Multi-Objective GA

Mh. Samadi

***Abstract*— In this essay we'll test a multi-objective GA called NSGA-II on multiple test functions in multiple dimensionalities.**

I. INTRODUCTION

NSGA-II is a well known, fast sorting and elite multi objective genetic algorithm. Process parameters such as cutting speed, feed rate, rotational speed etc. are the considerable conditions in order to optimize the machining operations in minimizing or maximizing the machining performances. Unlike the single objective optimization technique, NSGA-II simultaneously optimizes each objective without being dominated by any other solution.
An implementation of the algorithm in python is presented in the source code. The implementation is then tested with 6 different test functions each in 3 different dimensionalities. 10, 30 and 50.
Test functions are kursawe, schaffer, zdt1, zdt2, zdt3, zdt4 and zdt6.

II. IMPLEMENTATION

There are multiple modules used in our implementation. A problem module which represents a multi-objective optimization problem. The main evolution module which does the optimization itself and contains the main loop of our EA. And then the test module for all the test functions.
The source code is presented alongside with the essay.

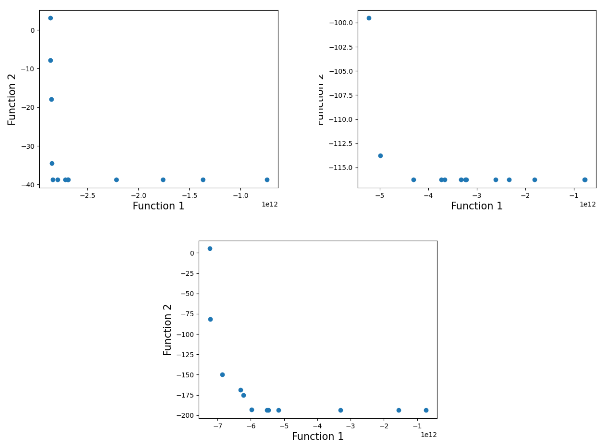
III. RESULTS

Here are the final fitness values from each objective using different test functions with different dimensionalities. Parameters of the algorithm are: number of generations = 1000, pop size = 100, tournament prob = 0.9, crossover param = 20 (20/100), mutation param = 2 (2/100).

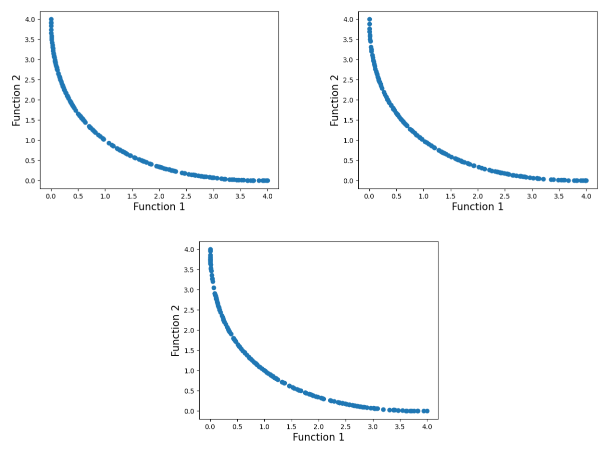
test	obj1	obj2
kur-10	-1.5e12	-38.75
kur-30	-1.9e12	-116.27
kur-50	-3.6e12	-193.78
schaffer-10	3.99	1.09
schaffer-30	3.99	2.26
schaffer-50	3.99	4.10
zdt1-10	10e5	-2.4e4
zdt1-30	10e5	-1.9e4
zdt1-50	7.8e4	-1.6e4
zdt3-10	98197	-1.2e4
zdt3-30	90078	-1e5
zdt3-50	94940	-1e6
zdt4-10	57681	-14035
zdt4-30	862	1.6e8
zdt4-50	580	1.3e9
zdt6-10	1.0	918
zdt6-30	1.0	2.2e8
zdt6-50	1.0	1.1e9

Table 1. The objectives after 1000 iterations

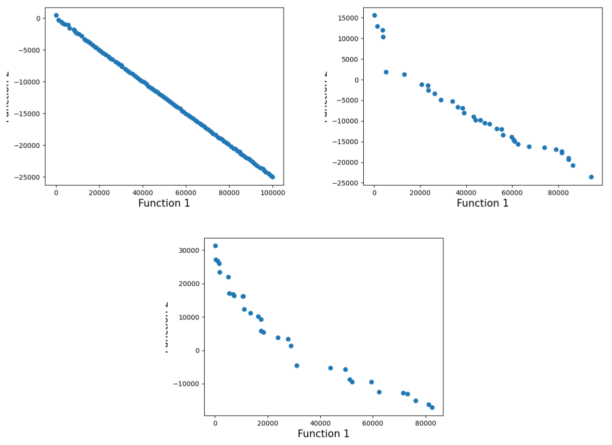
Here are the scatter plots of each run:



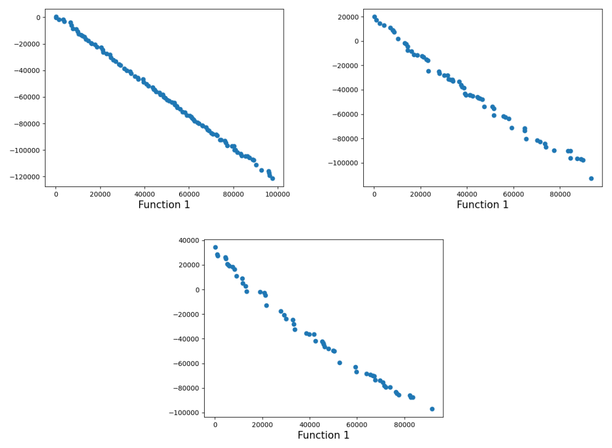
kur 10, 30, 50



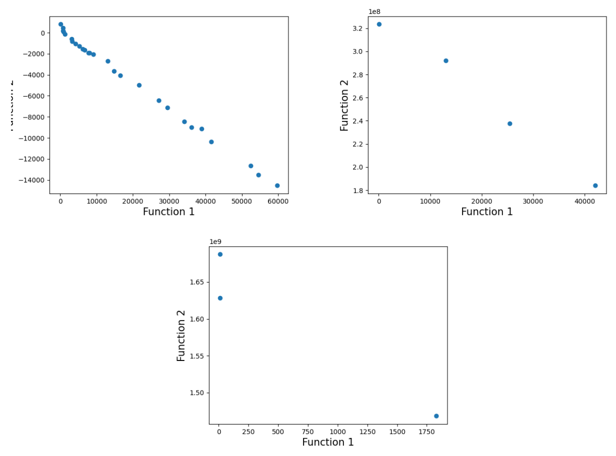
schaffer 10, 30, 50



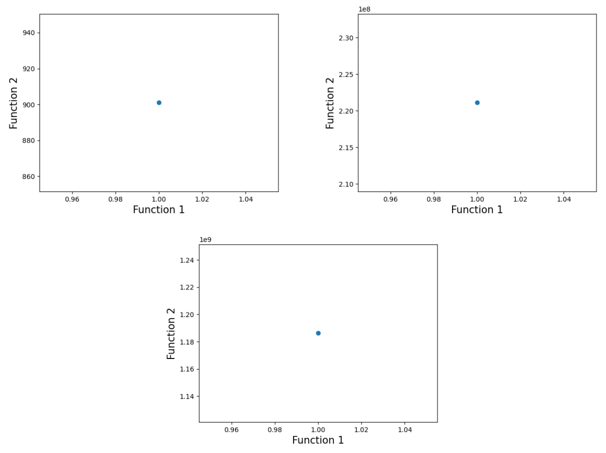
zdt1 10, 30, 50



zdt3 10, 30, 50



zdt4 10, 30, 50



zdt6 10, 30, 50

IV. Conclusion

As seen in the scatter plots, there's always a drop. As the plot's axis are function1 against function2, this means the algorithm is optimizing one objective in the cost of making the other objective worse. But with a closer look we can ses that the scale of the axis are not the same and are different by orders of magnitude. This means that the algorithm is choosing the axis with more impact and optimizes that axis with the cost of making the other axis slightly worse.