HW1 - GA in Numerical Optimization

Sheng-Hsuan Peng
PME, NTHU
107033588

I. Objectives

Practice and get familiar with the most widely used evolutionary algorithm — genetic algorithm (GA). In this assignment you need to make use of the taught subject matters about GA's representation, crossover, mutation, and survivor to solve the given problem.

II. Problem Description

Write efficient programs to implement GAs to find the minimal solution of the Schwefel function (SCH):

$$f_{SCH}(\vec{x}) = 418.98291N - \sum_{i=1}^{N} x_i \sin\left(\sqrt{|x_i|}\right)$$

where $-512 \leq x_i \leq 511$ and N = 10. This function is a continuous, multimodal, non-convex, deceptive, and N-dimensional function with a global minimum of 0.

TABLE I
PARAMETERS

	Binary GA	Real-valued GA
Representation	$c_i \in 2^{10}$	$c_i \in \mathbb{R}$
Population	Generation (size 100)	
Parent Selection	Tournament Selection($n = 2$)	
Crossover ($p_c = 0.9$)	Uniform	
	2-point	Whole-Arithmetic
Mutation ($p_m = 1/l$)	Bit-flip	Uniform
Survivor Selection	$\mu+\lambda$	
Termination	500 generations	

III. Result

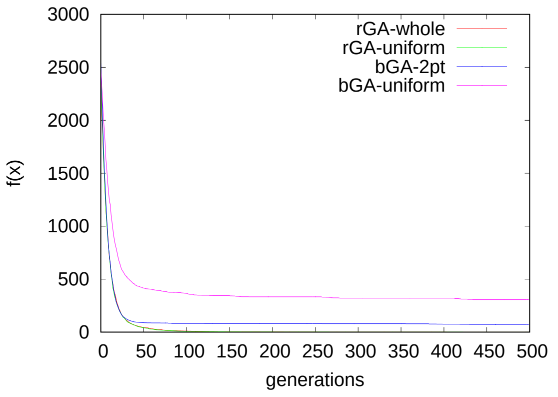

Fig. 1. Anytime behavior (averaged over 30 trials) of the above GAs

IV. Comparison

Compare convergence speed and solution quality between different representations and operators; give reasons why some combination performs better (or worse).

A. Convergence Speed

Real-valued GAs and Binary GA with 2-point crossover have same speed at the begin, and Binary GA with 2-point crossover converge first, then the 2 Real-valued GAs converge, then the Binary GA with uniform crossover.

B. Solution Quality

Real-valued GAs have better performance over the 4 GAs, and they could reach the optimum f(x) value (0). However, Binary GAs can't reach the optimum f(x) value, and Binary GA has the worst performance

C. Description

The representation of the above GAs lead to these outcomes. For Binary GAs, changing of bits would have different degrees of influence due to it's location. Take 0000000000 as example, this gene has it's fitness -512, if the first bit changes to 1 (1000000000), then value would be 0, in the meanwhile, the change of last (0000000001) bit would only lead to the change of 1. And for Real-valued GAs the method of crossover would lead the children be the mid-fitness of their parents, thus would tend to evolute into the mid-point of the search space, which is 0 the optimum point. Thus Real-valued GAs would have better convergence speed and solution quality in this problem.

V. Other Setting

In this section, I had tried different parameters setting for p_c, p_m, n, and different settings have different effect on diffent method:

A. Binary GA(Uniform Crossover)

p_c Owing to the symmetrical characteristics of the uniform crossover, the anytime behaviors are the alike with p_c and $(1 - p_c)$, and $p_c = 0.5$ converged faster and performed better than others.

p_m Mutation rate would take effect on both convergence speed and solution quality. In the experiment, $p_m = 0.05$ is the best parameter setting for the uniform crossover binary GA.

n In this experiment $n = 1$(ramdon selection) converge slower but have better solution quality.

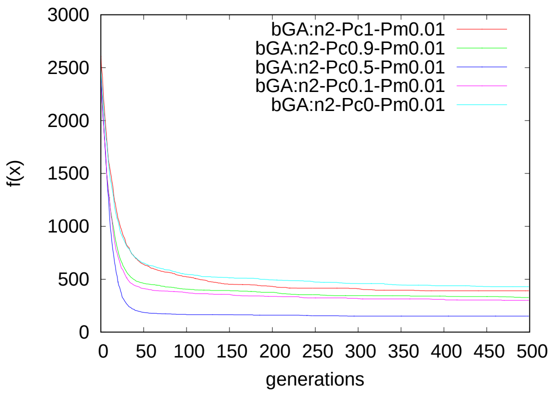

Fig. 2. Binary GA(Uniform Crossover) with different p_c.

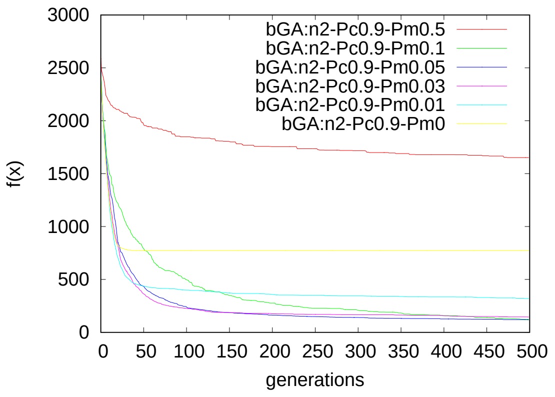

Fig. 3. Binary GA(Uniform Crossover) with different p_m.

B. Binary GA(2-point Crossover)

- p_c Convergence speed are same, and the higher the crossover rate the better solution quality.
- p_m Higher or zero mutation rate would lead to slowlier convergence speed.
- n In this experiment $n = 1$(ramdon selection) converge slower but have better solution quality.

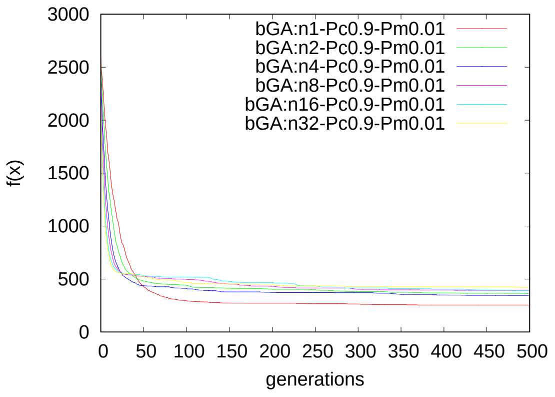

Fig. 4. Binary GA(Uniform Crossover) with different n-Tournament size.

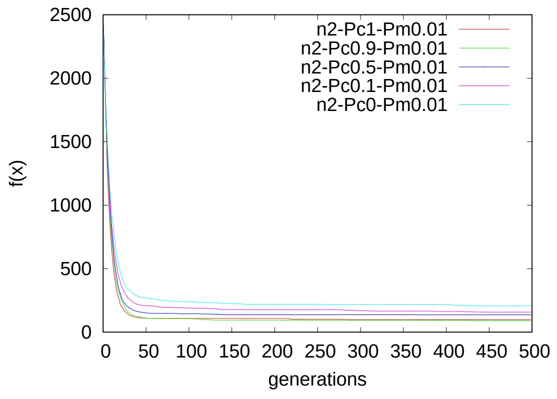

Fig. 5. Binary GA(2-point Crossover) with different p_c.

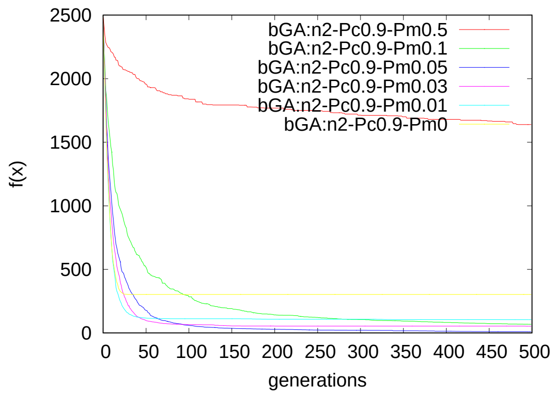

Fig. 6. Binary GA(2-point Crossover) with different p_m.

C. Real-valued GA(Uniform Crossover)

- p_c Both settings could get the optimum solution, and the higher crossover rate the faster convergence speed.
- p_m Zero mutation rate would converge too fast thus had poor solution quality.
- n Both settings could get the optimum solution, and the higher n-Tournament selection the faster convergence speed.

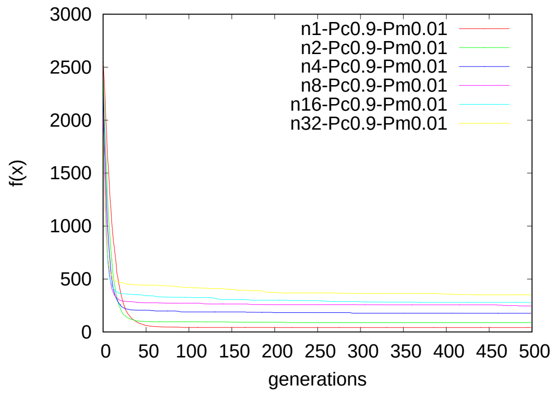

Fig. 7. Binary GA(2-point Crossover) with different n-Tournament size.

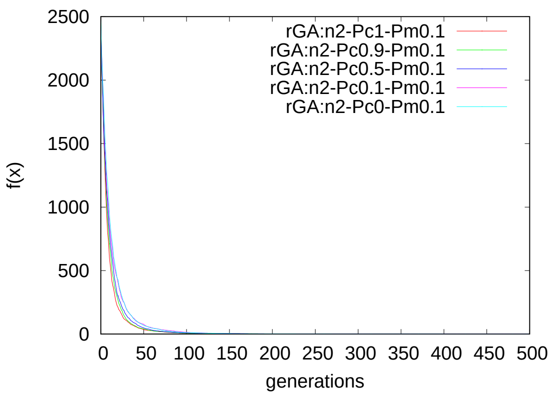

Fig. 8. Real-valued GA(Uniform Crossover) with different p_c.

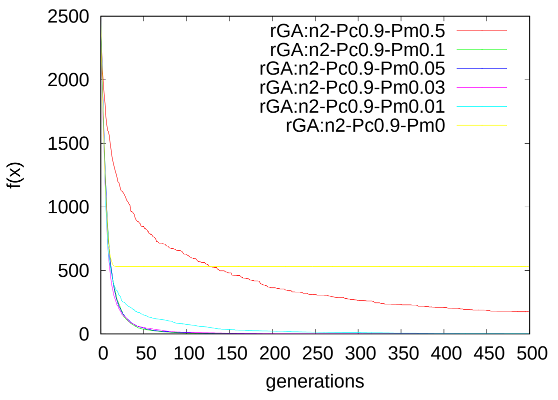

Fig. 9. Real-valued GA(Uniform Crossover) with different p_m.

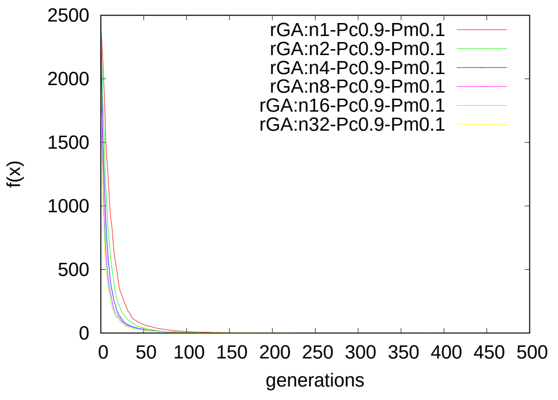

Fig. 10. Real-valued GA(Uniform Crossover) with different n-Tournament size.

D. *Real-valued GA(Whole Arithmetic Crossover)*

p_c	Both settings could get the optimum solution, and the higher crossover rate the faster convergence speed.
p_m	Zero mutation rate would converge too fast, thus had poor solution quality. The higher mutation rate would cause it converge slowlier.
n	Both settings could get the optimum solution, and the higher n-Tournament selection the faster convergence speed.
α	Different α settings for whole Arithmetic Crossover seems to heve no effect on the result.

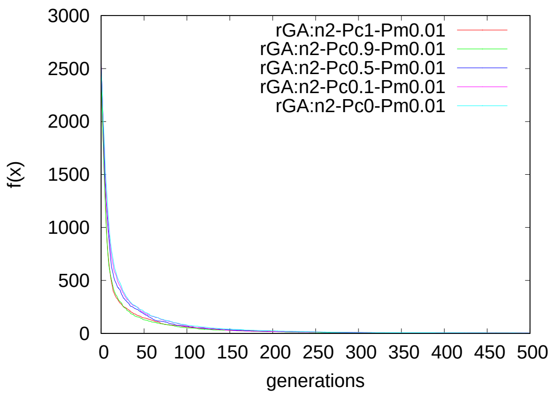

Fig. 11. Real-valued GA(Whole Arithmetic Crossover) with different p_c.

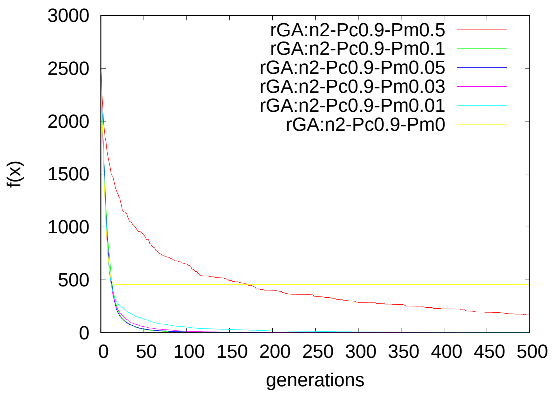

Fig. 12. Real-valued GA(Whole Arithmetic Crossover) with different p_m.

VI. Large-scale Problem

These GAs became harder to reach the optimal solution, so I enlarge the population size, and quality had 4 times enhanced, however the running time became 10 times longer.

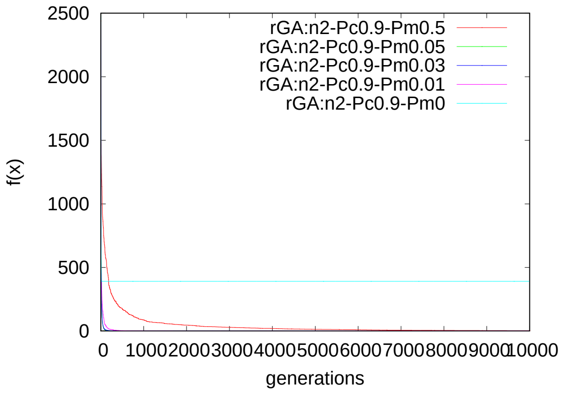

Fig. 13. Real-valued GA(Whole Arithmetic Crossover) with different p_m (10000 generations).

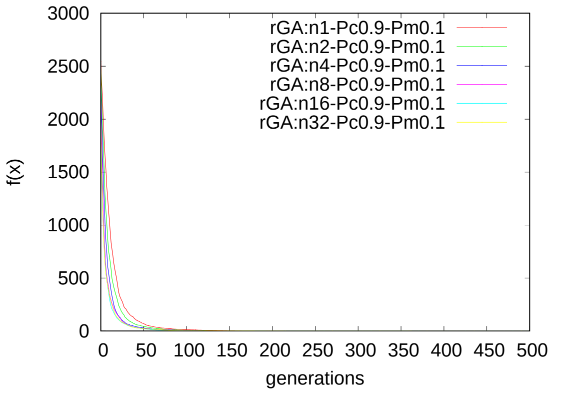

Fig. 14. Real-valued GA(Whole Arithmetic Crossover) with different n-Tournament size.

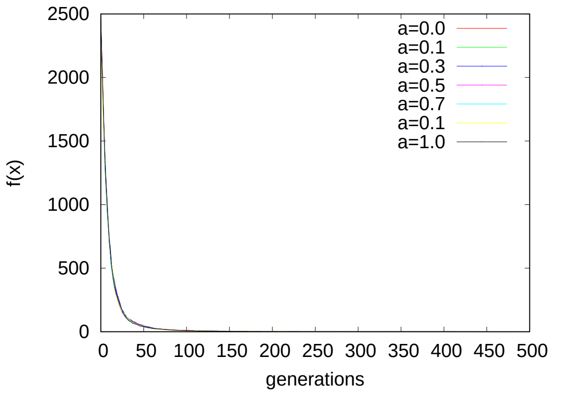

Fig. 15. Real-valued GA(Whole Arithmetic Crossover) with different α settings.

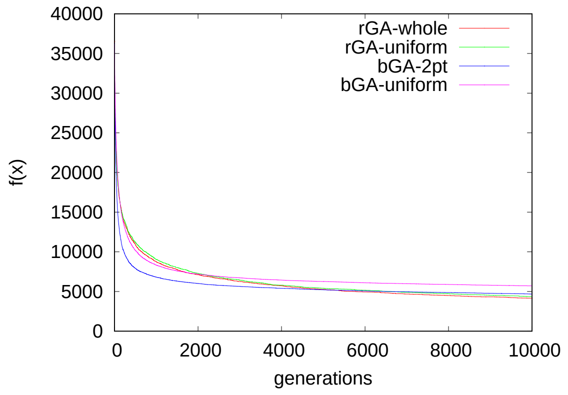

Fig. 16. Anytime behavior (averaged over 30 trials) of the above GAs for $N = 100$.

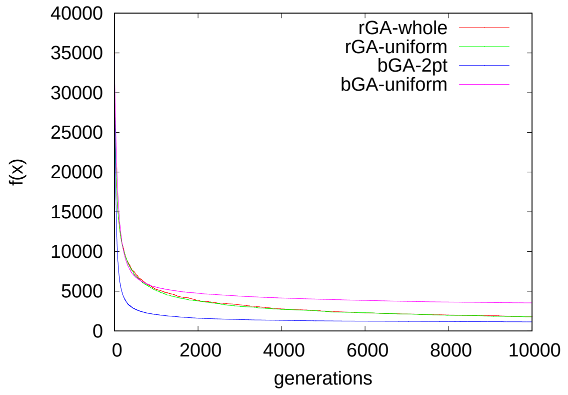

Fig. 17. GAs for $N = 100$ when population size is 1000