográficamente válidas pero sí se ha

utilizado en otras áreas de la criptografía; en este trabajo de investigación se utilizará con estos fines.

También se mostraron diversas aplicaciones de estos métodos en otras áreas así como algunas comparaciones entre ellos en dependencia del tipo de problema en que se esté trabajando.

# Capítulo 3: Nuevo algoritmo para la optimización de propiedades criptográficas de funciones booleanas

En este capítulo se desarrolla un nuevo algoritmo heurístico para la obtención de funciones booleanas con fuertes propiedades criptográficas. El algoritmo utiliza las dos técnicas fundamentales de construcción de funciones booleanas criptográficamente válidas: técnicas algebraicas y métodos heurísticos. Mediante la combinación de ambas técnicas se pueden alcanzar fuertes resultados en el ámbito de optimización de propiedades múltiples. La idea básica del algoritmo propuesto está en partir de una función suficientemente buena, aplicarle los métodos metaheurísticos clásicos de Recocido Simulado y Búsqueda Tabú de forma tal que se añadan nuevas propiedades a la función y se optimicen las ya existentes.

## 3.1 Propiedades deseadas para obtener funciones booleanas fuertes

Alta no linealidad es una de las propiedades más importantes requeridas, no solo para funciones booleanas criptográficamente fuertes, sino también para la seguridad de los sistemas de cifrado que incorporan esta componente; es la propiedad fundamental para proveer confusión a dicho sistema de cifrado. Ataques criptoanalíticos como Criptoanálisis Lineal (Wu & Steinbach, 2009), y Mejor Aproximación Afín (DING, 2011), son ejemplos de ataques que explotan no linealidades modestas.

Balance es una propiedad fundamental que las funciones booleanas deben satisfacer con el objetivo de evitar dependencia estadística entre los valores de entrada y salida, además de para resistir ataques como Aproximación por Función Constante.

Otra propiedad criptográfica de extrema importancia que deben cumplir las funciones booleanas es el Criterio de Propagación. El Criterio Estricto de Avalancha (SAC) fue introducido por Webster y Tavares en (Webster & Tavares, 1986), y este concepto fue previamente generalizado al Criterio de Propagación (PC) por Bart Preneel en (Preneel et al., 1990). Estas propiedades describen el comportamiento de una función cuando cierta cantidad de bits en

una cadena de entrada son complementados, por lo tanto, están relacionadas con la propiedad de difusión del sistema criptográfico que usa esta función.

Es conocido que funciones booleanas balanceadas que satisfacen Inmunidad de Correlación de orden $m$, son referidas como funciones $m$-resistentes. Esta definición fue introducida por Siegenthaler en (Thomas Siegenthaler, 1984), está relacionada a un ataque de generadores pseudo-aleatorios que usan funciones de combinaciones, conocido como Ataque de Correlación (Wu & Steinbach, 2009): si $f$ no es $m$-resistente, entonces existe una correlación entre la salida de la función y (como máximo) m bits de la entrada. Si $m$ es pequeño, existe un método conocido como Ataque Divide y Vencerás desarrollado por Siegenthaler en (T. Siegenthaler, 1985), y que posteriormente fue perfeccionado por varios autores, que usa esta debilidad para atacar al sistema.

Por otro lado, también es sabido que funciones booleanas con fines criptográficos deben poseer un alto grado algebraico. De hecho, todos los sistemas criptográficos que usan funciones booleanas para proveer de confusión al sistema en sí pueden ser atacados si estas funciones tienen bajo grado algebraico.

### 3.2 Metaheurísticas en la optimización de propiedades criptográficas

La primera aplicación de GA a la evolución de las funciones booleanas criptográficamente adecuadas surgió en 1997. Millan et al. experimentó con GA para desarrollar funciones booleanas con alta no linealidad (William Millan et al., 1997).

En su tesis, Clark presenta varias aplicaciones de técnicas de optimización en el campo de la criptología (A. J. Clark, 1998). Una de las aplicaciones es la evolución de las funciones booleanas con alta no linealidad utilizando GA y técnicas de escalada.

Millan et al. usan GA para desarrollar funciones booleanas que tienen alta no linealidad (William Millan, Clark, & Dawson, 1998b). En conjunto con el GA, se utiliza la escalada de colinas junto con un paso de reinicio para encontrar funciones booleanas con una mayor no linealidad. Ellos experimentaron con tamaños de función booleana de hasta 12 entradas y encontraron funciones

booleanas balanceadas con no linealidad 112 y la inmunidad de correlación igual a uno.

Millan et al. usan variaciones de un método de escalada para encontrar funciones booleanas que tienen alta no linealidad y baja autocorrelación (William Millan, Clark, & Dawson, 1999).

Clark y Jacob experimentaron con la optimización en dos etapas para generar funciones booleanas (John A. Clark & Jacob, 2000). Usaron una combinación de Recocido Simulado (SA) y escalada con una función de costo motivado por el teorema de Parseval con el fin de encontrar funciones con alta no linealidad y baja autocorrelación.

Clark et al. utilizó SA para generar funciones booleanas con propiedades criptográficamente relevantes (John A. Clark, Jacob, Stepney, Maitra, & Millan, 2002b). En su trabajo, ellos consideraron la función balanceada con alta no linealidad y con la inmunidad de correlación menor o igual a dos.

Kavut y Yücel desarrollan una función de costo mejorada para una búsqueda que combina SA y escalada (Kavut & Yücel, 2003). En su enfoque, los autores pudieron encontrar algunas funciones de ocho y nueve entradas que tienen una combinación de no linealidad y valores de autocorrelación no alcanzados previamente. También experimentaron con un método de optimización de tres etapas que SA y combina dos algoritmos de escalada de colina que combina diferentes objetivos.

Clark et al. experimentaron con SA para diseñar funciones booleanas usando inversión espectral (J. A. Clark, Jacob, Maitra, & Stanica, 2003). Ellos observaron que muchas propiedades criptográficas de interés se definen en términos de los valores de transformación de Walsh-Hadamard. Sobre la base del teorema de Parseval, uno puede inferir qué valores deberían estar en un espectro de Walsh-Hadamard. Sin embargo, es imposible decir cuáles deberían ser las posiciones. Por lo tanto, al generar un espectro de Walsh-Hadamard es necesario realizar una transformación inversa para verificar que el espectro efectivamente se corresponda con una función booleana.

Burnett et al. presentaron dos métodos heurísticos donde el objetivo del primer método era generar funciones booleanas balanceadas con alta no linealidad y baja autocorrelación. El segundo método apunta a generar funciones

resistentes con alta no linealidad y grado algebraico que maximiza la desigualdad de Siegenthaler (L Burnett, Millan, Dawson, & Clark, 2004).

Millan et al. propusieron una nueva estrategia de adaptación para un algoritmo de búsqueda local para la generación de funciones booleanas con alta no linealidad (W. Millan, Fuller, & Dawson, 2003). Además, se introdujo la noción de la gráfica de las clases de equivalencia de funciones booleanas afines.

Burnett en su tesis usó tres técnicas heurísticas para desarrollar funciones booleanas. El primer método destinado a evolucionar funciones balanceadas con alta no linealidad. El segundo método se utilizó para encontrar funciones booleanas balanceadas con alta no linealidad que son inmune de correlación. El último método se utilizó para encontrar funciones equilibradas con alta no linealidad y características de propagación diferente de cero (Linda Burnett, 2005). Además, experimentó con la evolución de S-boxes.

Aguirre et al. utilizó un escalador de bits aleatorio multiobjetivo para buscar funciones booleanas balanceadas de tamaño hasta ocho entradas que tienen alta no linealidad (Aguirre, Okazaki, & Fuwa, 2007). Los resultados indican que el enfoque multi-objetivo es altamente eficiente cuando genera funciones booleanas que tienen alta no linealidad.

Izbenko et al. usaron un algoritmo de escalada modificado para transformar las funciones de bent a funciones booleanas balanceadas con alta no linealidad (Izbenko, Kovtun, & Kuznetsov, 2009).

McLaughlin y Clark, por otro lado, usaron SA para generar funciones booleanas que tienen valores óptimos de algebraico inmunidad, resistencia algebraica rápida y grado algebraico (McLaughlin & Clark, 2013). En su trabajo, experimentaron con funciones booleanas con hasta 16 entradas.

Picek, Jakobovic y Golub experimentaron con GA y GP (programación genética: especialización de GA) para encontrar funciones booleanas que poseen varias propiedades óptimas (Picek, Jakobovic, & Golub, 2013). Por lo que saben los autores, esta es la primera aplicación de GP para la evolución criptográficamente adecuada Funciones Booleanas.

Con el objetivo de encontrar los valores máximos de no linealidad de las funciones booleanas, Picek et al. experimentado con varios algoritmos evolutivos (Picek, Marchiori, Batina, & Jakobovic, 2014). Además, combinaron técnicas de optimización con construcciones algebraicas para mejorar la

búsqueda. Aunque no pudieron encontrar una función booleana equilibrada con no linealidad igual a 118, presentaron varias posibles avenidas de trabajo a seguir cuando se buscan funciones booleanas equilibradas no lineales.

Finalmente, Picek et al. investigaron varios algoritmos evolutivos para funciones booleanas con diferentes valores de la correlación de inmunidad. En el mismo documento, los autores también discuten el problema de encontrar funciones inmune de correlación con un peso Hamming mínimo, pero solo experimentaron con funciones booleanas que tienen ocho entradas(Picek, Carlet, Jakobovic, Miller, & Batina, 2015).

### 3.3 Construcción algebraica de funciones booleanas m-resilentes, con alta no linealidad y alto grado algebraico.

Sea $Q$ un operador $Q$: $\boldsymbol{\Omega}_n X \boldsymbol{\Omega}_n \rightarrow \boldsymbol{\Omega}_{n+1}$ ($\boldsymbol{\Omega}_n$ es el conjunto de las funciones booleanas de $n$-variables) definido como

Para $\mathrm{f}(\mathrm{x_n}, \ldots., \mathrm{x_1}), \mathrm{g}(\mathrm{x_n}, \ldots., \mathrm{x_1}) \in \Omega_\mathrm{n}$ ,

$$\mathrm{Q}\big(\mathrm{f}(\mathrm{x_n}, \ldots., \mathrm{x_1}), \mathrm{g}(\mathrm{x_n}, \ldots., \mathrm{x_1})\big) = \mathrm{F}(\mathrm{x_{n+1}}, \ldots., \mathrm{x_1}) = (1 \oplus \mathrm{x_{n+1}})\mathrm{f}(\mathrm{x_n}, \ldots., \mathrm{x_1}) \oplus \mathrm{x_{n+1}}\mathrm{g}(\mathrm{x_n}, \ldots., \mathrm{x_1})$$

Sea $f$ una función booleana de $n$-variables, $m$-resilente, de grado algebraico $d$ y no linealidad $x$. Definimos $F(x_{n+1}, \ldots., x_1)$ una función de $n+1$ variables como:

$$F(x_{n+1}, \ldots., x_1) = Q(f(x_n, \ldots., x_1), a \oplus f(b \oplus x_n, \ldots., b \oplus x_1),$$

Donde, $a, b \in \{0,1\}$, si $m$ es par $a \neq b$ y si $m$ es impar, $a = 1$ y $b$ puede ser $0$ o $1$.

Entonces $F(x_{n+1}, x_n \ldots., x_1)$ es $m+1$ resilente, de grado algebraico $d$ y no linealidad $2x$ (Maitra & Sarkar, 2002).

En este artículo en el que se presenta tal construcción, se considera la construcción de funciones resistentes con el máximo posible grado algebraico y alta no linealidad. Se utilizan métodos algebraicos para demostrar que con esta construcción se pueden encontrar funciones resistentes de orden superior con el máximo posible grado y muy alta no linealidad. Tales funciones tienen amplias aplicaciones para transmitir cifrado de flujo criptográfico.

### 3.4 Algoritmo Heurístico

Como se muestra a continuación se expone el algoritmo desarrollado para encontrar funciones criptográficamente válidas. Este algoritmo es utilizado para encontrar funciones booleanas $m$-resistentes ($m = 1$ o $m = 2$) con alta no linealidad, un alto grado algebraico y que satisfacen un alto orden de Criterio de Propagación. Aunque con una simple modificación en la función de costo, se puede obtener un orden mayor de funciones resistentes, lo que conocida la afectación que puede traer sobre otros criterios, en esta tesis solo se buscarán de orden 1 o 2.

La idea básica de este algoritmo es concentrar el proceso de búsqueda heurística a regiones del espacio $n-$dimensional de funciones booleanas donde se espera que se exhiban propiedades conjuntas de alta no linealidad, así como alto Criterio de Propagación. Por definición, estas regiones contienen funciones que están a grandes distancias del conjunto de las funciones booleanas afines. La necesidad de concentrar la búsqueda se hace cada vez más evidente a medida que crece la dimensión $n$ del espacio, ya que el cardinal del espacio de funciones booleanas de dimensión $n$ es $2^{2^n}$.

El primer paso de gran importancia es comenzar con una buena función inicial $F$, con propósitos de eficiencia. Esta función inicial debe exhibir ya de por sí buenas propiedades criptográficas, con el objetivo de que en pasos posteriores del algoritmo se descubran funciones booleanas con propiedades criptográficas cada vez mejores.

En el algoritmo expuesto se construye la función inicial $F$ según se definió en la sección anterior. Como ya se demostró, esta función $F$ es m+1-resistente, depende del orden de resistencia de $f$, lo que en nuestro caso tomaremos las funciones $f$ 1-resistentes, y en dependencia del grado algebraico de $f$, tendremos el de $F$, además si la no linealidad de $f$ es $x$, la de $F$ es $2x$.

Posteriormente a partir de la función $F$ se crean, iterativamente, nuevas funciones mediante el intercambio de bits $1$ y $0$, o sea moviéndose en el espacio de búsqueda, de forma tal que se aceptan aquellas funciones que mejoren la propiedad de no linealidad con respecto a la función guardada anteriormente como “máximo”, además de que mantengan un orden de

inmunidad de correlación de 1 o 2. Este proceso se realiza mediante la evaluación de las nuevas funciones en una función de costo y comparando éste con respecto a la función guardada, si existe una mejora en las propiedades de la función, se acepta la nueva; existe la posibilidad también de aceptar funciones que empeoren levemente la propiedad de no linealidad de forma estocástica, éste proceso en general se realiza mediante el uso de Recocido Simulado.

La función de costo puede ser utilizada también como herramienta para tratar de moverse en el espacio de búsqueda hacia funciones que exhiban nuevas características además de las ya existentes, siempre optimizando con respecto a las ya existentes por supuesto. Como ejemplo de esto, en el algoritmo planteado se usa la función de costo reflejada a continuación (John A. Clark et al., 2002a)):

$$cost(f) = A * \sum_{1 \le |s| \le n-4} \left|\hat{r}_f(s)\right|^R + B * \sum_{|w| \le 2} \left|\hat{F}_f(w)\right|^R + C * \max_{\omega} \left|\hat{F}_f(w)\right|$$

Esta función de costo está destinada a tratar de obtener un orden alto del criterio de propagación mediante la primera sumatoria del miembro derecho, Inmunidad de Correlación de orden $1\ o\ 2$, mediante la segunda sumatoria expresada en el miembro derecho, ya que se sabe que una función es Inmune de Correlación de orden $1$ si y sólo si $\hat{\mathrm{F}}_{\mathrm{f}}(\omega) = 0\ \forall \omega \in \mathrm{V}_{\mathrm{n}} | \mathrm{wt}(\omega) \le 1$ o Inmune de Correlación de orden $2$ si y sólo si $\hat{\mathrm{F}}_{\mathrm{f}}(\omega) = 0\ \forall \omega \in \mathrm{V}_{\mathrm{n}} | \mathrm{wt}(\omega) \le 2$ ; R se toma como un valor entre $2$ y $3$. A la vez esta función de costo intenta optimizar la función booleana en cuestión con respecto a la propiedad de no linealidad como se refleja en el tercer sumando del miembro derecho, o sea, se trata de reducir el valor máximo absoluto en el espectro de Walsh-Hadamard, usando el parámetro $C$ como la ponderación que tiene la no linealidad con respecto a la propiedad de inmunidad de correlación y el criterio de propagación $A$ y $B$ son ponderaciones al igual que $C$, respecto a las otras dos propiedades de la función de costo respectivamente.

La función que resulta de este proceso es optimizada con respecto a la no linealidad mediante el método de Búsqueda Tabú, siempre prestando especial atención a no perder la propiedad de inmunidad de correlación del orden que

alcanzó dicha función anteriormente, así como la propiedad de tener un alto criterio de propagación se va a ir mejorando a la vez que optimizamos la no linealidad ya que se había demostrado que una influye positivamente sobre la otra. Este proceso de Búsqueda Tabú no va a tener un costo computacional tan alto ya que fue empleado después del Recocido Simulado que impulsó la búsqueda de la solución. La función de costo que utilizará la Búsqueda Tabú para maximizar la no linealidad es:

$$cost(f) = max\left|\hat{F}_f(w)\right|$$

(J.A. Clark, Jacob, & Stepney, 2004)

A continuación se expone el algoritmo general para generar funciones booleanas $m$-resistentes ($m = 1$ o $m = 2$) con alta no linealidad, un alto grado algebraico y que satisfacen un alto orden de Criterio de Propagación.

Algoritmo 1: Algoritmo General.

1. Datos de entrada del algoritmo: $f$ (dimensión de las funciones booleanas requeridas), $c$ (cantidad de funciones booleanas requeridas), $n_1$ (cantidad de iteraciones del ciclo interior).
2. Construir un conjunto $G$ de $c$ funciones aleatorias $m$-resilentes, de grado algebraico $d$ y no linealidad $x$.
3. Crear un conjunto vacío $H$ para el almacenamiento de las $c$ funciones booleanas de salida del algoritmo.
4. Se aplica Algoritmo 2 a cada una de las funciones del conjunto $\mathrm{G}$.
5. Se aplica Algoritmo 3 a cada una de las funciones del conjunto $\mathrm{H}$.
6. Datos de salida: $\mathrm{H}$ (conjunto de funciones booleanas optimizadas).

Algoritmo 2: Algoritmo Recocido Simulado.

1. Datos de entrada: $f$ (función del conjunto $G$)
2. Construir la función inicial $F$ (a partir de las construcciones especificadas en el epígrafe anterior y la dimensión del espacio de búsqueda).
3. Sea $\mathrm{T}_0$ la temperatura inicial. Aumentar la temperatura hasta que el porcentaje de movimientos aceptados dentro de un ciclo interior de $\mathrm{n}$ ensayos haya superado cierto umbral (se sugiere $95\%$).

**4.** Hacer $\mathrm{NI} = 0$ (número de iteraciones), $\mathrm{terminado} = \mathrm{falso}$ y $\mathrm{CIDUA} = 0$ (ciclos interiores desde última aceptación) y tomar como solución inicial $\hat{\mathrm{f}}_{\mathrm{act}} = \mathrm{F}$.

**5.** Mientras (no $\mathrm{terminado}$), hacer desde 5.a hasta 5.d

a. Ciclo interior: repetir $\mathrm{n}_1$ veces.

i. $\hat{\mathrm{f}}_{\mathrm{new}} = \mathrm{generarMovimientoDesde}(\hat{\mathrm{f}}_{\mathrm{act}})$

ii. Calcular cambio en el costo:

$\Delta_{\mathrm{cost}} = \mathrm{cost}(\hat{\mathrm{f}}_{\mathrm{new}}) - \mathrm{cost}(\hat{\mathrm{f}}_{\mathrm{act}})$

iii. Si $\Delta_{\mathrm{cost}} < 0$ y $\forall \omega: 1 \leq \mathrm{wt}(\omega) \leq 2$ se cumple que $\hat{\mathrm{F}}(\omega) = 0$, entonces aceptar el movimiento, o sea, $\hat{\mathrm{f}}_{\mathrm{act}} = \hat{\mathrm{f}}_{\mathrm{new}}$.

iv. De otra forma generar un valor $\mathrm{u}$ de una variable aleatoria distribuida uniformemente en el intervalo $(0,1)$. Si $\mathrm{e}^{-\Delta_{\mathrm{cost}}/\mathrm{t}} > \mathrm{u}$ y y $\forall \boldsymbol{\omega}: 1 \leq wt(\boldsymbol{\omega}) \leq 2$ se cumple que $\hat{F}(\boldsymbol{\omega}) = 0$, entonces aceptar el movimiento, de otra forma rechazarlo.

b. Si ningún movimiento ha sido aceptado en el ciclo interior más reciente, entonces:

$$CIDUA = CIDUA + 1$$

c. $\mathrm{T} = \mathrm{T} * \beta$, $\mathrm{NI} = \mathrm{NI} + 1$ (donde $\beta$ es la razón de enfriamiento, se sugiere entre 0.95-0.99)

d. Si $(\mathrm{CIDUA} > \mathrm{CiclosMáximosFallidos})$ o $(\mathrm{NI} > \mathrm{n}_1)$ entonces $\mathrm{terminado} = \mathrm{verdadero}$.

**6.** Datos de salida: $\hat{\mathrm{f}}_{\mathrm{act}}$ (solución final obtenida después del proceso de iteración), se inserta en el conjunto $\mathrm{H}$ de soluciones.

Posteriormente se expone el Método de Búsqueda Tabú al cual se hace referencia en el algoritmo anterior:

Algoritmo 3: Algoritmo de Búsqueda Tabú

1. Entrada: $f$ (función booleana a optimizar).

2. Inicializar parámetros: el tamaño de la lista tabú STABU (cantidad de funciones del conjunto H), el tamaño de la lista de posibilidades consideradas en cada iteración SPOSS y el número máximo de iteraciones MAX.

3. Inicialice la lista de tabú con las funciones del conjunto H y calcule el costo de cada función en la lista tabú.

4. Para I = 1, ..., MAX hacer:

a. Encuentra la mejor función con el costo más bajo en la lista tabú actual, KBEST.

b. Para j = 1, ..., SPOSS hacer:

i. b.1 Seleccionar dos posiciones de bit, $\mathrm{i}$ y $\mathrm{j}$ , donde $\mathrm{f(i)} \neq \mathrm{f(j)}$.

b.2 $g = \mathrm{GenerarMovimientoDesde(f)}$

b.3 Si $\forall \boldsymbol{\omega}: 1 \leq wt(\boldsymbol{\omega}) \leq 2$ se cumple que $\hat{G}(\boldsymbol{\omega}) = 0$, $g = KNEW$, si no, volver a 4 (b) i.

ii. Compruebe si KNEW ya se encuentra en la lista de posibilidades generadas para esta iteración o la lista tabú. Si es así, regrese al paso 4 (b) i.

iii. Agregue KNEW a la lista de posibilidades para esta iteración.

c. De la lista de posibilidades para esta iteración, encuentre la función con el costo más bajo, PBEST.

d. En la lista de tabú, busque la función con el costo más alto, TWORST.

e. mientras el costo de PBEST es menor que el costo de TWORST:

i. Reemplace TWORST con PBEST.

ii. Encuentra el nuevo PBEST.

iii. Encuentra el nuevo TWORST.

5. Obtenga la mejor solución de la lista de tabú, KBEST (la que tiene el menor costo).

Algoritmo 4: GenerarMovimientoDesde

Datos de entrada: $\mathrm{f}$ (función booleana a la cual se le va a generar un movimiento).

1. Construir una nueva función $\mathrm{g(x)} = \mathrm{f(x)}$, donde $\mathrm{x} = 0, \ldots, \mathrm{i} - 1, \mathrm{i} + 1, \ldots, \mathrm{j} - 1, \mathrm{j} + 1, \ldots, 2^{\mathrm{n}} - 1$; y $\mathrm{g(i)} = \mathrm{f(i)} \oplus 1$, $\mathrm{g(j)} = \mathrm{f(j)} \oplus 1$.
2. Datos de salida: $\mathrm{g}$ (nueva función booleana).

En la sección anterior fue demostrado que las construcciones iniciales de funciones booleanas satisfacían ciertas propiedades, exactamente: resistencia, alta no linealidad y alto grado algebraico. La propiedad de balance nunca se

pierde a lo largo de todo el proceso de optimización ya que la forma de generar nuevas funciones establecida en el Algoritmo 4 claramente mantiene el peso de Hamming en $2^{n-1}$. Para la propiedades de no linealidad se establece cierto umbral por parte del usuario en forma de parámetros de entrada del algoritmo, de forma tal que el algoritmo se mueva en el espacio de las funciones booleanas con propiedades mejores a este mínimo pre-especificado. La propiedad de un alto criterio de propagación es alcanzada mediante el Algoritmo 2 con el uso de la función de costo previamente establecida. De experimentos numéricos con la función de costo anteriormente expresada, se tiene evidencia de que se pueden alcanzar funciones booleanas con propiedad de Inmunidad de Correlación de diversos órdenes (John A. Clark et al., 2002b); aunque en el Algoritmo 1 se limita la búsqueda solamente a funciones booleanas de Inmunidad de Correlación de Orden 1 o 2, para que la existencia de esta propiedad no influya negativamente en la optimización de la no linealidad y el Criterio de Propagación. Por último, se llegó a la conclusión de que la optimización de propiedades como no linealidad influye positivamente en el Criterio de Propagación y en el grado algebraico de la función booleana, de hecho, las funciones resultantes del proceso de búsqueda pueden poseer un grado algebraico de hasta $n - 2$ en virtud de la Desigualdad de Siegenthaler, siendo $n$ la dimensión del espacio de búsqueda.

A continuación se ofrece una comparación entre el Algoritmo 1 propuesto y el método de búsqueda desarrollado en (Keily Alejandro Vicente, 2016):

- En (Keily Alejandro Vicente, 2016) se usa como función inicial del proceso una función construida algebraicamente a partir de una función bent, con las propiedades de balance, alta no linealidad y alto Criterio de Propagación, aquí se usa una función 1-resistente, con alta no linealidad y alto grado algebraico.
- En (Keily Alejandro Vicente, 2016) se busca balance, alta no linealidad, alto Criterio de Propagación e Inmunidad de Correlación de orden 1, aquí se busca balance, alta no linealidad, alto criterio de propagación e inmunidad de correlación de orden 1 o 2, lo cual mejora ya que buscamos un orden de inmunidad de correlación más alto.

- En (Keily Alejandro Vicente, 2016) se usa como función de costo:
  - $cost(f) = \sum_{wt(\omega)\leq 1}\left|\hat{F}_f(\omega)\right|^R + A \cdot \max_{\omega}\left|\hat{F}_f(\omega)\right|$

  Mientras que aquí se usa:
  - $cost(f) = A * \sum_{1\leq|s|\leq n-4}\left|\hat{r}_f(s)\right|^R + B * \sum_{|w|\leq 2}\left|\hat{F}_f(w)\right|^R + C * \max_{\omega}\left|\hat{F}_f(w)\right|$

    Por lo que se observa, que a través de la primera función de costo se optimizan dos propiedades y a través de la segunda se optimizan tres propiedades.
- En (Keily Alejandro Vicente, 2016) se usa un método híbrido de Recocido Simulado y Hill Climbing, mientras que aquí se usa un método hibrido entre Recocido Simulado y Búsqueda Tabú, lo cual es una ventaja ya que Hill Climbing puede quedar atrapado en un óptimo local y la Búsqueda Tabú no .

### 3.5 Conclusiones

Se ha realizado una exposición de un nuevo algoritmo heurístico para la optimización de propiedades criptográficas de funciones booleanas y se ha llevado a cabo una breve discusión de trabajos científicos realizados por otros investigadores del tema.

El método de búsqueda presentado está destinado a encontrar funciones booleanas que poseen la propiedad de ser $\mathrm{m}$-resistentes ($\mathrm{m} = 1$ o $\mathrm{m} = 2$) y alta no linealidad, así como alto orden de Criterio de Propagación. La idea clave del algoritmo es partir de una función suficientemente fuerte y moverse en la vecindad de esta, siempre impidiendo que sus propiedades bajen de un umbral prefijado.

Con este nuevo algoritmo heurístico se demuestra cómo con diferentes métodos heurísticos pueden ser utilizados en conjunto para explotar las virtudes de cada uno, de forma tal que se logre un buen proceso de búsqueda a lo largo de todo el espacio. Por otro lado, se demuestra la fuerza que puede llegar a tener un algoritmo que inicia con una construcción algebraica para posteriormente modificar las propiedades de ésta con métodos heurísticos; además vemos la superioridad de este nuevo algoritmo al emplear la Búsqueda Tabú después de haber aplicado Recocido Simulado. Así podemos ver una de

las soluciones al problema de optimización de propiedades criptográficas de funciones booleanas atendiendo a múltiples criterios.

## Conclusiones

A través de la presente investigación se adquirió un conocimiento sólido de la Teoría de las Funciones Booleanas, así como de sus propiedades criptográficas y las relaciones entre éstas.

Se establecieron definiciones y teoremas pertinentes a la Teoría de las Funciones Booleanas, así como la relación entre algunas de éstas necesarias para nuestra investigación. Además se presentaron métodos heurísticos existentes que han sido aplicadas en investigaciones relacionadas con la obtención de funciones booleanas criptográficamente válidas.

Se diseñó un nuevo algoritmo heurístico híbrido entre Recocido Simulado y la Búsqueda Tabú, en el que se toma como punto de partida una función booleana construida algebraicamente de forma tal que satisfaga determinadas propiedades criptográficas suficientemente fuertes, para así mediante las métodos heurísticos optimizar las propiedades ya existentes e incluir otras nuevas, aprovechando las fortalezas de cada una de estas técnicas heurísticas.

En el desarrollo de nuestro algoritmo para obtener funciones booleanas criptográficamente fuertes; se aplicaron métodos heurísticos conocidos en conjunto con construcciones algebraicas previamente desarrolladas para optimizar propiedades criptográficas.

Con el presente trabajo se contribuye a resolver el problema de obtención de funciones booleanas criptográficamente fuertes para aumentar la seguridad de los sistemas criptográficos de cifrado.

## Recomendaciones

Se propone el desarrollo de nuevos métodos que deberían estar dirigidos a optimizar un mayor número de propiedades criptográficas que dependerán en las limitaciones de coexistencia de estas propiedades. Igualmente posee gran importancia el desarrollo de métodos capaces de efectuar búsquedas dirigidas a lo largo de grandes espacios, de forma tal que se obtengan funciones booleanas criptográficamente válidas con gran número de variables de entrada.

Además es recomendable trabajar en el área de las construcciones algebraicas así como en la creación de nuevas construcciones de funciones booleanas criptográficamente fuertes ya sea para utilizar en conjunto con métodos heurísticos o para explotar más fortalezas en esta teoría. El desarrollo de métodos para construir funciones criptográficamente válidas que son combinación de métodos heurísticos y construcciones algebraicas, es un área amplia y prometedora.

Se pueden considerar dos enfoques con esta idea. Primero, los métodos heurísticos se pueden usar en la generación de un gran número de funciones suficientemente buenas, las cuales se pueden utilizar en construcciones algebraicas para lograr múltiples funciones con propiedades óptimas, y posiblemente, sin debilidades estructurales. Otro enfoque sería, usar construcciones algebraicas para obtener funciones que se usen como el punto de partida de métodos heurísticos que dirijan la búsqueda a áreas específicas que exhiban propiedades criptográficas deseadas.

## Referencias Bibliográficas

Aguirre, H., Okazaki, H., & Fuwa, Y. (2007). An Evolutionary Multiobjective Approach to Design Highly Non-linear Boolean Functions. En Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (pp. 749–756). New York, NY, USA: ACM. https://doi.org/10.1145/1276958.1277112

Burnett, L, Millan, W., Dawson, E., & Clark, A. (2004). Simpler methods for generating better Boolean functions with good cryptographic properties, 17.

Burnett, Linda. (2005). Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography, 226.

Canteaut, A., Carlet, C., Charpin, P., & Fontaine, C. (2000). Propagation Characteristics and Correlation-Immunity of Highly Nonlinear Boolean Functions. En B. Preneel (Ed.), Advances in Cryptology — EUROCRYPT 2000 (Vol. 1807, pp. 507-522). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45539-6_36

Carlet, C. (2002). On the Coset Weight Divisibility and Nonlinearity of Resilient and Correlation-Immune Functions. En Sequences and their Applications (pp. 131-144). Springer, London. https://doi.org/10.1007/978-1-4471-0673-9_9

Clark, A. J. (1998). Optimisation Heuristics for Cryptology, 164.

Clark, J. A., Jacob, J. L., Maitra, S., & Stanica, P. (2003). Almost Boolean functions: the design of Boolean functions by spectral inversion. En The 2003 Congress on Evolutionary Computation, 2003. CEC ’03 (Vol. 3, pp. 2173-2180 Vol.3). https://doi.org/10.1109/CEC.2003.1299941

Clark, J.A., Jacob, J. L., & Stepney, S. (2004). Searching for cost functions (pp. 1517-1524). IEEE. https://doi.org/10.1109/CEC.2004.1331076

Clark, John A., & Jacob, J. L. (2000). Two-Stage Optimisation in the Design of Boolean Functions. En Information Security and Privacy (pp. 242-254). Springer, Berlin, Heidelberg. https://doi.org/10.1007/10718964_20

Clark, John A., Jacob, J. L., Stepney, S., Maitra, S., & Millan, W. (2002a). Evolving Boolean Functions Satisfying Multiple Criteria. En A. Menezes & P. Sarkar (Eds.), Progress in Cryptology — INDOCRYPT 2002 (Vol. 2551, pp. 246-259). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36231-2_20

Clark, John A., Jacob, J. L., Stepney, S., Maitra, S., & Millan, W. (2002b). Evolving Boolean Functions Satisfying Multiple Criteria. En A. Menezes & P. Sarkar (Eds.), Progress in Cryptology — INDOCRYPT 2002 (Vol. 2551, pp. 246-259). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36231-2_20

Cusick, T. W., & Stănică, P. (2009). Cryptographic Boolean functions and applications (1st ed). Amsterdam: Academic Press/Elsevier.

DING, C. (2011). The Stability Theory of Stream Ciphers. Hong Kong, 39.

Elhosary, A. M., Hamdy, N., Farag, I. A.-G., & Rohiem, A. E. (2013). State of the ART in Boolean Functions Cryptographic Assessment, 7.

Henderson, D., Jacobson, S. H., & Johnson, A. W. (2003). The Theory and Practice of Simulated Annealing. En F. Glover & G. A. Kochenberger (Eds.), Handbook of Metaheuristics (Vol. 57, pp. 287-319). Boston: Kluwer Academic Publishers. https://doi.org/10.1007/0-306-48056-5_10

Izbenko, Y., Kovtun, V., & Kuznetsov, A. (2009). The Design of Boolean Functions by Modified Hill Climbing Method. En 2009 Sixth International Conference on Information Technology: New Generations (pp. 356-361). https://doi.org/10.1109/ITNG.2009.102

Kaddouri, Z., & Omary, F. (2017). Application of the Tabu Search Algorithm to Cryptography. International Journal of Advanced Computer Science and Applications, 8(7). https://doi.org/10.14569/IJACSA.2017.080712

Kavut, S., & Yücel, M. D. (2003). Improved Cost Function in the Design of Boolean Functions Satisfying Multiple Criteria. En Progress in Cryptology - INDOCRYPT 2003 (pp. 121-134). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24582-7_9

Keily Alejandro Vicente. (2016). Optimización de funciones booleanas criptográficamente válidas mediante el uso de Metaheurísticas. Unpublished. https://doi.org/10.13140/rg.2.1.5132.9520

Kocak, O., Kurt, O., & ztop, N. O. (2012). Notes on Bent Functions in Polynomial Forms, 6.

Maitra, S., & Sarkar, P. (2002). Cryptographically signiÿcant Boolean functions with ÿve valued Walsh spectra. Theoretical Computer Science, 14.

McLaughlin, J., & Clark, J. A. (2013). Evolving balanced Boolean functions with optimal resistance to algebraic and fast algebraic attacks, maximal algebraic degree, and very high nonlinearity., 19.

Millan, W., Fuller, J., & Dawson, E. (2003). New concepts in evolutionary search for Boolean functions in cryptology. En The 2003 Congress on Evolutionary Computation, 2003. CEC ’03 (Vol. 3, pp. 2157-2164 Vol.3). https://doi.org/10.1109/CEC.2003.1299939

Millan, William, Clark, A., & Dawson, E. (1997). An effective genetic algorithm for finding highly nonlinear boolean functions. En Information and Communications Security (pp. 149-158). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0028471

Millan, William, Clark, A., & Dawson, E. (1998a). Heuristic design of cryptographically strong balanced Boolean functions. En Advances in Cryptology — EUROCRYPT'98 (pp. 489-499). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0054148

Millan, William, Clark, A., & Dawson, E. (1998b). Heuristic design of cryptographically strong balanced Boolean functions. En Advances in Cryptology — EUROCRYPT'98 (pp. 489-499). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0054148

Millan, William, Clark, A., & Dawson, E. (1999). Boolean Function Design Using Hill Climbing Methods. En Information Security and Privacy (pp. 1-11). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48970-3_1

Miller, J. F., & Thomson, P. (1996). Restricted evaluation genetic algorithms with Tabu search for optimising Boolean functions as multi-level AND-EXOR networks. En T. C. Fogarty (Ed.), Evolutionary Computing (Vol. 1143, pp. 85-101). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0032775

Picek, S., Carlet, C., Jakobovic, D., Miller, J. F., & Batina, L. (2015). Correlation immunity of Boolean functions: An evolutionary algorithms perspective. GECCO 2015 - Proceedings of the 2015 Genetic and Evolutionary Computation Conference, 1095-1102.

Picek, S., Jakobovic, D., & Golub, M. (2013). Evolving cryptographically sound boolean functions (p. 191). ACM Press. https://doi.org/10.1145/2464576.2464671

Picek, S., Jakobovic, D., Miller, J. F., Batina, L., & Cupic, M. (2016). Cryptographic Boolean functions: One output, many design criteria. Applied Soft Computing, 40, 635–653.

Picek, S., Marchiori, E., Batina, L., & Jakobovic, D. (2014). Combining Evolutionary Computation and Algebraic Constructions to Find Cryptography-Relevant Boolean Functions. En Parallel Problem Solving from Nature – PPSN XIII (pp. 822-831). Springer, Cham. https://doi.org/10.1007/978-3-319-10762-2_81

Preneel, B., Leekwijck, W. V., Linden, L. V., Govaerts, R., Vandewalle, J., & Mercierlaan, K. (1990). Propagation Characteristics of Boolean Functions. Lecture Notes in Computer Science, 12.

Reeves, C. R. (Ed.). (1995). Modern Heuristic Techniques for Combinatorial Problems. New York, NY, USA: John Wiley & Sons, Inc.

Rothaus, O. S. (1976). On “bent” functions. Journal of Combinatorial Theory, Series A, 20(3), 300-305. https://doi.org/10.1016/0097-3165(76)90024-8

Sarkar, P., & Maitra, S. (2000). Nonlinearity Bounds and Constructions of Resilient Boolean Functions. En M. Bellare (Ed.), Advances in Cryptology — CRYPTO 2000 (Vol. 1880, pp. 515-532). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44598-6_32

Saveetha, P., Arumugam, D. S., & Kiruthikadevi, K. (2014). Cryptography and the Optimization Heuristics Techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 6.

Siegenthaler, T. (1985). Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Transactions on Computers, C-34(1), 81-85. https://doi.org/10.1109/TC.1985.1676518

Siegenthaler, Thomas. (1984). Correlation-immunity of nonlinear combining functions for cryptographic applications (Corresp.). Information Theory, IEEE Transactions on, 30, 776-780. https://doi.org/10.1109/TIT.1984.1056949

Sigauke, C., & Talukder, H. M. (2003). A modified Osman's simulated annealing and tabu search algorithm for the vehicle routing problem, 22(3), 7.

Tao Zhang, C. E. (2012). Research on Optimization Strategy of Construction Methods of Boolean Functions.

Tarannikov, Y., & Kirienko, D. (2001). Spectral analysis of high order correlation immune functions (p. 69). IEEE. https://doi.org/10.1109/ISIT.2001.935932

Tarannikov, Y. V. (2000). On Resilient Boolean Functions with Maximal Possible Nonlinearity. En B. Roy & E. Okamoto (Eds.), Progress in Cryptology —INDOCRYPT 2000 (Vol. 1977, pp. 19-30). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44495-5_3

Webster, A. F., & Tavares, S. E. (1986). On The Design Of S-Boxes (pp. 523–534). Springer-Verlag.

Wu, C., & Steinbach, B. (2009). Applications of Boolean Functions in Cryptography, 10.

Youssef, H., Sait, S. M., & Adiche, H. (2001). Evolutionary algorithms, simulated annealing and tabu search: a comparative study. Engineering Applications of Artificial Intelligence, 15.

Zhang, X.-M., & Zheng, Y. (1996). Auto-Correlations and New Bounds on the Nonlinearity of Boolean Functions. En Advances in Cryptology — EUROCRYPT ’96 (pp. 294-306). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-68339-9_26

Zheng, Y., & Zhang, X.-M. (2000). On Relationships among Avalanche, Nonlinearity, and Correlation Immunity. En Advances in Cryptology — ASIACRYPT 2000 (pp. 470-482). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44448-3_36

Zheng, Y., & Zhang, X.-M. (2001). Improved Upper Bound on the Nonlinearity of High Order Correlation Immune Functions. En D. R. Stinson & S. Tavares (Eds.), Selected Areas in Cryptography (Vol. 2012, pp. 262-274). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44983-3_19

Zheng, Y., & Zhang, X.-M. (2003). Connections among nonlinearity, avalanche and correlation immunity. Theoretical Computer Science, 292(3), 697-710. https://doi.org/10.1016/S0304-3975(02)00319-5