Heuristic Search for Random k-SAT Using Hill Climbing, Beam Search, and VND

Gajipara Nikunj, Parmar Divyraj
M.Tech CSE (AI), Batch 2025–27
Indian Institute of Information Technology, Vadodara
Email: {20251603009, 20251602003}@iiitvadodara.ac.in

***Abstract*—This experiment explores the use of heuristic search techniques to solve randomly generated Boolean satisfiability (k-SAT) problems. Uniform random 3-SAT instances were generated programmatically, and three local search algorithms—Hill Climbing, Beam Search, and Variable-Neighborhood-Descent (VND)—were implemented. The performance of these algorithms was compared across multiple problem sizes using two heuristic evaluation functions and a penetrance metric to measure the fraction of unsatisfied clauses resolved.**

***Index Terms*—k-SAT, Hill Climbing, Beam Search, Variable-Neighborhood Descent, Penetrance**

I. Objective

- To generate uniform random k-SAT instances with given parameters (k, m, n).
- To implement heuristic algorithms (Hill Climbing, Beam Search, VND) for solving 3-SAT.
- To use and compare two heuristic functions in Beam Search.
- To compute and analyze the penetrance of each algorithm.

II. Problem Definition

A Boolean satisfiability problem (SAT) asks if there exists a truth assignment for Boolean variables that satisfies a formula composed of m clauses with k literals each. Each clause C_i is a disjunction of literals, and the formula is in Conjunctive Normal Form (CNF):

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m.$$

The 3-SAT problem is NP-complete, and heuristic search provides approximate solutions within practical time for randomly generated instances.

III. Methodology

GitHub Repository: https://github.com/NikunjGajipara27/Lab-Assignment

A. Random k-SAT Generation

Each instance is generated by:

- Selecting k distinct variables from $\{x_1, x_2, \ldots, x_n\}$.
- Negating each variable independently with 50% probability.

This produces uniform random k-SAT formulas.

B. Algorithms Implemented

1) Hill Climbing:

- Start with a random assignment of truth values.
- Iteratively flip one variable if it increases the number of satisfied clauses.
- Restart if a local optimum is reached.

2) Beam Search:

- Maintain k best partial assignments (beam width 3 or 4).
- Expand each by assigning the next variable.
- Rank partial states using heuristic scores.
- Compare results for two heuristic functions:
  - Heuristic A: $H_A = \text{full} + 0.5 \times \text{partial}$.
  - Heuristic B: $H_B = \text{full} + 0.3 \times \text{undecided} + 0.2 \times \text{partial}$.

3) Variable-Neighborhood Descent (VND):

- Defines three neighborhood structures:
  - Flip one variable.
  - Flip two random variables.
  - Flip three random variables.
- Switches between neighborhoods adaptively when local optima are reached.

IV. Penetrance Metric

Penetrance quantifies the improvement in satisfied clauses from an initial random assignment to the final optimized assignment:

$$P = \frac{S_{\text{final}} - S_{\text{initial}}}{S_{\text{total}} - S_{\text{initial}}},$$

where S_{final} is the number of satisfied clauses after the algorithm terminates and S_{total} is the total number of clauses.

V. Experimental Setup

Table I
Experimental Setup Parameters

Number of variables (n)	20
Clauses per instance (m)	40, 60
Clause length (k)	3
Beam widths	3, 4
Restarts	10 (Hill Climb), 6 (VND)
Heuristic functions	A and B (Beam Search)
Trials per setting	5

VI. Results and Analysis

Table II summarizes the average performance across five trials.

Table II
AVERAGE PENETRANCE AND RUNTIME

m	Method	Beam	Heuristic	Mean Penetrance	Mean Time (s)
40	Hill Climb	-	-	1.000	0.002
40	Beam	3	A	0.995	0.012
40	Beam	4	B	0.998	0.016
40	VND	-	-	1.000	0.008
60	Hill Climb	-	-	0.999	0.004
60	Beam	3	A	0.982	0.022
60	Beam	4	B	0.991	0.027
60	VND	-	-	1.000	0.010

Observations

- Hill Climb and VND consistently reached full clause satisfaction for smaller m.
- Beam Search performed slightly slower but demonstrated robustness with Heuristic B.
- Higher penetrance values (> 0.98) indicate that most unsatisfied clauses were eventually satisfied.
- Heuristic B (optimistic weighting) improved penetrance for dense instances ($m = 60$).

VII. Discussion

The results show that local search techniques can efficiently handle random SAT instances.

- Hill Climb converges rapidly but may stagnate without restarts.
- Beam Search balances exploration and exploitation; Heuristic B yields better generalization.
- VND leverages multiple neighborhoods to escape local optima and achieve near-perfect satisfaction.

The penetrance metric demonstrates each algorithm's relative efficiency in resolving unsatisfied clauses.

VIII. Conclusion

This experiment successfully demonstrates heuristic approaches for solving random k-SAT problems. Uniform random 3-SAT instances were generated and solved using Hill Climb, Beam Search (beam widths 3 and 4, two heuristics), and VND. Among these, VND achieved the most stable convergence, while Hill Climb was the fastest. The penetrance comparison confirmed that both Heuristic A and B yielded high satisfaction rates, with minor runtime differences.

IX. Sample Output

A sample output summary (mean final satisfied, mean penetrance, mean time) is:

m	method	beam_width	heuristic	mean_final	mean_penetrance	mean_time
40	Beam	3	A	39.9	0.995	0.012
40	Beam	4	A	40.0	0.998	0.016
40	HillClimb	–	B	40.0	1.000	0.002
40	VND	–	–	40.0	1.000	0.008

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.

[2] H. Kautz and B. Selman, "Planning as Satisfiability," in *Proc. ECAI*, 1992.

[3] D. Mitchell, B. Selman, and H. Levesque, "Hard and Easy Distributions of SAT Problems," in *Proc. AAAI*, 1992.