MINIMUM DISTANCE COMPUTATION OF LINEAR CODES VIA GENETIC ALGORITHMS WITH PERMUTATION ENCODING

José Gómez Torrecillas
F. J. Lobillo
Gabriel Navarro

UNIVERSIDAD DE GRANADA

NP-HARD PROBLEM

Vardy

Compute the minimum Hamming distance of an arbitrary linear code over a finite field, or, compute the minimum Hamming weight of an arbitrary linear code over a finite field

Linear Algebra-based algorithms

Brouwer-Zimmermann Algorithm

Lisonek Trummer

or, variations of BZ Algorithm...

Use an intelligent exhaustive search. Exponential time with respect to dimension and bit-size of elements

Metaheuristic-based methods

Heuristic algorithm templates for solving optimization problems

Genetic Algorithms

Simulated Annealing

For binary codes

Probabilistic approaches

Leon

Chabaud

Foster

Stern

Exhaustive search for some random vector spaces. Impracticable for large finite fields

Needed for working over large finite fields

Key-tool: Permutation Encoding

Theorem. Let G be a generating matrix of an $[n,k]$-linear code C over a finite field $\mathbb{F}$. There exists a permutation P in the symmetric group S_n and a row of $\mathrm{rref}(GM_P)$ whose weight equals the minimum distance of C, where $\mathrm{rref}(GM_P)$ is the row reduced echelon form of GM_P and M_P is the permutation matrix of P.

Reformulation of the problem

Compute the minimum of fit: $S_n \longrightarrow \mathbb{N}$, mapping P to the minimum weight of the rows of $\mathrm{rref}(GM_P)$

This research has been supported by grant MTM2016-78364-P from the Spanish Agencia Estatal de Investigacion and FEDER.

Genetic Algorithms

Talbi

Holland

Genetic algorithms are a class of heuristic methods that follows a biologically inspired search model based on the process of genetic recombination

Terminology

- **Population**. A finite set of chromosomes
- **Chromosome**. A possible solution of the problem encoded by certain codification (permutation, binary vector, real vector, ...)

1	0	1	1	0	1	0	0

- **Crossover**. An n-ary operator whose role is to inherit some characteristic of the parents to generate the offspring

parents

1	0	1	1	0	1	0	0
0	0	1	0	0	1	0	1

1	0	1	1	0	1	0	1
0	0	1	0	0	1	0	0

offspring

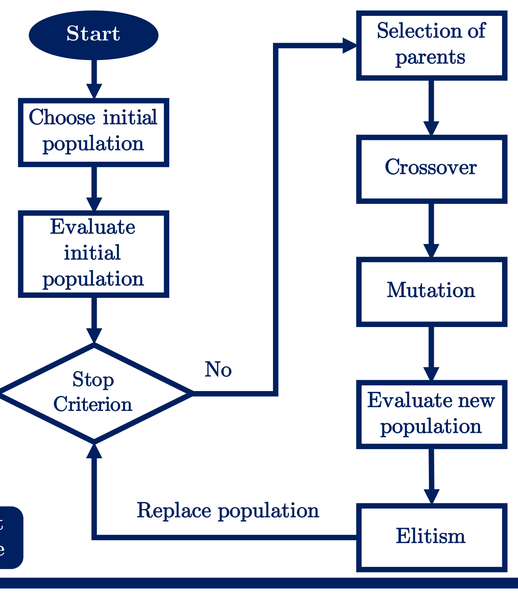
- **Mutation**. Unary operator doing small changes in a chromosome

1	0	1	1	0	1	0	0

1	0	1	0	0	1	0	0

- **Fitness**. Image of a chromosome by the map to be optimized
- **Elitism**. Add best parents to the new population of children

Standard scheme of a generational genetic algorithm



GLN Genetic Algorithm

Components

- **Representation**. Permutation encoding
- **Initial population**. Selected randomly
- **Selection**. Random selection.
- **Fitness**. Given by fit: $S_n \longrightarrow \mathbb{N}$ above
- **Algebraic crossover**. For any r, AX_r is defined as follows: from r chromosomes $p_1, p_2, \ldots, p_r$, select the best r elements of the set
$$\mathcal{T} = \{p_{\tau(1)} \circ p_{\tau(2)} \circ \cdots \circ p_{\tau(r)} \mid \tau \in \mathcal{S}_r\}$$

Send non-information columns to first positions

3	1	2	4
1	3	4	2

4	1	3	2
3	2	4	1

- **Mutation**. Change one of the first k columns with other of the remaining n-k (=multiply by a transposition)

3	1	2	4

4	1	2	3

- **Elitism**. Change worst of the new population by best of the old population

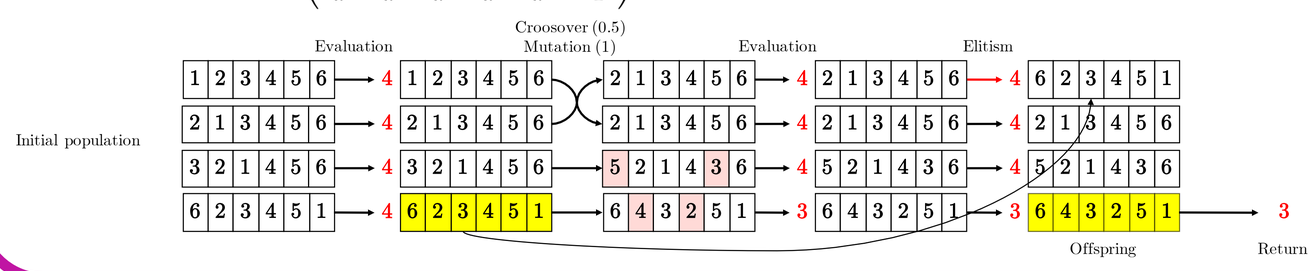
Pseudocode

Input: G, generating matrix; p_c, crossover probability; p_m, mutation probability; r, crossover step; c or t, iterations or execution time; p, population size.
Output: d upper bound of the minimum distance

```
P(0) ← random initial population of size p.
i ← 1
While i < c (or time < t) do
    best ← minimum of fit in P(i).
    X ← ∅
    While #P(i) ≥ r do
        S ← {r random chromosomes of P(i)}.
        P(i) ← P(i) − S
        S ← AX_r(S) with probability p_c.
        If S was not crossed then
            for s ∈ S do
                s ← s ∘ (p_1, p_2) for random p_1 ≤ k and p_2 > k ,with prob. p_m.
        X ← X ∪ S
    X ← X ∪ P(i)
    Change the worst of X by best
    i ← i + 1
Return best
```

Small Example

Let $\mathcal{C}$ generated by $G = \begin{pmatrix} a^5 & 0 & a^5 & a^6 & a & 0 \\ a^4 & a & 1 & 0 & a & a^2 \\ a^5 & a^4 & a^6 & a^4 & a^2 & 1 \end{pmatrix} \in \mathcal{M}_{3\times 6}(\mathbb{F})$, where $\mathbb{F} = \mathbb{F}_8 = \mathbb{F}_2(a)$



Experimental Results

Databases

Grassl

Grassl

Saouter-Le Mestre

Leon

Results

field	length	dim.	dist.	bound†	pop. size	loop‡	time (s)‡
$\mathbb{F}_8$	30	14	12	**12**(100)	5	0	~ 0
$\mathbb{F}_8$	60	30	20	**20**(100)	5	0	~ 0
$\mathbb{F}_8$	90	19	49	**49**(100)	10	61	0.03
$\mathbb{F}_8$	90	50	21	22(49)	20	423	1.69
$\mathbb{F}_8$	90	60	16	**16**(94)	30	284	1.91
$\mathbb{F}_8$	130	75	28	**28**(10)	150	524	40.35
$\mathbb{F}_8$	130	85	23	**23**(3)	150	471	40.11
$\mathbb{F}_8$	130	95	18	**18**(91)	50	320	9.58
$\mathbb{F}_2$	QR(223)	112	31	**31**(100)	5	39	0.7

Stop: Until the distance is reached, or 1000 iterations. Crossover: AX_2, p_c=0'7. Mutation: p_m=1.

† 100 repetitions per code. Between parenthesis, number of times that the best bound was reached

‡ Average number of iterations and mean time whenever the best bound is reached

GLN-GA vs Randomness

Distributions of the distances obtained for some codes of the table for GLN-GA and the random selection of 1000 generations