Hill Climbers and Microbial GAs

Sector: Artificial Intelligence and Adaptive behaviour.

Author: Tonino O'Leary
CandNo.197524
Sussex University Informatics Sector.

May 2020

Abstract

The principal aim of this paper is to explore the concepts of hill climbers and microbial Genetic Algorithms. Two particular questions are addressed: "What are the differences between a Hill climber and microbial GA?" and "How do factors like mutation rate (parameters) affect performance?" The findings will determine which optimisation techniques perform best. The comparative aspects of the paper will focus on the core components of the algorithms, Including mutations rates, crossover rates and a raw comparison of the Hill climbers versus GAs.

1 Introduction

Hill climbers and GAs are fundamentally the same. They are optimisation techniques used to find the best solution to a given problem. The difference lies in their approach. What are the quantitive differences between the two algorithms that makes their performance different?

1.1 The Problem

A given scenario is as follows (proposed problem):

The knapsack problem is a combinatorial optimisation problem. The knapsack has positive integer volume (or capacity) V. There are a series of n items that can be placed in the knapsack. Item i has a positive integer benefit denoted Bi and a positive integer volume denoted Vi. In this problem, we will consider each item available once. The inclusion of an item within the knapsack will be represented in binary form (0 or 1).

The goal is to maximise value:

- $\sum_i^N B_i$

under the given constraint (the knapsack's maximum capacity):

- $\sum_i^N B_i$

Item	a	b	c	d	e	f	g	h	i	j
B	5	6	1	9	2	8	4	3	7	10
V	3	2	4	5	8	9	10	1	6	7

Table 1: Given data for the proposed problem

2 Hill Climber and Microbial

2.1 What makes up a hill climber?

The hill climber takes the form of a genotype G (our list of binary ins and outs of the bag). Each of these binary options is thought of as a phenotype Gi. With each iteration a new genotype is created from the current one and a percentage of $Gi...Gn$ are mutated. If G performs better than the G without breaking the given constraints, it replaces it. This occurs for n iterations.

Fundamentals of a hill-climber:

Fitness function:

- The fitness function determines how well the knapsack is packed.
- We will consider the following:
  - The total benefit of the packed items
  - Whether the capacity has overflowed

Mutation function:

- For our mutation function, we will randomly change the "states" of our genotype. We will do this with a given mutation rate (as a percentage).

"Birth" function:

- This will manage our genotypes.
- If the new genotype is worse than the genotype it was mutated from, it will be discarded. Otherwise, it will become our main genotype.

Stopping condition:

- stop when you flat out or set the number of iterations

The Microbial algorithm is very similar to this. It uses a population of individuals. It picks an individual and its neighbour at random based on a neighbour boundary. The best individual is then copied over to the weaker one by a crossover percentage and mutated.

To analyse these algorithms we use the matplotlib library and its .pyplot capabilities to plot results and experiment with the algorithms.

2.2 Local Maxima

The search space of a hill climber can be thought of as a series of hills, each with their own local maxima. In an ideal situation the hill climber will reach the best local maxima available but it may get stuck on non-optimal local maxima. This is where tweaking mutation rates and iteration numbers will provide the best opportunity for achieving the optimum local maxima.

2.3 Growth Rates

The growth rates of hill climbers can differ drastically and often they are not stable. The genotype(s) are set at random, giving them a chance to be high on instantiation. This gives the genotype the possibility of growing faster than normal or getting stuck in a local maxima for a period. On the other hand, the Microbial GA often has a high initial fitness. This is because it is a population-based algorithm. The top individual's performance gives a higher and more stable starting point for growth.

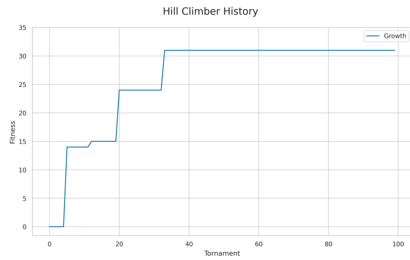

Figure 1: Growth of an instance of a hill climber over 100 generations, with a mutation rate of 0.5

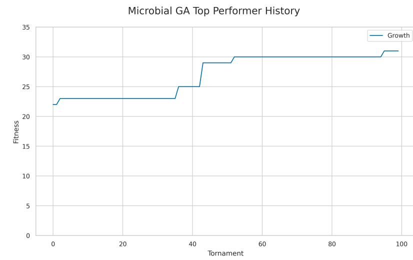

Figure 2: Growth of an instance of a microbial over 100 generations, with a mutation rate of 0.5

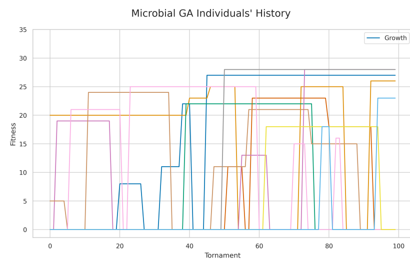

Figure 3: Growth of individuals withing a Microbial GA over 100 generations, with a mutation rate of 0.5

At first glance, both algorithms seem quite similar but the microbial is erratic in comparison to the hill climber. Its individuals are constantly changing/fluctuating (seen in Figure 1), unlike the linear trajectory of the hill climber.

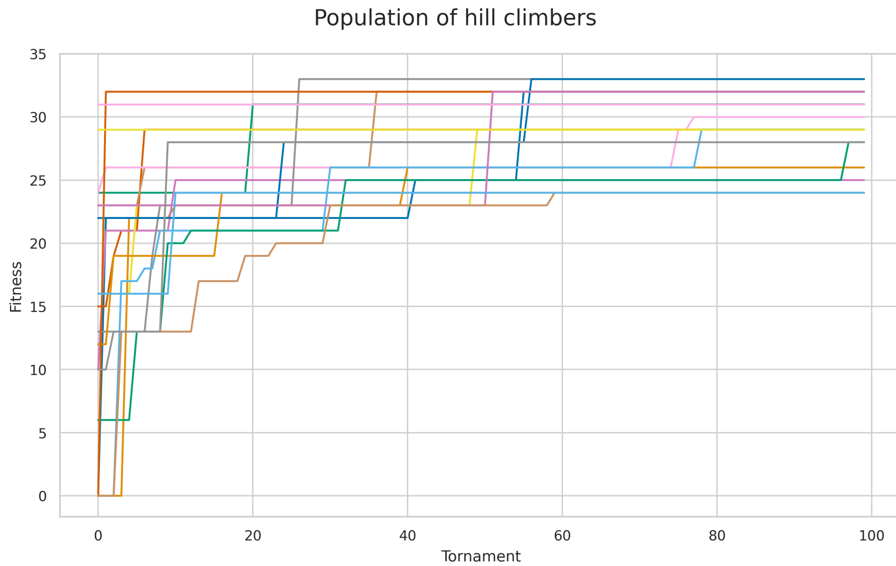

Figure 4: Plot of the growth of fitness for a series of hill climbers over 100 generations, with a mutation rate of 0.5

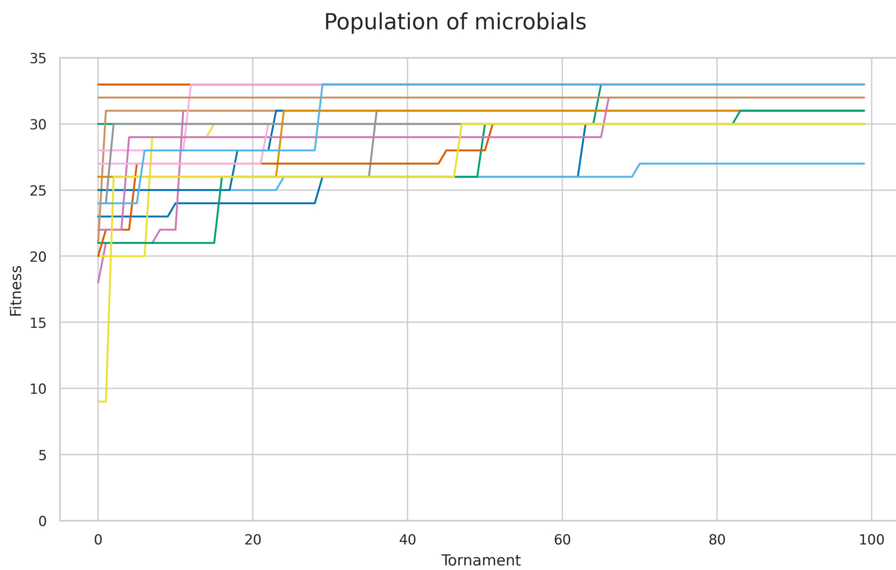

Figure 5: Plot of the growth of fitness for a series of microbials over 100 generations, with a mutation rate of 0.5

The trajectory of hill climbers and microbials are very similar in the short term. They both increase in a purely positive trend overall. To see the difference in optimisation more clearly, an exploration of different mutation rates and tournament numbers will be needed.

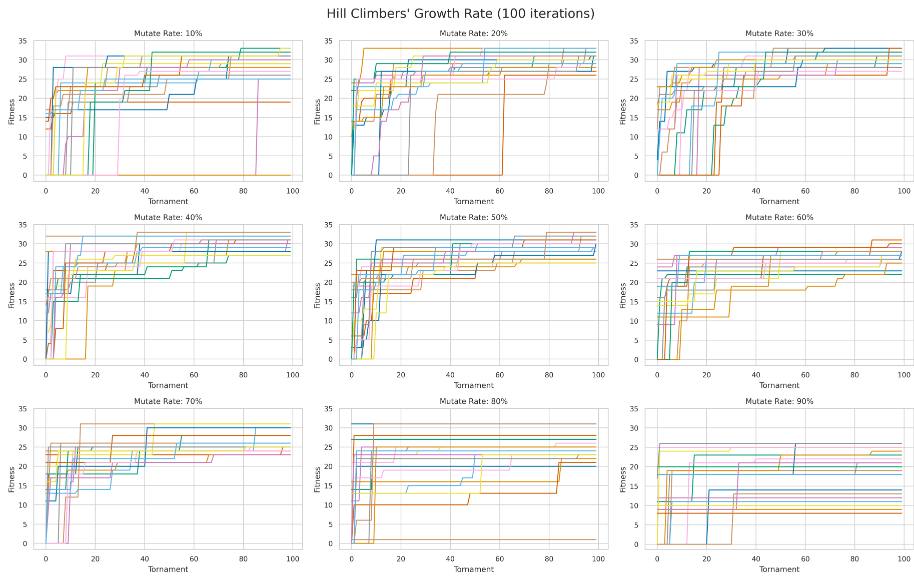

Figure 6: Plots of the fitness growth of a series of hill climbers over 100 iterations. Shown using a variety of mutation rates.

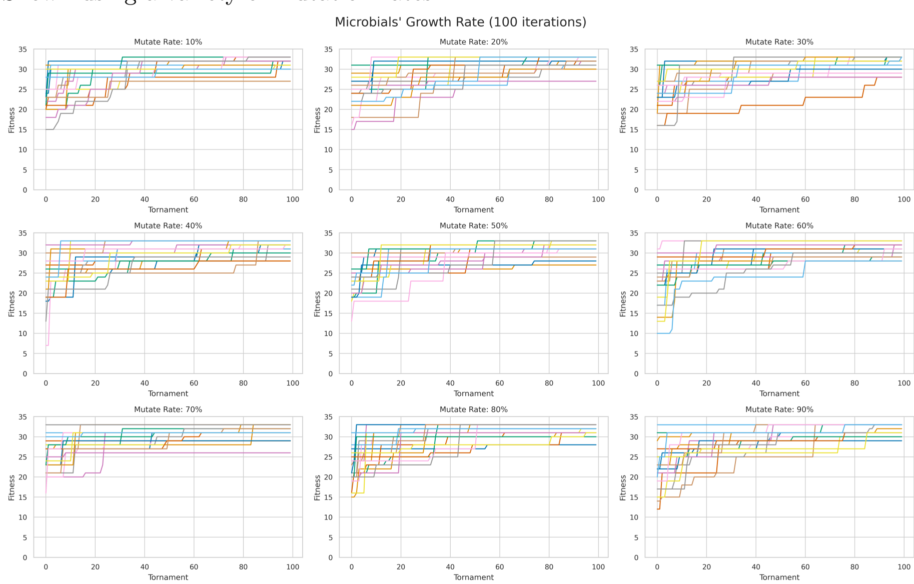

Figure 7: Plots of the fitness growth of a series of microbials over 100 iterations. Shown using a variety of mutation rates.

Stability is a big difference in the optimisation of the two algorithms. The hill climber is much more unstable in relation to shifting parameters. In Figure 6 the most beneficial mutation rate is around 0.3. Anywhere above a mutation rate of 0.7 or below 0.3 you start to get severe inconsistencies in the results. In figure 7 these inconsistencies aren't as apparent and they do not differ much from one to the other. Over a greater number of iterations, these inconsistencies can be seen with more clarity as shown in figures 8 and 9.

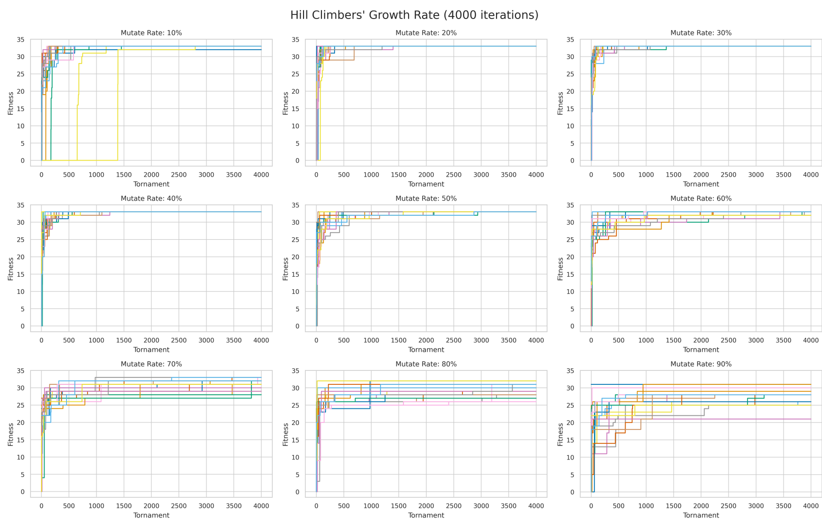

Figure 8: Plots of the fitness growth of a series of hill climbers over 4000 iterations. Shown using a variety of mutation rates.

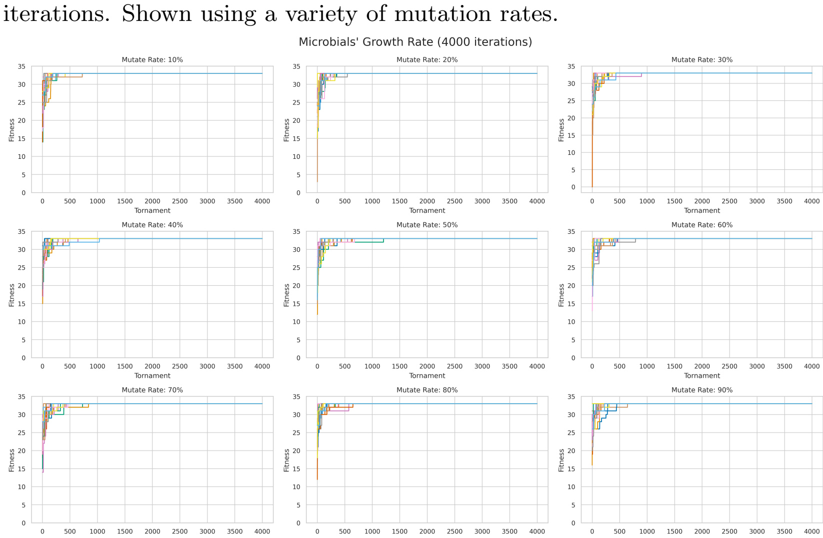

Figure 9: Plots of the fitness growth of a series of microbials over 4000 iterations. Shown using varying mutation rates.

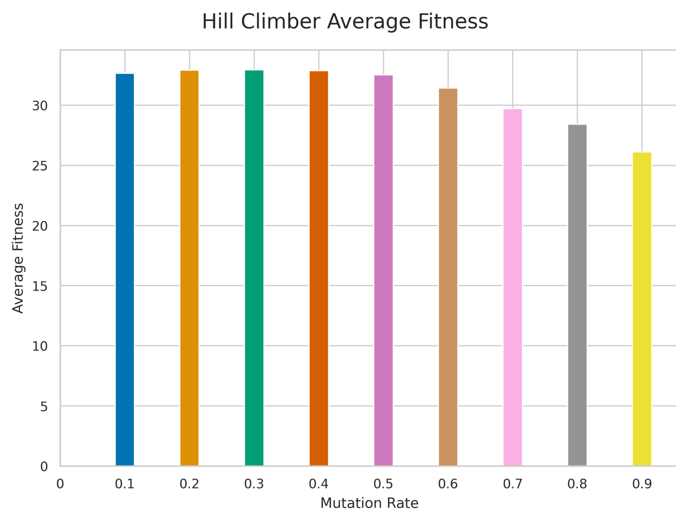

Figure 10: Plots the average fitness of 150 hill climbers over 1000 iterations for a variety of mutation rates.

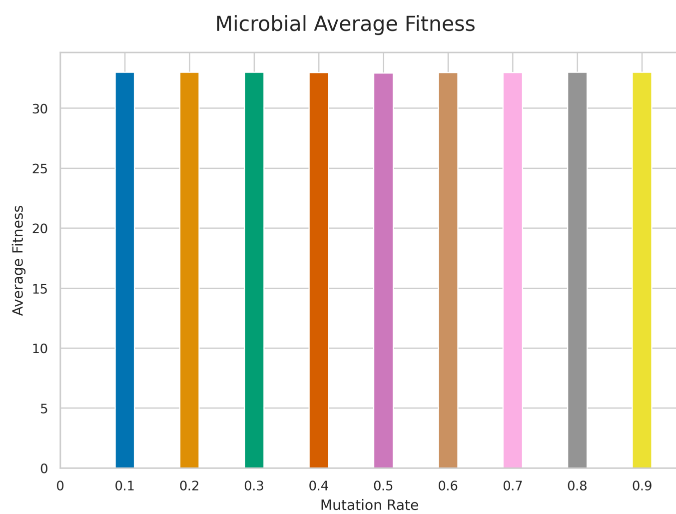

Figure 11: Plots the average fitness of a 150 microbials over 1000 iterations for a variety of mutation rates.

The average fitness for the hill climber fluctuates whereas the fitness of the microbial was almost always at optimum.

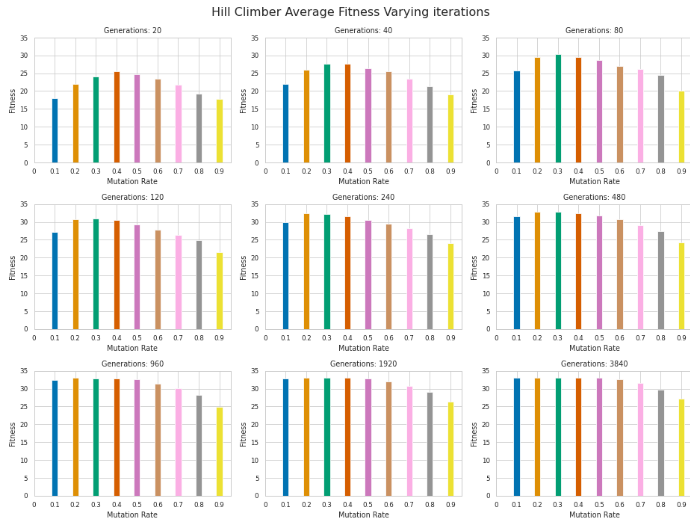

Figure 12: Plots the average fitness of 150 hill climbers over different iterations for a variety of mutation rates.

Figure 13: Plots the average fitness of 150 microbials over different iterations for a variety of mutation rates.

The optimum was much easier to achieve in lower numbers of iterations for the microbial. The hill climber changed a lot with different mutations. Therefore, there was increased reliability in the microbial algorithm. The number of iterations could be even further reduced with experimentation with crossover rates values. Overall, this makes it easier to optimise than the hill climber.

3 Crossover rate

Comparing features that both algorithms include has kept the comparison balances, but additional features that aren't available in the hill climber give the microbial a greater opportunity for fine tuning and optimisation. For instance the crossover rates allow the microbial to pass on its 'genes' from one individual to another with mutations within a given mutation rate.

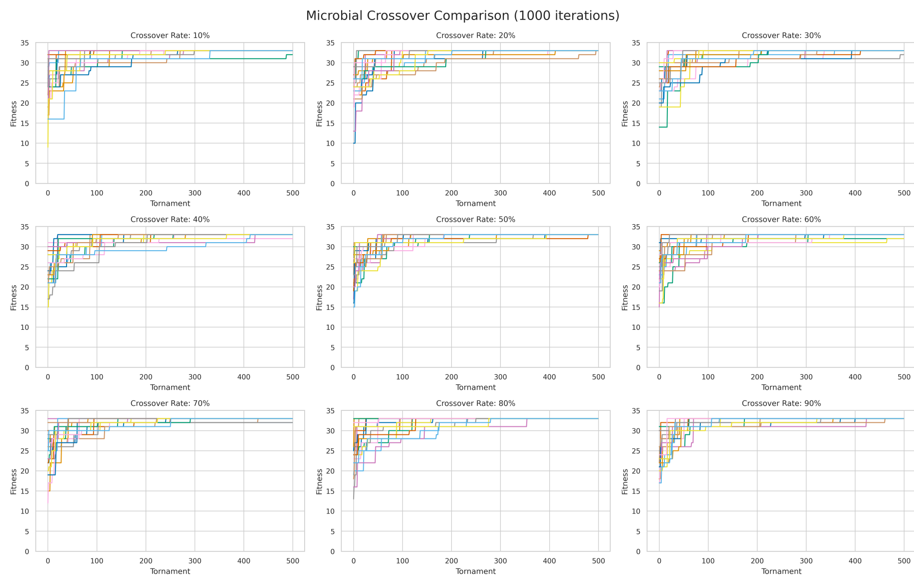

Figure 14: A plot of the fitness growth of a series of microbials over 500 iterations. Shown using varying crossover rates.

4 Discussion

The microbial and hill climber performed generally as expected. Stability was found in the microbial due to its fundamental structure. A population provided a group of evolving individuals that fed off of the stronger of the group. The hill climber being non-population based had a linear dynamic. Its results varied much with changes to parameters, whereas unexpectedly, the microbial kept somewhat stable even when the parameters where changed. In choosing an optimisation technique, it's worth considering that Microbial GAs provide greater stability and will reach optimum local maxima in fewer iterations, being invariant to parameters to some extent. Cross over rates also provide better optimisation for the algorithm. Therefore, the microbial provides a more reliable solution to a local search optimization technique than the hill climber. Generally, a hill climber's iteration will have a faster running time but benefit isn't sufficient to override the the positives of a microbial.

The following was used for information gathering and research [1]

References

[1] Inman Harvey. *The Microbial Genetic Algorithm*. Sussex University.