1. Problem 1

1.1. In this study, several typical parameter sets have been chosen to explore their effect on solution quality and convergence speed (Pedersen, 2010). They include the following: $\{(w = 0.9, a_1 = a_2 = 2)$, $(w = 0.7, a_1 = a_2 = 2.0)$, $(w = 0.7, a_1 = a_2 = 1.4)$, $(w = 0.7, a_1 = 1.6, a_2 = 0.6)$, $(w = 0.5, a_1 = a_2 = 2)$, $(w = 0.3, a_1 = a_2 = 2.0)$, $(w = 0.1, a_1 = a_2 = 2\}$. Additionally, a termination condition needs to be specified to ensure convergence, which is specified as follows

$$|f(p_k^g) - f(p_{k-q}^g)| \le \epsilon \qquad q = 1, 2, ..., S \tag{1}$$

where $f(p_k^g)$ is fitness of the global best particle at the kth iteration. In other words, the optimization terminates if for some specified S number of iterations, the difference between each subsequent pairs of global best fitness is less than some error bound. This means that termination is sufficient if there is no significant changes in the fitness, which suggests convergence. In this study, the experiment parameters are set as follows, for the sphere function , the search space dimension is set to: 10, 30 particles, 1000 iterations, error bound: 0.0001. For the Rastrigin function, the iterations is set to 5000 with all other parameters fixed. A total of 30 simulations will be run for both the sphere and Rastrigin function and average results are reported. It is worth noticing that the S parameter shall only be set after an inspection of the range of fitness for each set of parameter. This is because smaller values of c_1 and c_2 will lead to a much faster convergence as there is less emphasis on local and global best attractions. However, this means setting the same S for small and large c_1, c_2 will not be suitable since it could either terminate early with large differences in convergence value, or terminate many steps after convergence is already reached. From Figure 1, it is clear the best parameter set for the sphere function is $(w = 0.7, c1 = 1.6, c2 = 0.6)$. This is not surprising given that the smaller c_1, c_2 values have lead to faster convergence. By setting the S at 3, the difference in fitness is 0.469 for an exchange of terminating 16 steps earlier. Although the difference is not exactly minute, given the average convergence step of 39 steps, increasing S slightly may actually push the termination condition beyond convergence. For the Rastrigin function, Figure 2 showed that parameters $(w = 0.7, c1 = c2 = 2.0)$ obtained the best results. By setting $S = 250$, the difference in fitness is only 1.06 while finishing 726 steps earlier. This is a very good result since it is often difficult for high dimensional Rastrigin function to minimise to 0 given the large amount of local minima.

1.2. Changing the number of particles N simply means having more particles looking for the solution at each iteration which means a higher likelihood in finding the optimum solution. The search space dimension on the other hand describes the magnitude of the search space that contains the solution. The greater is d, the longer it will take for the algorithm to converge as there are more spaces to search for. For the sphere function, changing the number of particles N or the search space dimension d has much less significant effect than the Rastrigin function which is much more complex and contains many local minima. Testing on the best parameters for both functions on a list N and d values in Figure 3 and 4, it is observed that having a low population and a high search dimension has lead to the longest convergent time.

1.3. The modification on the PSO is the addition of a repulsive term to the velocity which becomes

$$v_i = wv_i + \alpha_1 r_1 \circ (p_i - x_i) + \alpha_2 r_2 \circ (g - x_i) + \alpha_3 Z \tag{2}$$

where

$$Z = -\sum_{k \ne i} \frac{r_3 \circ (x_k - x_i)}{||x_k - x_i||^2} \tag{3}$$

Z acts as a repulsive term such that if x_k and x_i is far apart, then the repulsion effect will be small, however, if they are close, the repulsion effect will be larger. This inclusion aims to balance the exploitation and exploration of PSO, namely being attracted to the global best whilst being repulsed by nearby particles. Nonetheless, in practice, this implementation had actually lead to worse results, thus, the graphs will be in the appendix.

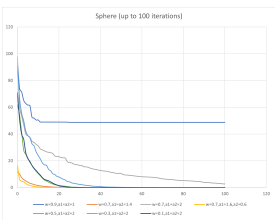

Figure 1: Sphere fitness for different parameters

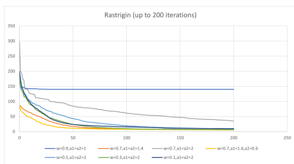

Figure 2: Rastrigin fitness for different parameters

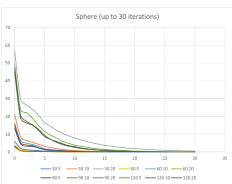

Figure 3: Sphere fitness for different particle size and dimension

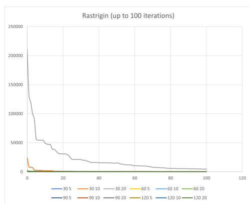

Figure 4: Rastrigin fitness for different particle size and dimension

2. Two datasets each containing 100 data points are generated for both functions. The Genetic Programming (GP) implemented in this study roughly follows the same structure from Sipper (2019)'s TinyGP. For the sphere function, its set of non-terminals is given by

$$T = [+, x] \tag{4}$$

and the set of terminals is given by

$$NT = [x1, x2] \tag{5}$$

For the Rastrigin function, its set of non-terminals is given by

$$T = [+, -, x, cos] \tag{6}$$

and the set of terminals is given by

$$NT = [x1, x2, -10, 2, \pi, 10] \tag{7}$$

The GP will be using a tree adaptation to represent programs and the fitness will be evaluated by the normalized inverse mean absolute error. For the selection method, double tournament will be used to control bloat. In contrast to a regular tournament selection, it hosts two sequential tournaments to select trees; first on fitness and second on tree size. The selection method contains two parameters F and D. F is the size of the fitness tournament which is analogous to regular tournament selection. However, D is the tournament size for tree size, also known as the parsimony tournament size. For example, if $D = 2$, there are two trees in this size tournament, for a probability of $\frac{D}{2}$, the tree with smaller size wins, otherwise, the larger tree wins. From the findings of Luke and Panait (2004), they established that $D \in [1.0, 2.0]$ such that if $D = 1$, the probability to select the smaller tree becomes $\frac{1}{2}$ and the tournament becomes a random selection where both sizes are likely to be selected. If $D = 2$, the probability to select the smaller tree becomes $\frac{2}{2} = 1$, then the tournament always selects the smaller tree which reflects the full parsimony pressure.

The list of parameters for the GP is shown as follows, for the sphere function, the population size is 60, minimum depth is 2, maximum depth is 3, generations is 50, tournament size is 7, crossover rate is 0.8 and mutation probability is 0.2. The selection of the parameters above are fairly standard and that changing them will not generate much effect on the outcome, given that the sphere function is simple to produce. The choice in tournament size of 7 orignates from a typical choice in literature (Qi *et al.,* 2013). For the Rastrigin function, the population size is 100, minimum depth is 2, maximum depth is 7, generations is 200, tournament size is 7, crossover rate is 0.8 and mutation probability is 0.2. Due to complexity of the Rastrigin function, the population size has been increased to search for more solutions, while the generation is increased to enable longer exploration of better solutions. Each GP is performed for 30 simulations and average results are reported.

Results showed that for the sphere function, the average generation to reach a fitness of 1 is approximately 3.45. However, for the Rastrigin function, given its complexity, the GP was not able to find a solution of fitness 1, rather the highest average fitness solution is only 0.113. The tree size of this solution is 194. Figure 5 shows the average fitness levels of the GP at each generation.

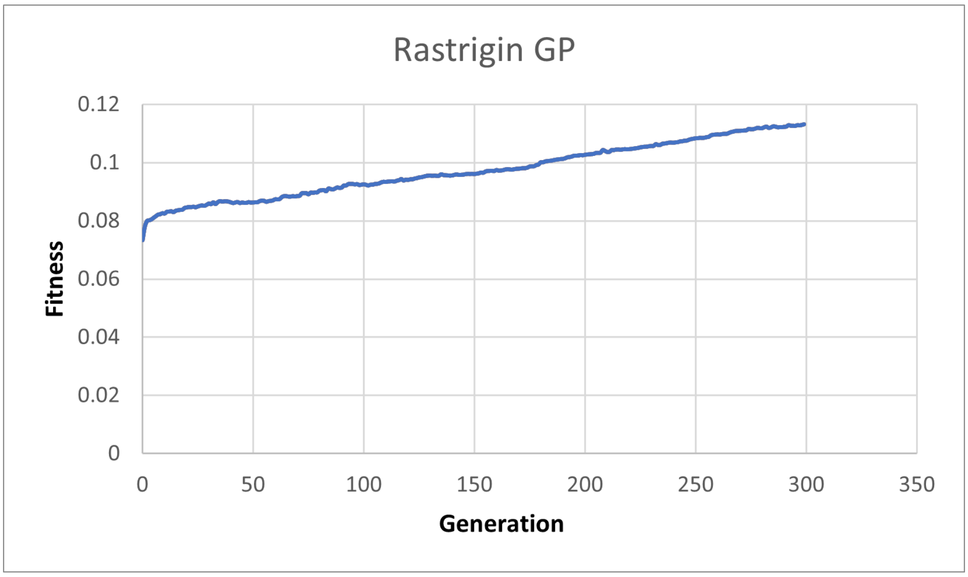

Figure 5: Rastrigin fitness for GP

3. For this report, the combination of GP and PSO follows the architecture of the Hybrid Genetic Programming with Particle Swarm Optimization model (HGPPSO) (Qi *et al.*, 2013). In this model, the PSO component is modified such that the particle position update is now

$$x_i = RWS(x_i, x_i - p_i, x_i - g) \tag{8}$$

where RWS is the roulette wheel selection on the particle fitness. $x_i - p_i$ tree encoding of the difference between the particle's current position and the local best, while $x_i - g$ concerns with the global best. They are defined as $Select(Cross(x_i, p_i))$ and $Select(Cross(x_i, g))$, respectively; $Cross$ is the crossover operator for the tree encoding in GP; $Select$ is the selection method for tree fitness and the standard tournament selection method will be used. Once all three parameters have been computed, they are summed to obtain a population fitness P, then for each parameter in the population, their share of the population fitness is $\frac{pr_i}{P}$ for i in population, where $pr_i \leq P$. Finally, a random selection occurs to select a tree as solution for the current generation.

Taking the example of the sphere and Rastrigin function from the GP study, HGPPSO has managed to outperform GP in both fitness and tree size. For the sphere function, it took on average 1.8 generations to reach a fitness level 1, which is nearly half the generations in the GP study. For the Rastrigin function, the average highest fitness solution was recorded at 0.114 when compared to 0.113 when using GP. Furthermore, the average tree size of the highest fitness solution is 183 which is noticeably smaller than the GP study. Figure **??** shows the fitness to generation graph for the HGPPSO model on the Rastrigin function. Thus, by introducing the an additional optimization procedure by PSO and incorporating the double tournament selection method, there are improvements for both fitness and tree size.

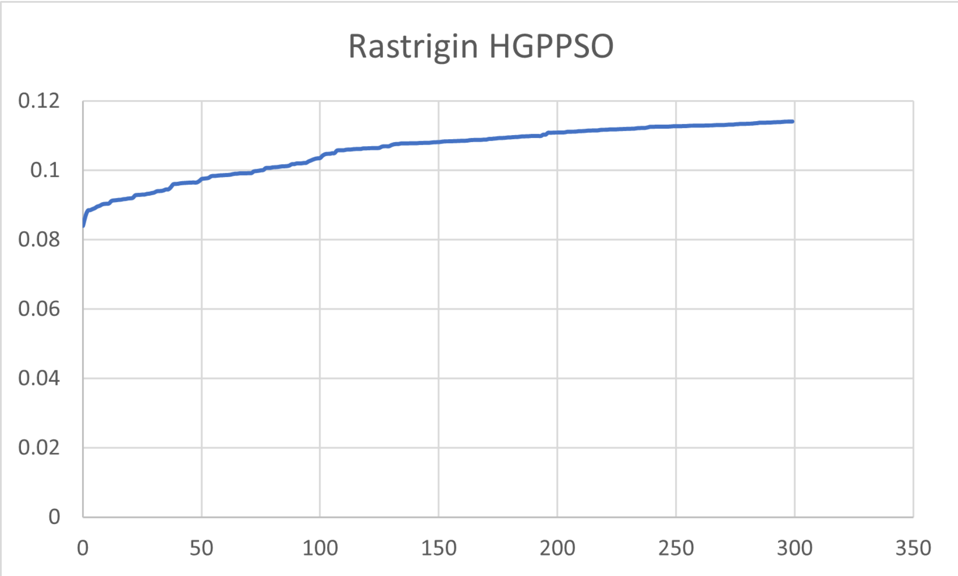

Figure 6: Rastrigin fitness for HGPPSO

References

Luke, S. and Panait, L. (2006). *"A comparison of bloat control methods for genetic programming."* Evol Comput. 2006 Fall;14(3):309-44. doi:10.1162/evco.2006.14.3.309. PMID: 16903796.

Pedersen, M. E. H. (2010). *"Good Parameters for Particle Swarm Optimization."*

Qi, F., Ma, Y., Liu, X and Ji, G. (2013). *"A Hybrid Genetic Programming with Particle Swarm Optimization"*. Proceedings 4th International Conference on Advances in Swarm Intelligence, ICSI 2013, Part II.

Sipper, M. (2019). *"Tiny Genetic Programming in Python"*. Github, Github repository.

Appendix

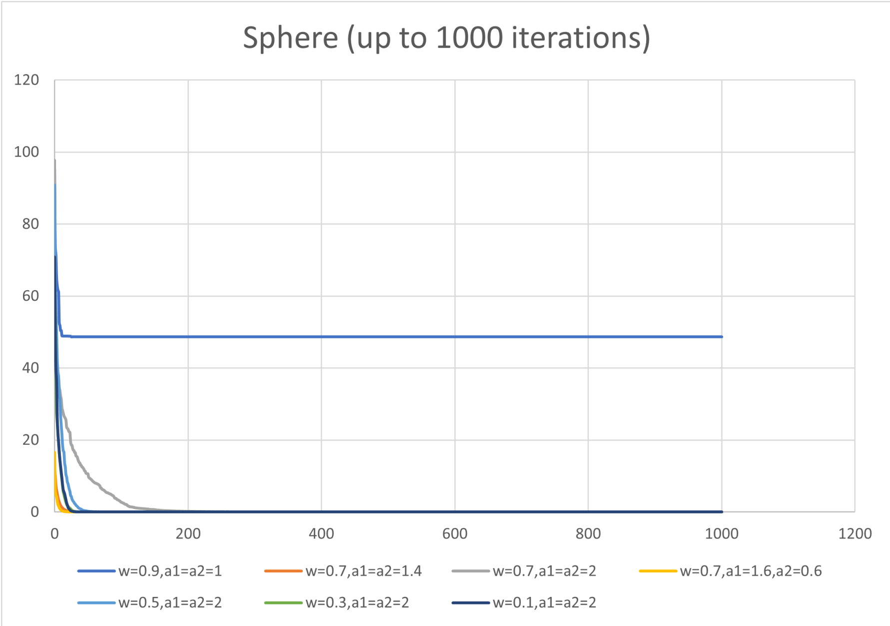

Figure 7: Sphere fitness for different parameters

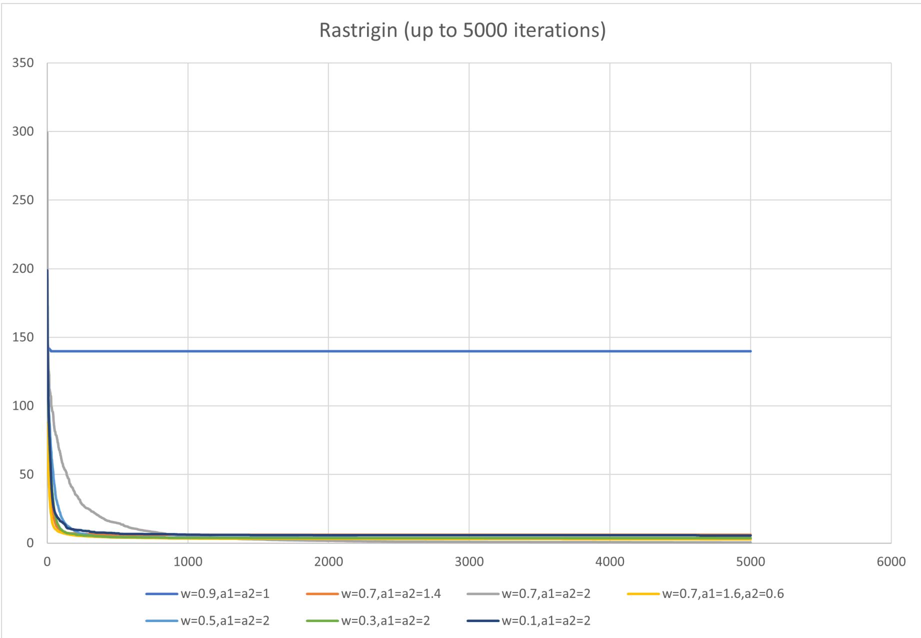

Figure 8: Rastrigin fitness for different parameters

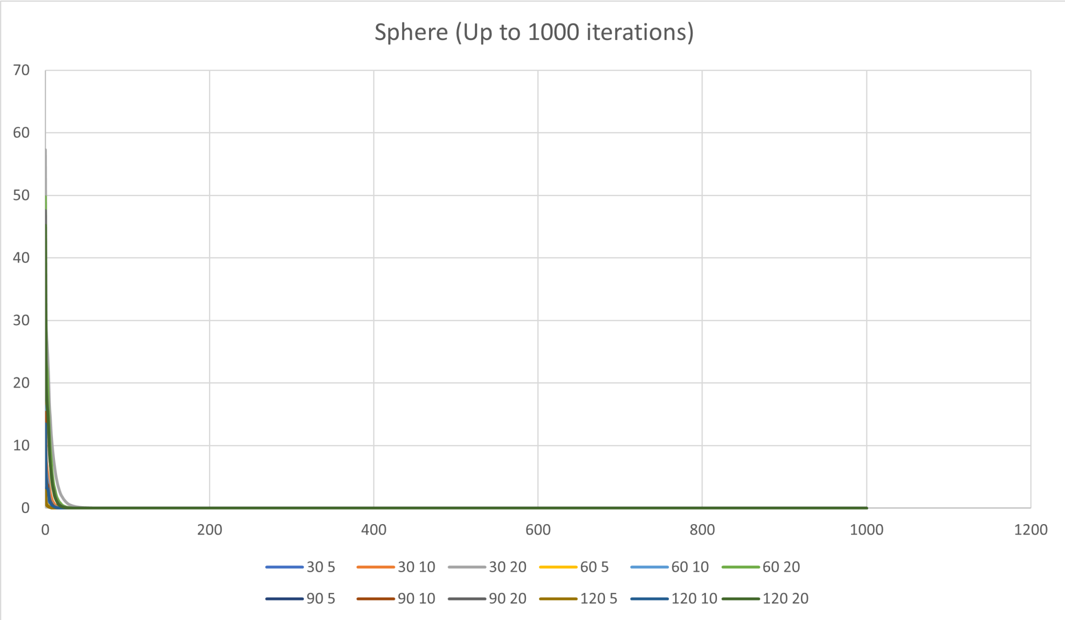

Figure 9: Sphere fitness for different particle size and dimension

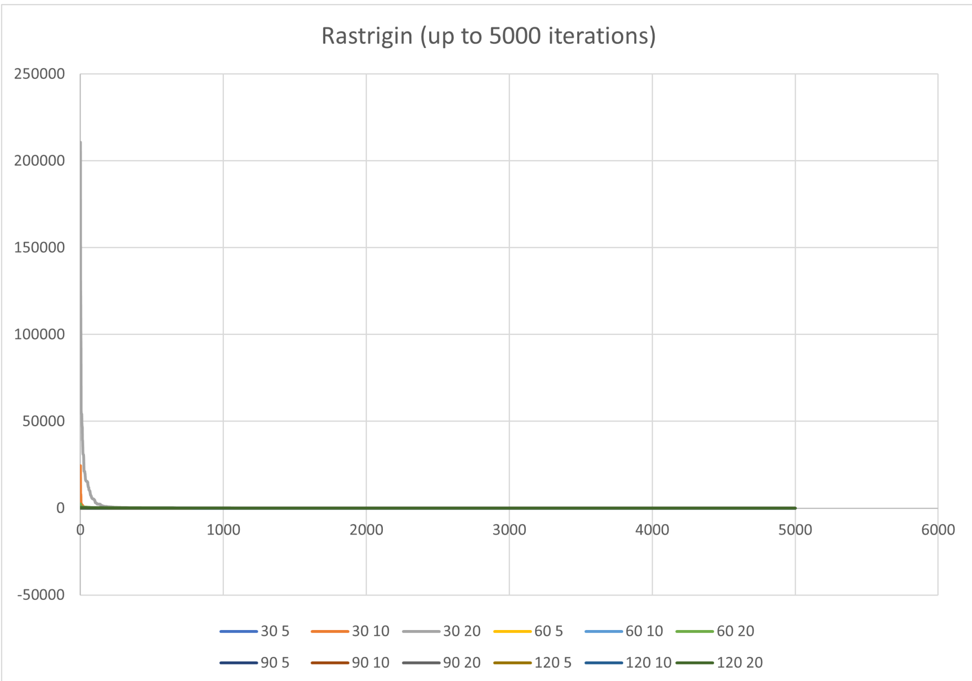

Figure 10: Rastrigin fitness for different particle size and dimension

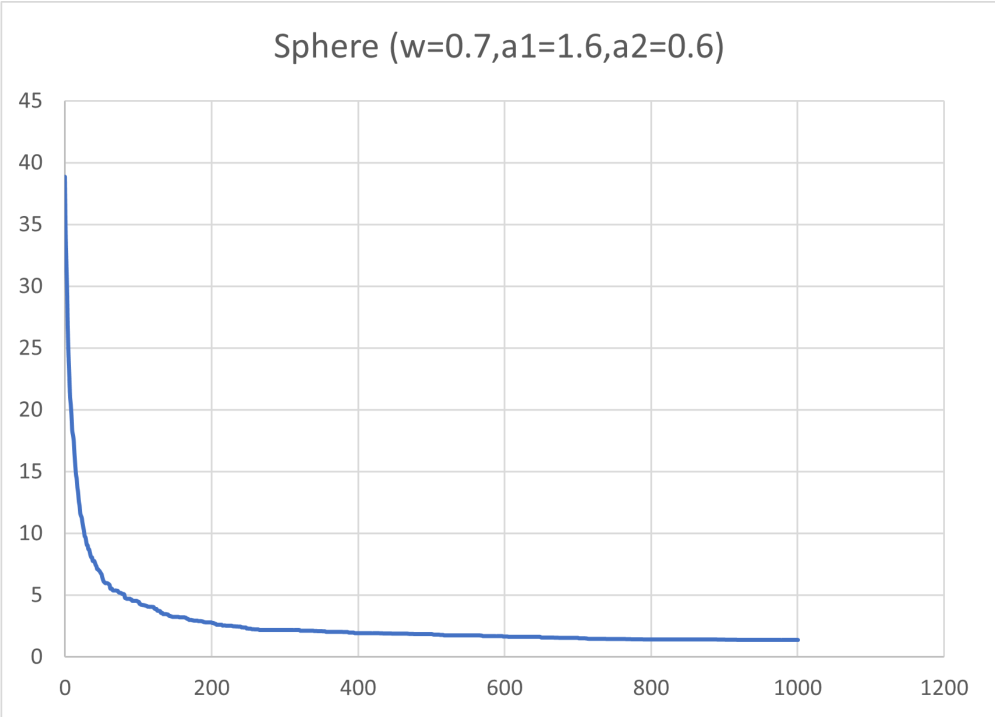

Figure 11: Sphere fitness with repulsion

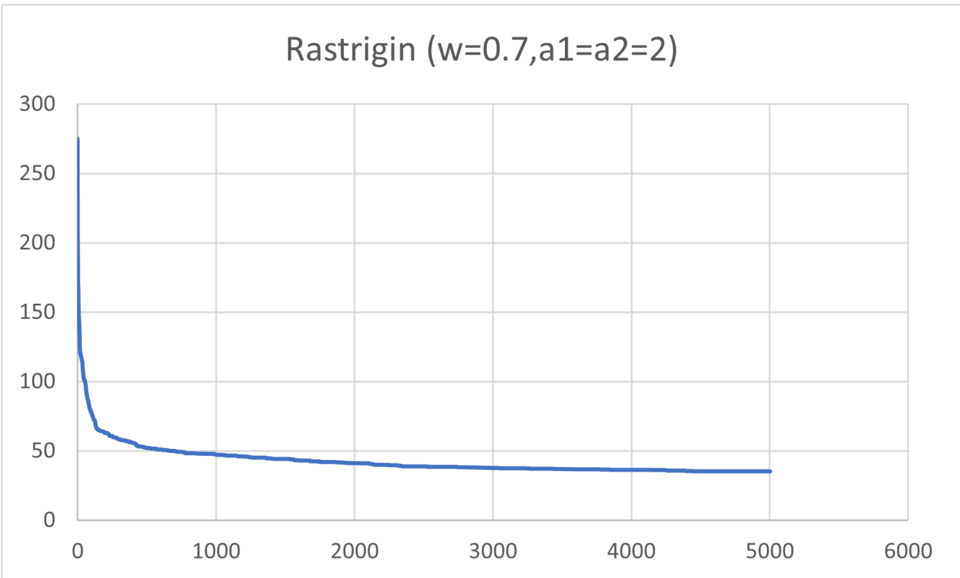

Figure 12: Rastrigin fitness with repulsion