1 Methodology

In this chapter, the simulator used to investigate the research questions is described. An overview of the system is presented in section 2. Section 3 includes implementation details, and design decisions made when implementing the genetic algorithm which is the foundation for all the population distributed genetic algorithms. Sections 4-6 contains implementation details of the population distributed genetic algorithms. The wind scenarios used to evaluate the different population distributed genetic algorithm are described in section 7, and the choice of implementing the genetic algorithm from scratch is defended in section 8.

2 System Architecture

The program is implemented in Java and the interactions between the different classes of the program are shown in figure 1. The GeneticAlgorithm class is extended by the three population distributed genetic algorithm classes: IslandModel, CellularModel and PoolModel. In addition, the GeneticAlgorithm class is also implemented as instances in all three population distributed algorithms. The main loop of the program is contained in the GeneticAlgorithm class. It uses instances of the classes WindScenario, WindFarmLayoutEvaluator, Population, AdultSelection, ParentSelection, Crossover and Mutation. AdultSelection, ParentSeleciton and Crossover are interfaces that needs to be implemented if new methods are to be added to the program. Mutation is a class containing four different mutation methods.

3 Genetic Algorithm

As mentioned in reference chapter background, the genetic algorithm consists of five steps: Adult selection, parent selection, recombination and fitness evaluation as shown in figure 2. The implementation details of each step is described below.

3.1 Adult Selection

Adult selection is the process of selecting which individuals that are allowed to step into the adult pool and thereby become potential parents for the next generation of individuals. Three adult selection mechanisms were implemented in this thesis: Full generational replacement, generational mixing, and overproduction. Each method was tested in order to decide which adult selection method was more suitable for solving the wind farm layout optimization problem.

Full generational replacement, is the simplest adult selection mechanism consisting of replacing all the individuals in the previous adult population with the

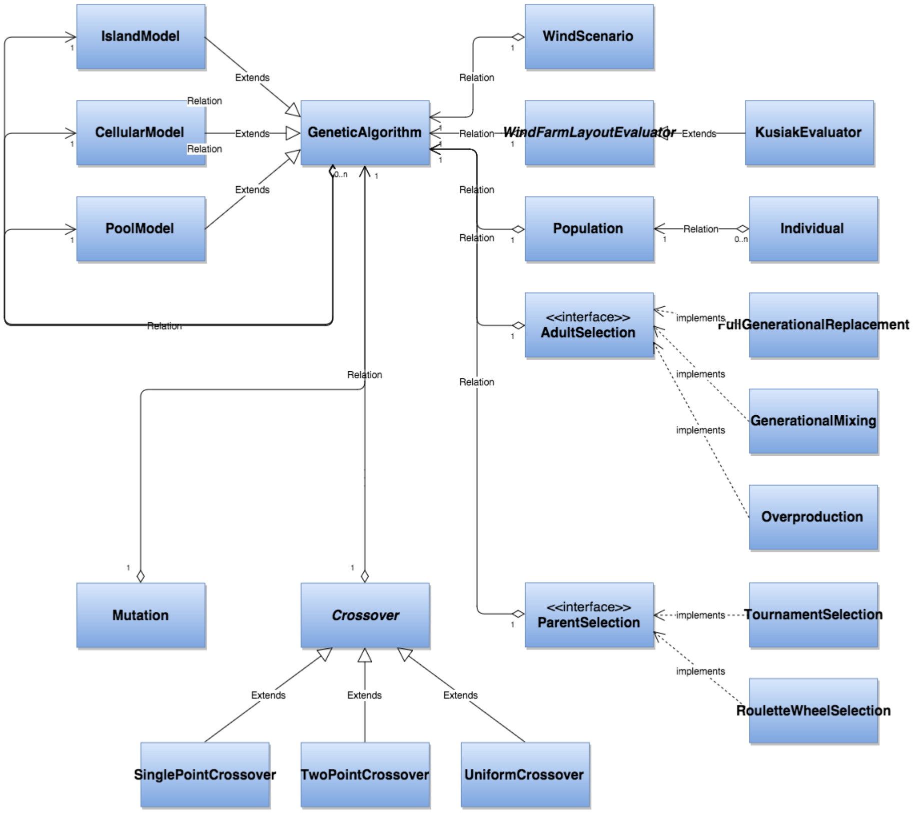

Figure 1: Class Diagram.

newly generated child population.

The second method, generational mixing, is illustrated in figure 3. As can be seen in the figure, the four best individuals (individuals with lowest fitness) from the pool consisting of all the newly generated children and the previous adult population are selected as the new adult population.

Overproduction, the third adult selection mechanism, is illustrated in figure 4. The newly generated child population consist of twice as many individuals than in the adult pool. Therefore, the child population have to compete against each other for the spots in the adult pool and only those with better fitness are able to survive.

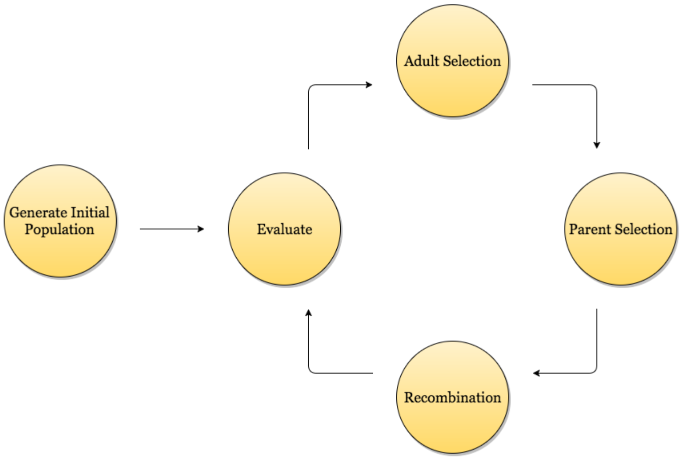

Figure 2: Genetic algorithm.

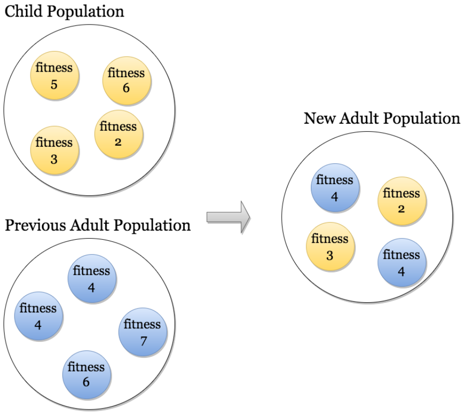

Figure 3: Generational mixing. The best individuals, those with lowest fitness, from the previous adult pool and the new child population are selected to represent the new adult pool.

3.2 Parent Selection

Parent selection is the selection of which adults become parents for the next child generation. When choosing parent selection method there are a few concerns that needs to be addressed. First, it is important that parents with good genes, i.e. higher fitness, gets their genes transferred to the next generation. However, it is also important to keep diversity in the population so that one does not end wit a sub-optimal solution; a local maxima. Two parent selection methods are implemented for the genetic algorithm: Tournament selection and

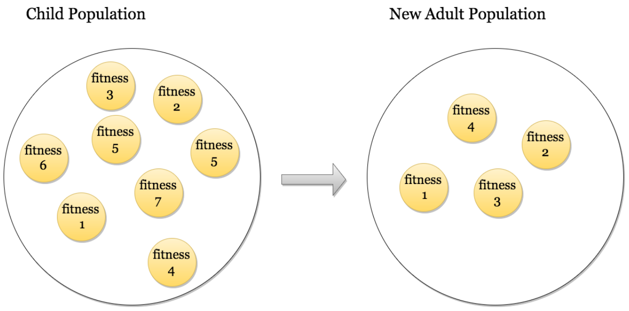

Figure 4: Overproduction. The newly generated child population consist of twice as many individuals as there are room for in the adult population, therefore only the fittest individuals from the large child population grow up into adults.

roulette wheel selection.

In tournament selection, a given number of individuals are drawn randomly from the population. The number of individuals drawn is decided by the variable "tournament size". These individuals compete in a tournament for one spot in the parent pool. The individual with "best", i.e. lowest fitness is selected as a parent. These tournaments continue until the adult pool is full. Figure 5 shows how tournament selection works. As can be seen in the figure, three individuals are drawn randomly from the adult pool, meaning that the tournament size in this example is 3. The "best" individual, the individual with fitness 4 is the tournament winner and is allowed to enter the adult pool.

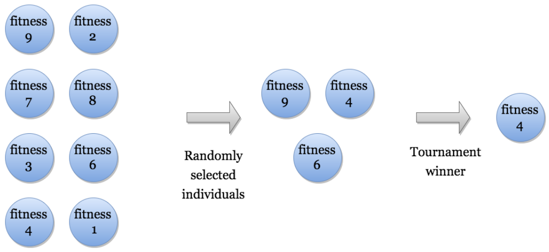

Figure 5: Tournament selection.

Different values of "tournament size" needs to be tested in order to find how settings that allow the algorithm to explore different solutions, but that also prioritize the best solutions. In chapter reference results chapter, results obtained for different tournament sizes are compared to find which is best for the wind farm layout optimization problem.

Roulette wheel selection assigns a probability of being chosen as parent to each individuals which is proportional to its fitness compared to all other solutions. Therefore, individuals with better fitness will be more likely to selected for the parent pool. Figure 6 shows how roulette wheel selection works. The roulette wheel on the left shows the probability of each of the four individuals been selected. Since individual$_4$ has the best fitness, it has a larger probability of being selected than the others.

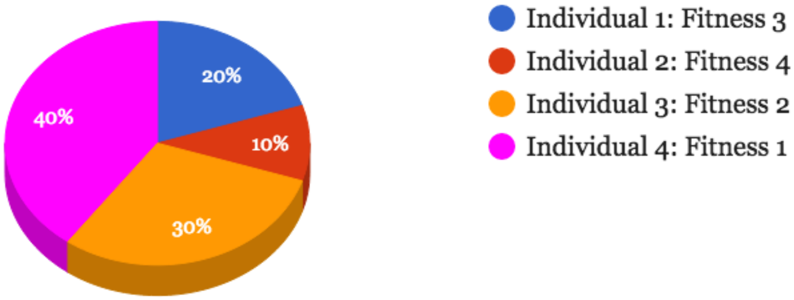

Figure 6: Roulette wheel selection. The roulette wheel is shown to the left, the four individuals to the right. Individual$_4$ has a four times better fitness than individual$_2$ and therefore has a four times larger probability of being selected.

3.3 Genetic Operations

This subsection gives an overview over the genetic operations used to produce the next child generation. Four crossover methods, elitism and four mutation methods are implemented and will be presented below.

3.3.1 Crossover and Elitism

Crossover is the recombination method utilized by the genetic algorithm to perform sexual reproduction. A crossover operation produced two children by recombination of the genes of tho parent individuals. The genetic algorithm implemented for this thesis has three crossover methods to chose from: Single point crossover, two point crossover and uniform crossover.

3.3.2 Mutation

3.4 Wind-, Wake- and Power Model

3.5 Fitness Function

4 Island Model

5 Cellular Model

6 Pool Model

7 Scenarios

8 Motivation