

Algorithm: Random Selection

Input: Original pool(Sorted by fitness), Pool of offsprings, Network data, Size

Output: Next Generation

Procedure: Random Selection(original pool, pool of offsprings, network data, size)

Start procedure

 Crossover(original pool, pool of offsprings, network data)

 Mutation(pool of offsprings, network data, size)

 Copy the pool of offsprings to the original pool replacing the old population

End procedure

Algorithm: Crossover

RAND_MAX := Upper bound of the range from which a random number is generated by the library. This is a predefined constant in the standard C library.

Input: Original pool, Intermediate pool, Pool of offsprings, Network data, Size

Output: Next Generation

Procedure: Crossover(original pool, intermediate pool, pool of offsprings, network data, size)

Start procedure

 random_value = random() / RAND_MAX

Label: Repeat forever

 IF the new offspring pool is filled completely THEN

 Break the loop

 END IF

 IF error_flag = 1

 random_value = random() / RAND_MAX

 END IF

 crossover points = random() {for the range between 1 and total number of new links}

 index1 = random()

 Keep picking index2 = random() until it is not EQUAL to index1

 IF random_value > crossover probability THEN

 Do not crossover the 2 candidates at positions index1 and index2. Skip the rest of the loop and jump back to Label.

 random_value = random() / RAND_MAX

 END IF

 Do either a single point / two point crossover operation on 2 candidates

 at positions index1 and index2 in intermediate pool and store in temporary memory

 IF the above offspring is non zero AND

 the above offspring is budget feasible AND

 the above offspring is not a duplicate from original pool AND

 the above offspring is not a duplicate from the current offspring pool THEN

 Add this to the offspring pool

 ELSE

 IF MAX_ATTEMPTS reached THEN

 Set error_flag = 1 and copy over the best of the 2 parents to the pool

 Skip the rest of the loop

 ELSE

 Repeat the above process till MAX_ATTEMPTS by skipping the rest of the loop and jumping back to Label.

 END IF

 END IF

 Jump to Label

End procedure