Bisection for sGA solving OneMax and Trap-5

Truong Minh Chau - 19521281

October 2021

1 OneMax Problem

This is the simplest optimization task that only performs the maximum of the computed value from a string binary number, which means going to count the number of bit 1 presented in the string of each instance

1.1 Mean

This is the MEAN of MRPS and EVALUATIONS for Onemax Problem extracted from file "Total Bisection Data".

	OneMax			
	sGA-1X		sGA-UX	
Problem size	MRPS	Evaluations	MRPS	Evaluations
10	29.4	201.13	13.2	99.37
20	64.6	758.52	21.2	224.9
40	163.2	2965.88	34	524.55
80	571.2	15581.88	74.8	1606.53
160	2828.8	113672.59	122.4	3722.5

1.2 Standard Deviation

The STANDARD DEVIATION of MRPS and EVALUATIONS for Onemax Problem extracted from files "std_onemax_1x_data" and "std_onemax_ux_data" in folder "std" of "data_onemax".

1.2.1 STD of 1x

Problem size	std_MRPS	std_Eval
10	14.91	95.1
20	7.21	93.56
40	27.2	441.24
80	67.32	1605.21
160	249.29	9086.85

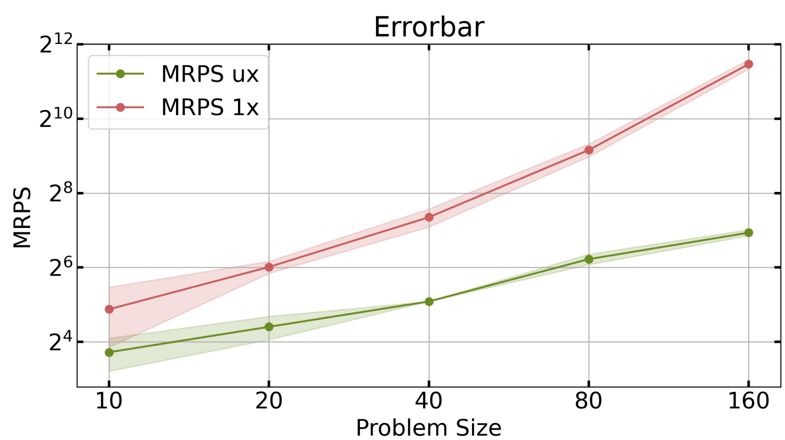
1.2.2 STD of ux

Problem size	std_MRPS	std_Eval
10	3.92	28.69
20	4.53	41.46
40	0	9.97
80	7.21	141.08
160	6.8	193.78

1.3 Line Graph

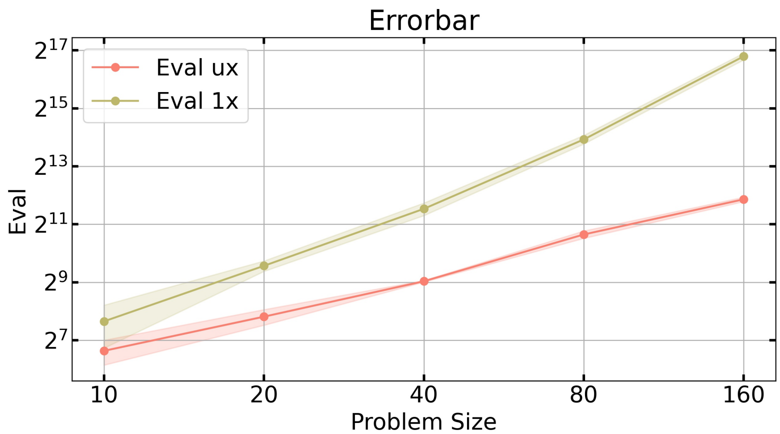
1.3.1 MRPS Graph

The resulting graph shows the MRPS as the problem size increases.



1.3.2 Evaluation Graph

The resulting graph shows the number of calls to fitness evaluations as the problem size increases.



2 Trap Problem

"Trap" functions were known as a method to check how GAs behave on functions where sampling most of the search space would provide pressure for the algorithm to move in the wrong direction (wrong in the sense of away from the global optimum).

2.1 Mean

Similar to the above problem, we also calculate the MEAN of MRPS and EVALUATIONS for Trap Problem given from file "Total Bisection Data".

	Trap-5			
	sGA-1X		sGA-UX	
Problem size	MRPS	Evaluations	MRPS	Evaluations
10	85	678.72	340	3515.6
20	231.2	2829.21	2176	48647.82
40	625.6	11881.54	#DIV/0!	#DIV/0!
80	2284.8	67052.19	#DIV/0!	#DIV/0!
160	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!

2.2 Standard Deviation

The STANDARD DEVIATION of MRPS and EVALUATIONS for Trap given from files "std_trap_1x_data" and "std_trap_ux_data" in folder "std" of "data_trap".

2.2.1 STD of 1x

Problem size	std_MRPS	std_Eval
10	34.84	287.81
20	44.07	514.8
40	122.4	2389.26
80	269.27	7314.8

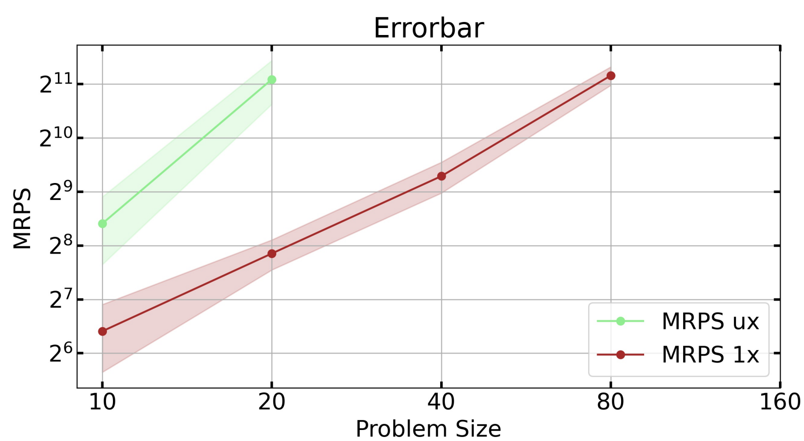
2.2.2 STD of ux

Problem size	std_MRPS	std_Eval
10	139.36	595.92
20	1520.85	14268.07

2.3 Line Graph

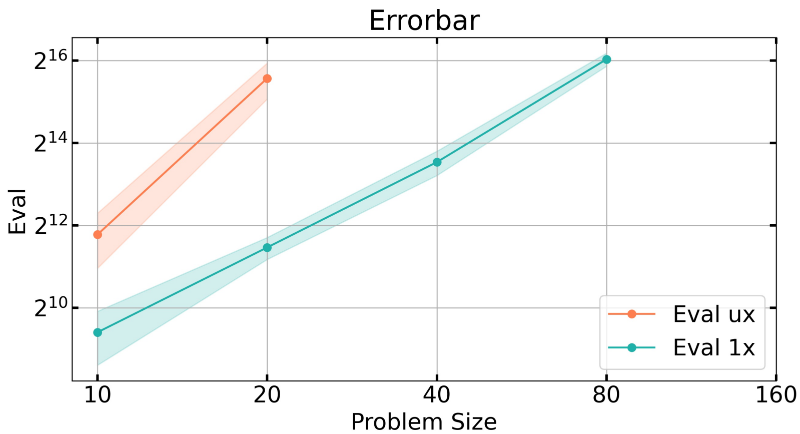
2.3.1 MRPS Graph

The resulting graph shows the MRPS with 1X and UX.



2.3.2 Evaluation Graph

The resulting graph shows the number of calls to fitness evaluations with 1X and UX.



3 Comparison

From the above 4 graphs, we can clearly see that as the problem size increases, the Trap problem performs the fitness evaluations function to try to find the population that converges with the individuals whose value is 1. So the line tends to spike.
When it comes to the Onemax problem, running the model with both crossovers returns optimal results in a short period of time for whole population size. So the line graph increases steadily through each population size.

3.1 Onemax

This is a function for the sum of digits, so the implementation of finding the number of instances that converges at the best result simply adds bits and returns only when the result is equal to the length of the string.
Because of the simple implement, no matter what crossover it is, we got a much smaller and faster result. Especially for U-crossover, both the number of individuals and the number of Evaluation Fitness function calls are many times smaller than that of a 1-point one.
Then, confirm that the uniform crossover used for this problem will be more efficient and less computational cost than the other.

3.2 Trap

Based on the definition of trap as well as the graph of the problem, we can see whether the choice of crossovers in variation can make the problem easy or not. To come up with the uniform crossover, the model can return the number of individuals converging when increasing the population size to 80, but that is difficult for single point crossover. It took almost 24 hours but still cannot

return results when the population size is 40 if there is no stopping condition (n_upper is over 8192).
Then, confirm that the one point crossover used for this problem will be more efficient and less computational cost than the other.