Heuristic Search by Particle Swarm Optimization of Boolean Functions for Cryptographic Applications

Luca Mariot, Alberto Leporati
l.mariot@campus.unimib.it, alberto.leporati@unimib.it
Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi Milano-Bicocca (Milano, Italy)

Problem

Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ used in stream and block ciphers must satisfy several *cryptographic properties*, such as:

- *Balancedness*: the truth table of f is composed of an equal number of 0s and 1s
- *Algebraic degree*, $deg(f)$: the degree of the *Algebraic Normal Form* (ANF) of f should be as high as possible
- *Nonlinearity*, $Nl(f)$: the Hamming distance of f from the set of *affine functions* should be as high as possible
- *Correlation Immunity*, $CI(k)$: by fixing i input variables for $1 \leq i \leq k$, the truth tables of the restrictions of f all have the same Hamming weight. CID_k denotes deviation from $CI(k)$. f is *k-Resilient* if it is both balanced and $CI(k)$
- *Strict Avalanche Criterion*, SAC: by complementing a single input variable, the value of f changes with probability $1/2$. $SACD$ denotes deviation from the SAC
- *Absolute Indicator*, AC_{max}: the maximum absolute value of the *autocorrelation function* of f must be as low as possible

Trade-offs among some of these criteria:

- *Siegenthaler's bound*: $deg(f) \leq n - k - 1$
- *Tarannikov's bound*: $Nl(f) \leq 2^{n-1} - 2^{k+1}$

We propose a *discrete Particle Swarm optimizer* to solve the combinatorial optimization problem of finding boolean functions with good cryptographic properties.

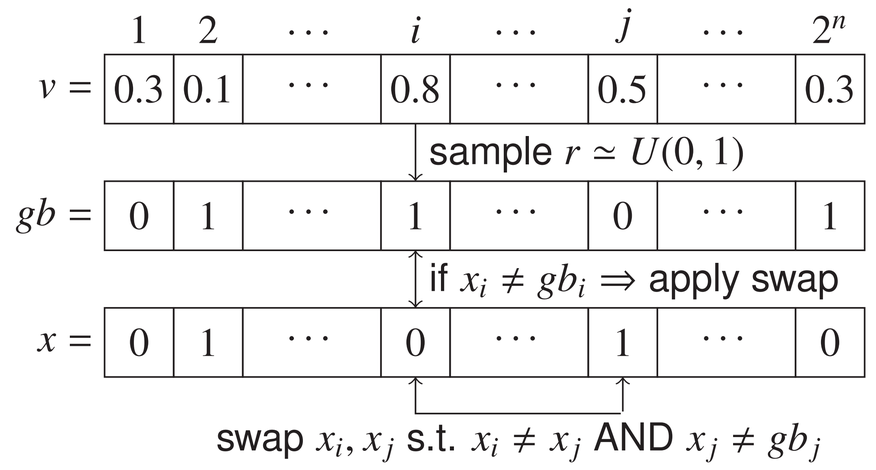
PSO Algorithm - Main Features

- *Baseline algorithm*: *discrete PSO* designed by Kennedy and Eberhart [2], the particles positions are 2^n-bit vectors representing the truth tables of boolean functions of n variables
- *velocity vector v*: specifies the *update probabilities* for the particle positions coordinates. Vector v is updated by classic PSO velocity equation normalised with the *logistic function*
- *position update*: balancedness-preserving, inspired by Hu, Eberhart and Shi's swap operator [1] (see next frame)
- *Hill Climbing step*: performed after position update using the technique proposed by Millan, Clark and Dawson [3], which increases $Nl(f)$ and decreases deviation from $CI(k)$
- Considered Fitness functions:

$$fit_1(f) = Nl(f) - [CID_1(f)/4] - [SACD(f)/8]$$
$$fit_2(f) = Nl(f) - CID_2(f)$$
$$fit_3(f) = Nl(f) - AC_{max}(f)$$

Position Update Operator

- For each coordinate i of particle x, a random number $r \simeq U(0,1)$ is sampled. If $r < v_i$, the value x_i is swapped with another value x_j in such a way that the Hamming distance from the *global best gb* is reduced
- The update process is repeated using the *local best lb* of x



Experiments and Results

- We tested our PSO algorithm on the spaces of boolean functions from $n = 7$ to $n = 12$ variables
- For each value of n, we set the numbers of particles P, PSO iterations I and PSO runs R respectively to $P = 200$, $I = 400$ and $R = 100$
- Properties of the best functions found over all R runs:

fit_j	Property	7	8	9	10	11	12
fit_1	Nl	56	112	236	480	972	1972
	deg	5	6	7	8	9	10
	CID_1	0	0	0	0	0	0
	$SACD$	0	0	8	8	8	8
fit_2	Nl	56	112	232	476	972	1972
	deg	4	6	7	8	9	10
	CID_1	0	8	8	8	8	16
	CID_2	0	8	8	8	8	16
fit_3	Nl	56	116	236	480	976	1972
	deg	5	6	7	9	10	11
	AC_{max}	16	32	48	80	128	208

Conclusions and Future Developments

- Our PSO algorithm finds boolean functions with good combinations of nonlinearity, 1-resiliency and SAC, reaching in some cases Siegenthaler's and Tarannikov's bounds, while it does not perform well when it minimizes CID_2 or AC_{max}
- Possible future development: integrate the Hill Climbing optimization step inside the position update operator, in order to perform only those swaps which increase nonlinearity while decreasing deviation from $CI(k)$

References

[1] X. Hu, R. C. Eberhart, and Y. Shi. Swarm intelligence for permutation optimization: Case study of *n*-queens problem. In *Proceedings of the IEEE Swarm Intelligence Symposium, (Indianapolis, IN, April 24-26, 2003)*, pages 243–246, 2003.

[2] J. Kennedy and R. C. Eberhart. A discrete binary version of the particle swarm algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, (Orlando, FL, October 12-15, 1997)*, pages 4104–4108, 1997.

[3] W. Millan and A. J. Clark. Heuristic design of cryptographically strong balanced boolean functions. In *Proceedings of EUROCRYPT'98 (Espoo, Finland, May 31–June 4, 1998), Lecture Notes in Computer Science vol. 1403*, pages 489–499, 1998.