

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, ParentSelection 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

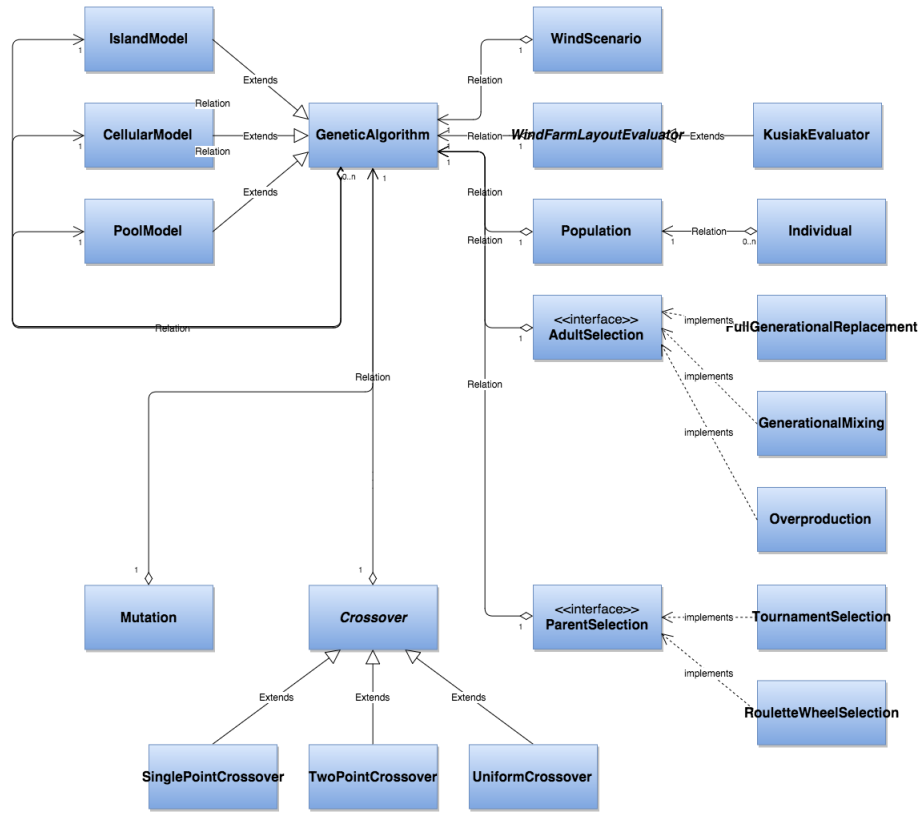


Figure 1: Class Diagram.

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.

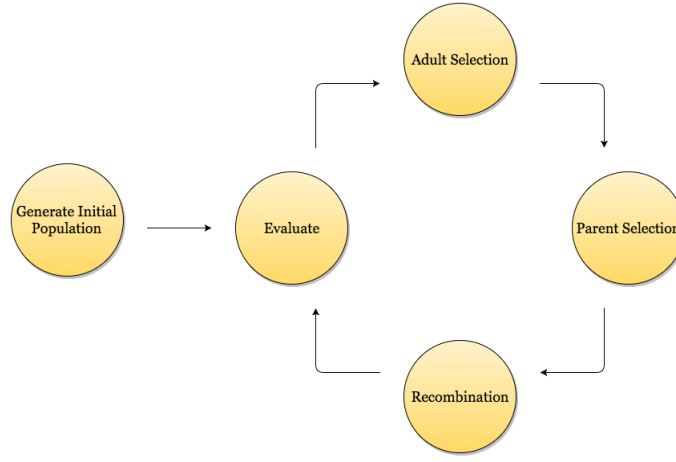


Figure 2: Genetic algorithm.

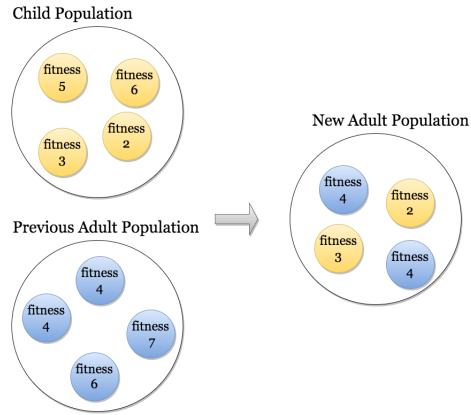


Figure 3: Generational mixing. The best individuals, those with lowest fitness, from the previous adult pool and the new child population are selected to represent the new adult pool.

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

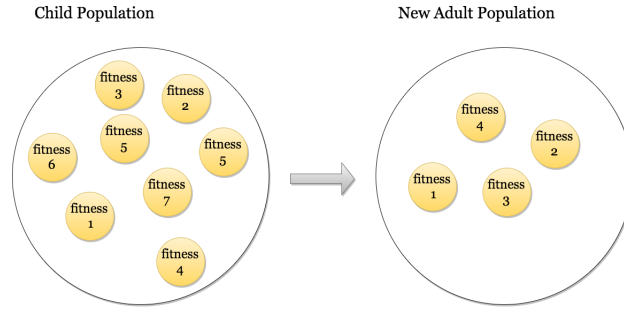


Figure 4: Overproduction. The newly generated child population consist of twice as many individuals as there are room for in the adult population, therefore only the fittest individuals from the large child population grow up into adults.

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.

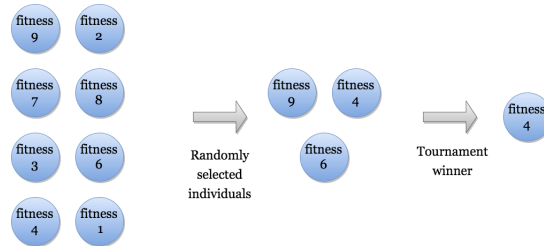


Figure 5: Tournament selection.

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 individual₄ has the best fitness, it has a larger probability of being selected than the others.

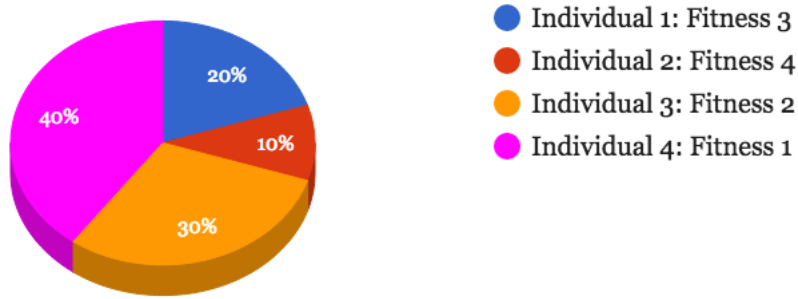


Figure 6: Roulette wheel selection. The roulette wheel is shown to the left, the four individuals to the right. Individual₄ has a four times better fitness than individual₂ and therefore has a four times larger probability of being selected.

3.3 Genetic Operations

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

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 recombining genes of two 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. These are all presented in figure 7. As shown in figure 7a, single point crossover is a method where a random position, called the crossover point, is generated. All genes from the first parent prior to the crossover point are copied to the first child, and all genes after the crossover point are copied to the second child, and all genes from the second parent prior to the crossover point is copied to the second child, and all genes after the crossover point is copied to the first child. Two point crossover is similar to single point crossover, except that it uses to crossover points instead of one.

This is shown in figure 7b. In uniform crossover, there is a fifty percent probability that each gene will be drawn from each parent as shown in figure 7c. This crossover method results in child individuals that might not inherit the gene patterns from their parents the same way as in the other two methods. Since the relative positions between wind turbines are very important when solving the wind farm layout optimization problem, uniform crossover might not be the best choice. **Is this true? If it is, talk about this in the results part, remove it from here.**

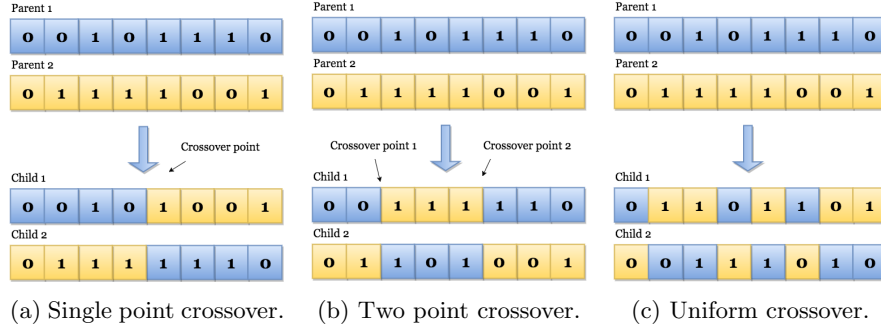


Figure 7: Crossover methods.

3.3.2 Mutation

Although crossover is a powerful genetic operator, which able child individuals to inherit genes from the best parent individuals, it is not enough on its own. Without any way to mutate genes, one can end up with a population where every individual has the same gene value in the same position, this means that no recombination will be able to change the value of that gene. Mutation is the operation used by the genetic algorithm to make sure that the population do not get sterile. In nature, genes can be mutated in numerous ways. In this thesis, four mutation methods are implemented to mimic the mutations methods that can happen in genes, these are called: Flip mutation,

3.4 Wind-, Wake- and Power Model

The evaluation class uses the same wake-, wind- and power model as [?]. The wake model used is the classical Jensen model Jensen, which is used in almost every study of the wind farm layout optimization problem, as can be seen in table ??.

Wind distribution is modeled using the Weibull distribution, a continuous probability distribution shown to model wind distribution quite well Justus. The probability density function is shown in equation ??

$$f(x; c, k) = \begin{cases} kc \left(\frac{x}{c}\right)^{k-1} e^{-\left(\frac{x}{c}\right)^k} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1)$$

where k is called the shape parameter and c is the scale parameter, and $k, c > 0$. In most of the wind scenarios provided by GECCO 2015, $k \approx 2$, this is shown empirically to be a good value for wind speed distribution Justus. On the other hand, the shape parameter vary for each wind direction. Figure ?? shows the Weibull distribution plotted for $k = 2$ and for different values of c .

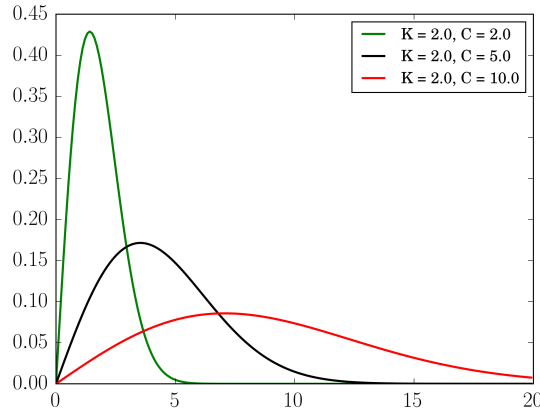


Figure 8: The Weibull distribution plotted for $k = 2$ for different values of the scale parameter c .

The wind scenarios used in this thesis are therefore a specification of the shape- and scale parameters for every wind direction, where wind direction is partitioned into 24 different directions. Twenty wind scenarios are provided by GECCO 2015, ten which simply specify wind distribution parameters, and ten that specify wind distribution parameters and locations of obstacles.

The power curve used is also the same as used in [?], it is a linear function, shown in equation ??,

$$f(v) = \begin{cases} 0 & \text{if } v < v_{cut-in} \\ \lambda v + \eta & \text{if } v_{cut-in} \leq v \leq v_{rated} \\ P_{rated} & \text{if } v_{cut-out} > v > v_{rated} \end{cases} \quad (2)$$

Here λ is the slope parameter, v the wind speed, η the intercept parameter, P_{rated} is the fixed power output, and v_{cut-in} is the cut-in speed; the minimum speed for which the turbine produces power, and $v_{cut-out}$ is the cut-out speed; the maximum wind speed for which the turbine is kept on.

3.5 Fitness Function

The main task of the evaluation classes is to calculate the fitness of each individual based on the fitness function. The fitness function to be optimized is provided by GECCO, and is displayed in equation ??.

$$fitness = \frac{(c_t \cdot n + c_s \cdot \frac{n}{m}) \left(\frac{2}{3} + \frac{1}{3} \cdot e^{-0.00174n^2} \right) + c_{OM} \cdot n}{\left(\frac{1-(1+r)^{-y}}{r} \right)} \cdot \frac{1}{8760 \cdot P} + \frac{0.1}{n} \quad (3)$$

Description and numerical values of all parameters given in equation ?? are displayed in table ?. As can be seen in this table, the values of n , the number of turbines, and P , farm energy output, are not given. This is because the number of turbines, together with the turbine positions, are the parameters to be optimized by the genetic algorithm. Farm energy output is the indirect parameter that we are trying to optimize. It is dependent on turbines count, position, wind scenario and so on, and is of course therefore not provided in table ? either.

Table 1: Description and value of each parameter used in the objective function provided by GECCO 2015.

| Parameter | Description | Value |
|-----------|------------------------------------|-----------|
| c_t | Turbine cost (usd) | 750 000 |
| c_s | Substation cost (usd) | 8 000 000 |
| m | Turbines per substation | 30 |
| r | Interest rate | 0.03 |
| y | Farm lifetime (years) | 20 |
| c_{OM} | Yearly operating costs per turbine | 20 000 |
| n | Number of turbines | |
| m | Farm energy output | |

Intuitively, the objective function can be divided into different parts. The first parenthesis in the nominator of the first fraction is the construction cost, while the second parenthesis is the economies of scale and the third part of the nominator is yearly operating costs. The denominator represents the interests. The denominator of the second fraction describes yearly power output, while the number 0.1 in the nominator of the last fraction is a farm size coefficient.

- 4 Island Model
- 5 Cellular Model
- 6 Pool Model
- 7 Scenarios
- 8 Motivation