

CPSC 322: Introduction to Artificial Intelligence (Section 2)

CSP: Stochastic Local Search Solution

Do this exercise in pairs. If there's an odd number, do it in a group of 3.

Submit the sheet before leaving.

Name of Student (last, first)	Student Number

In this activity, you'll be giving us the stopping criteria and variable selection method for the following algorithms.

1. Random sampling
2. Random walk
3. Greedy descent
4. Greedy descent with random walk
5. Greedy descent with random restart

Read the algorithm on the next page carefully and fill in the table below.

Procedure Local-Search(V, dom, C)

Inputs

V : a set of variables

dom : a function such that $\text{dom}(X)$ is the domain of variable X

C : set of constraints to be satisfied

Output

complete assignment that satisfies the constraints

Local

$A[V]$ an array of values indexed by V

```
1:  repeat
2:    for each variable  $X$  do
3:       $A[X] \leftarrow$  a random value in  $\text{dom}(X)$ ;
4:
5:    while (stopping criterion not met &  $A$  is not a satisfying assignment):
6:      select a variable  $Y$  and a value  $V \in \text{dom}(Y)$ 
7:      set  $A[Y] \leftarrow V$ 
8:    if ( $A$  is a satisfying assignment) then
9:      return  $A$ 
10:
11:  until termination
```

Local search algorithm	Stopping criteria condition (line 5)	Variable and value selection (line 6)
Random sampling	“Stopping criteria not met condition” in the while loop is always False. (We never go in the while loop.)	NA
Random walk	“Stopping criteria not met condition” in the while loop is always True. (We only get out of the while loop when A is a satisfying assignment.)	Random step Pick a neighbour at random
Greedy descent	until A is a satisfying assignment or # steps is reached	Greedy steps Pick the “best” scoring neighbour (e.g., the one with minimum number of constraint violations)
Greedy descent with random walk	until A is a satisfying assignment or # steps is reached	Combination of greedy and random steps <ul style="list-style-type: none"> - Sometimes pick the “best” scoring neighbour - Sometime pick a random neighbour
Greedy descent with random restart	until A is a satisfying assignment or # steps is reached	Combination of greedy and random sampling <ul style="list-style-type: none"> - Sometimes pick the “best” scoring neighbour - Sometime pick a random possible world