CSPs

(Project1)

Nguyen Quoc Huy

23020082

CS 188

Method

Research Design

- Object: Understand constraint satisfaction problems (CSPs)

Results

Search problems and CSPs

- Search problems is a type of planning problem, problems in which the path to the goal is the important thing.
- Constraint satisfaction problems (CSPs) are a type of identification problem, problems in which we must simply identify whether a state is a goal state or not, with no regard to how we arrive at that goal.

CSPs overview

- Constraint satisfaction problems are NP-hard.
- The successor function for a CSP state outputs all states with one new variable assigned.
- Some types of constraints:
    + Unary Constraints
    + Binary Constraints
    + Higher-order Constraints
- We can improve the performance of solving constraint satisfaction problems by filtering, ordering, and exploiting their structure.

Backtracking Search

- Is a vast improvement over the depth-first search.
- It only selects values that don't conflict with previously assigned values. If no such values exist, backtrack and change its value.

Filtering

- Filtering is forward checking: crossing off values that violate a constraint when added to the existing assignment.

- Arc consistency: An arc X -> Y is consistent if for every x in the tail(X) there is some y in the head(Y) which could be assigned without violating a constraint.

- Arc consistency is implemented with the AC-3 algorithm.

- K-consistency: we enforced guarantees that for any set of k nodes in the CSP, a consistent assignment to any sub set of k - 1 nodes guarantees that the kth node will have at least one consistent value.

Ordering

Two principles:

- Minimum Remaining Values (MRV): using when selecting which variable to assign next.

- Least Constraining Value (LCV): using when selecting which value to assign next.

Structure

- With tree-structured CSP (one that has no loops in its constraint graph): pick a node as root then linearize (or topologically sort), performing a backward pass of arc consistency and finally performing a forward assignment.

- Cutset conditioning: We can extend the structured algorithm to CSPs by cutset conditioning. We need to find the smallest subset of variables in a constraint graph such that their removal results in a tree (such a subset is known as a cutset for the graph).

Local Search

- Local search selects a random conflicted variable and reassigns its value to the one that violates the fewest constraints until no more constraint violations exist.

- Local search is both incomplete and suboptimal

Hill-Climbing Search

- This algorithm iteratively moves to a state with a higher objective value until no such progress is possible.

Simulated Annealing Search

- Simulated Annealing Search combines random walk (randomly move to nearby states) and hill-climbing.

Genetic Algorithms

- Genetic algorithms begin as beam search with k randomly initialized states called the population
- States are individuals
- Each individual is evaluated using an evaluation function (fitness function)
- Selecting pairs of states to reproduce
- Offsprings are generated by crossing over the parent at the crossover point.
- Each offspring is susceptible to some random mutation with independent probability.