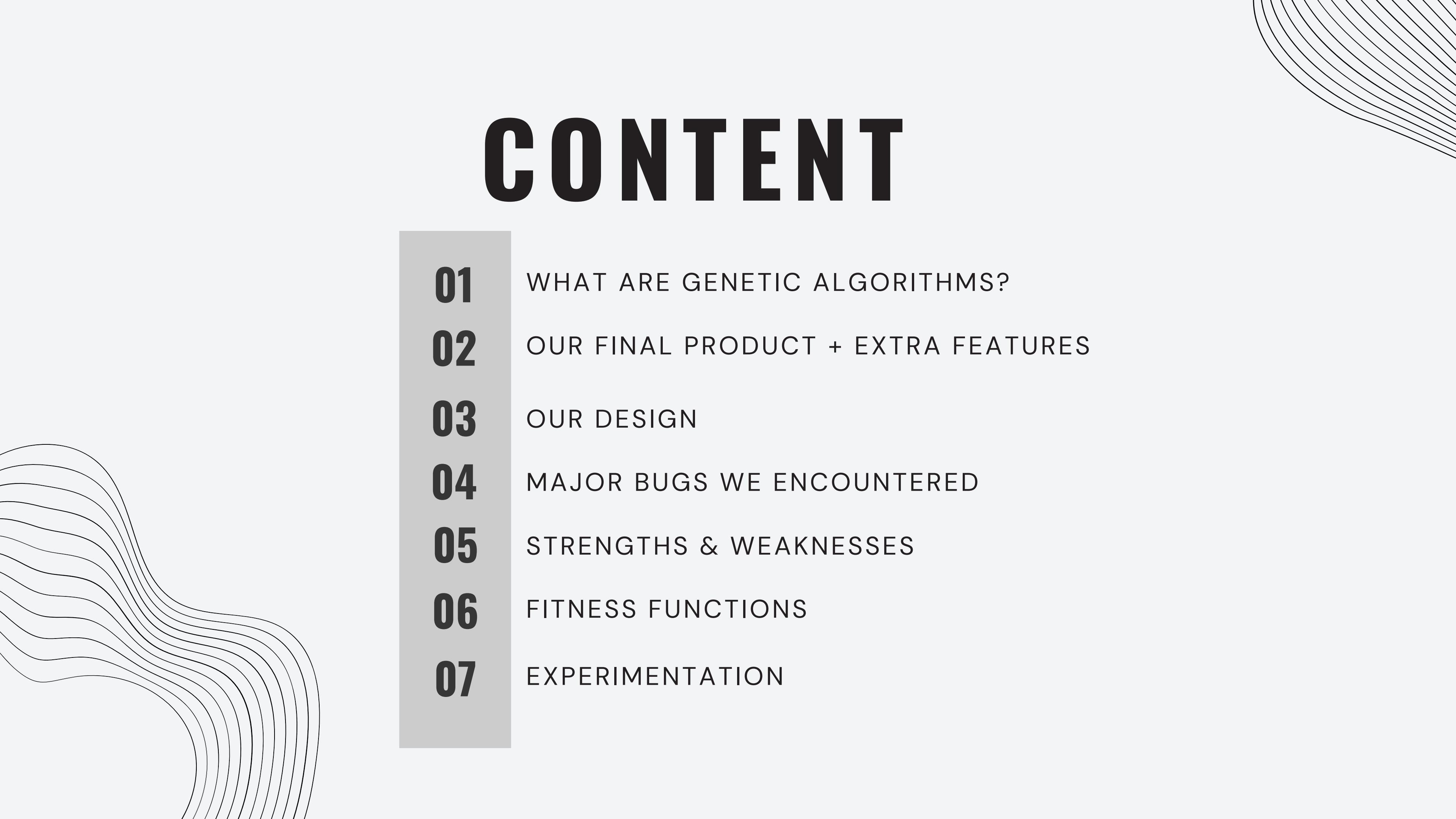


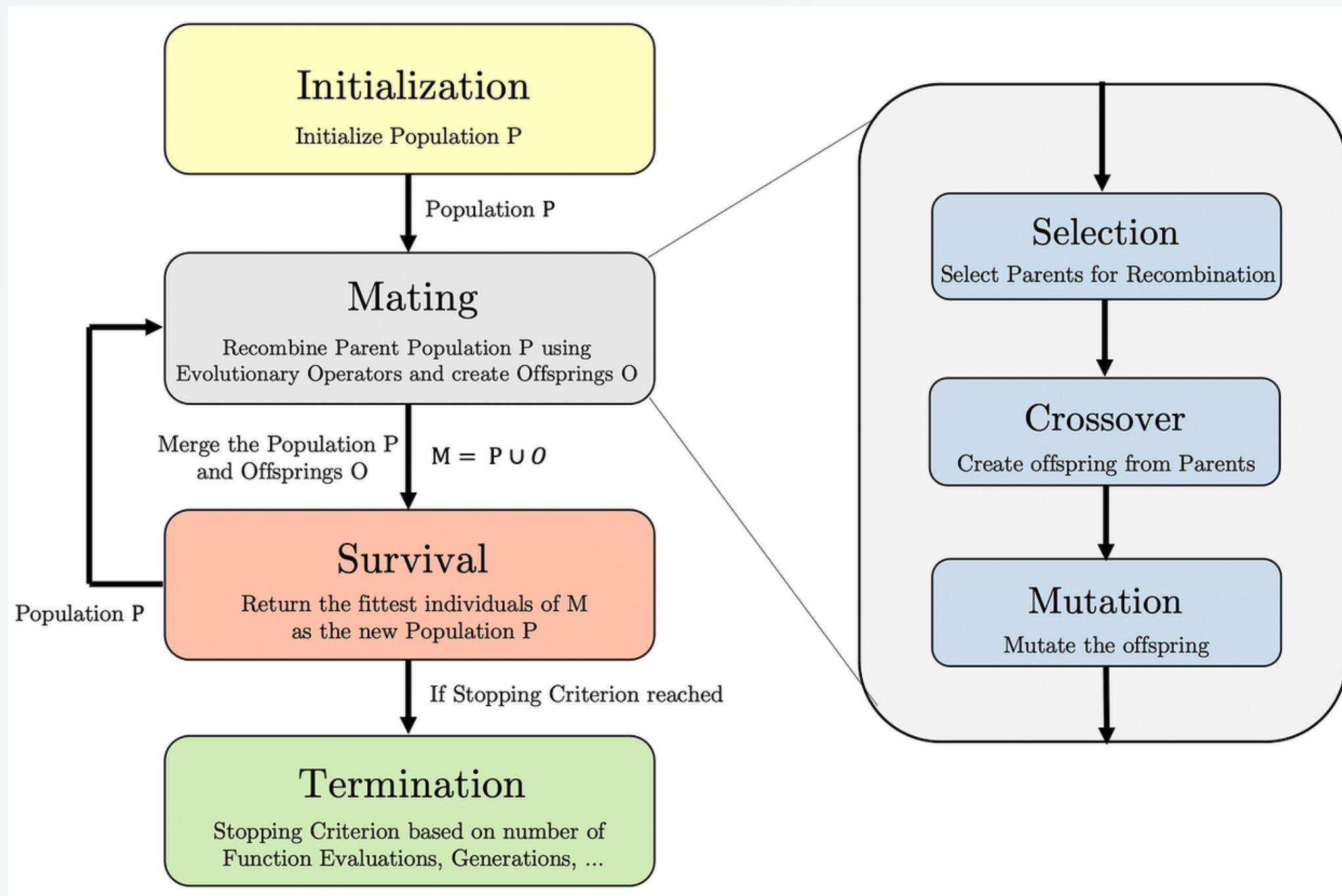
GENETIC ALGORITHM RESEARCH PROJECT (GARP)

MARK, PEI-JEN, JASON, ALEX

CONTENT

- 
- 01** WHAT ARE GENETIC ALGORITHMS?
 - 02** OUR FINAL PRODUCT + EXTRA FEATURES
 - 03** OUR DESIGN
 - 04** MAJOR BUGS WE ENCOUNTERED
 - 05** STRENGTHS & WEAKNESSES
 - 06** FITNESS FUNCTIONS
 - 07** EXPERIMENTATION

WHAT ARE GENETIC ALGORITHMS?

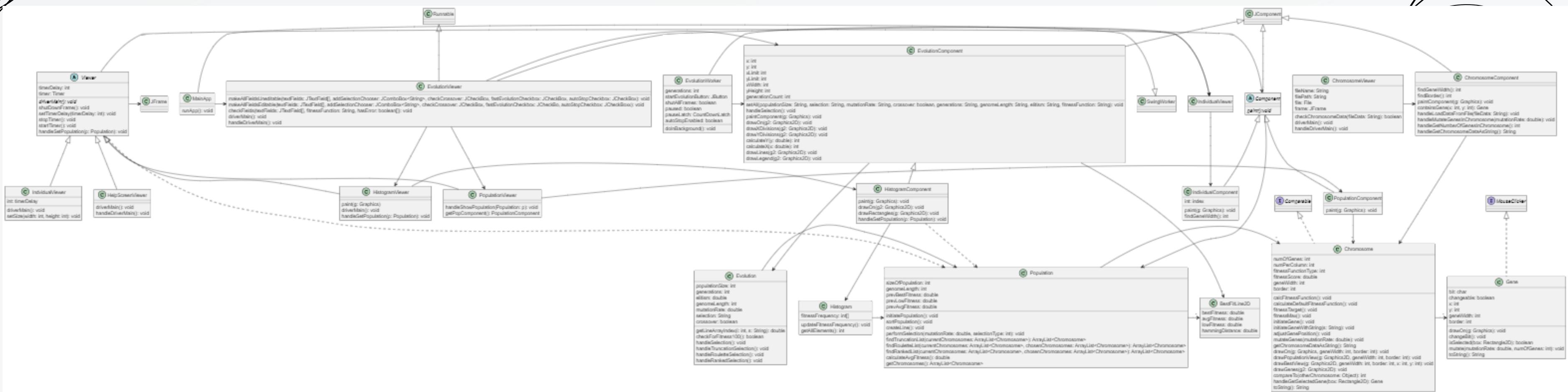




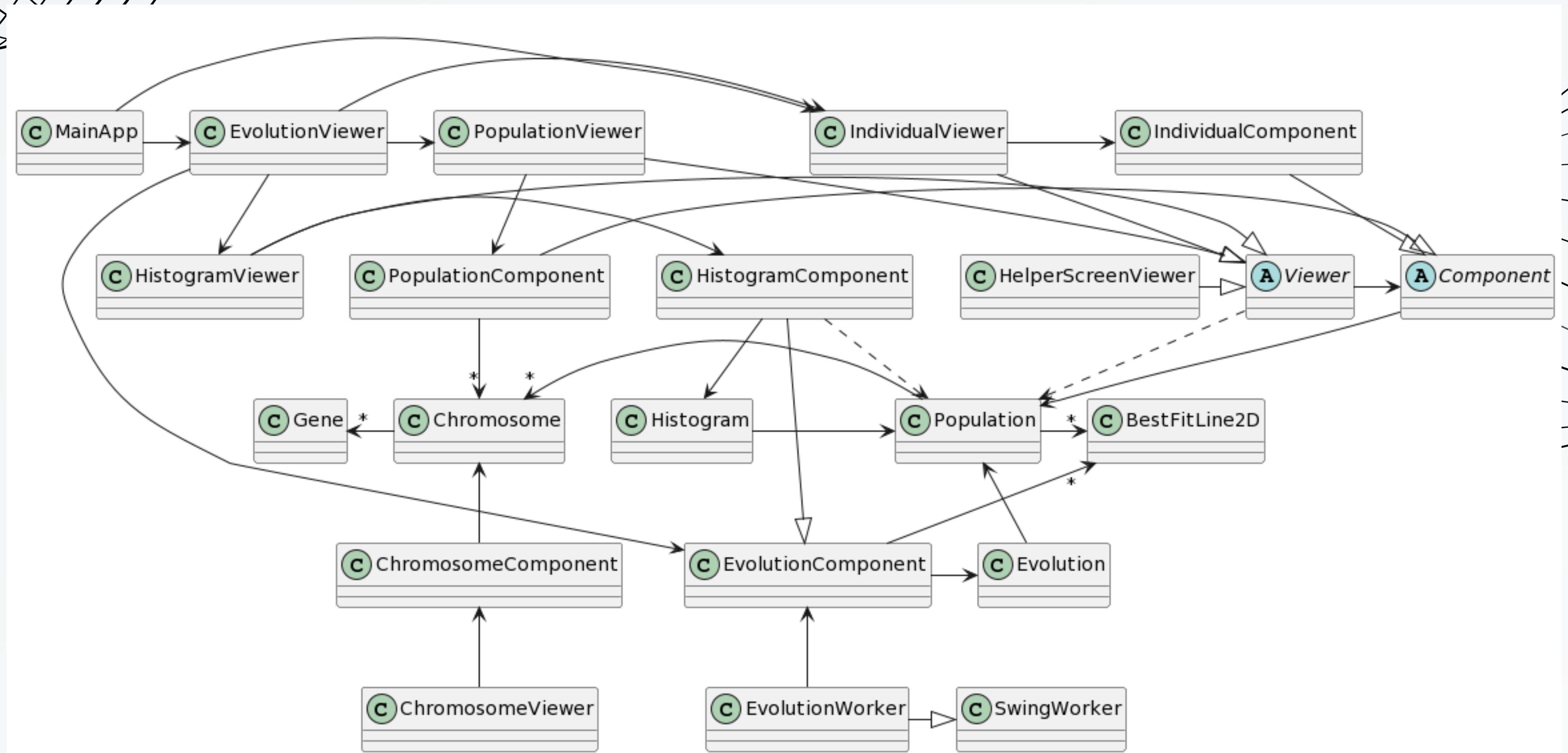
OUR FINAL PRODUCT

OUR DESIGN

OUR DESIGN



OUR DESIGN (SIMPLIFIED)



FEATURES THAT WE RAN INTO

Fast Evolution

- used 2 different action listeners, which created bugs when we only added a function to one of the listeners
- when we added new features for M4 - had to spend a lot more time on it to ensure it works on fast

Fitness Function Dropdown

- When we added more than one fitness function, we decided to let users decide which one they want to choose, and the fun started
- It took us 3-4 hours to figure that out, and, as always, it was just one line

Initializing Chromosome's Genome

- We used the wrong variable for the initialization of the genes in the chromosome
- It duplicated genes in the chromosome's genome

STRENGTHS & WEAKNESSES

Strengths

- Reduced coupling
- Eliminated message chains
- Each class appears to have a clear and single responsibility
Encapsulated related functionalities within classes
- Created classes to abstract away the implementation details and provide a higher-level interface
- Multiple usage of inheritance

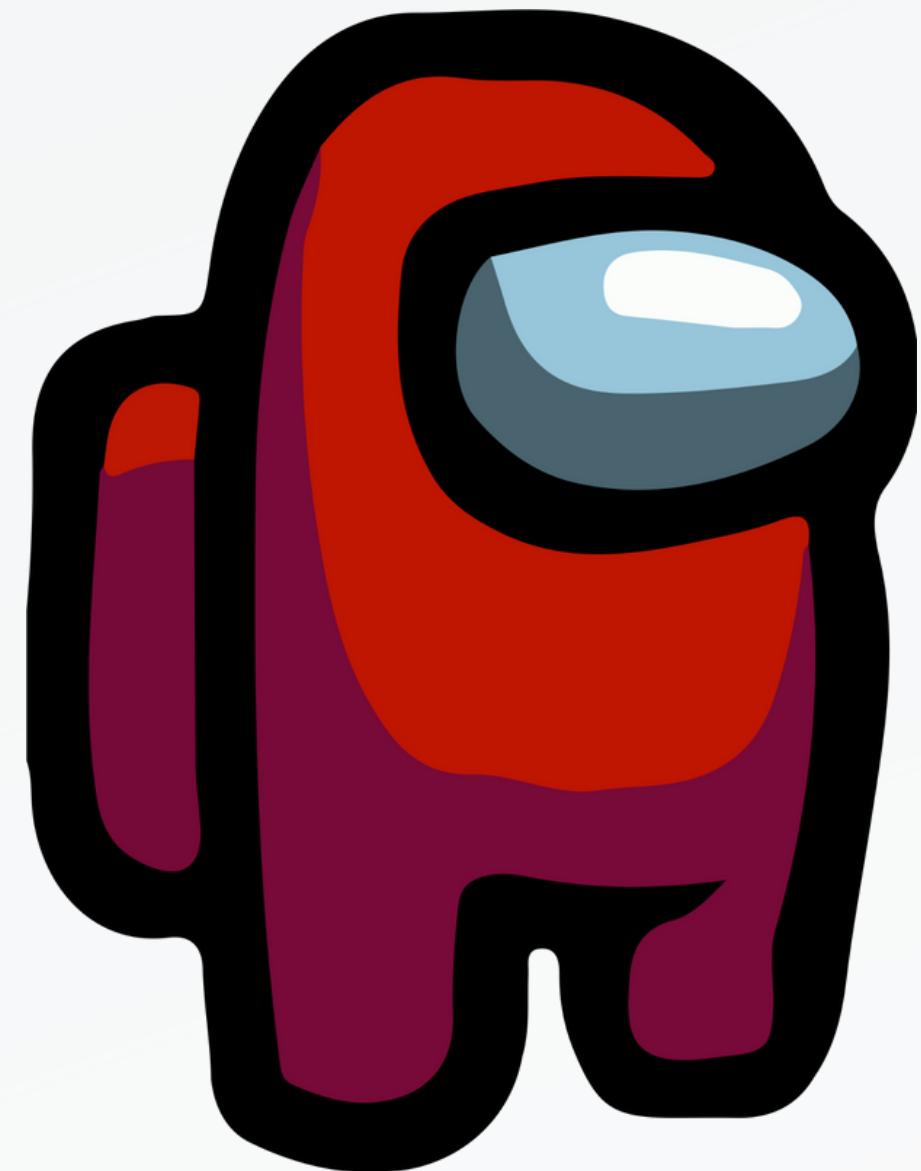
Weaknesses

- Minor code duplication across classes and method
- Bigger classes can be broken down into smaller ones
- Add more inheritance
- Assess whether common design patterns are appropriately applied in the code
- Lack of some refactored code

FITNESS FUNCTIONS

OUR ORIGINAL FITNESS FUNCTION

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99



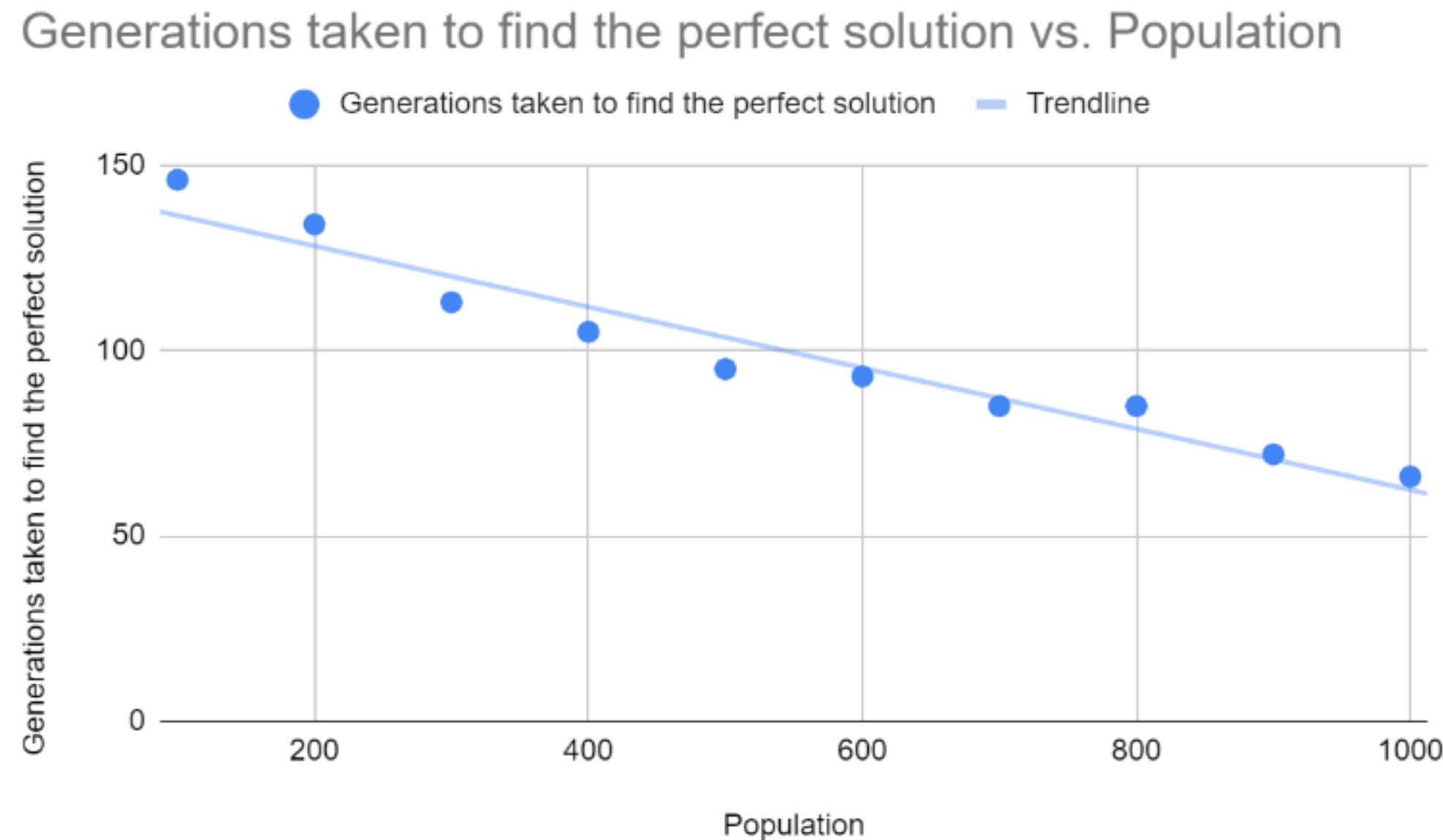
EXPERIMENTATION

EXPERIMENT

parameter	value
bit string length	100
population size	100 (initial starting size)
selection	Truncation Selection
elitism	1% (i.e. best chromosome cloned)
mutation rate	1% (i.e. $1.0/N$)
crossover	true
generations	500 (terminate at max fitness)

THE EFFECT OF POPULATION SIZE ON THE
NUMBER OF GENERATIONS IT TAKES TO
FIND THE PERFECT SOLUTION

RESULTS



Victory

