**Solution representation:**

We represent solution as a 7-genes chromosome, because we have 4-body part and we are allowed to selected up to (2-elbow, 2-upper arm, 2-lower leg, and 1-wrist) exercises. Solution represented ad [elbow1, elbow2, upper arm 1, upper arm 2, lower leg 1, lower leg 2, wrist].

For exercise not selected we set -1, so for (elbow, upper arm, and lower leg) the range from -1 to 11, where for wrist the range from -1 to 10.

Example solution representation: [1, 2, -1, 3, 0, -1, 5]

Represent [elbow1, elbow2, arm1, arm2, leg1, leg2, wrist]

| **Gene Number** | **Exercises** | **Range** |
| --- | --- | --- |
| 1 | Elbow1 | 0 to 11 integers  -1 if not selected |
| 2 | Elbow2 | 0 to 11 integers  -1 if not selected |
| 3 | Arm1 | 0 to 11 integers  -1 if not selected |
| 4 | Arm2 | 0 to 11 integers  -1 if not selected |
| 5 | Leg1 | 0 to 11 integers  -1 if not selected |
| 6 | Leg2 | 0 to 11 integers  -1 if not selected |
| 7 | Wrist | 0 to 10 integers  -1 if not selected |

**Fitness function:**

Fitness function includes three parameters; Age Category, Condition Type and number of exercises. A solution evaluated in the following steps:

1. Condition Type

We calculate the correct condition type for 4-body parts as following:

* 1 exercise for body part, and solution has 1- exercise as well. If condition type of that exercise match user input, then we add 1
* 2 exercises for body part, and solution has 2- exercises as well. We check if both exercises condition type match user input, then add 1
* Wrist body part is a special case, we add 1 if user input 0 and gene 7 is set as -1.
* Return sum of correct, a value between 0 to 4

1. Age category

In the same way, we calculate the correct age category for 4-body parts as we did with condition type. For each body part if correct match user input for age category, we add 1. Returns a value between 0 to 4

1. Number of exercises

* For 2-genes body parts {elbow, arm, and leg} if user input number of exercises is equal to the number of exercises in the solution, then we add 1.
* For wrist, if user input 0 and gene 7 is -1, we add 1. Other case when both user input and solution have 1 exercise.
* Return sum of correct, a value between 0 to 4

We divide correct matches by 4

w\_ConditionType = Condition Type / 4

w\_AgeCategory = Age Category / 4

w\_Exercises = Number of Exercises / 4

Then, weighted condition type by 0.5, age category and number of exercises 0.25,

Fitness = (w\_ConditionType \* 0.5) + (w\_AgeCategory \* 0.25) + (w\_Exercises \* 0.25)

**Genetic operators:**

* Crossover

We used one point crossover, when random value less than crossover rate. The crossover applied on 2 parents; a random point chosen from 1 to 5 because genes indexed from 0 to 6. The result creates 2 children, child 1 (takes first part of parent 1 and second part of parent 2), where child 2 (takes first part of parent 2 and second part of parent 1)

* Mutation

When a random number less than mutation rate, we apply mutation on gene from valid range values. This is an iterative method from index 0 to 6, it applies based on random number and mutation rate. We take into account; the valid range should not include the gene value itself and the value of other gene on same body part.

* Roulette wheel selection

Firstly, total sum population’s fitness is calculated, then we draw a random number (between 0 and total fitness). Roulette wheel is iterating from solution 1 to N, in which partial sum of fitness is compare with the random number we draw before, if partial sum more than random number then the current solution is selected.

* Replacement

Based on roulette wheel selection, 2 parents selected, crossover creates 2 children. After mutation on children, we replace parent 1 by child 1 and parent 2 by child 2 on same population indexes.

**Termination condition:**

We run each experimental settings till one of the following conditions is met:

1. Reaches to maximum number of iterations 20,000
2. The error difference between the optimal solution and best solution fitness is 10-8

Optimal objective is 1

Error = 1 – best solution’s fitness

If error < 10-8

Terminate

**Experimental Settings:**

* Each experiment runs for 20 times, each run with maximum 20,000 iterations
* Each run initialized with new random population.
* We plot genetic performance generations vs fitness, average fitness over 20 times
* A combination of 33 parameters of population sizes, crossover rates, and mutation rates. Total 27 experiments.

| **GA parameter** | **Values** |
| --- | --- |
| Population sizes | 10, 20, 30 |
| Crossover rates | 0.95, 0.8, 0.65 |
| Mutation rates | 0.1, 0.25, 0.4 |

Parameter combinations of the 27 experiments:

[(10, 0.95, 0.1), (10, 0.95, 0.25), (10, 0.95, 0.4), (10, 0.8, 0.1), (10, 0.8, 0.25), (10, 0.8, 0.4), (10, 0.65, 0.1), (10, 0.65, 0.25), (10, 0.65, 0.4), (20, 0.95, 0.1), (20, 0.95, 0.25), (20, 0.95, 0.4), (20, 0.8, 0.1), (20, 0.8, 0.25), (20, 0.8, 0.4), (20, 0.65, 0.1), (20, 0.65, 0.25), (20, 0.65, 0.4), (30, 0.95, 0.1), (30, 0.95, 0.25), (30, 0.95, 0.4), (30, 0.8, 0.1), (30, 0.8, 0.25), (30, 0.8, 0.4), (30, 0.65, 0.1), (30, 0.65, 0.25), (30, 0.65, 0.4)]

**GA performance results:**

| pop\_size = 10 c\_rate = 0.95 m\_rate = 0.1 | pop\_size = 10 c\_rate = 0.8 m\_rate = 0.1 |
| --- | --- |
|  |  |
| pop\_size = 10 c\_rate = 0.95 m\_rate = 0.25 | pop\_size = 10 c\_rate = 0.8 m\_rate = 0.25 |
|  |  |
| pop\_size = 10 c\_rate = 0.95 m\_rate = 0.4 | pop\_size = 10 c\_rate = 0.8 m\_rate = 0.4 |
|  |  |

| pop\_size = 10 c\_rate = 0.65 m\_rate = 0.1 | pop\_size = 20 c\_rate = 0.95 m\_rate = 0.1 |
| --- | --- |
|  |  |
| pop\_size = 10 c\_rate = 0.65 m\_rate = 0.25 | pop\_size = 20 c\_rate = 0.95 m\_rate = 0.25 |
|  |  |
| pop\_size = 10 c\_rate = 0.65 m\_rate = 0.4 | pop\_size = 20 c\_rate = 0.95 m\_rate = 0.4 |
|  |  |

| pop\_size = 20 c\_rate = 0.8 m\_rate = 0.1 | pop\_size = 20 c\_rate = 0.65 m\_rate = 0.1 |
| --- | --- |
|  |  |
| pop\_size = 20 c\_rate = 0.8 m\_rate = 0.25 | pop\_size = 20 c\_rate = 0.65 m\_rate = 0.25 |
|  |  |
| pop\_size = 20 c\_rate = 0.8 m\_rate = 0.4 | pop\_size = 20 c\_rate = 0.65 m\_rate = 0.4 |
|  |  |

| pop\_size = 30 c\_rate = 0.95 m\_rate = 0.1 | pop\_size = 30 c\_rate = 0.8 m\_rate = 0.1 |
| --- | --- |
|  |  |
| pop\_size = 30 c\_rate = 0.95 m\_rate = 0.25 | pop\_size = 30 c\_rate = 0.8 m\_rate = 0.25 |
|  |  |
| pop\_size = 30 c\_rate = 0.95 m\_rate = 0.4 | pop\_size = 30 c\_rate = 0.8 m\_rate = 0.4 |
|  |  |

| pop\_size = 30 c\_rate = 0.65 m\_rate = 0.1 |
| --- |
|  |
| pop\_size = 30 c\_rate = 0.65 m\_rate = 0.25 |
|  |
| pop\_size = 30 c\_rate = 0.65 m\_rate = 0.4 |
|  |

**Analysis of GA performance:**

To analysis genetic algorithm performance of fitness function vs iteration, we look to the 3-parameters as fixed then as a combination of different values.

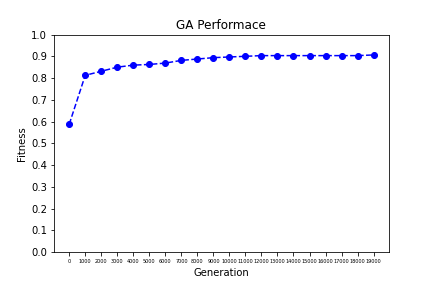
* Population size 30 performance of fitness function is much better than population sizes 10, 20
* Crossover rate 0.95 performance is better as compared to 0.8, 0.65
* Mutation rate 0.1 performance is better as compared to 0.25, 0.4

For combinations:

* Pop\_size 30, c\_rate 0.95, and m\_rate 0.1 is the best combination. It implies the above overall findings.
* Pop\_size 30, c\_rate 0.95, m\_rate 0.4 is comparable to Pop\_size 30, c\_rate 0.8, m\_rate 0.25 which show that decrease in crossover is equivalent to decrease of mutation rate, and vice versa.
* For all combinations of pop\_size 10, 20. The search space is not covered by population size, that yields to the need for large number of iterations to reach optimal fitness.

Best performance is given by parameters combination:

Pop\_size = 30, c\_rate = 0.95, m\_rate = 0.1



**Explained solution:**

Experiment for GA using best parameter combination:

Population size 30, Crossover rate 0.95, and Mutation rate 0.1

We give user inputs as following:

* Age Category: C for child
* Condition Type: S for stroke
* 2 elbow exercises
* 1 upper arm exercises
* 2 lower leg exercise
* 1 wrist exercise

Example of best solution found: [1, 3, 3, -1, 3, 1, 3]

Fitness: 1.0

Your plan is presented below with 6 exercises per day

Elbow: 1. Crawling 2. Bear walking

Upper Arm: 1. Crawling

Knee/ Lower: 1. Seated walking 2. Squatting against a wall

wrist: 1. Finger Flexor using grip device

We note, that genes values indexes start from 0. For example, elbow2 has value 3, which equal to 4th elbow exercise “Bear walking” on table.

We compared solution genes, with table of exercises and found that:

* Condition type is “S stroke” for (elbow1,2, upper arm1, lower leg 1,2, and wrist)
* Age category “C child “also met for all (elbow1,2, upper arm2, lower leg 1,2, and wrist)
* Number of exercises is 6 which met user inputs

Condition Type = (1 + 1 + 1 + 1) / 4 = 1

Age Category = (1 + 1 + 1 + 1) / 4 = 1

Exercises = (1 + 1 + 1 + 1)/ 4 = 1

Weighted condition type by 0.5, age category and number of exercises 0.25,

Fitness = (Condition Type \* 0.5) + (Age Category \* 0.25) + (Exercises \* 0.25)

Fitness = (1 \* 0.5) + (1 \* 0.25) + (1 \* 0.25) = 1