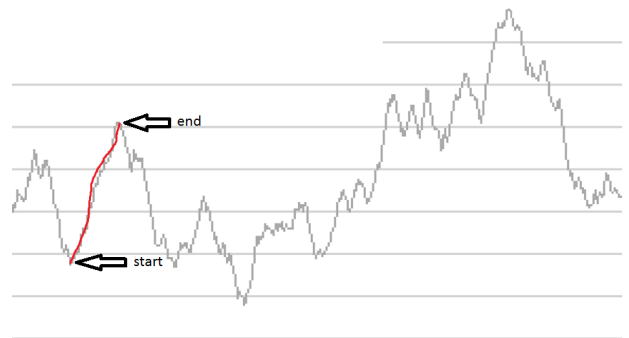# Simple Hillclimbing

New current solution = the first fitter point it finds in the neighbourhood of the current point

```
Start with any point p
repeat
  q = chooseNextPoint(p)
  if p == q then return p as final solution

chooseNextPoint(p):
for each neighbour q of p
   if q is fitter than p return q
return p
```
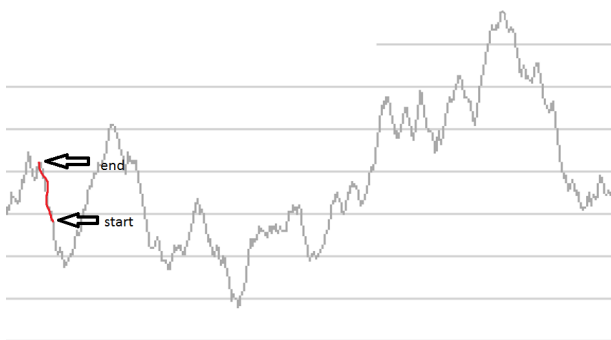
Steepest ascent hillclimbing

Simple Hillclimbing

Steepest ascent hillclimbing

Evaluates all neighbours of the current point and selects the fittest of them

```
start with any point p
repeat
  q = chooseNextPoint(p)
  if p == q then return p as final solution

chooseNextPoint(p):
fittest = p
for each neighbour q of p
  if q is fitter than fittest then fittest = q
return fittest
```

Simulated annealing

Randomly choose a different solution that is better than current solution, or even worse than current solution if temperature is higher.

Gradually reduce temperature.

https://en.wikipedia.org/wiki/Simulated_annealing

The explore/exploit balance

Hillclimbing **exploits** current knowledge of search space.

Simulated annealing also **explores** search space.

The explore/exploit balance is very important in RL (later).