

Homework 1

Connor Taffe

September 29, 2015

The following outlines a report of Artificial Intelligence's Homework One. I include both the `main.c` solution (Compiles with `make` on *Linux*, *BSD*, and *OS X*. Very impressive 0.161s execution time for 1000 restarts on my machine. Optional Runge-Kutta classic simulation and PCG random numbers.), and the hastily programmed `C++` version which compiles on *Windows*.

I encourage you to run the `C++` version, but investigate the `C` version for more aesthetically appealing code and functional identity.

1 Compiling the code

1. Open `main.cc` in Visual Studio.
2. Hit F5.

2 Running the code

1. Hit F5 again?

3 Plot

The following is the scatter plot with the cost (y) as a function of the number of evals (x), $y = f(x)$. Seen in figure 1.

4 Report

Here follows the report sections as outlined in the spec.

- (a) The local search implements *Greedy-Descent* (*Steepest-Descent*), so it pursues the lowest error neighbors ($O(n)$ testing) towards a local minimum. Shoulders would end the search as would plateaus. Any rise in the error landscape would terminate the search. The only redeeming quality of the program is that since it randomly sampled so much, many times landing on local minima slopes, the landscape was somewhat navigated (applies to all local search).

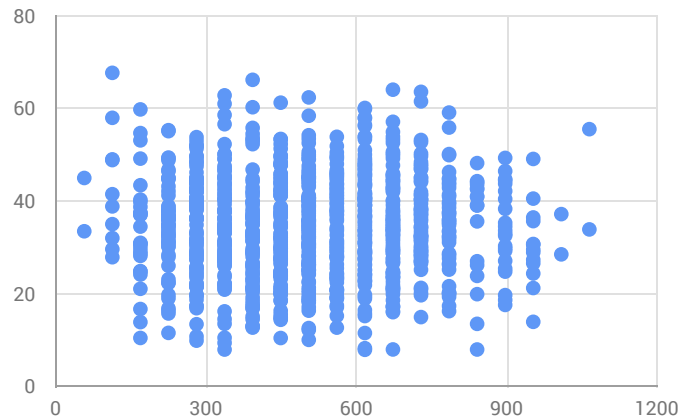


Figure 1: Scatter Plot

- (b) With 1000 random restarts and an average neighbor-follow length of around 9 hamming distances, I would assume the local search had a fairly good navigation of even such a large space.
- (c) An intermediate solution is a solution that is not a terminating local maxima. These are solutions that have one hamming distance neighbors with a lower cost. They are on a slope of the error landscape.
- (d) The error function defines the error landscape, or how the local search moves throughout the set of possible solutions.
- (e) Possibly, if it provided a more easily followable slope for local search. For example, an error function which creates many local minimas but one large global minima which are unassociated (rigid, sharply pitted, one deep pit) compared to one which has few far reaching slopes leading to local minima, one of which being global (smoother, bowl-like). The second is much better for local search, giving it a much larger error landscape surface to land on which leads to the global maxima.

5 Suggestions

The “Programming Language Policy” section of the syllabus outlines which “base languages” are approved for use in this course. `C` is listed as an acceptable language, which (to me) suggests that a recent standard version of `C` would be appropriate for use.

As the required environment (Visual Studio on Windows) does not actually have a `C` compiler available, it is fairly impossible to use a `C` language program as a submission. To be fair, the `C++` compiler available will compile the `C` standard version from 1989, but I know of nothing except `git` that is programmed in `C`

'89. Therefore, perhaps the accepted language specification should include this information.