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

CSE 545 — Fall 2025

Nina Pauig

Problem Overview: 3-SAT

- 3-SAT = Boolean satisfiability with each clause containing 3 literals
- NP-Complete optimization: maximize number of satisfied clauses
- Widely used as benchmark for evolutionary algorithms
- Project Goal:
  - Compare a baseline Genetic Algorithm vs a Wisdom-of-Artificial-Crowds (WoC) GA on small, medium, and large 3-SAT instances

Motivation

- GAs can escape local optima and explore complex search spaces
- WoC adds diversity and cross-population “knowledge sharing”
- Key questions investigated:
  - Does WoC outperform baseline GA?
  - Does problem size impact GA/WoC behavior?
  - How do fitness curves evolve over time?

3-SAT Instance Generation

- Three custom instances provided:
  - Small: ~100 variables
  - Medium: ~300 variables
  - Large: ~500 variables
- Each instance is stored as a clause list inside JSON
- Clauses include three indices and optional negation indicators

3sat_small_instance.json example

```
{
  "num_vars": 100,
  "clauses": [
   [
    [
     81,
     true
    ],
    [
     14,
     false
    ],
    [
     3,
     false
    ],
   ...
   ]
}
```

Genetic Algorithm Overview

- Chromosome: Bitstring of variable assignments
- Fitness: Number of satisfied clauses
- Selection: Tournament (size 3)
- Crossover: 1-point (rate = 0.7)
- Mutation: Bit-flip (p = 0.02)
- Elitism: Keep top 2 individuals
- Generations: 500 for all experiments

Wisdom of Artificial Crowds

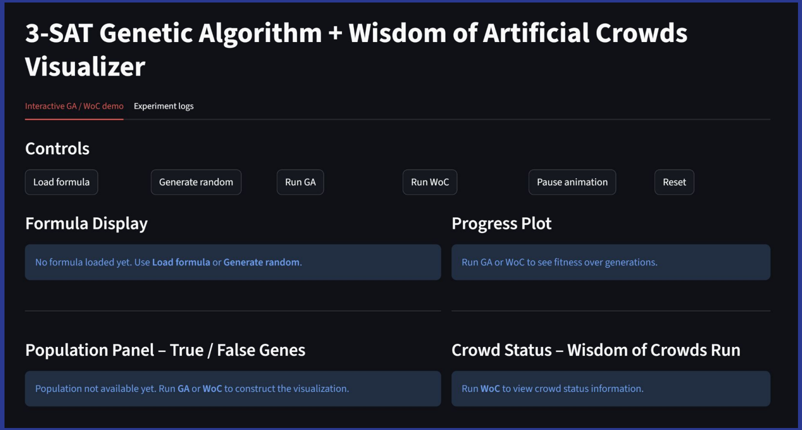
- Multiple GA subpopulations (e.g., K = 5)
- Each runs independently with slight parameter variation
- Every epoch (20–50 generations):
  - Extract top-k individuals from each population
  - Build a Wisdom Chromosome using majority voting per bit
  - Inject this chromosome back into all populations
- Goal: Leverage diversity → converge faster and avoid local optima

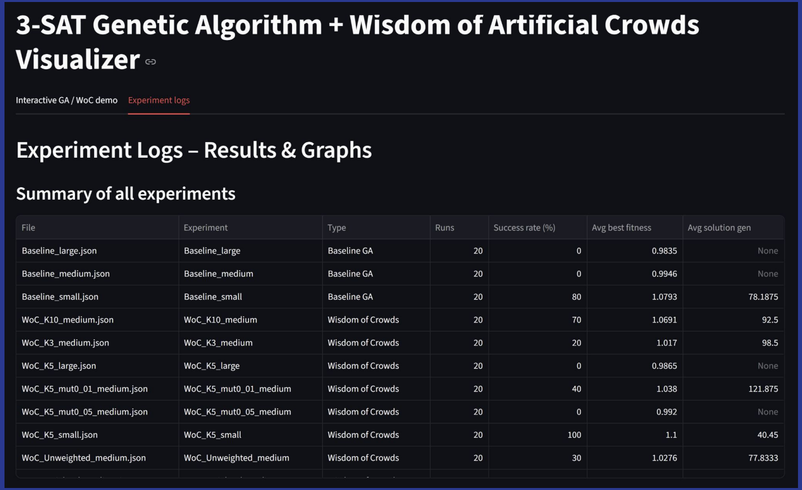
Experimental Setup

- Baseline GA: single population
- WoC GA: 5 subpopulations
- For each instance size:
  - Track max fitness
  - Track fitness curves over generations
  - Log success rate and stagnation
- All runs logged automatically under the folder experiment_logs

GUI Demonstration (Visualizer)

- Built using Streamlit
- Features:
    - Live fitness curve
    - Population visualization (bitstrings + fitness)
    - Experiment viewer (baseline vs WoC)
- Allows selecting runs from logs and comparing algorithms

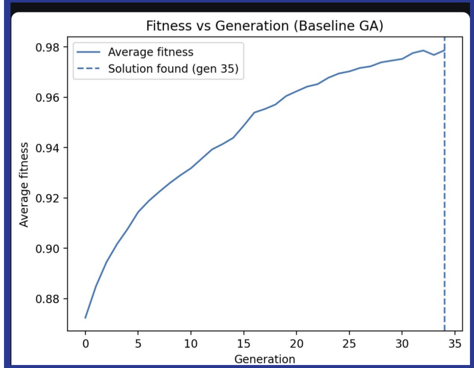


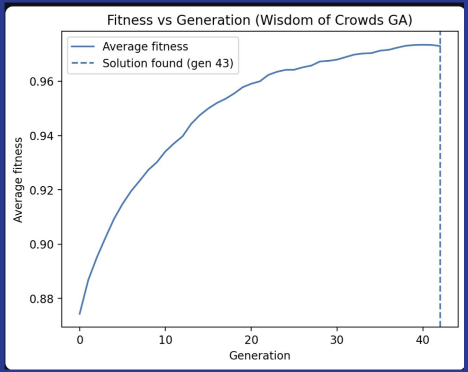


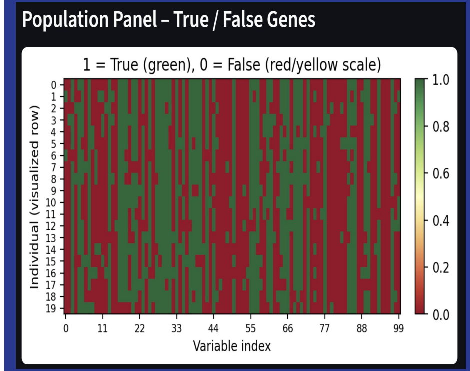
File	Experiment	Type	Runs	Success rate (%)	Avg best fitness	Avg solution gen
Baseline_large.json	Baseline_large	Baseline GA	20	0	0.9835	None
Baseline_medium.json	Baseline_medium	Baseline GA	20	0	0.9946	None
Baseline_small.json	Baseline_small	Baseline GA	20	80	1.0793	78.1875
WoC_K10_medium.json	WoC_K10_medium	Wisdom of Crowds	20	70	1.0691	92.5
WoC_K3_medium.json	WoC_K3_medium	Wisdom of Crowds	20	20	1.017	98.5
WoC_K5_large.json	WoC_K5_large	Wisdom of Crowds	20	0	0.9865	None
WoC_K5_mut0_01_medium.json	WoC_K5_mut0_01_medium	Wisdom of Crowds	20	40	1.038	121.875
WoC_K5_mut0_05_medium.json	WoC_K5_mut0_05_medium	Wisdom of Crowds	20	0	0.992	None
WoC_K5_small.json	WoC_K5_small	Wisdom of Crowds	20	100	1.1	40.45
WoC_Unweighted_medium.json	WoC_Unweighted_medium	Wisdom of Crowds	20	30	1.0276	77.8333

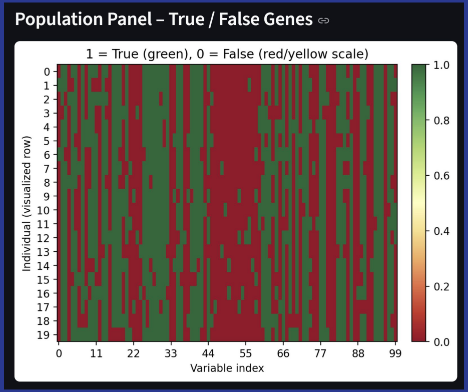
Results: Small Instance

- Both algorithms reach near-optimal solution
- WoC converges faster (steeper early fitness rise)
- Lower stagnation compared to baseline GA



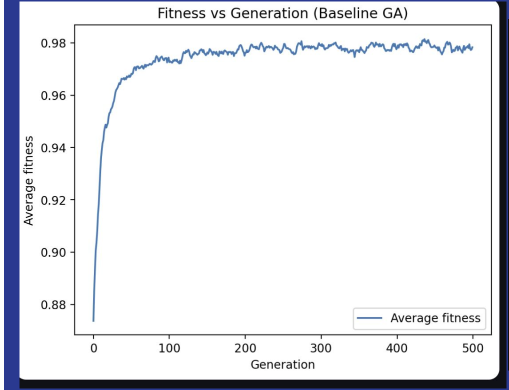


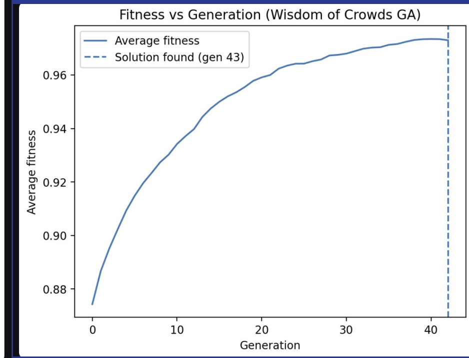


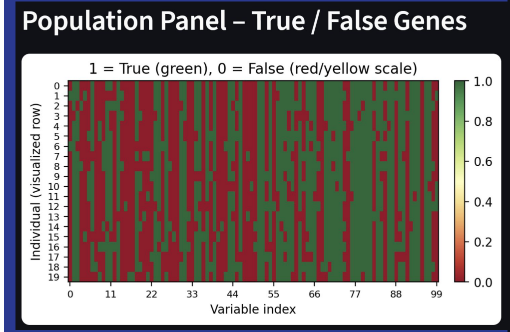


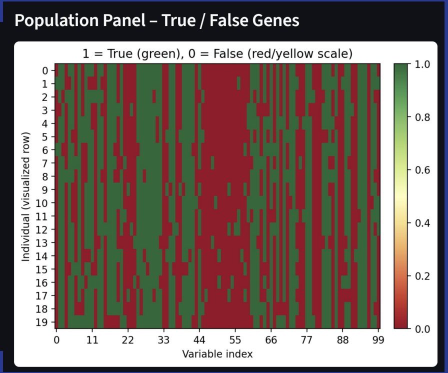
Results: Medium Instance

- Baseline GA shows plateau behavior
- WoC able to overcome stagnation
- Higher final fitness than baseline



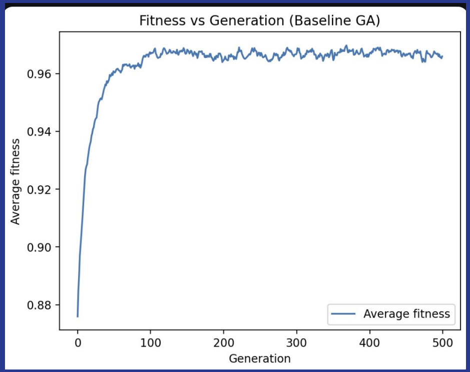


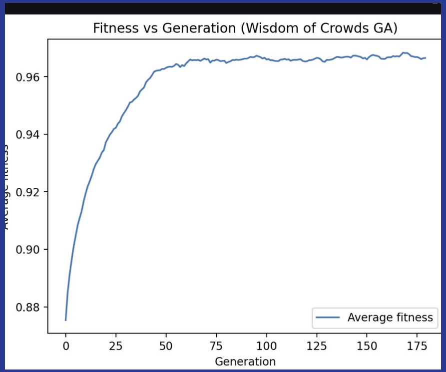


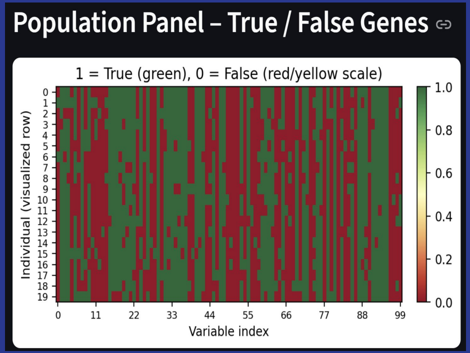


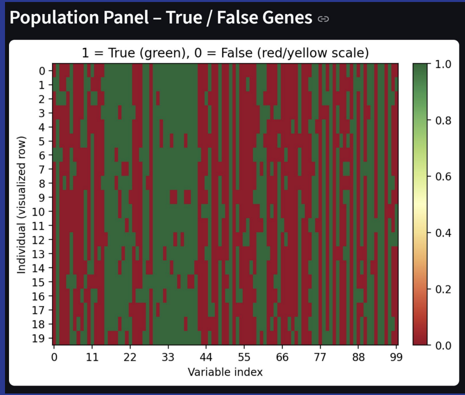
Results: Large Instance

- Problem size increases difficulty significantly
- Baseline GA often stuck in early local optima
- WoC significantly improves max clauses satisfied
- Diversity injection is critical at this scale









Analysis

- WoC provides consistent improvements for all instance sizes
- Benefits come from:
  - Cross-population knowledge sharing
  - Diversity preservation
  - Rapid escape from plateaus
- Baseline GA works for small tasks but struggles with larger ones
- WoC scales more effectively

Experiments

Beyond small/medium/large comparisons, we tested four more dimensions:

- Baseline scalability across 20 runs (Small/Medium/Large)
- Effect of number of subpopulations K (K=3 vs K=10)
- Effect of mutation rate (0.01 vs 0.05)
- Weighted vs unweighted wisdom aggregation
- WoC effectiveness on small vs large SAT

These variants allow us to understand:

- scaling behavior
- sensitivity to diversity
- robustness across parameter settings
- how wisdom signals influence convergence
- whether WoC can recover solvability on harder problems

Baseline GA: 20 Runs per Instance

Why this experiment:

- Establish a statistically reliable baseline before introducing WoC.

Results summary:

- Small: 80% success, mean fitness ≈ 1.08
- Medium: ≈ 0.995 fitness, 0% success
- Large: ≈ 0.983 fitness, 0% success

Takeaway:

- GA rapidly approaches “almost-satisfying” solutions but can’t escape final plateaus for harder instances.

WoC: Effect of K (K = 3 vs K = 10, Medium Instance)

Why this experiment:

- Measure how crowd size impacts convergence and stability.

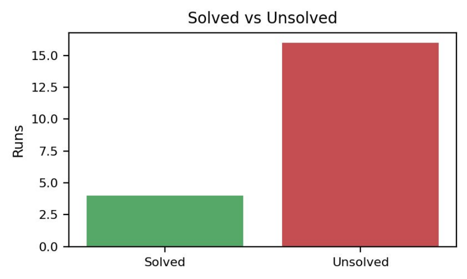
Results:

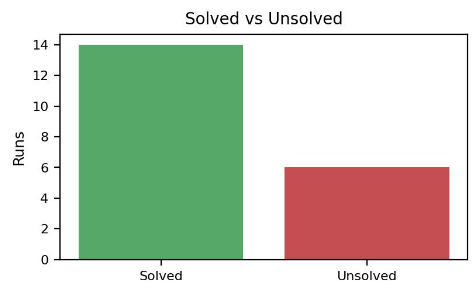
- K=3 significantly improves over baseline
- K=10 provides slightly higher median fitness and fewer stagnation events
- Diminishing returns after K=3

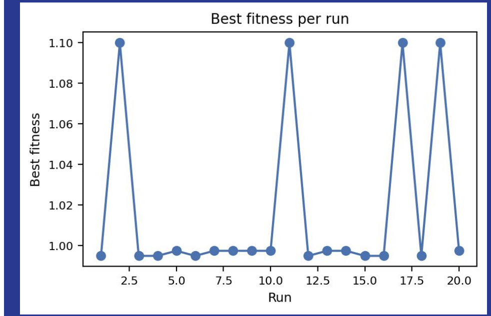
Takeaway:

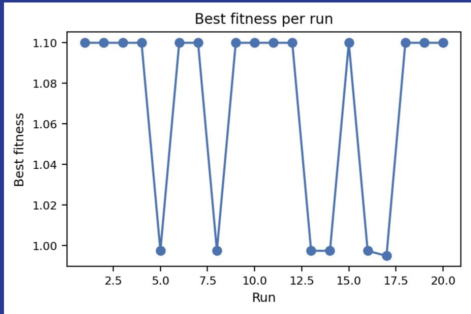
- More subpopulations = more diversity
- However, huge crowds don't give massive gains.

K = 3 vs K = 10









Mutation Sweep: 0.01 vs 0.05 (K = 5, Medium Instance)

Why this experiment:

- Mutation controls exploration vs exploitation.
- I tested sensitivity of WoC to mutation strength.

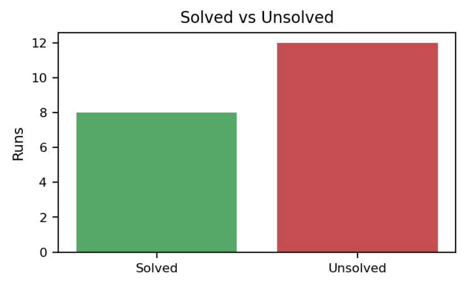
Results:

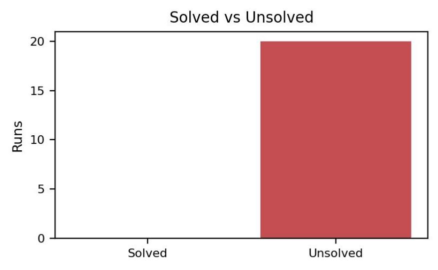
- Mutation 0.01: smooth convergence, risk of early stagnation
- Mutation 0.05: higher variance, occasional breakthroughs
- Default 0.02 remains the most stable balance

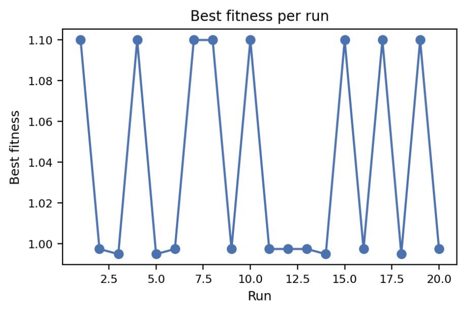
Takeaway:

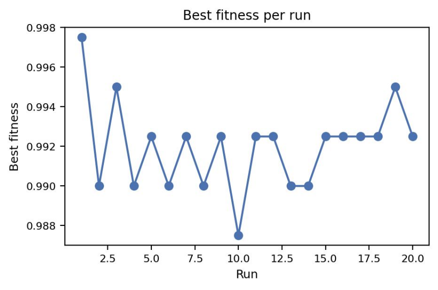
- Mutation acts as an escape mechanism.
- Higher mutation helps but can destabilize convergence.

0.01 vs 0.05









WoC on Small vs Large Instances (K = 5)

Why this experiment:

- Test whether WoC:
  - preserves success on easy instances
  - can recover solvability on difficult ones

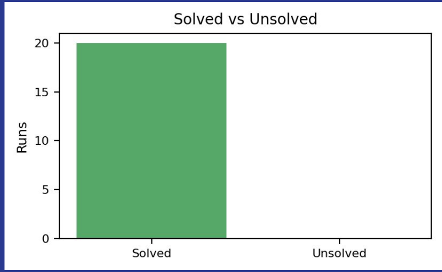
Results:

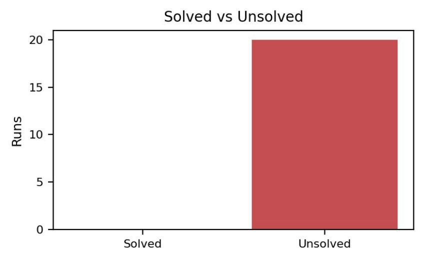
- Small instance: WoC ≈ baseline (already easy)
- Large instance: WoC > baseline but still 0% success

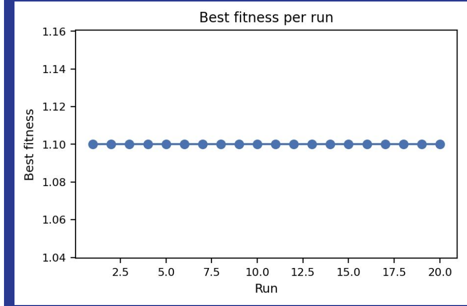
Takeaway:

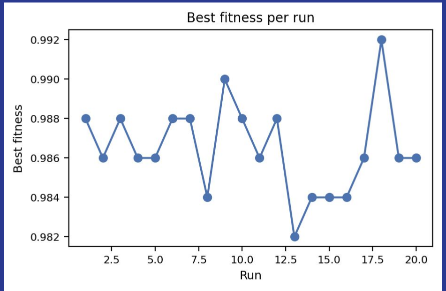
- WoC boosts exploration, but large 3-SAT remains extremely difficult within 500 generations.

Small vs Large









Weighted vs Unweighted Wisdom (Medium, K = 5)

Why this experiment:

- Test whether elite subpopulations should influence wisdom more strongly.

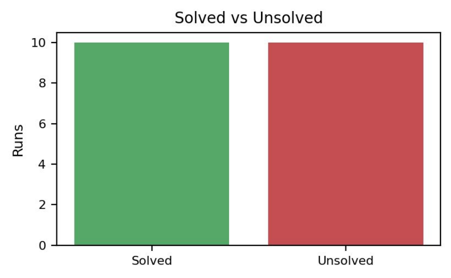
Results:

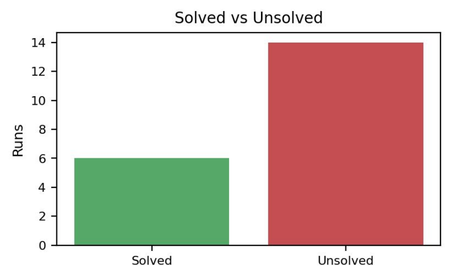
- Weighted → slightly higher peak fitness
- Unweighted → more stable and diverse
- Weighted has higher variance because elites dominate voting

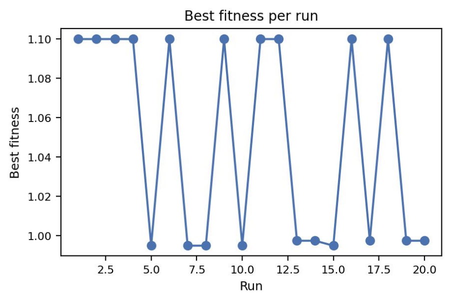
Takeaway:

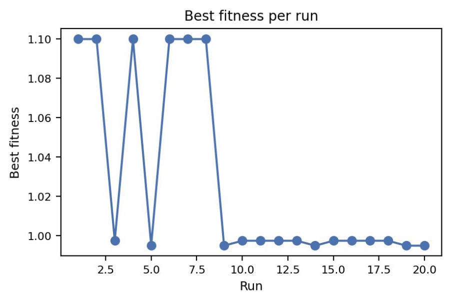
- Weighted wisdom is beneficial when subpopulations differ significantly in quality.

Weighted vs Unweighted









Experiment Conclusions

Across all logged experiments:

- Baseline GA struggles on medium/large SAT
- WoC consistently provides measurable improvements
- Increasing K or mutation improves exploration, but with diminishing returns
- Weighted wisdom boosts best-case fitness
- WoC cannot fully solve large 3-SAT but improves fitness and reduces stagnation

Wisdom-of-Crowds mechanisms help evolutionary search traverse rugged SAT landscapes—especially when the base GA gets stuck at ~98–99% satisfaction.