Bisection for sGA solving Onemax and Trap-5

19520208 - Huynh Phuong Nhu

October 2021

Chapter 1

Introduction

1.1 Experiment

In this experiment, we will try find the MRPS which means minimum required population size by running bisection on sGA.
The experiment is in two part:

1. Running it on Onemax, a function return the highest fitness when all of its variables has the value of 1; and the more variables equal to 1 the higher the fitness is.

$$\sum_{i}^{n} u_i; \text{ where } u_i \in [0,1] \text{ and n is the number of variables}$$

2. Running sGA on Trap-k function and k = 5. A formula for one trap is the below equation:

$$f(x) = f_{TRAP}u$$

$$u = \sum_{i=1}^{k} x_i$$

$$f_{TRAP}u = \begin{cases} kifu = k \\ k - 1 - uifu < k \end{cases}$$

Trap-k is an concatenated trap design to misleading the direction of sGA.

1.2 Source code

For this experiment, I basically use all of the source code from previous exercise of POPOP as the implementation for sGA.
The requirement is we use this with tournament selection with the size s = 4,

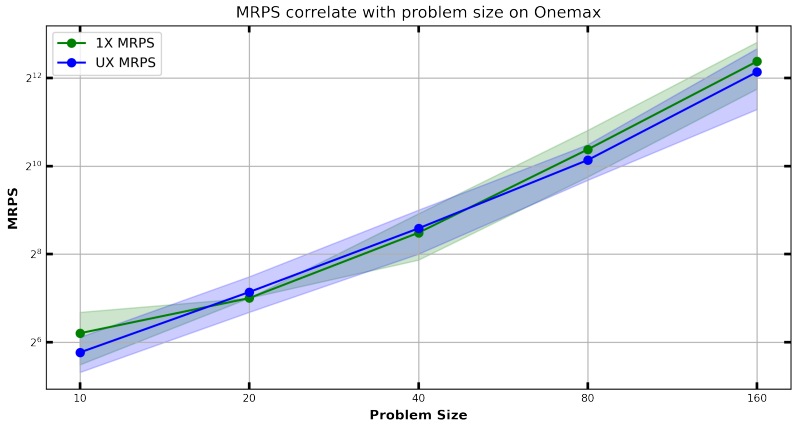
using only crossover without mutation.
Using the increment in problem size l doubling each time from 10 upto 160, we first calculate the upper bound to solve 10 seeds completely and then it comes the second period of finding out the MRPS and the average amount of evaluations in 10 seeds solving times. Talking about my own implementation, as regards the bisection applied with Trap-5 function as fitness function, I first initialized population then building Mariginal Product Model (MPM) for the current population in order to make it converge faster than only using the implementation of POPOP, to find upper bound. After that is the step of creating new population from the existing one and repeat this process until all individuals in the population converge to one. Then the assessments came, if they are all converge into the optimal value, the test on that seed is done successfully and if the 10 seeds are executed perfectly then we have found our upper bound. Vice versa, the seed is done but in unsuccessful way and we need to move up the bound.
After getting the upper bound, we try to move up the lower one in order to find a smaller option for upper bound as MRPS. Hence the first $lower = upper/2$, constantly testing the seeds like in the first period and eventually we would find the new upper bound as MRPS or the old one has already been the MRPS for this subset of seeds.

Chapter 2

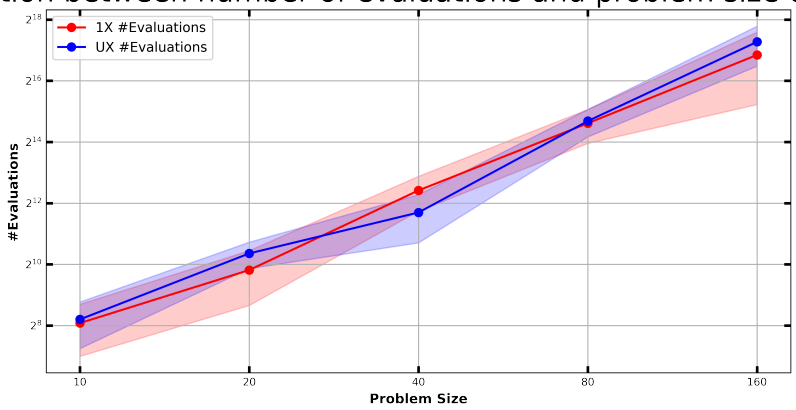
Result

2.1 Onemax

After running the code, we arrive at some statistics. As can be clearly seen from the table 2.1 and two graphics, there is a jump in both MRPS and the number of evaluations made by the rise of problem size.

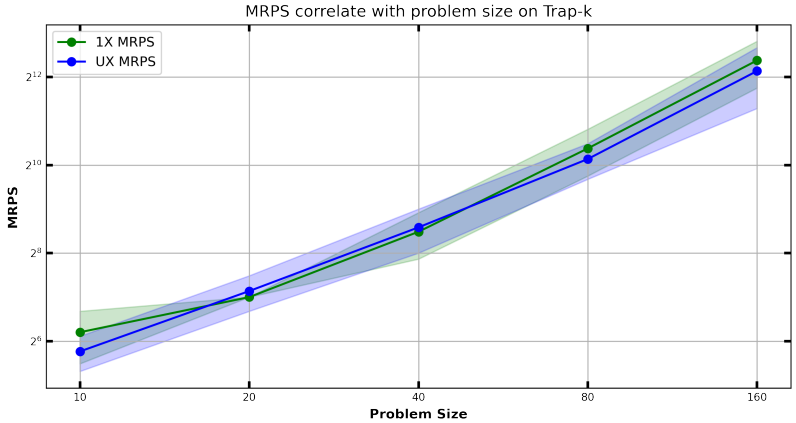


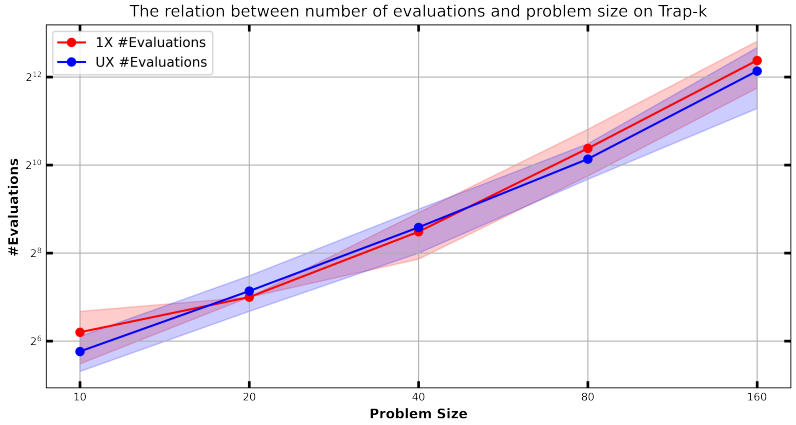
e relation between number of evaluations and problem size on One



2.2 Trap-5

Overall, the trend of MRPS and evaluations time is both the same although 1X seems to take more time and resources. The reason for this may come from the amount of variation that 1X may have on the population, it easily compromised the better individuals by making changes on multiple individuals simultaneously.





2.3 Conclusion

To conclude, trap-k fitness function make the algorithm much harder to deal with, this can be seen from the MRPS or the time it took to run. With the help of model building using MPM, trap function is better in convergence although the time it used to build the model of itself is still long.

OneMax								
type	**sGA-1X**				**sGA-UX**			
Problem size	MRPS	std(MRPS)	#Evaluations	std(#Evaluations)	MRPS	std(MRPS)	#Evaluations	std(#Evaluations)
10	54.4	14.66	270.88	143.47	51.2	15.67	294.41	142.41
20	108.8	29.32	900.6	496.63	140.8	38.4	1312.8	381.26
40	409.6	125.41	5472.0	2052.27	281.6	76.8	3319.2	1652.10
80	1228.8	409.6	25056.0	9181.43	1228.8	409.6	26332.8	7938.92
160	4710.4	1843.2	117830.4	79530.37	5324.8	1877.02	158553.6	67254.4

Table 2.1: MRPS and the figure for evaluations of applying sGA on Onemax

Trap-5								
type	**sGA-1X**				**sGA-UX**			
Problem size	MRPS	std(MRPS)	#Evaluations	std(#Evaluations)	MRPS	std(MRPS)	#Evaluations	std(#Evaluations)
10	73.6	28.8	450.98	191.39	54.4	14.66	327.18	123.44
20	128.0	0.0	1221.6	68.68	140.8	38.4	1168.8	561.39
40	358.4	125.41	4682.4	2092.80	384.0	128.0	5037.6	2289.38
80	1331.2	469.25	27139.2	9677.07	1126.4	307.2	24768.0	6227.43
160	5324.8	1877.02	139968.0	87769.87	4505.6	2006.62	121958.4	80720.59

Table 2.2: MRPS and the figure for evaluations of applying sGA on Trap-5