**Computational Complexities of Approaches**

**For GAC, the computational complexity can be represented as follows:**

The chromosome length is defined as , where , , and represents the number of collection centers, warehouses, and demand points, respectively. In the population, each individual is initialized, leading to a time complexity of , where is the population size. For each individual, fitness is evaluated across all scenarios, resulting in a complexity of where represents the time to compute fitness for one scenario of one individual. Sorting for elitism requires . Roulette wheel selection typically requires . Assuming each crossover operation processes the entire chromosome, the complexity is . Similarly, assuming each mutation operation processes the entire chromosome, the complexity is also . Replacing population with new offspring is typically a linear operation with a complexity of . The initialization phase complexity is . The total time complexity per iteration is , which simplifies to . Overall, for iterations, the computational complexity is .

**For GACRFNI, the computational complexity can be represented as follows:**

The RF is trained using only the current population in each iteration. The population initialization leads to a time complexity of . Fitness evaluation is resulting in a complexity of . RF training requires . Sorting for elitism requires . Roulette wheel selection requires to select individuals for the mating pool. For crossover, the complexity is . In each iteration, fitness values are predicted for the current population. The time complexity for per iteration is , where is the maximum depth of the trees. Each prediction involves traversing each tree in the RF, which takes . Removing worst individuals based on predicted fitness requires . Updating population takes . For the mutation operation, the complexity is . The initialization phase complexity is . The total time complexity per iteration is by considering . Overall, for iterations, the computational complexity is .

**For GACRFI, the computational complexity can be represented as follows:**

The RF is trained using only the current population in each iteration. The population initialization leads to a time complexity of . Fitness evaluation is resulting in a complexity of . For initialization, RF training requires , where . The total training data size can be computed as . Sorting for elitism requires . Roulette wheel selection requires to select individuals for the mating pool. For crossover, the complexity is . In each iteration, fitness values are predicted for the current population. The time complexity for per iteration is , where is the maximum depth of the trees. Each prediction involves traversing each tree in the RF, which takes . Filtering worst individuals based on predicted fitness requires . Updating population takes . For the mutation operation, the complexity is . Select best individuals for RF requires . The time complexity for RF training during iterations is . The initialization phase complexity is . The total time complexity per iteration is . Overall, for iterations, the computational complexity is .

GACRFNI is computationally less intensive in terms of RF training since it only uses the current population, while GACRFI has a higher computational complexity due to the increasing size of the training data over iterations.