Hill climbing is a variant of the generate-and-test algorithm. It uses feedback from the test procedure to determine the direction to move in the search space. The test function is augmented with a heuristic function that provides an estimate of how close a given state is to a goal state. The generate procedure can exploit this heuristic function.

Key Concepts

- Heuristic Function: A function that estimates the distance from a given state to the goal state.
- Search Space: The set of all possible states that can be reached from the initial state.
- Goal State: The desired state or solution.
- Local Maximum: A state that is better than its neighbors but not the global maximum.

Types of Hill Climbing

- Simple Hill Climbing: Consider a single move from the current state and select the first one that is better.
- Steepest-Ascent Hill Climbing: Consider all moves from the current state and select the best one.

Algorithm for Simple Hill Climbing

1. Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
2. Loop until a solution is found or until there are no new operators left to be applied in the current state:
    - Select an operator that has not yet been applied to the current state and apply it to produce a new state.

- Evaluate the new state.
    - If it is a goal state, then return it and quit.
    - If it is not a goal state but it is better than the current state, then make it the current state.
    - If it is not better than the current state, then continue in the loop.

Algorithm for Steepest-Ascent Hill Climbing

1. Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
2. Loop until a solution is found or until a complete iteration produces no change to current state:
    - Let SUCC be a state such that any possible successor of the current state will be better than SUCC.
    - For each operator that applies to the current state do:
        - Apply the operator and generate a new state.
        - Evaluate the new state. If it is a goal state, then return it and quit. If not, compare it to SUCC. If it is better, then set SUCC to this state. If it is not better, leave SUCC alone.
    - If the SUCC is better than the current state, then set the current state to SUCC.

Advantages and Disadvantages

- Advantages
    - Simple to implement.
    - Can be effective for certain types of problems.
- Disadvantages

    - Can get stuck in local maxima.
    - May not find the optimal solution.

Applications

- Robotics.
- Machine learning.
- Optimization problems.

Conclusion

- Hill climbing is a useful search algorithm with various applications.
- It is important to understand its limitations.
- Future work: Explore more advanced search algorithms.

References:

- Russell, S., and Norvig, P. Artificial Intelligence: A Modern Approach, 4th Edition, 2020, Pearson.
- Rich, E., Knight, K., & Nair, S. B. Artificial Intelligence. McGraw-Hill International.
- Nilsson, N. J. Artificial Intelligence: A New Synthesis. Morgan Kaufmann.

Note: This content was generated with the assistance of Google’s Gemini AI.

Related posts:

1. Artificial Intelligence Intelligence Tutorial for Beginners
2. Difference between Supervised vs Unsupervised vs Reinforcement learning
3. What is training data in Machine learning
4. What other technologies do I need to master AI?
5. How Artificial Intelligence (AI) Impacts Your Daily Life ?
6. Like machine learning, what are other approaches in AI ?
7. Best First Search in AI
8. Heuristic Search Algorithm
9. A* and AO* Search Algorithm
10. Knowledge Representation in AI
11. Propositional Logic and Predicate Logic
12. Resolution and refutation in AI
13. Deduction, theorem proving and inferencing in AI
14. Monotonic and non-monotonic reasoning in AI
15. Probabilistic reasoning in AI
16. Bayes’ Theorem
17. Artificial Intelligence Short exam Notes
18. Transformer Architecture in LLM
19. Input Embedding in Transformers
20. Positional Encoding in Transformers
21. Multi-Head Attention in Transformers