

Optimization is Hard, but Hillclimbing is Easy – Right?

Johannes Lengler
Institute of Theoretical Computer Science
Department of Computer Science
ETH Zürich, Switzerland
johannes.lengler@inf.ethz.ch
<https://www.ti.ethz.ch/>

Abstract

Hillclimbing is an integral part of every optimization algorithm. One would expect that our standard hillclimbing algorithms can solve nice, unimodal functions efficiently. For example, strictly monotone functions: consider the search space $S = \{0,1\}^n$ of n -bit strings; then a strictly monotone function $f : S \rightarrow R$ is a function that satisfies $f(x) < f(y)$ for all different strings x, y in S such that $x_i \leq y_i$ holds for all $1 \leq i \leq n$. Strictly monotone functions are the easiest imaginable testbed for hillclimbers: The function f always prefers one-bits over zero-bits, so there are always short and clearly labelled paths from any start point to the unique global optimum 11..1. Surprisingly, it has become clear in the last years that many of our standard hillclimbers fail miserably on some strictly monotone functions. In many instances, they need exponential time to find the optimum. This devastating result has deepened our understanding of hillclimbing techniques, and has given new insights into the mechanisms that are behind some hillclimbing techniques, like selective pressure, step size policies, and population effects. These insights have also improved our understanding of optimization beyond hillclimbing, in particular of optimization in dynamic environments.

Keywords: Optimization, Hillclimbing