MI Sheet 4

4.1 Give the name of the algo. that is a special case of

a) Local beam search with $K=1$

1. Start with K initial states
2. generate all successors of each $\Rightarrow$ if any is a goal then halt
3. Take top K successors

repeat

. Clearly hill climbing for $K=1$ (but performs goal test before updating the node; same behav.)

b) Local beam search with 1 initial state and no limit on K

1. Start with initial state
2. generate all successors for each $\Rightarrow$ halt if goal

repeat for each successor as initial state

. This is clearly BFS (retains all states)

c) Simulated annealing with $T=0$ and no termination test

$T==0$

. for $t=1$ to ∞

next = random successor

if $\Delta E > 0$ then current = next

else current = next with Prob. $= e^{\Delta E/T}$

• when $T=0$ and $\Delta E < 0$ then
→ Current = next with Prob $\underbrace{e^{-|\Delta E|/0}}_{e^{-\infty}=0}$ } i.e., will never happen

⇒ hence, it applies Current = next whenever $\Delta E > 0$ only.

- Takes first successor of higher value
- hence, its first-choice hill climbing (after adding the goal test)

d) Simulated Annealing with $T=\infty$ at all times

→ This time $e^{-|\Delta E|/T} = e^0 = 1$

1. for t = 1 to ∞
{ next = random successor
Current = next // else is useless
}

Assuredly, this is a random walk

e) Genetic algo. with Population size N = 1
. Call that state S

1. Selection of 2 states randomly from Population
→ Both will be S

2. Crossover the two states ↙ example
→ cross(1234, 1234) = 1234 = S

3. Mutation $\underbrace{\text{in S representation}}$
→ randomly flip each digit with some low Prob.

- So we have
  repeat until State is good enough
    if (Small Prob.) then mutate(& State)

→ Its a random walk if we think that mutate will likely give us a neighboring State (otherwise, its more of a 'random jump')

2) Design an Objective function

a) 8-Queens

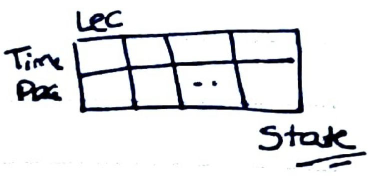
→ No. of Pairs of queens not attacking each other either directly or indirectly.
→ i.e. even if another queen is in the middle*

or

→ No. of queens that can't be attacked

- Notice that both of these are maximum when the problem is solved (all queens not attacked)

b) Creating Course Schedule



→ $\underbrace{\text{Cost function}}_{\text{want to minimize}}$ is no. of lectures taking place at same time and place

(to maximize, consider $\underbrace{-\text{Cost function}}_{\text{Obj. function}}$)

→ Clearly minimum (zero), once solved

c) Solving Equations

e.g. find $\overbrace{(x, y, z, w)}^{\text{State}}$ that satisfy

$$x^2 + z = y\cos w$$
$$xy + yz = e^x$$

- Cost function can be $\sum_{\text{eqns}} |LHS - RHS|$

→ Clearly minimum (zero) once Problem is solved.

d) Winning at chess

→ It's not so easy to arrive at integer that would be min/max upon winning (e.g. no. of dead Pieces)

لقد حان وقت الفزلكة

→ let the State be the Piece of Code that Plays Chess
- action = modify a line of the Code

→ let the Objective function be the % of games it wins vs. the best Player.

// local Search will keep changing the code until its maximized.

3. Suppose we want to minimize the function

$$f(x,y) = |2x| + |3y| + 4|x-y|$$

- using Hill Climbing and the 4 actions

$$(x,y) \longrightarrow (x+1, y) \quad // \text{inc. } x$$
$$(x,y) \longrightarrow (x-1, y) \quad // \text{dec. } x$$
$$(x,y) \longrightarrow (x, y+1) \quad \ldots$$
$$(x,y) \longrightarrow (x, y-1)$$

- Start from
$$(x,y) = (4, 3)$$

1. At which point will it terminate

Cost function at	State	Successors
	$f(4,3) = 21$	• $f(4+1,3) = 27$
		• $f(4-1,3) = 15 \Leftarrow$ best
		• $f(4, 3+1) = 20$
		• $f(4, 3-1) = 22$
	$f(3,3) = 15$	• $f(3+1,3) = 21$
		• $f(3-1, 3) = 17$
		• $f(3, 3+1) = 22$
		• $f(3,3-1) = 16$

>> all of them are higher than current state so we halt

- Final state: $(x,y) = (3,3)$

b. It's not the global min. (its local min.)

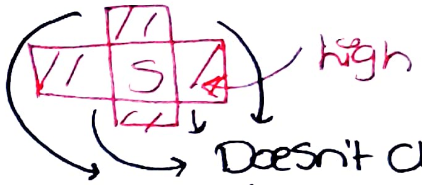
→ global min. is at $(x,y) = (0,0)$ giving

$f(0,0) = 0$ (clearly $f(x,y) \geq 0$)

C. What can we do to reach min from any initial state.

→ it might have gotten stuck in a ridge:

- only tries to go down from the horizontal or vertical direction



Doesn't check diagonal directions although they maybe lower

hence, consider adding actions

$$(x, y) \longrightarrow (x \pm 1, y \pm 1) \quad \} \text{ 4 actions}$$

→ it might just need a way to escape local min

→ Consider adding stochasticity (randomness)

e.g. Simulated annealing