

Algorithm: Default Selection

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

Output: Next Generation

Procedure: Default 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, pool size)

Label: Repeat till loop_index < pool size

 Model_problem(pool of offsprings[loop_index])

 lptimize(model)

 Pool of offsprings[loop_index].fitness = fitness(pool of offsprings[loop_index])

Jump to Label

Sort(pool of offsprings, pool size)

Mutation(pool of offsprings, network data, size)

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

Merge of both the original pool and offspring pool into a new generation considering the best candidates from both and discarding the others

End procedure

Algorithm: Crossover

RAND_MAX := Upper bound of the range from which random number is generated. 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

randomvalue = random() / RAND_MAX

Label: Repeat forever

 IF MAX_ATTEMPTS reached THEN

 randomvalue = random() / RAND_MAX

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

 index1 = random()

 index2 = random()

 IF randomvalue > 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.

 IF index1 == index2 THEN skip the remaining part of loop and jump back to Label

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 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 AND

the above offspring is non zero THEN

 Add this to the offspring pool

ELSE

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

End procedure