**Beyond Search Homework**

1. Simulated Annealing with T=0 and omitting termination test

**function** SimulatedAnnealing(*problem*) **returns** solution state:

*current* = node with problem initial state

**for** *t* = 1 **to infinity do:**

*next* = random successor for current

*Delta E* **=** *next*.Value – *current*.Value

**if** Δ*E* > 0 **then** *current* = *next*

**else** *current = next*

2. Genetic Algorithm with population of 1

*population =* set with population size 1

**function** GeneticAlgorithm(*population*) **returns** individual:

*new\_population* = empty set

**for** i = 1 to SIZE(population) **do**:

*x* = SELECT-ONE(*population*)

**if** (small random probability) **then** *x* = MUTATE(*x*)

*population = x*

**return** *x*

3. Hill Climbing with random restarts

**function** RandomRestartHillClimbing(*problem*) returns state of the best maximum:

*start* = CHOSE-RANDOM(*problem*) // randomly chosen starting node

best\_run = current highest value state

**loop do:**

*local\_maximum* = HillClimbing(*start*)

**if** *local\_maximum*.VALUE > *best\_run*.VALUE

**then** *best\_run* = *local\_maximum*

**until** sufficient time has passed or a goal value is reached

***return*** *best\_run*

**function** HillClimbing(*problem*) **returns** a state that is a local maximum:

*current* = MAKE-NODE(*problem*.INITIALSTATE)

**loop do:**

*neighbor* = highest value successor of current

**if** *neighbor*.VALUE <= *current*.VALUE **then return** *current*.STATE

*current* = *neighbor*