

Self-Adaptive EA analysis

For this assignment, I decided to make mutation self-adaptive. I achieved this by implementing the *Uncorrelated Mutation with n step sizes* algorithm as described on page 60 of the textbook. In this implementation, each allele in my genotype is assigned a σ value that acts as the standard deviation value to a normal distribution. This sigma co-evolves with the mutation rate, allowing each allele to tune its own perfect mutation rate.

Overall, 7 experiments were carried out, namely being: a control experiment on datasets 1&2, a 1-elitist restart without self-adaptivity performed on datasets 1&2, a 2-elitist restart without self-adaptivity using only dataset 1, and finally a self-adaptive mutation experiment without restarts performed on datasets 1&2. In the following paragraphs, we will go over each experiment type in its own section and explain what was effective and what was not.

For the control experiments, they turned exactly as one would expect: a nice curve that converges. As is shown in graph 1, the average fitness quickly jumps up and then plateaus due to convergence.

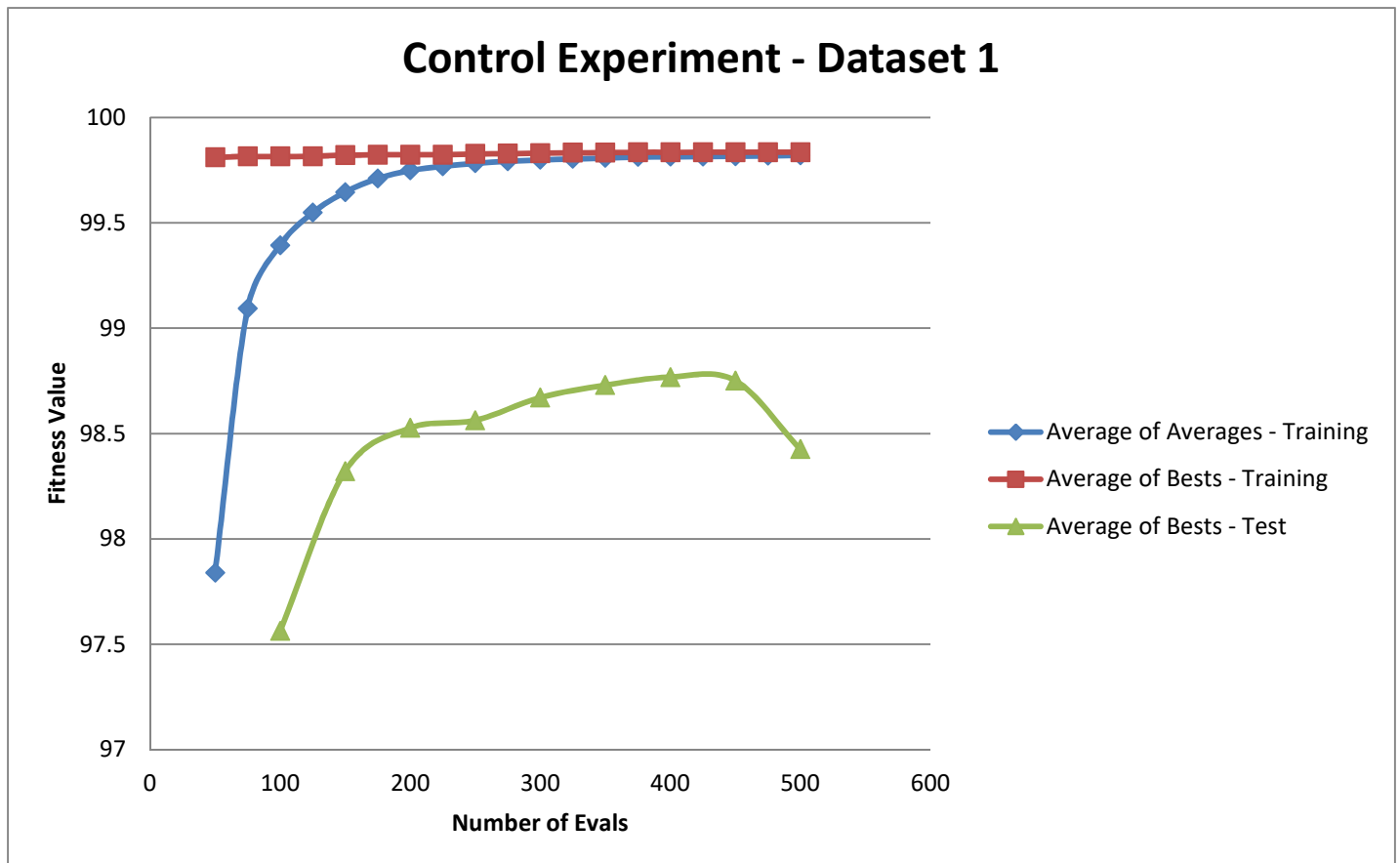


Figure 1 – Control Experiment on dataset 1

The actual experimentation came with the 1-elitist restart experiments performed on datasets 1 and 2. Unfortunately, the trend shown to us by the r-elitism on dataset 1 is lost because of a configuration error that set the number of stagnant generations to determine convergence too high – therefore the algorithm never actually restarted. On the second dataset however, the algorithm restarted multiple times and the effects were quite interesting. Every peak on the average fitness line shows us where the population was deemed to have converged, was nuked, and then started over again. However, every time a restart happens, the peak is higher than it was previously, indicating that r-elitist restarts help to push the population towards a better solution.

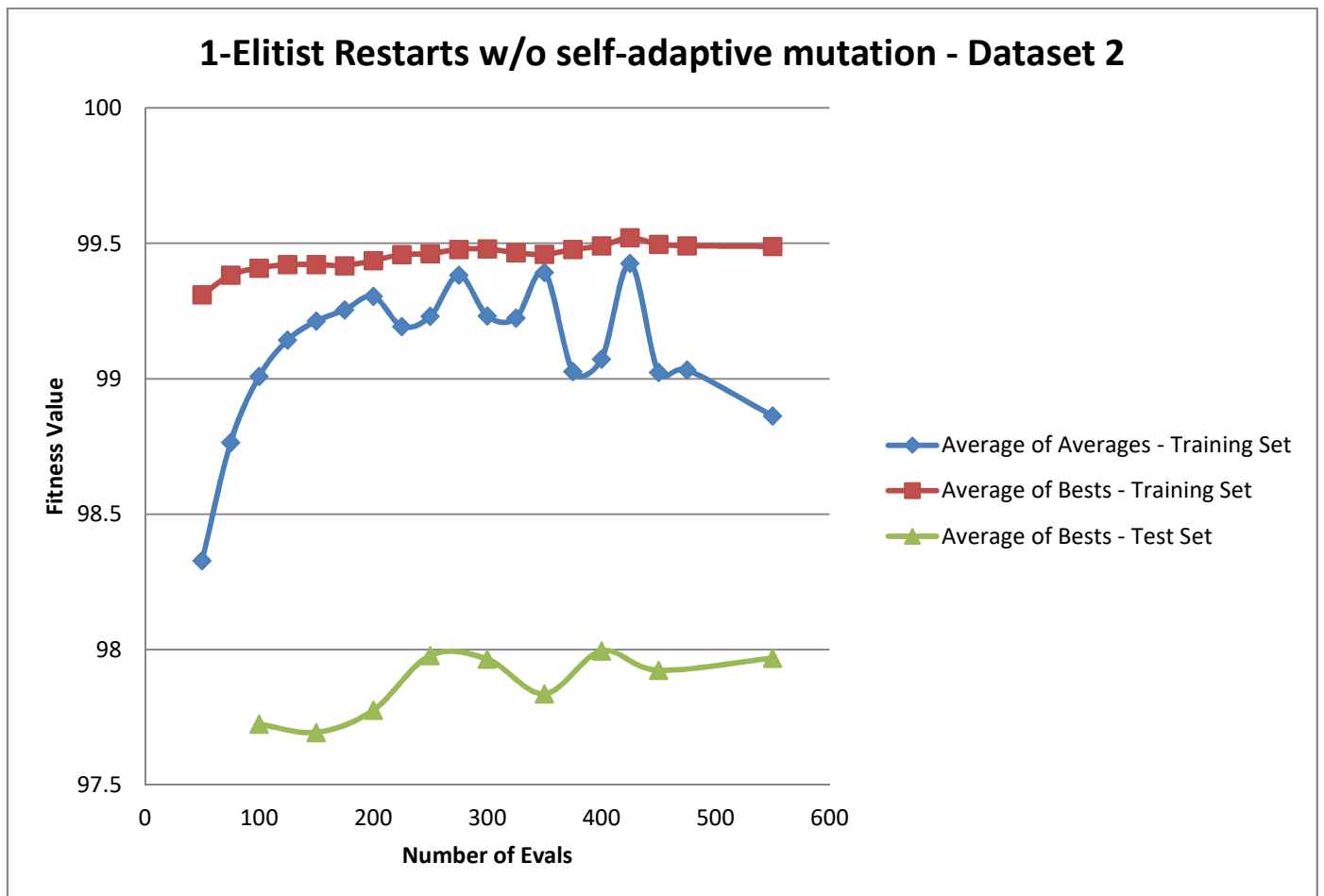


Figure 2 – 1-Elitist restart performed on Dataset 2

Strangely enough, the self-adaptive mutation didn't really seem to have much of an effect on the overall population. It did, however drastically slow down the rate of convergence. It seems that altering mutation as convergence inches closer helps to stave it off and maintain a healthy diversity in the population

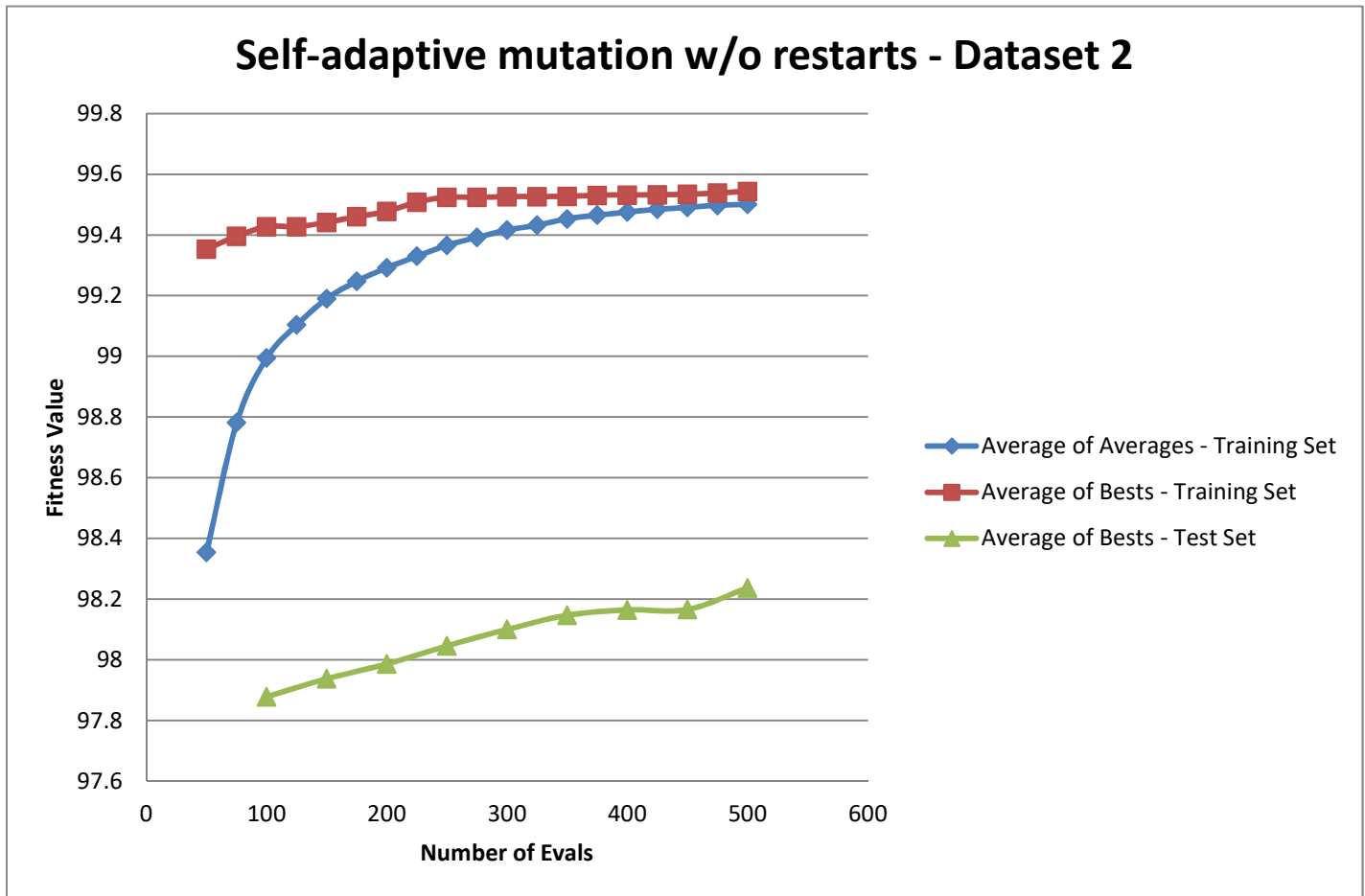


Figure 3 – Slow convergence using self-adaptive mutation rate on dataset 2

Now on to the important part, were any of the methods better than any other ones? Well, in short, no. Not a single one of my statistical comparisons was able to prove that using fancy r-elitist restarts and self-adaptive mutation was *actually* better at all. My guess is this is due to the relatively low number of generation that each EA went through. Since the number of fitness evaluations was capped at 500, none of the algorithms really had much time to get off the ground and really let the differences in their implementations show. If these experiments were run again with a cap of 5000 evaluations instead of 500, then my guess is that we would see some wildly different results. All statistics can be found in the included *Statistics.xlsx* file. In there, there are f-tests to determine if the variances can be assumed equal, and then t-tests to attempt to reject the null hypothesis and prove that the algorithm with the larger average is indeed better.