Summer Internships 2017

Schedule Optimiser

Weekly Report - 17/07/2017-21/07/2017 (Week 2)

Trainee: Daniel Ramos

Mentor: Afonso Ribeiro

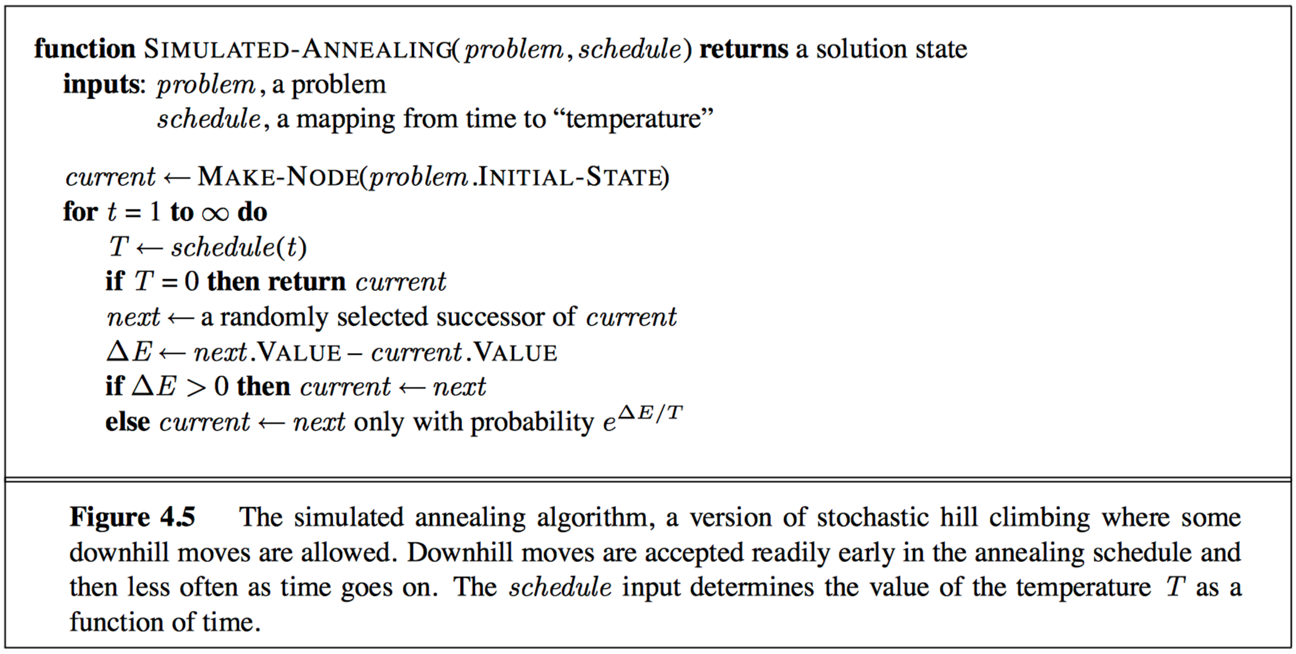
Co-Mentor: Vítor Martins

Summary:

This week main focus was on Local Search Algorithms, particularly Simulated Annealing. Simulated Annealing is an algorithm useful in hard optimization problems, it's inspired on the process of annealing in metallurgy (a process in which a material is subjected to temperature oscillations to change its physical structure).

To change the physical structure of a material, firstly it has to be heated to its melting point, this allows its atoms to move and build more stable crystals, after a while the temperature starts to be slowly reduced, so the material can retain the acquired properties.

In simulated annealing we make an analogy with this process, the temperature is kept as a variable to simulate the process. Initially the temperature is set high and then it gradually cools at each iteration of the algorithm. When the temperature is high we allow moves that might seem bad to happen frequently. This gives the algorithm the chance of jumping out of any local optimums it might find during early stages of execution. As the temperature is reduced so is the chance of accepting bad moves, allowing the algorithm to start focusing on a specific area of search (example of execution).



```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
          schedule, a mapping from time to "temperature"

  current ← MAKE-NODE(problem.INITIAL-STATE)
  for $t$ = 1 to $\infty$ do
      $T \leftarrow$ schedule($t$)
      if $T = 0$ then return current
      next ← a randomly selected successor of current
      $\Delta E \leftarrow$ next.VALUE – current.VALUE
      if $\Delta E > 0$ then current ← next
      else current ← next only with probability $e^{\Delta E/T}$
```

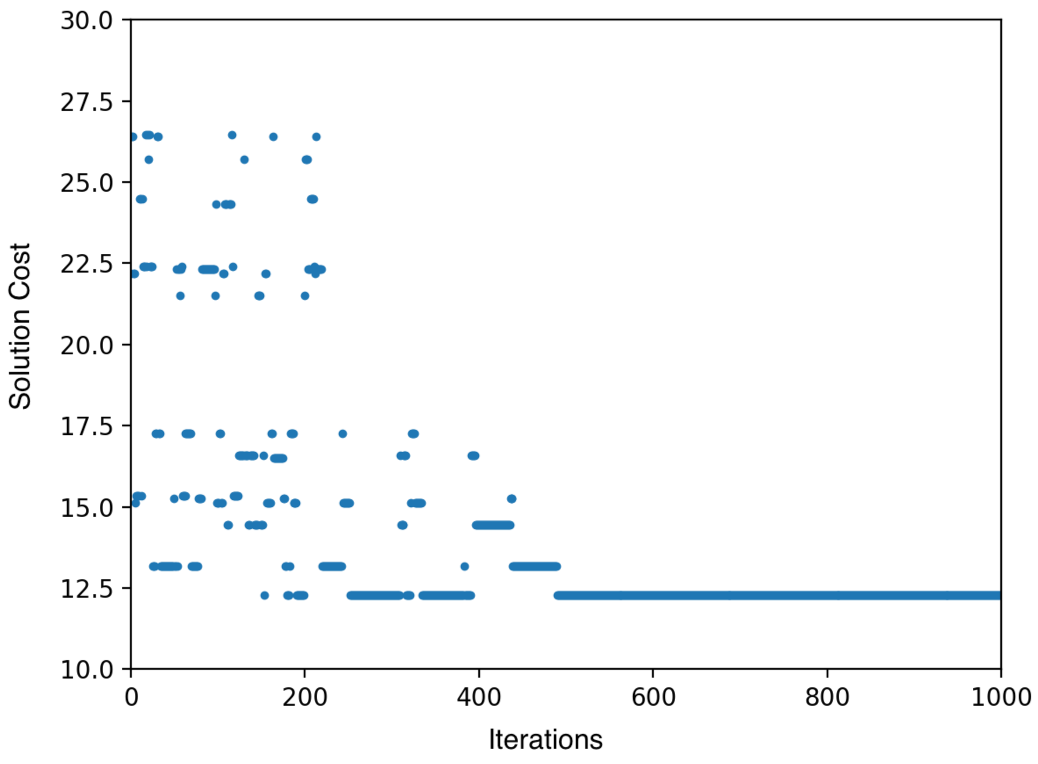
Figure 4.5 The simulated annealing algorithm, a version of stochastic hill climbing where some downhill moves are allowed. Downhill moves are accepted readily early in the annealing schedule and then less often as time goes on. The *schedule* input determines the value of the temperature T as a function of time.

Artificial Intelligence: A Modern Approach. Russell. Norvig.

To better understand the algorithm, I decided to implement a prototype of it. I used a relaxed problem (a version of the problem that is easier to solve) where the following was assumed:

- Teams don't have working days (they are always available);
- Teams don't have work areas (they can go anywhere);
- Every team can do everything.

The dataset was small and can be found in the project wiki, alongside the algorithm's implementation.
The results were as follows (the lower the cost, the better):



The parameters (initial temperature and its schedule) used on the algorithm were not ideal. The optimal solution could be found in 5-10 moves, the algorithm bounced back and forth because the temperature was set too high for the dataset size.

Completed tasks:

- Set-up the work environment;
- Analyse and choose one (or more) algorithm(s) to implement;
- Start implementing the algorithm(s).

To-do:

- Learn more about C#.
- Implement the algorithm (now in C#) and integrate it into existing codebase.