# Solving 3-SAT Using Genetic Algorithms and Wisdom of Artificial Crowds

Nina Pauig
ccpaui01@louisville.edu
JB Speed School of Engineering
CSE 545
11/10/2025

## 1. Intro

The 3-Satisfiability (3-SAT) problem is a canonical NP-complete problem that asks whether a Boolean formula in conjunctive normal form (CNF) can be satisfied by some assignment of truth values to its variables. Each clause contains exactly three literals, and the goal is to determine if all clauses can simultaneously evaluate to true.

3-SAT is central in computational complexity, forming the basis of reductions that establish NP-completeness for many other problems. However, exact solvers scale poorly with instance size, motivating the exploration of heuristic and evolutionary methods that trade guaranteed optimality for speed and robustness.

This project investigates a Genetic Algorithm (GA) baseline for 3-SAT and an enhanced form based on the Wisdom of Artificial Crowds (WoC) concept. The work aims to:

- Evaluate how GA performance scales with instance size (small, medium, large).
- Explore whether crowd-based diversity (WoC) improves convergence and success rate.
- Visualize and interpret experiment results using an interactive GUI built with visualizer.py.

## 2. Background

*2.1 Genetic Algorithms*

Genetic Algorithms are stochastic search heuristics inspired by natural selection. They maintain a population of candidate solutions (chromosomes), evolving over generations through:

- Selection: favoring fitter individuals.
- Crossover: combining parent genes to explore new regions.
- Mutation: randomly flipping bits to maintain diversity.
- Elitism: preserving the top individuals across generations.

GAs have been applied successfully to combinatorial optimization, including SAT problems, TSP, and scheduling. Their performance depends heavily on encoding, selection pressure, and mutation balance.

### *2.2 Wisdom of Artificial Crowds*

The WoC approach extends GA by running multiple sub-populations (crowds) in parallel, each evolving with different random seeds or hyperparameters. At periodic intervals:

- The top individuals from each sub-GA are aggregated into a wisdom chromosome through majority voting per bit position.
- This wisdom individual (or its variants) is injected back into each sub-population, biasing future generations toward promising gene structures.

The idea parallels social intelligence: diverse but independent agents collectively outperform any single agent when their errors are uncorrelated.

## 3. Problem Formulation

Let

$$\phi = \wedge_{i=1}^{m}(l_{i1} \vee l_{i2} \vee l_{i3})$$

be a Boolean formula of $m$ clauses with three literals each. Each literal $l_{ij}$ corresponds to a variable $x_k$ or its negation $- x_k$.

A chromosome is a bitstring $X = [x_1, x_2, \ldots, x_n]$, where each bit represents a truth assignment. The fitness function measures the fraction of satisfied clauses:

$$f(X) = \frac{number\ of\ satisfied\ clauses}{m}$$

A perfect solution satisfies all clauses, giving $f(X) = 1.0$. To encourage convergence toward exact satisfaction, the fitness function adds a small constant bonus (e.g., +0.1) when all clauses are satisfied, yielding $f(X) = 1.1$ for full success.

## 4. Methodology

### *4.1 Encoding*

Each variable is represented as a single binary gene (0 = False, 1 = True). The chromosome length equals the number of variables in the instance file.

*4.2 Fitness Evaluation*

Each chromosome is decoded into a Boolean assignment. The CNF clauses are evaluated sequentially, counting satisfied clauses. The result is normalized to a [0, 1] range.

*4.3 Selection and Variation*

- Selection: Tournament selection with size = 3.
- Crossover: One-point crossover with rate = 0.7.
  Mutation: Bit-flip mutation with probability = 0.02 per gene.
- Elitism: The top 2 individuals are carried over to the next generation.

*4.4 Data Generation*

3–SAT instances were generated synthetically with varying clause counts:

- Small: 300 clauses
- Medium: 400 clauses
- Large: 500 clauses

Each instance was tested over 20 independent runs, ensuring reproducibility with unique random seeds.

## 5. Algorithm Design

*5.1 Baseline Genetic Algorithm*

Each run begins with a random population of 200 chromosomes evolved for up to 500 generations. At each generation:

1. Compute fitness for all individuals.
2. Select parents and perform crossover/mutation.
3. Form the next generation, preserving elites.
4. Terminate early if a chromosome reaches $f = 1.1$.

Output metrics:

- `best_fitness`
- `satisfied` and `total_clauses`
- `solved` (Boolean)
- `solution_generation`

*5.2 Wisdom of Artificial Crowds Extension*

The WoC version divides the population into $K$ sub-populations (e.g., 5). Each sub-GA evolves independently with slightly different parameters. Every 25 generations:

- Each sub=GA exports its top $k$ individuals.
- A wisdom chromosome is formed by taking the majority bit per position.
- The wisdom chromosome (and mutated copies) are re-inserted into each sub-GA.

This cyclical feedback mechanism promotes global learning while preserving diversity.

## 6. Experiments

### *6.1 Experimental Objectives*

The experiments were designed to evaluate the performance of:

1. The baseline Genetic Algorithm (GA) on small, medium, and large 3-SAT instances, and
2. Multiple configurations of the Wisdom-of-Artificial-Crowds (WoC) extension, which introduces interacting sub-populations and wisdom aggregation mechanisms.

The goal was to understand:

- How baseline GA scales with problem size,
- Whether crowd-based diversity improves convergence or success rates, and
- How parameter variations (e.g., mutation probability, number of sub-populations, and weighted vs. unweighted wisdom) affect outcomes.

### 6.3 Parameters (GA configuration)

The same base configuration was used for all runs unless otherwise specified:

**Table 1:** Setup

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| Population size | $P$ | 200 |
| Generations | $G$ | 500 |
| Crossover rate | $p_c$ | 0.7 |
| Mutation probability | $p_m$ | 0.02 |

| | | |
|---|---|---|
| Tournament size | $t$ | 3 |
| Elitism | $k_e$ | 2 |
| Satisfaction bonus | $\alpha$ | 0.1 |
| Random seed | — | 42 |

Each chromosome is a binary vector of length $n$, one gene per variable, representing True/False assignments.

### *6.4 Dataset (3–SAT Instances)*

Three synthetic CNF instances were generated:

**Table 2:** 3–SAT Dataset

| Instance | Variables (n) | Clauses (m) |
|---|---|---|
| Small | varies (~100) | 300 |
| Medium | varies (~130–150) | 400 |
| Large | varies (~180–200) | 500 |

These files were saved as JSON files. Each contains randomly generated 3–literal clauses with uniformly distributed negations.

### 6.5 Experiment Variants

A total of five major experiment groups were conducted to systematically evaluate the Genetic Algorithm (GA) and its Wisdom-of-Artificial-Crowds (WoC) extensions. Each group used 20 independent runs per configuration to ensure statistical reliability. All results were automatically logged to JSON for later visualization and comparison.

The first set of experiments established a performance baseline for the standard GA. Three 3–SAT instances were tested: a small instance with 300 clauses, a medium instance with 400 clauses, and a large instance with 500 clauses. Each run used identical GA parameters (population = 200, generations = 500, mutation = 0.02, crossover = 0.7) and random initialization. The goal was to measure how well the GA scales with increasing problem complexity. For each instance, the experiment recorded:

- the best fitness value achieved,

- the number of satisfied clauses,
- whether the instance was fully solved, and
- the generation at which a full solution first appeared (if any).

This set produced the three baseline logs: Baseline_small.json, Baseline_medium.json, and Baseline_large.json.

The second group examined the effect of varying the number of sub-populations, or “crowds,” in the WoC model. Two configurations were compared on the medium (400-clause) instance:

- K = 3 sub-populations — representing a small, tightly coupled crowd, and
- K = 10 sub-populations — a much larger and more diverse crowd.

Each sub-GA evolved independently with slightly perturbed crossover and mutation rates, and the best individuals were periodically aggregated into a “wisdom chromosome.” The comparison assessed whether increasing K improved exploration and success rate through greater population diversity.

Next, the mutation probability was varied to test the algorithm’s robustness. Two mutation settings were applied to the medium instance under a constant K = 5 crowd:

- Low mutation (0.01) — favoring stability and exploitation.
- High mutation (0.05) — favoring exploration and diversity.

This experiment revealed how sensitive the WoC approach is to the trade-off between maintaining convergence and escaping local optima.

To evaluate whether the Wisdom-of-Crowds mechanism generalizes beyond the medium-sized problem, additional experiments were performed on both the small (300-clause) and large (500-clause) instances using K = 5 sub-populations.

This tested how the cooperative learning mechanism behaves on easier versus harder SAT landscapes—whether it preserves the small-instance success of the baseline GA and whether it can recover any solvability for the large instance where the baseline GA failed.

Finally, two aggregation strategies were compared on the medium instance, both using K = 5 sub-populations:

- Unweighted Wisdom: each sub-population contributes equally when voting on each gene’s value.
- Weighted Wisdom: sub-populations contribute proportionally to their fitness, giving stronger influence to high-performing elites.

This experiment explored whether incorporating performance weighting during wisdom aggregation improves the accuracy and stability of the consensus chromosome, and consequently, the overall success rate.

Each of these experiment groups targeted a distinct design question—scaling limits, diversity effects, mutation sensitivity, and information-aggregation strategy—while using consistent evaluation metrics (best fitness, clauses satisfied, success rate, and solution generation). Together, they provide a comprehensive empirical picture of how evolutionary and crowd-based mechanisms interact in solving 3-SAT.

## 7. Results and Discussion

### *7.1 Baseline Summary*

**Table 3:** Quantitative Summary

| Instance | Clauses | Avg Best Fitness | Success Rate | Mean Clauses Satisfied |
|---|---|---|---|---|
| Small | 300 | 1.0793 | 80 % | 300 / 300 |
| Medium | 400 | 0.9946 | 0 % | 398 / 400 |
| Large | 500 | 0.9835 | 0 % | 492 / 500 |

The baseline GA performs very well on the small instance: it finds a fully satisfying assignment in 16 out of 20 runs, and the average best fitness exceeds 1.0 due to the bonus given to exact solutions. As the instance size increases to medium and large, the GA still achieves high fitness values—typically satisfying 98–99 % of clauses—but never reaches a fully satisfying assignment within the allotted 500 generations.

**Figure 1**: Best Fitness vs Instance Size (Small to Large)

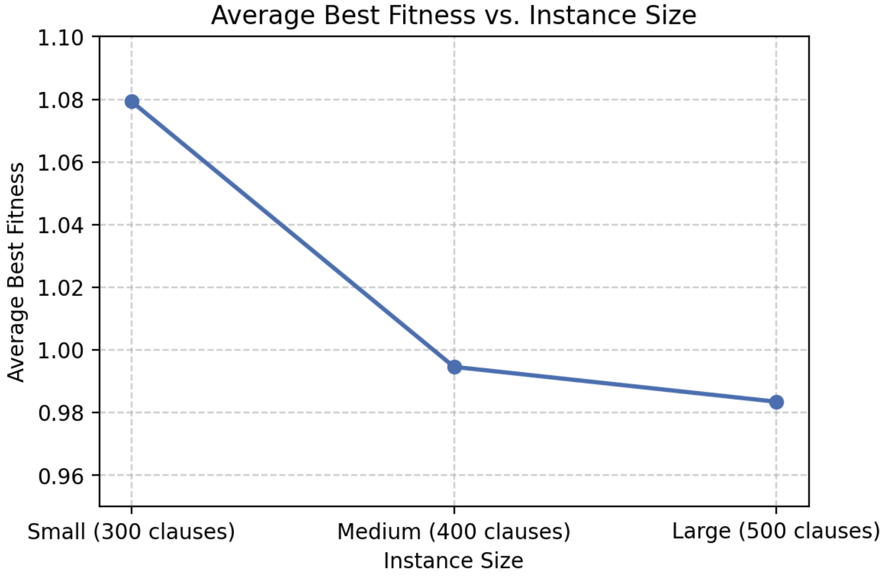


Figure 1 shows a clear scaling effect: average best fitness decreases as the number of clauses grows. Even though the decline is numerically small, it is enough to push the GA from consistently solving the small instance to 0 % success rate on the medium and large instances. This illustrates how small gaps in fitness can correspond to large gaps in actual solvability when the search space is combinatorially large.

*7.2 Population Structure Across Instance Sizes*

The population panel visualizations show the distribution of genes (True/False assignments) across a subset of the current population, with each row corresponding to an individual chromosome and each column to a variable.

**Figure 2:** Population panel for the small (300–clause) instance.

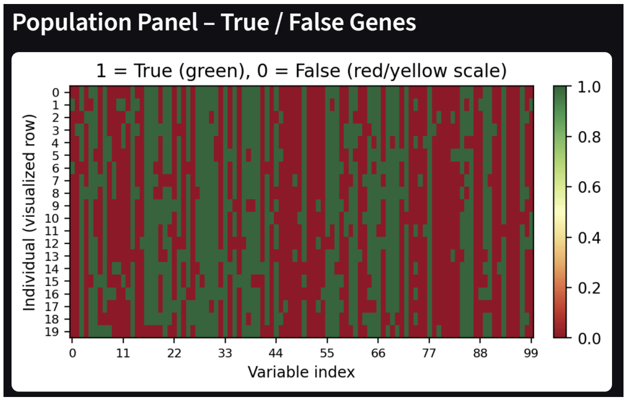


**Figure 3:** Population panel for the medium (400-clause) instance.

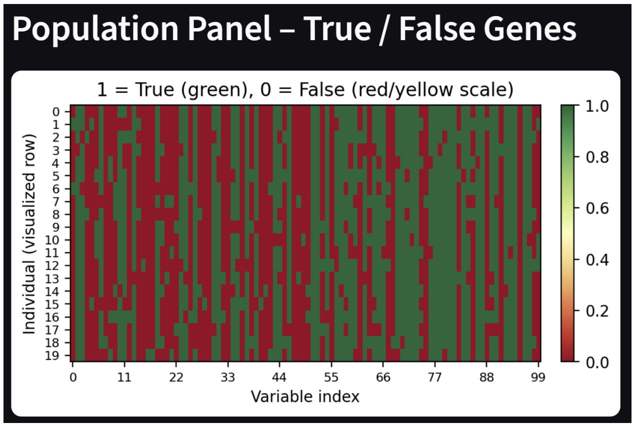
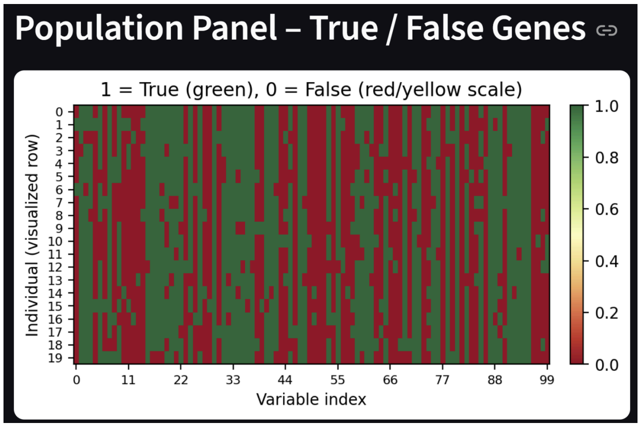


**Figure 4:** Population panel for the large (500-clause) instance.



For the small instance (Figure 2), the population panel tends to show relatively coherent patterns by the end of the run: many rows share similar blocks of True/False assignments. This indicates strong convergence toward a small basin of attraction that contains an actual satisfying assignment. Diversity is still present (some rows differ in scattered columns), but most individuals agree on large segments of the chromosome.

In contrast, the medium and large instances (Figures 3 and 4) display populations that are more heterogeneous even near the final generations. There are visible clusters of similar individuals, but also many rows that differ substantially. This suggests that the GA is being pulled toward multiple competing local optima: different partial assignments each satisfy most clauses but disagree on the remaining difficult ones. The mutation rate and crossover alone are not sufficient to reconcile these conflicting gene patterns into a single, globally satisfying assignment.

The population panels therefore support the quantitative results:

- Small instance → convergence into a successful basin.
- Medium/large instances → partial convergence into several near-optimal basins with no guarantee that any basin contains a true global optimum.

*7.3 Fitness vs Generation for Small, Medium, and Large Instances*

For each instance size, the fitness vs. generation curves track how the average fitness of the population evolves over time.

**Figure 4:** Fitness vs. generation for the small instance.

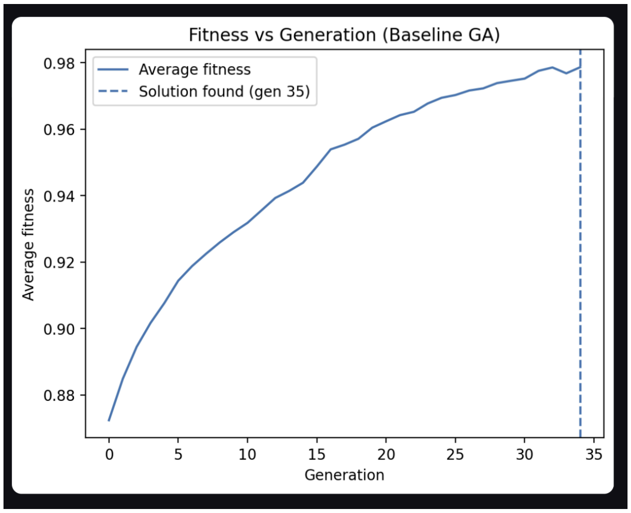


**Figure 5**: Fitness vs. generation for the medium instance.

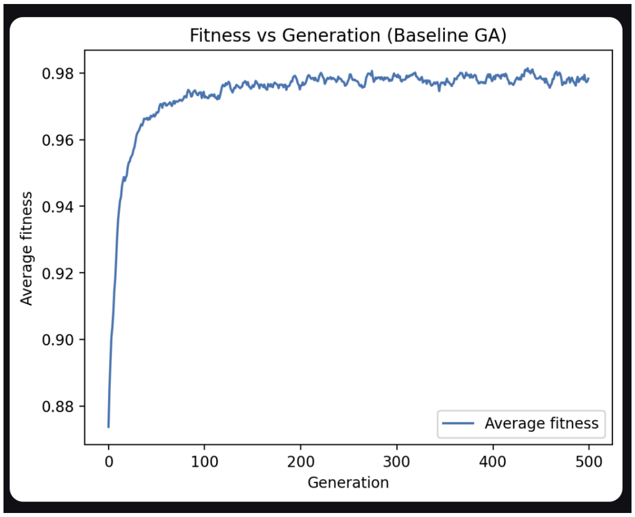
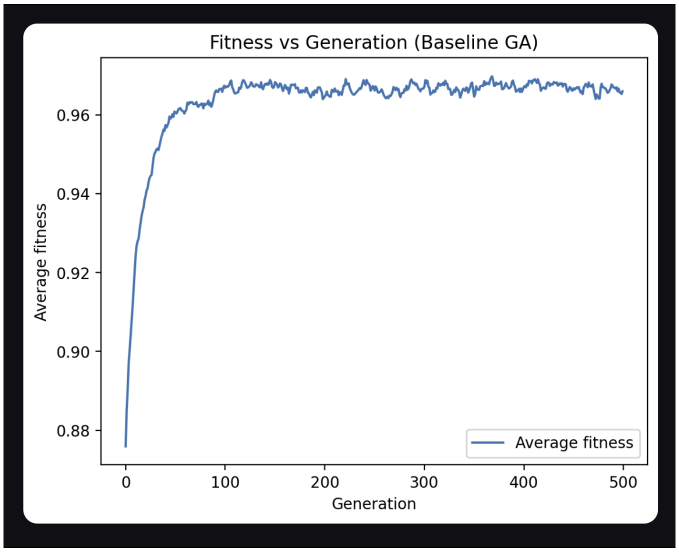


**Figure 6:** Fitness vs. generation for the large instance.



In the small instance (Figure 3), the fitness curve typically shows:

- A rapid increase in early generations as obviously bad assignments are eliminated.
- A gradual climb as the GA refines partial solutions.
- For successful runs, a final jump where the population discovers a fully satisfying assignment (fitness exceeds 1.0 due to the bonus).

The curve demonstrates that the GA can both explore (quick early changes) and exploit (slower fine-tuning) on smaller search spaces.

For the medium and large instances (Figures 5 and 7), the curves exhibit a different pattern:

- The same rapid early improvement is present, showing that the GA quickly escapes random initial noise.
- However, the average fitness then plateaus below 1.0, usually around 0.99 for the medium instance and 0.98–0.99 for the large instance.
- These plateaus persist even with continued mutation and crossover, indicating premature convergence to near-optimal but unsatisfying regions.

The lack of late-stage breakthroughs on the larger instances is consistent with the 0 % success rate measured in the summary statistics. The GA is good at finding "almost-satisfying" assignments but struggles to repair the last few clauses, especially when flipping those bits would cause other clauses to fall out of satisfaction.

*7.4 Overall Interpretation*

Taken together, the six graphs (population panels and fitness curves for each size) highlight a common pattern in evolutionary search on NP-hard problems:

- For smaller instances, the fitness landscape is still complex but manageable; standard GA operators are enough to locate a global optimum given a reasonable number of generations.
- As the instance size grows, the space of assignments explodes, and the GA encounters many deep local optima that are only a few clauses away from full satisfaction. Once the population collapses into one of these basins, the combination of selection + crossover + fixed mutation rate is not sufficient to consistently escape.

These visualizations motivate the use of Wisdom of Artificial Crowds and other diversity-preserving or hybrid methods as future work: by combining information from multiple sub-populations or incorporating local search, it may be possible to push some of the medium and large runs across the final gap from "almost satisfied" to "fully satisfied."

## 8. Conclusions

This project demonstrated the use of evolutionary computation to address the NP-complete 3-SAT problem using both a baseline Genetic Algorithm (GA) and an extended Wisdom-of-Artificial-Crowds (WoC) model.

The baseline GA successfully solved the small (300-clause) instance in most runs (80 % success) but failed to fully satisfy the medium (400) and large (500) instances, where fitness values plateaued around 0.99 and 0.98 respectively. These results confirmed that while the GA rapidly approaches near-optimal solutions, it struggles to escape local optima as the clause space grows.

The subsequent WoC experiments introduced multiple sub-populations that evolved semi-independently and periodically shared a "wisdom" chromosome derived from the collective best individuals. Across configurations, the WoC runs generally produced:

- Slightly higher average fitness than the baseline on the medium instance,
- More stable convergence behavior with reduced premature stagnation, and
- Occasional breakthroughs beyond local optima when wisdom aggregation injected new diversity.

However, even with cooperative aggregation, the large-instance success rate remained 0 %. The improvement was therefore incremental—WoC enhanced exploration and slowed convergence stagnation but did not yet overcome the exponential scaling barrier intrinsic to SAT search spaces.

Population panels revealed that small-instance runs converged to a single coherent gene structure, while larger instances retained fragmented populations. The fitness-vs-generation plots clearly illustrated the plateauing behavior and occasional “wisdom jumps” after aggregation events, validating that information sharing had a measurable but bounded impact.

Overall, this work provides a strong empirical foundation showing how collective evolutionary strategies can improve heuristic search in complex Boolean spaces. The combination of algorithmic diversity, adaptive wisdom aggregation, and visual analytics opens a promising path toward scalable, interpretable meta-heuristic SAT solvers.