

# A Software Tool to discover Linear Codes with Large Minimum Distance

## Advisors

Pedro I. Rivera Vega, UPRM  
Heeralal Janwa, UPRRP

## Student:

Víctor Kareh-Fábregas

# Project Objective

Development of a software tool

- For finding better linear error-correcting codes
- Based on genetic algorithms
- Allow user interaction while executing

# Error Correcting Code

- Allows to detect and correct errors during the transmission of digital data.
- Receiver can detect if errors have occurred, and correct them if possible
  - Avoid retransmission of messages

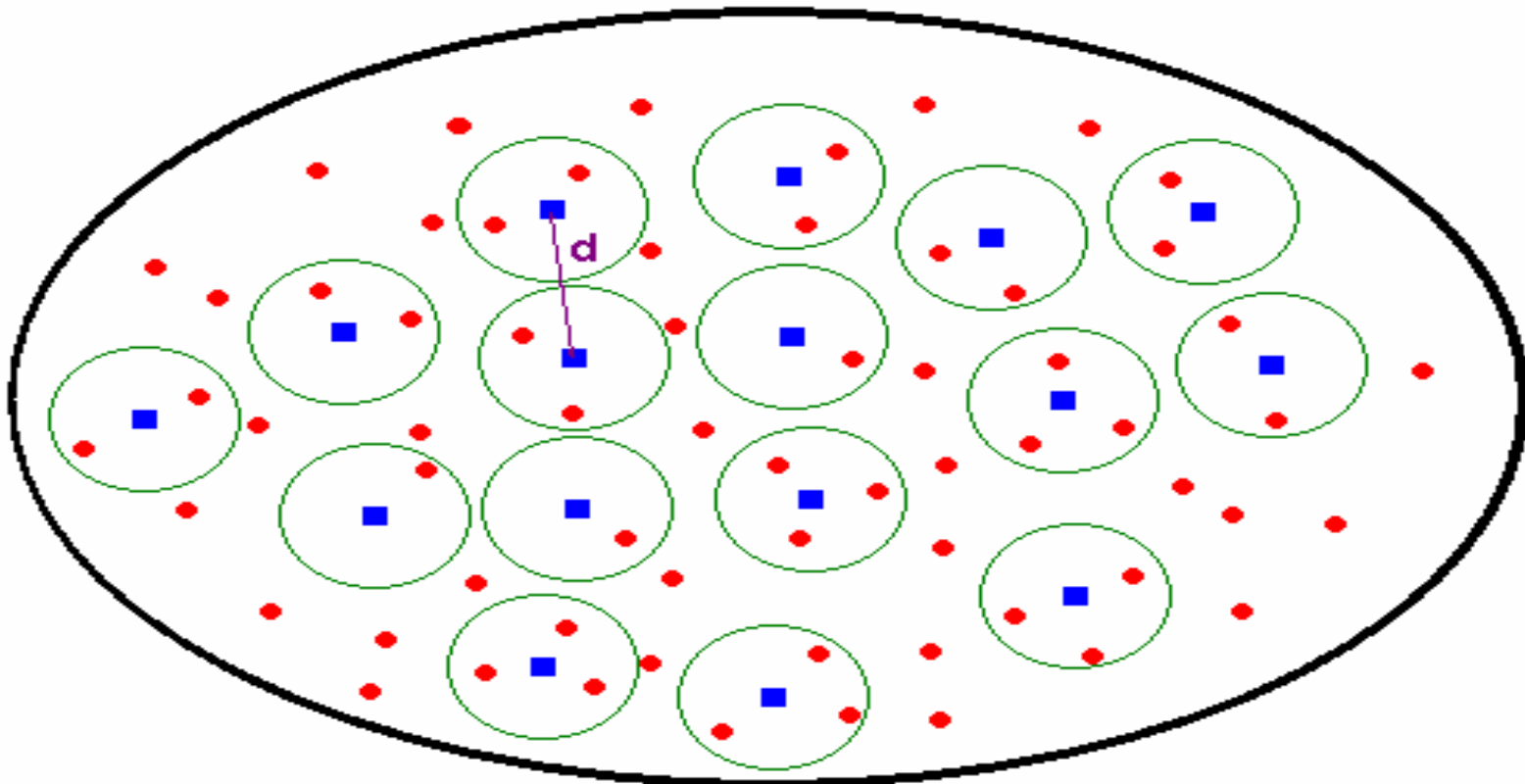
# Genetic Algorithm

- **Genetic Algorithm** – applies the principles of the evolution process to solve an optimization problem.
  - Generates different populations of possible solutions
  - In general converges to good solutions

# Definitions

- **Hamming weight**
  - the number of 1's in a bit string
- **Hamming distance**
  - number of positions in which two bit strings differ
- **Distance (d) of a linear code**
  - minimum Hamming weight among non-zero elements
  - code can detect up to  $d-1$  errors
  - code can correct up to  $\text{floor}((d-1)/2)$

# Graphical Illustration



■: element of the code of distance  $d$

# Project Major Tasks (1 of 2)

1. Study of the basic concepts of the coding theory and genetic algorithms.
2. Adapt the genetic algorithm technique to our problem.
3. Develop and analyze different data structures and access mechanisms to be used in the final implementation.
4. Implementation of the different data structures being suggested using C++.

## Project Major Tasks (2 of 2)

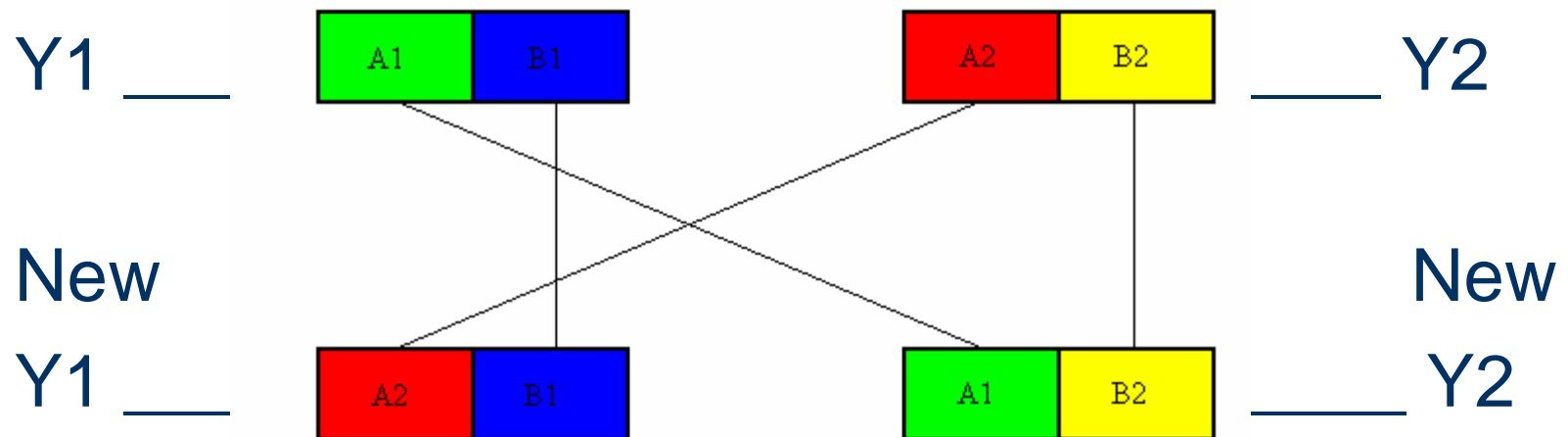
5. Implementation of the genetic algorithm developed using C++ and the library of data structures previously implemented.
6. Implement the graphical user interface
7. Experimentation and analysis of the results.

# Template for a Genetic Algorithm

1. Start with an Initial Population of candidate solutions
2. Repeat as needed:
  - a. Determine fitness of each member of current population
  - b. Based on fitness select some of those elements
  - c. For each pair of elements selected
    - i. Perform crossover and produce two new elements
    - ii. Add them to the new population
  - d. To each member of the new population, apply a random mutation

# Idea of the Crossover Operation in a Genetic Algorithm

Elements Y1 and Y2 undergoing Crossover



# Major Attributes in Our Genetic Algorithm

- Each member of a population is a linear code
  - Encoded by using one of its basis
- The fitness of each of the members is its distance
  - Estimated by using a second genetic algorithm

# Binary Representation of a Member of a Population

- Let  $n = 7$  and  $k = 4$
- Without loss of generality, we can assume that the basis of the subspace is of the form:

Vector 1	1	0	0	0	$S_1$	$S_2$	$S_3$
Vector 2	0	1	0	0	$S_4$	$S_5$	$S_6$
Vector 3	0	0	1	0	$S_7$	$S_8$	$S_9$
Vector 4	0	0	0	1	$S_{10}$	$S_{11}$	$S_{12}$

Where  
 $S_i \in \{0,1\}$ ,  
for  $i = 1, \dots, 12$

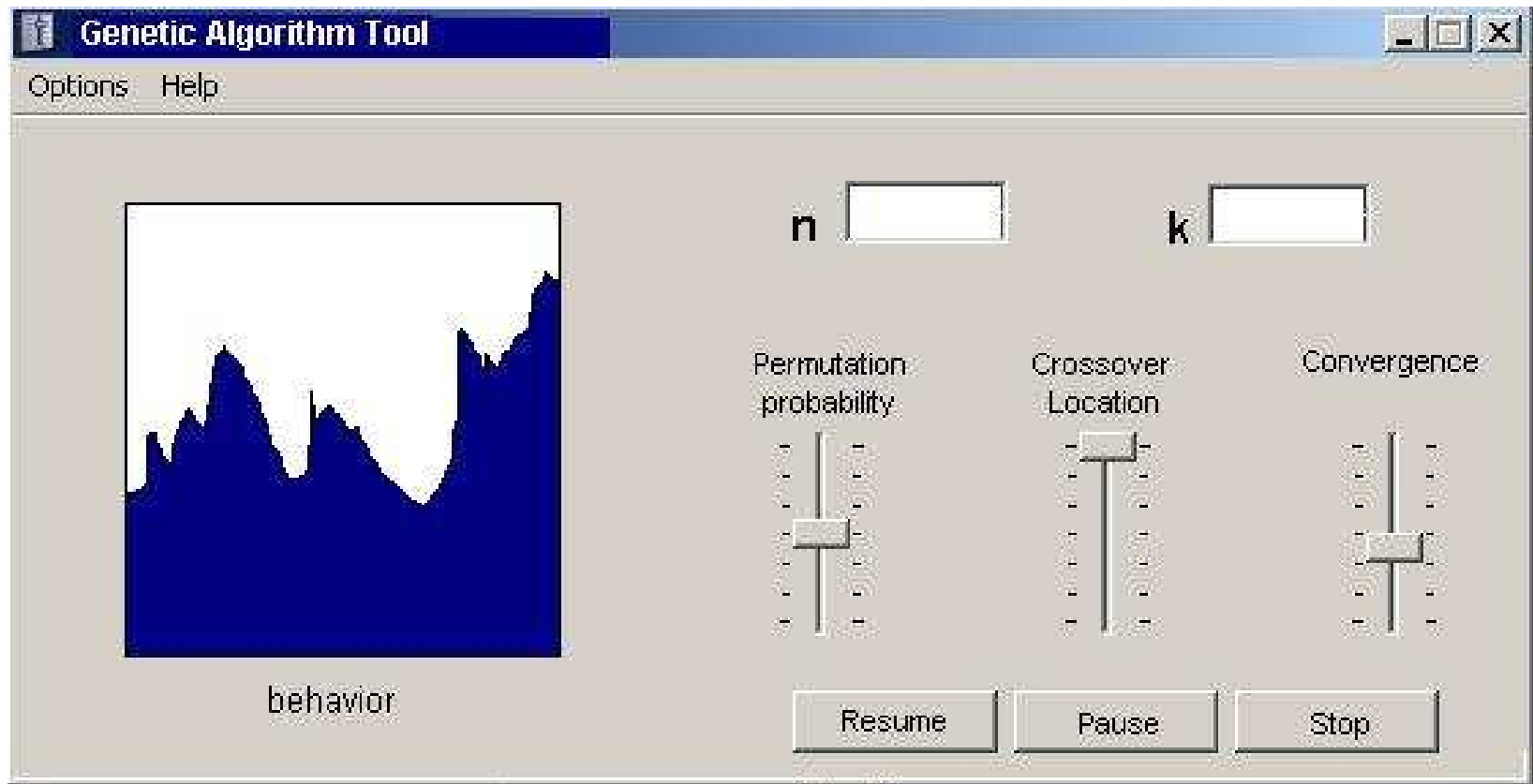
- For such a subspace its representation is the binary string:

$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------

# Fitness of each Linear Code

- Ideal: the distance of the corresponding code.
  - But this is too costly to compute.
- Alternative: approximation of the distance.
  - For this we use another genetic algorithm
    - Members of the population are elements of the linear space.
    - Fitness of an element is the *Hamming weight*.
    - Representation of vector  $X$ : bit string formed by the  $k$  coefficients of the linear combination of the elements in the basis of the linear space that produces  $X$ .

# Graphical User Interface



# Current Status

- The algorithms have been analyzed.
- We have implemented major data structures
  - To manipulate bit strings of any length
  - To encode the vector spaces
  - To perform genetic algorithm operations
- We need to implement the GUI.
- We need to experiment with this tool and compare results with current results in literature.