Simulated Annealing - Basic Algorithm

```
Current ← Initial state
T ← a large positive value
while T > 0 do
    next ← a random neighbour for current
    ΔE ← current.cost - next.cost
    if ΔE > 0 then
        current ← next
    else
        current ← next with probability p = e^(ΔE/T)
    end if
    decrease T
end while
return current
```

$$p = e^{\Delta E / T}$$

Algorithm:

```
current ← randomly generated initial state
current_cost ← cost (current)
T ← a large positive value
while T > 0 and current_cost > 0
    neighbour ← generated neighbour of current state
    neighbour_cost = cost (neighbour)
    cost_diff = current_cost - neighbour_cost
    if cost_diff > 0:
        current ← neighbour
        current_cost ← neighbour_cost
    T = T - 1
end while
return current, current_state
```