Comparison between Microbial GA and RMHC

Contents

[Abstract 1](#_Toc4354819)

[The Problem 2](#_Toc4354820)

[Solutions 2](#_Toc4354821)

[Genotype 2](#_Toc4354822)

[Fitness Function 3](#_Toc4354823)

[RMHC 3](#_Toc4354824)

[RMHC mutation rate 3](#_Toc4354825)

[Microbial GA 4](#_Toc4354826)

[Microbial Mutation Rate 4](#_Toc4354827)

[Microbial crossover rate 5](#_Toc4354828)

[Hybrid 6](#_Toc4354829)

[Comparisons 6](#_Toc4354830)

[Large problem set 6](#_Toc4354831)

[Small problem set 6](#_Toc4354832)

[Conclusion 7](#_Toc4354833)

[Bibliography 7](#_Toc4354834)

# Abstract

I compare 3 algorithms use for solving the 0-1 knapsack problem. The RMHC(random mutation hill climber), microbial and hybrid algorithms are compared over 2 example problem sets. A random mutation hillclimber performs well at exploring a problem space thoroughly (Mitchell, 1998), better suited to problem spaces with one peak. The microbial can span large sections of the problem space. My hybrid algorithm is an attempt to utilize the strengths of both and using RMHC properties to slow convergence of a microbial to increase its viability as a general solution (Hart & Belew, 1996). I test over varying population sizes and multiple problem sets to get a good grasp of the algorithm’s performance during my comparison.

The main investigation I perform in this report is to explore how viable a hybrid of hill climbing and microbial GA is. My intuition is that the hybrid can use the Microbes ability to explore large areas of a problem space while utilising elements from the hillclimber to control convergence. My findings where counter intuitive, while my assumption was that the hill climber’s attributes would be initially useful in the early stages of the hybrid, I found that the algorithm performed better when the microbe was used in the early stages to quickly find possible peaks then the hillclimber was used in the later stages to explore those peaks.

# The Problem

The problem I used to perform this investigation is the 0-1 knapsack problem.

The 0-1 knapsack problem gives a set of items which each have a weight and value, the solution needs to maximise the combined values of a subset of the items while not letting the combined weights exceed a maximum weight value.

I used 2 problem sets, a larger one to test my results and a smaller one to validate my observations. The large set contains 100 items in which creates an incredibly large and rugged problem space. There is possible solutions to the large problem set and for the small set.

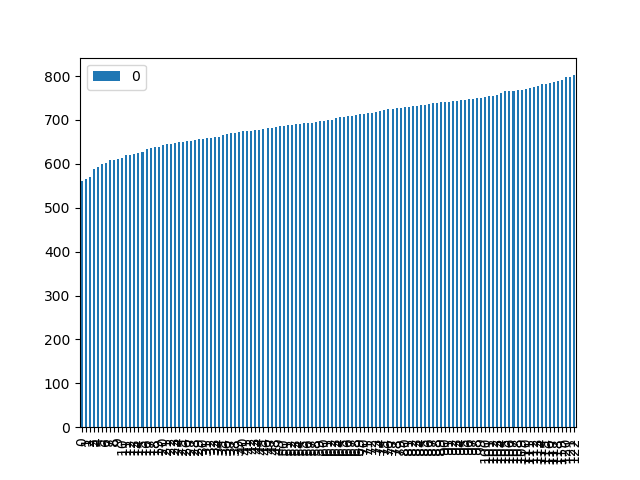
The solution space is particularly rugged, this can be shown below from the results of running 300 separate hill climbers with a small mutation rate over 3000 generations. Below is a graph showing all the different peaks identified during this test.

Figure 1

The set of all solution space peaks discovered

As you can see in figure 1, with a population size of 300, over 3000 generations, a large set of peaks were found.

# Solutions

### Genotype

The genotypes for my solution algorithms where encoded using binary mapping, with one binary digit per item given in the problem set. A 1 indicated the item is contained in the solution and a 0 indicates it is not.

The reason I mapped a binary genotype is because multiples of items are not allowed in the solution of the problem, this was one way to solve that while keeping a simple minimalist genotype.

### Fitness Function

My fitness levels allow for negative fitness, this is so I can start from a completely randomised set of hill climbers where they do exceed the total weight for the problem. I did this to ensure they can span as much of the potential problem space to begin with.

It is important to note that the performance of my graphs show what appear to be large jumps in fitness, but this is actually the fitness of the solutions transitioning from negative fitness to positive fitness, because positive and negative fitness are measured differently. This is not an issue in comparison, because a positive solution will always beat a negative solution, it does not affect the comparison if they are measured differently.

Allowing for negative weights is necessary when performing tournament recombination with my microbials. Without negative weights, the initially randomised population has no way of fairly deciding winners through tournaments.

## RMHC

Below is a pseudo code description of my RMHC:

My RMHC populations run as stochastic hill climbers running parallel to each other. Hill climbers are good at zeroing in on different optimal regions of the search space (Mitchell, 1998), this makes them a good base case comparison for my microbials. I have tested a range of populations due to the hill climber showing its strength with larger populations.

The hill climber’s mutation rate is mapped to the genotype as a percentage chance of mutation per each gene. The climber only mutates if the new mutation has a better or equal fitness. The reason I had them mutate if the fitness is equal is because this can potentially give them a small ability to traverse the problem space horizontally to further avoid local optima where possible.

### RMHC mutation rate

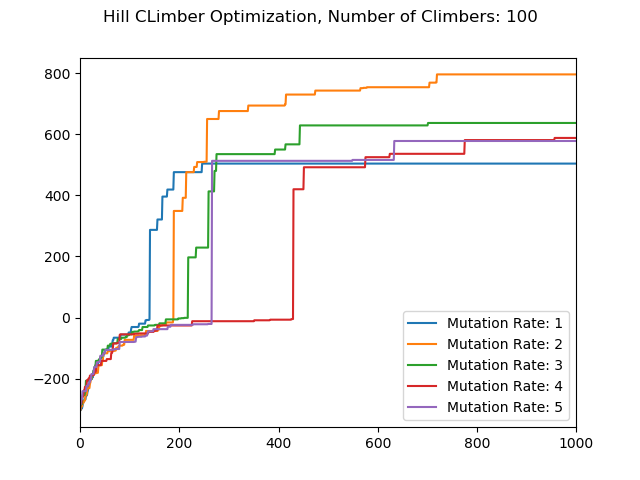
I performed tests with the intention of optimizing my RMHC for the knapsack problem by altering the variables. Each set of tests was performed on the same original seed of starting values to ensure a fair comparison. Multiple tests were done with the same results, the graphs are just examples. The optimization observations were proved on a separate problem set.My optimization test for the RMHCs were successful. I was able to identify a mutation rate that consistently outperformed others in both large and small populations.

Figure 2

RMHC mutation rate comparison

Pictured in figure 1 are examples of my tests where a mutation rate of 2 clearly outperformed other mutation rates. While these tests were performed on my initial knapsack problem of 50 items, I replicated them with the same results on a secondary set of 20 items. With less items however, the variance in performance was less apparent.

It is worth noting that initially a mutation rate of 1 performed best but in later generations was outperformed significantly by a rate of 2. This likely due to the 2 ability to make bigger jumps when mutating 2 genes per mutation.

## Microbial GA

Below is a pseudo code description of my microbial GA:

:

I used a population of microbial Gas to use as a comparison with my hill climbers. These Microbes will have a crossover probability and a mutation probability.

For my selection process I will use tournament style selection. It works well as a good minimal style, easy to implement selection method for microbial GAs (Harvey, 1996). The way I use microbial inheritance with a tournament, provides free elitism (Harvey, 1996). The random pick of my tournament selection, while to do with ranking, will provide variance to the picks (Harvey, 1996). My microbial GA does not overwrite anything when one member of the population is paired against itself in a tournament, this is done so maximum fitness is not lost over time.

My microbials will be able to cover a large search space while still benefiting from elitism to maintain the best individuals and using mutation to explore this space.

I performed tests with the intention of optimizing my MGA for the knapsack problem by altering the variables. Each set of tests was performed on the same original seed of starting values to ensure a fair comparison. Multiple tests were done with the same results, the graphs are just examples. The optimization observations were proved on a separate problem set.

For the microbial, the rates are both measured as an integer out of 100 which equate to a percentage of the crossover/mutation occurring per gene during a recombination. For my RMHC the mutation rate is a fixed amount of mutations performed by flipping a random gene per mutation.

My optimisation of the microbial GA was partially successful, while I was able to identify parameters which had the least variation in results, I was unable to determine one set of parameters that clearly and consistently outperformed others.

It is likely that there is no one rate that performs best, due to the nature of the problem space and seeding a set of tests with randomised population.

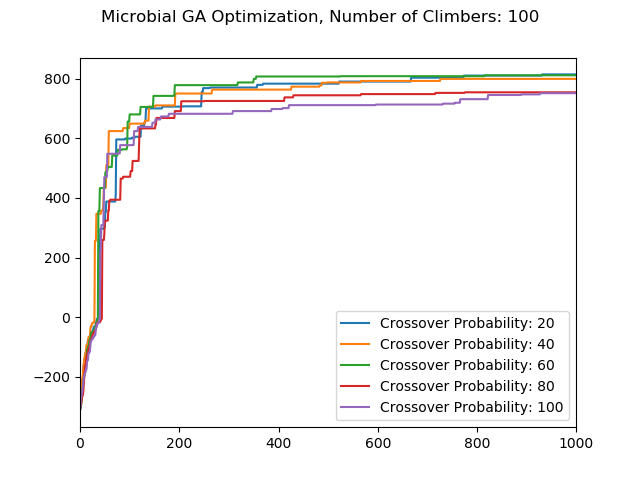
### Microbial Mutation Rate

Figure 3

Comparison of microbial GA mutation rate tests

In figure 3 is an example of a test performed while investigating mutation rate for microbes. During my tests for optimising mutation rates, I found there was a high amount of variance for optimal rate when using a low number of climbers and less variance when using a higher number of climbers. The most consistent performing rate is a mutation rate of 2% in my 50 gene solution. This equates to roughly 1 mutation per crossover.

### Microbial crossover rate

My optimising rate tests were more successful than mutation rates, I was able to determine that a high percent crossover rate performed better with smaller populations rates around the 40% mark performed better with large populations. There was variance to these tests that likely depend on the original randomised seed data, however, this trend still followed.When testing on a small population, the higher crossover rates of 80% and 100% performed best. Figure 4 shows examples of my tests on crossover rates for large populations. The most consistent performing rates was 40% during these tests. The results coincide with the crossover rate recommended in (Harvey, 1996) when using microbial algorithms.

Figure

Comparison of tests on crossover probability

## Hybrid

Below is a pseudocode description on the hybrid implementation I settles on:

:

The third algorithm I implemented is a hybrid between RMHCs and Microbials. This investigation was to see if I could use the best properties from both to achieve better results

Initially I implemented the hybrid by running a population of RMHCs that perform microbial recombination x percentage of the time, with x being a controlled variable. My intentions for this investigation are to utilise the ability of hill climbers to search their search space thoroughly (Hart & Belew, 1996) while taking advantage of the microbials ability to bridge gaps in local optima and as a way of controlling the microbials convergence. Problem space dependant, GA augmented hill climbers have been shown to outperform GAs alone (Hart & Belew, 1996).

I implemented different variations of the hybrid. First, I attempted to have a microbial tournament probability rate which was mildly successful. However, I later settled on having the first half of the generations microbial, ending with the second half as hillclimbers. This method came about through multiple test of many possible implementations. I believe this is counter intuitive however my reasoning is discussed in the comparison

# Comparisons

Figure 5

Comparison using large problem set

### Large problem set

Figure 5 pictures the comparison I ran on my algorithms using the large problem set.

When the large problem set was used, the RMHC consistently underperformed early on to a large degree, and generally underperformed overall.

The Hybrid method and microbial method both performed well, often with quite similar results.

I repeat the tests with a maximum weight value of 200, from this I was able to determine that the results are similar regardless of the change in maximum weight.

### Small problem set

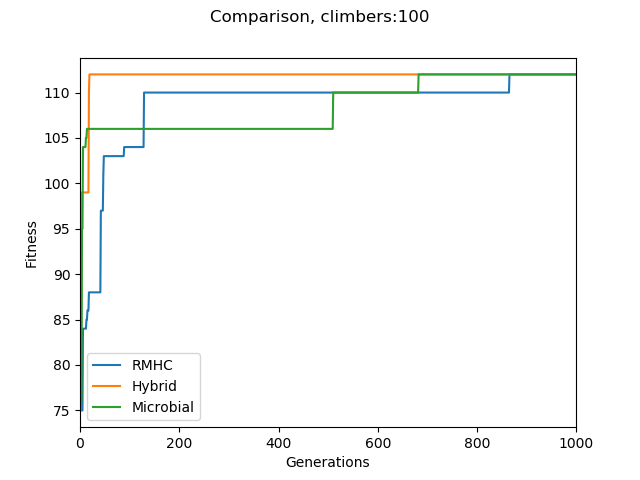


Figure 6

Comparison using small problem set

In figure 6 are the results of my comparisons on the smaller problem set.

The smaller problem set had an optimal solution of 112, for these comparisons the best performing algorithm was the first to have a member of the population to reach this maximum. For this smaller problem set, the RMHC and the hybrid consistently outperformed the microbial.

### Conclusion

The microbial was much better than the RMHCs when a large problem set was used, this is likely to do with the ruggedness of the solution space, the microbials were able to explore many peaks due to the nature of their crossovers and mutations.

The RMHCs performed best in the smaller problem set, this is likely due to their innate ability to their gradient climbing nature.

The microbial in nearly all examples was the fastest algorithm to reach a positive weight in a large solution space. The hybrid performed best when convergence was limited at the end. Taking advantage of the microbials ability to quickly find good solutions, then utilising traits from the RMHC to explore these peaks. I believe with further investigation; a heuristic can be determined that to control the transition from microbe to RMHC.

The hybrid was shown to be a much more robust solution than either the RMHC or the microbial, due to its ability to perform well in the 2 vastly different problem spaces. Whereas the other algorithms each performed better in 1.

It appears that the success of each algorithm does vary based upon the problem set used, however during all my tests, the hybrid was the most overall consistently performing algorithm over all problem sets. These findings are supported by other investigations done on GA modified hill climbers (Harvey, 1996). These tests show that there is the potential for GA modified hill climbers to outperform both hill climbers and pure genetic algorithms. My results showed that the hybrid was a overall a more robust algorithm capable of handling different scale problems well, however further tests are needed prove this.

# Bibliography

De Jong, K. & Sarma, J., 1995. *On Decentralising Selection Algorithms.* San Francisco CA, Morgan Kaufmann .

Hart, W. & Belew, R., 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms.* Boston: Westview Press.

Harvey, I., 1996. *The Microbial Genetic Algorithm,* Brighton: School of Cognitive and Computing Sciences.

Mitchell, M., 1998. *An Introduction to Genetic Algorithms.* reprint edition ed. Cambridge, Massachusetts: MIT Press.