

# 1 Local Search

1. Give the name of the algorithm that results from each of the following special cases:

(a) Local beam search with  $k = 1$ .

Local beam search with  $k = 1$  is hill-climbing search.

(b) Local beam search with one initial state and no limit on the number of states retained.

Local beam search with one initial state and no limit on the number of states retained, resembles breadth-first search in that it adds one complete layer of nodes before adding the next layer. Starting from one state, the algorithm would be essentially identical to breadth-first search except that each layer is generated all at once.

(c) Simulated annealing with  $T = 0$  at all times (and omitting the termination test).

Simulated annealing with  $T = 0$  at all times: ignoring the fact that the termination step would be triggered immediately, the search would be identical to first-choice hill climbing because every downward successor would be rejected with probability 1.

(d) Simulated annealing with  $T = \infty$  at all times.

Simulated annealing with  $T = \infty$  at all times is a random-walk search: it always accepts a new state.

(e) Genetic algorithm with population size  $N = 1$ .

Genetic algorithm with population size  $N = 1$ : if the population size is 1, then the two selected parents will be the same individual; crossover yields an exact copy of the individual; then there is a small chance of mutation. Thus, the algorithm executes a random walk in the space of individuals.

2. When might local search (i.e. hill climbing) be better than using A\* search? When might it be worse? There are many possible answers.

Many answers are possible here: local search helps to solve computationally difficult problems like TSP (shown below), local search can converge to a local optimum, but not global optimum.

Another useful characteristic of local searches is that if something changes (e.g., a bridge is closed or a flight is cancelled), one can simply resume the search and get a good solution that is close to the current one. A\* has to be restarted and may generate a quite different path.

## Q2. Propositional Logic

(a) Provide justification for whether each of the following are correct or incorrect.

(i)  $(X \vee Y) \models Y$

Incorrect:  $(X \vee Y) \models Y$  if and only if  $(X \vee Y) \wedge \neg Y$  is unsatisfiable; however, the latter is satisfied by  $X = \text{true}$  and  $Y = \text{false}$ .

(ii)  $\neg X \vee (Y \wedge Z) \models (X \implies Y)$

Correct: Via the same reasoning as the previous part, we can attempt to show that  $(\neg X \vee (Y \wedge Z)) \wedge \neg(X \implies Y)$  is unsatisfiable as follows:

- $(\neg X \vee (Y \wedge Z)) \wedge \neg(X \implies Y)$
- $(\neg X \vee (Y \wedge Z)) \wedge \neg(\neg X \vee Y)$
- $(\neg X \vee (Y \wedge Z)) \wedge (X \wedge \neg Y)$

Its clear that for the RHS to evaluate to true,  $X = \text{true}$  and  $Y = \text{false}$ . However, setting that automatically makes the LHS evaluate to false. Thus, the whole thing is unsatisfiable, so the original must be correct.

(iii)  $(X \vee Y) \wedge (Z \vee \neg Y) \models (X \vee Z)$

Correct: In general,  $A \models B$  if and only if  $A \implies B$  is valid. To show that this works for this problem, we can write it as an implication and prove by counterexample that the statement is always valid. Consider  $(X \vee Y) \wedge (Z \vee \neg Y) \implies (X \vee Z)$ . In general,  $A \implies B$  only evaluates to false if  $A = \text{true}$  and  $B = \text{false}$ . In order for the RHS to evaluate to false,  $X = \text{false}$  and  $Z = \text{false}$ . Plugging those in, the LHS evaluates to  $(\text{false} \vee Y) \wedge (\text{false} \vee \neg Y)$ , which can never evaluate to true. Therefore, that case never holds, so the implication is valid.

(b) Consider the following sentence:

$$[(\text{Food} \implies \text{Party}) \vee (\text{Drinks} \implies \text{Party})] \implies [(\text{Food} \wedge \text{Drinks}) \implies \text{Party}] .$$

(i) Determine, using enumeration, whether this sentence is valid, satisfiable (but not valid), or unsatisfiable.

A simple truth table has eight rows, and shows that the sentence is true for all models and hence valid.

(ii) Convert the left-hand and right-hand sides of the main implication into CNF.

For the left-hand side we have:

- $(\text{Food} \implies \text{Party}) \vee (\text{Drinks} \implies \text{Party})$
- $(\neg \text{Food} \vee \text{Party}) \vee (\neg \text{Drinks} \vee \text{Party})$
- $(\neg \text{Food} \vee \text{Party} \vee \neg \text{Drinks} \vee \text{Party})$
- $(\neg \text{Food} \vee \neg \text{Drinks} \vee \text{Party})$

For the right-hand side we have:

- $(\text{Food} \wedge \text{Drinks}) \implies \text{Party}$
- $\neg(\text{Food} \wedge \text{Drinks}) \vee \text{Party}$
- $(\neg \text{Food} \vee \neg \text{Drinks}) \vee \text{Party}$
- $(\neg \text{Food} \vee \neg \text{Drinks} \vee \text{Party})$

(iii) What do you observe about the LHS and RHS after converting to CNF? Explain how your results prove the answer to part b.i.

The two sides are identical in CNF, and hence the original sentence is of the form  $P \implies P$ , which is valid for any  $P$ .