**Functional Requirements: Version #1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req No.** | **Description** | **Comments** | **Review Left Column** |
|  | Create a genetic program for target function (x\*x-1)/2 with fitness measure zero | 1. Target function should be configurable to allow for change in requirement 2. Program needs to generate a fitness measure = 0 or as close to 0 as possible within 15 minutes | -Doesn’t have to be 0 exactly at least close to 0.01.  -Did we want to use his exact equation? |
|  | Randomly initialize the first generation to be evaluated | 1. Population size needs to be configurable 2. Population size should be large enough to have diversity but small enough to generate meaningful results in 15 minutes execution time ( Generally 51-100) 3. Training data set – negative and positive decimal numbers | -Is 51-100 enough? Could we go higher 51-150? |
|  | Create successive generation(s) based on:  Fitness  Crossover  Mutation | Details in next three requirements | -We should identify this as ‘Tree Encoding’ |
|  | Fitness: Probabilistically determine the fittest individuals from the initial population of that generation | 1. Probabilistically select (1-r)p individuals 2. The criterion for defining “fitness” can change, but not likely. Configurability or the ability to quickly change code would be good. | -What is #1? Is that the objective function |
|  | Crossover: For each pair of fittest individuals, generate a pair of offspring using the Crossover operator | 1. What is the Crossover operator? Substituting portions of the individual binary trees with portions of binary trees representing (x\*x-1)/2? 2. The criterion for crossover can change (but not likely). Configurability or ability to quickly change code would be good. | -We would have to show individual as parents. Parent A and Parent B = Offspring of next generation. |
|  | Mutation: Invert a randomly selected bit in random members from population generated after Crossover | 1. Invert a randomly selected bit in m.p random members of new generation generated so far 2. The criterion for mutation can change, but not likely. Configurability or ability to easily change code would be good. | -Does this require picking a different operator (+, -, \*, /) node to select from the mutated parents? |
|  | Use Binary Trees as individuals where –  Operators: +, -, \*, /  Operands: Digits 1-9 | 1. Tree depth needs to be configurable 2. Define behavior for division by zero (Is zero an operand?) 3. User specified system requirement that can change, so configurability would be good | -Don’t we need ones for if, else, then statements? |
|  | Execution time = 15 minutes | 1. User specified performance requirement 2. Should be configurable to allow for change by user | -The customer or client specified this runtime. |
|  | Termination criterion – Solution found matching Target function or Execution time is over |  | -Something should be added here. We shouldn’t leave this blank. |

*Note: Phetsy and Greg also included some nice system requirements that I have not included in here yet to keep meeting focused on user requirements. I will add a separate section for System Requirements in this document later.*