CS410.L11.KHTN WEEK 3 REPORT

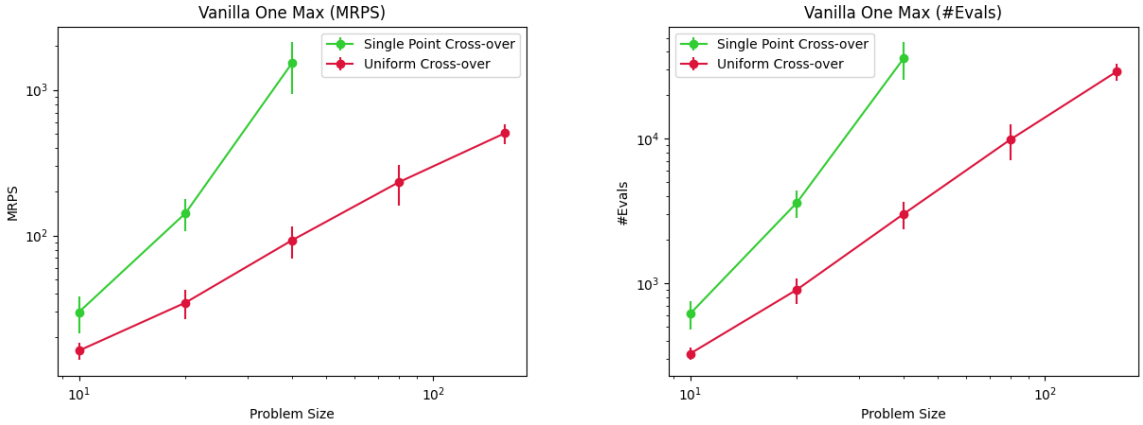
Nguyen Trung Hieu - 18520750

Experiment Results

Vanilla One Max Problem In One Max Problem, our objective is to maximize the number of 1-s appearance in the given bit number. The optimal case is when all bit become 1.

	sGA-1X		sGA-UX	
Problem size	MRPS	Num of Evaluations	MRPS	Num of Evaluations
10	29.8 ± 8.36	616 ± 140	16.2 ± 2.27	326.4 ± 32.11
20	142.4 ± 34.46	3616 ± 758.3	34.6 ± 7.8	904.6 ± 182.42
40	1529.6 ± 591.057	35959.2 ± 10526.47	92.8 ± 22.96	3020.4 ± 664.1
80			232 ± 72	9878.4 ± 2723.4
160			505.6 ± 79.4	29294.4 ± 4014.64

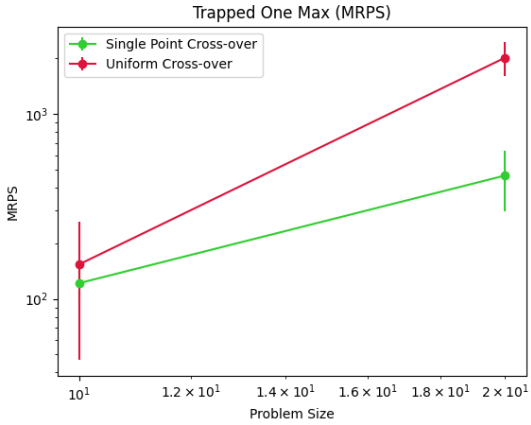
Table 2. One Max Problem.

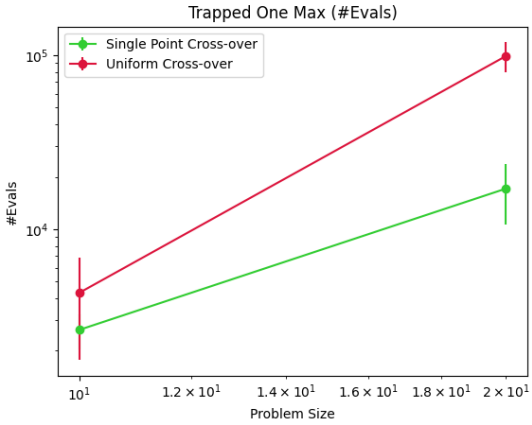


Trapped One Max Problem Trapped One Max Problem is a modified version of One Max Problem, proposed to discuss the behaviour of genetic algorithm. The "trap" lies in how the problem deceives genetic algorithm to get far away from the optimal solution by penalizing the algorithm as it gets nearer to the optimal case. In this experiment, several trap ($k = 5$) fitness function are tightly connected to calculate the real fitness function

	sGA-1X		sGA-UX	
Problem size	MRPS	Num of Evaluations	MRPS	Num of Evaluations
10	121.6 ± 29	2635.52 ± 591	153.6 ± 107	4301.76 ± 2535.64
20	465.6 ± 167.27	17111 ± 6541.15	2022.4 ± 423.7	98938.8 ± 19544.391

Table 4. Trapped One Max Problem.





Observation

The experiment generates insightful results, in which different cross-over methods significantly affects performance. In One Max Problem, uniform cross-over finds the optimal solution much faster than single point cross-over. Under limited population size and evaluation function usage, uniform cross-over methods solve all problem size while single point cross-over was not able to solve 80-bit case and 160-bit case. Using the same implementation and change only the fitness function, in trapped One Max Problem, simple genetic algorithm can only solve 10-bit case and 20-bit case. In contrast to the vanilla One Max Problem, uniform cross-over performs less well than single-point cross-over