Journal of Ocean and Wind Energy (ISSN 2310-3604)

Vol. 1, No. 4, November 2014, pp. 209–216

http://www.isope.org/publications

Reanalysis of Jacket Support Structure for Computer-Aided Optimization of Offshore Wind Turbines with a Genetic Algorithm

Sebastian Schafhirt, Daniel Zwick and Michael Muskulus*
Department of Civil and Transport Engineering, Norwegian University of Science and Technology
Trondheim, Norway

The optimization of jacket support structures for offshore wind turbines is a nontrivial task. Due to nonlinear and time history-dependent effects, the analysis is simulation-based. Structural optimization using time-domain simulations is computationally demanding and difficult and typically requires a gradient-free approach. A genetic algorithm can be used for this purpose, but is limited by the computational resources available. This paper presents an approach to modifying the standard genetic algorithm for the automatic optimization of offshore wind turbines with jacket support structures under fatigue constraints. It is shown that performing a reanalysis of the jacket structures, by using performance data from earlier analyses in parallel with the simulation-based analysis process, requires a smaller number of iterations to obtain improved designs. Thus, the use of reanalysis within the genetic algorithm speeds up the algorithm significantly.

INTRODUCTION

Offshore wind energy plays a central role in energy planning around the world. The United States has excellent wind resources, and many projects for offshore wind farms are under development. China's target is to install 30 GW offshore, and Europe aims to produce 20% of final electricity consumption with renewables by 2020 (Davey et al., 2012). However, the costs of this form of energy are still high. A cost reduction for electricity from offshore wind would be more than welcome and will be crucial for the future success of offshore wind energy in sustainability-focused international policies. A reduction in cost can be achieved by an optimized design for the support structure, since it constitutes a significant part of the capital costs. For offshore wind energy projects, the supply and installation of the substructure represent around 20% of the capital costs (Azau and Casey, 2011). Designs are typically adapted for the specific site conditions encountered, but a limiting factor is the time and effort needed to determine optimal solutions for each site. Especially for multi-member support structures such as jackets, the optimization process becomes a challenging task due to the large number of parameters, the complex numerical models, and the time-consuming time-domain analyses (Muskulus and Schafhirt, 2014). The latter is necessary to obtain reliable results, since offshore wind turbines (OWTs) are highly dynamic and tightly coupled systems and are subjected to nonlinear and time history-dependent effects, such as irregular waves and unsteady aerodynamic loading. Simulations in the time domain, therefore, are the most accurate analysis method currently available for integrated models of OWTs.

In the past, however, specialized and reliable simulation tools for OWTs were not available. Approaches using different analysis tools for the support structure and rotor, for example, typically led to a systematic error in the fatigue assessment, since dynamic interactions between the wind turbine and support structures were not captured accurately enough. A comprehensive, dynamic simulation of an entire OWT captures these dynamic interactions, but demands extensive computational effort. Therefore, it is hardly surprising that optimization methods based on static loads are still widely used in the industry, which frequently lead to suboptimal structures due to the use of a large factor of safety.

While static load optimization methods have been a basic component of research in the field, studies have more recently moved towards integrated simulations of the entire OWT under simultaneous aerodynamic and hydrodynamic loading. For example, King et al. (2013) used an integrated simulation of jacket support structures and thereby studied the potential for cost reduction under modification of the joint design. Ashuri (2012) optimized the rotor and tower of an OWT simultaneously by using time-domain simulations and gradients obtained by finite differences. Gradient-based optimization algorithms are the method of choice in structural optimization, but they become very difficult if gradients need to be estimated by finite differences, since the computational cost typically rises exponentially with the number of parameters.

Simulation-based optimization therefore often works with gradient-free methods (Gosavi, 2003). For example, Yoshida (2006) described the use of a genetic algorithm (GA) for the computer-aided optimization of a large tower using the aero-elastic simulation tool Bladed (GL Garrad Hassan, 2011). This is a stochastic search method based on the Darwinian principle of natural selection and survival of the fittest. Recently, we have successfully used such an algorithm for the optimization of jacket structures (Pasamontes et al., 2014); the method is based on modifications of member dimensions and structural topology. These calculations are not trivial, especially when considering fatigue constraints and including the nonlinearity of the fatigue damage calculation in the optimization of the system. The approach requires numerous iterations, making it currently impractical in an industrial setting. Even if the optimization is based only on one load case, each iteration step includes many lengthy analyses in the time domain. It is therefore important to reduce the number of time-domain simulations as much as possible. Reanalysis, in which data from previous time-domain simulations are used to approximately assess the performance of a new design, is an interesting possibility. This approach is widely used for linear static problems and has already been extended

*ISOPE Member.
Received April 9, 2014; updated and further revised manuscript received by the editors September 2, 2014. The original version (prior to the final updated and revised manuscript) was presented at the Twenty-fourth International Offshore and Polar Engineering Conference (ISOPE-2014), Busan, Korea, June 15–20, 2014.
KEY WORDS: Offshore wind turbines, support structures, structural optimization, genetic algorithm, reanalysis.

to nonlinear, dynamic, and nonlinear dynamic problems (Kirsch, 2010). In this paper, we describe the use of reanalysis and other modifications to tune the standard GA in order to speed up the optimization process. Results show that the number of iteration steps can be reduced to less than one third, while solutions similar to the ones obtained by the original algorithm are still achieved.

THE OFFSHORE WIND TURBINE

This method has been tested with the OC4 reference jacket (Vorpahl et al., 2011) and the well-known NREL 5-MW baseline wind turbine on top (Jonkman et al., 2009). This three-bladed upwind turbine has been used as a reference turbine in several research projects. The jacket support structure used in this study was originally designed by Rambøll AS (Vermula, 2010) for the UpWind project and has been used in the first phase of the Offshore Code Comparison Collaboration Continuation (OC4) project (Popko et al., 2012). The OC4 jacket is a four-legged lattice support structure with four bays and slightly inclined legs. The structure includes mud braces at the lowest bay and is clamped rigid at the mud line. The transition piece is modeled as a block of concrete that is penetrated by the upper parts of the four jacket legs. The design does not include any secondary steel, such as a j-tube, a boat landing, or anodes, and is designed for the 5-MW reference wind turbine. The total jacket height from the mud line to the bottom of the tower is 70.15 m, which leads to a hub height of 90.55 m over mean sea level. The design is not representative of commercial jacket structures and offers a lot of optimization potential. It is used here due to its simplicity and relatively widespread use as a reference design.

The flexible multibody simulation tool, FEDEM Windpower (Version R7.0.4, Fedem Technology AS, Trondheim), was used for a time-domain analysis of the offshore wind turbine. FEDEM Windpower allows a dynamic analysis of the entire wind turbine and has been verified in the OC3/OC4 project (Popko et al., 2012). The model of the OWT implemented in the FEDEM Windpower is shown in Fig. 1.

THE CONCEPT OF GENETIC ALGORITHM

The GA is a stochastic search method based on the Darwinian principle of natural selection and survival of the fittest. After Holland's (1975) pioneering work in the 1970s, which had immense influence on GA research, research interest in this field has grown exponentially. Today, the GA is widely used for optimization problems in several fields (e.g., Goldberg, 1989; Bäck, 1996; Gen and Cheng, 2007). In this study, it is used to optimize a support structure for an OWT.

Fig. 1 Offshore wind turbine model in FEDEM

The description of the GA closely follows biological terminology, in which the jacket structure is analogous to an individual organism. Each jacket has a different design. The geometry is kept constant, but the dimensions (diameter and thickness) of the elements (legs and braces) of the jacket structure are subjected to change, making the specific design variation of the four-bay OC4 jacket accurately described by 16 design variables. Each design variable represents a single gene in the terminology of GA. Integers, which define the diameter and thickness in mm, are selected for this purpose. The GA requires a single string, the so-called chromosome, in which the design variables are encoded. This process begins with transforming each design variable into a binary string and combining these binary strings to create a chromosome. The GA presented in this study uses 11 bits and 6 bits to encode the diameter and thickness, respectively. The number of bits used to encode each design variable in the binary strings determines the complete length of the chromosome, totaling 136 bits in this case. A minimum thickness of 10 mm is chosen, while the minimum diameter for the leg and brace is set to 800 mm and 600 mm, respectively. This leads to a maximum thickness of 73 mm (variable range of $2^6 = 64$ mm) for both legs and braces, and a maximum diameter of 2847 mm and 2647 mm (variable range of $2^{11} = 2048$ mm) for legs and braces, respectively.

In each iteration step, thought of as a new generation, a certain number of individuals are analyzed. The population size, i.e., the total number of individuals, is predefined and kept constant over the generations. The analysis of the individual leads to a fitness value, which describes the relative importance of the individual with respect to the current generation. A higher fitness of an individual implies a better jacket design. The basic idea of the GA is to generate offspring (a new population) from the current population in such a way that the average fitness of the population improves. The GA continues until the maximal number of generations is reached or a user-defined termination criterion is fulfilled (e.g., no improvement in the average fitness for a certain number of generations). Offspring can be generated by reproduction, crossover, and mutation. These elements of the GA will be explained in detail in the following sections.

Population Size

The choice of population size is a trade-off between efficiency and effectiveness. A small number of individuals do not allow for effective exploration of the search space, while too many individuals impair the efficiency of the algorithm such that no solution can be expected in a reasonable amount of computation (Reeves and Rowe, 2003). Many researchers (e.g., Goldberg, 1985; Goldberg and Deb, 1991; Grefenstette, 1986; Schaffer et al., 1989) have searched for a criterion for an optimal population size for a given string length. Goldberg recommended an optimal population size as an exponential (Goldberg, 1985) and linear (Goldberg and Deb, 1991) function of the string length, while empirical studies have subsequently demonstrated that a population size of 30 individuals is adequate in many cases. For the GA used in this study, a population size of 15 individuals was chosen due to simple system-specific reasons. The time-domain simulations were carried out in parallel on a 16-core workstation; thus, 15 analyses resulted in more or less the same computational cost that a smaller number of analyses would have produced. The remaining core was used to perform reanalysis simultaneously with the time-domain simulation. A population size of 15 individuals was considered sufficient, since the probability that every point in the search space is reachable from the initial population only by crossover is high enough. This was calculated by using the following formula derived from Reeves and Rowe (2003):

$$P = (1 - (1/2)^{N-1})^{l} \tag{1}$$

This probability exceeds 99.1% for the chosen population size ($N = 15$) and a string length of $l = 136$. At that string length, a population size of 19 would be necessary to ensure that the probability would exceed 99.9%.

Initial Population

It is possible to use the existing design of the OC4 jacket and seed the initial population with designs similar to this solution; however, it has been found in general that the GA tends to prematurely converge to suboptimal solutions, although seeding helps the GA find better solutions more quickly than it can from a random start (Surry and Radcliffe, 1996). Therefore, the initial population of designs for the algorithm is randomly generated, with each parameter being distributed uniformly within the limits mentioned above. The individuals used in the initial population were analyzed with FEDEM Windpower to ensure that all individuals fulfilled the design requirements.

Fitness Calculation

An optimization process is usually defined as the process of finding the optimum of some characteristic. A so-called objective function indicates the relative importance of a solution in terms of a single numerical value, the so-called fitness of the individual, with a higher fitness implying a better design. The objective function is often defined in terms of the weight, since the costs of material, manufacturing, transport, and foundations are closely associated (Wood, 2011). However, the objective function can also be defined to maximize the stiffness, optimize the eigenfrequencies, or optimize the use of the full damage capacity. In our case, the objective function used is a simple linear function that depends only on the weight of the individuals. The constraints on the design requirements are included by setting the fitness to zero for individuals that do not satisfy the performance requirements. This analysis is simulation-based and considers ultimate and fatigue limits as well as geometric conditions (e.g., constraints on the ratio of the thickness and diameter of the structural elements), as discussed in the following sections.

Fitness Scaling Function

Independent from the definition of the objective function, which computes a raw fitness score (f_{raw}) for each individual, fitness is scaled (f_{sc}) to improve the reproduction rate of the fitter individuals and to depress the reproduction rate of the weaker ones. The scale on which the fitness is measured is important. Fitness values between 10 and 20 are more clearly distinguished than values between 1010 and 1020, for example (Reeves and Rowe, 2003). Several different scaling methods can be found in the literature, such as ranked, power law, sigma truncation, exponential, or top fitness scaling, and have been analyzed in several empirical studies (e.g., Man et al., 1996; Sadjadi, 2004). We use one of the most popular methods, linear scaling (Goldberg, 1989):

$$f_{\text{sc}} = a \cdot f_{\text{raw}} + b \tag{2}$$

where the parameters a and b are obtained from the condition that the mean of f_{sc} and f_{raw} should be the same and that the maximum fitness is the double of its mean.

Selection

During the selection process, pairs of two individuals are randomly selected for a crossover operation. The selection of parents will determine the genetic diversity of the subsequent generation and can be guided to help avoid premature convergence. Baker (1987) developed stochastic universal selection, which corresponds to systematic random sampling (Lohr, 1999), from the viewpoint of statistical sampling theory (Reeves and Rowe, 2003). (See Chambers (1995) for a review.) This study uses a simpler selection method, in which a pair of parents must always consist of two different individuals. Hence, an individual will never mate with itself. The selection of the parents is related to the scaled fitness of each individual. Typically, GAs use a so-called roulette wheel selection, in which the selection probability of the individual is directly proportional to its scaled fitness. This approach works well for a large population size, but can rapidly lead to a loss of desired variability if the number of times that a certain individual is selected during the process is a significant part of the total population size. The selection of the parents, therefore, is carried out in such a way that every individual will become a parent for a new child and only the second parent is selected by using the roulette wheel selection described above. While certain individuals may still be under- or overrepresented in the selection process, this method prevents too rapid a loss of genetic material.

Crossover

After the parents for each child are selected, crossover operations are conducted to generate a new generation of offspring. During this process, the chromosomes of the parents are combined to create a hybrid chromosome for the child. The most common methods to perform the crossover are the one-cut-point and two-cut-points methods (Arora, 2012). The cut point defines the point where the chromosomes of the parents are split and is randomly selected within the chromosome. Figure 2 shows an example of a crossover operation with one cut point and a chromosome length of 25 bits. Since each chromosome consists of 136 bits, a multipoint crossover operation is used in this study. The number of cut points is randomly selected from a range from 6 to 8 for each crossover operation performed.

Mutation

Another possible way to generate offspring is through mutations, in which single bits within the string will be randomly changed based on the current value. The idea of this process is to encourage the development of new genetic material to avoid premature convergence. The mutation rate defines the number of mutations per chromosome. The GA used in this study starts with a baseline mutation rate of 0.05, which means that 5% of the genes of each chromosome will be switched from 0 to 1 or vice versa in each newly generated child. The mutation process is shown in Fig. 3.

Reproduction

In addition to crossover and mutation, reproduction is a third mechanism for defining a new generation. Reproduction requires

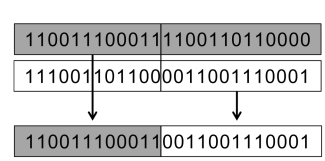

Fig. 2 One cut-point crossover for chromosome with 25 bits

1100111000111100110110000

↓

1100111000011100110110000

Fig. 3 Mutation of bit 11 for chromosome with 25 bits

selecting individuals from the current generation and carrying them unchanged into the new generation. An individual will be selected for reproduction if the fitness of the individual is higher than the fitness of a child generated by crossover or mutation. The reproduction process safeguards individuals with high fitness from extinction.

IMPROVEMENT OF THE GENETIC ALGORITHM

Our implementation of the GA consists of a design and analysis interface. The design interface is responsible for evaluation, reproduction, crossover, and mutation of the individuals, while the analysis interface performs the numerical analysis of the individuals (see Fig. 4). As already mentioned, FEDEM Windpower is used here for the analysis interface. Input files automatically created from a design of the current population are sent to FEDEM Windpower, where a time-domain simulation is carried out. The analysis of the design encompasses a single 90-second load case for the OWT under power production, of which the first 60 seconds are removed as transient. The load case is performed with a wind speed of 10 m/s, a turbulence intensity of 15.2% (normal turbulence model), and an irregular sea state with a significant wave height of 1.48 m and a peak spectral period of 5.74 s. This load case results in a major contribution to the total fatigue damage during power production due to the relatively large product of damage and probability of occurrence. The environmental conditions are in accordance with the UpWind design basis for a deepwater site (Fischer et al., 2010). After the simulation finishes, FEDEM Windpower returns time history response files for forces and moments for each member connected to a joint, which are then accessed by the design interface.

The design interface consists of custom-written scripts in MATLAB (Version R2013a, MathWorks, Kista). For the evaluation, the 30-second results from FEDEM Windpower are used to determine extreme loads and to extrapolate the fatigue damage of the structure for a lifetime of 20 years by using the hot-spot stress approach, as described in Zwick et al. (2012). Besides the fatigue damage and extreme load calculations, the check of geometrical validity with respect to certification guidelines (Det Norske Veritas, 2010) and the weight calculation are necessary to determine the fitness of the individual. The fitness is then scaled, and the selection, crossover, and mutation processes follow, leading to a new generation. The GA iterates until a maximum of 100 generations is reached. A special termination criterion is not defined for this algorithm.

Most of the optimization processes in which a GA is applied define a crossover and mutation probability and use these parameters to control the convergence properties and qualities of the solutions. In contrast to this typical use, the GA used in this study performs a crossover as well as a mutation for each individual.

Reanalysis

A new idea for the adaptation of GAs to our design problem is that the mutation process is used at two different stages in this study, while typically it is used after conducting the crossover and before analyzing the newly generated individuals only. The GA presented here also uses the mutation process simultaneously (in parallel) with the time-domain analysis of the wind turbine. While the analysis interface (in this case, FEDEM Windpower) performs the load case simulation of a certain population in the time domain (which might take around five minutes for a single 90-second load case on a state-of-the-art workstation), the algorithm conducts further mutation operations. The mutated individuals are each analyzed for fatigue damage and extreme loads based on time series results for forces and moments of the already simulated original (nonmutated) individual from previous generations. This reanalysis, therefore, can be performed without an additional simulation in FEDEM Windpower. However, only changes of the element dimensions are taken into account. Forces and moments derived from the original design are considered nonchanging. The dynamical behavior is thereby neglected, and the results are only approximately valid. The fitness of these mutated individuals will be evaluated later with more accurate time-domain simulations as part of the evaluation of the next generation if the individuals satisfy the design requirements. Since the mutation rate of 0.05 leads only to small structural modifications of the cross-section, the damage and extreme load calculations based on the known time series data result in a fairly good approximation of the structural performance of the mutated individuals. This locality principle has been already utilized in a previous study (Zwick et al., 2012).

The individuals used for the reanalysis are not randomly selected. The algorithm selects the fittest individuals out of the already calculated designs. However, an individual generated during this reanalysis process will not be selected for further reanalysis, since the original response time history is not available.

Figure 5 shows the flowchart of the reanalysis process. The mutation process for a single individual during a reanalysis continues until the mutated individual fulfills the extreme loads and fatigue damage constraints. However, this loop is limited to a certain number of executions, which depend on the degree to which the constraints are fulfilled. A mutated individual that is slightly above the design requirements will have the chance for an additional mutation, while individuals that exceed the design requirements enormously will not be used for mutation again. The reanalysis continues by mutating other individuals until the time-domain simulation has ended, and returns the individuals that were analyzed during the reanalysis and fulfilled the design requirements. In our case, typically around 15 individuals were mutated and reanalyzed in parallel with the time-domain simulations. These individuals will be considered as potential children for the new generation. Thus, individuals for a new generation can come not only from reproduction, crossover, and mutation of the current generation, but also from reanalysis, which is mutation of individuals from previous generations. They are selected based on their fitness. Since the population size N is kept constant, a new generation always consists of the N fittest individuals from reanalysis, crossover, and reproduction. Figure 4 shows the flowchart of the GA used in this study.

Precalculation of Fitness

The second modification implemented in this GA is a precalculation of the fitness value for individuals generated by crossover or mutation. As already mentioned, the idea of the GA is to generate offspring from the current individuals such that the average fitness of the offspring is improved. Hence, children will improve the average fitness only if their fitness is higher than the lowest fitness of the current generation. The fitness calculation is based only on the weight and is therefore much faster than performing an entire time-domain simulation for evaluating the performance constraints. Such a precalculation of the fitness is conducted for each individual before it is considered as a potential child, and a time-domain

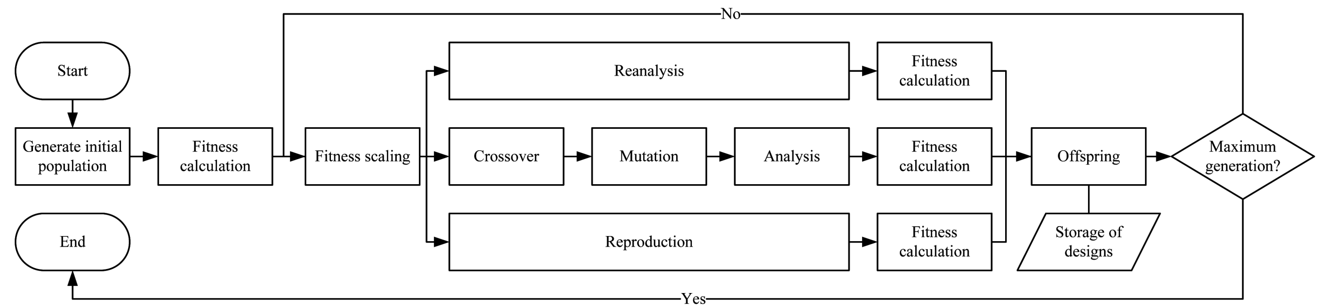

Fig. 4 Flowchart of genetic algorithm

simulation or reanalysis is only performed for children with a precalculated fitness higher than the lowest fitness of the current generation.

Similarity Check

The last modification of the standard GA also makes use of already calculated data and thereby avoids time-consuming simulations in the time domain. The algorithm compares newly generated individuals with already analyzed designs before a time-domain simulation or fatigue damage calculation is performed. For this purpose, each analyzed design is stored with fitness and performance in a database. The algorithm compares the chromosome of a newly generated individual with chromosomes of already existing designs bit by bit and calculates the degree of conformity. The degree of conformity is basically the ratio between the number of bits that are the same and the total length of the string. The algorithm refuses an individual if the degree of conformity exceeds 95% for fatigue damage calculation based on an original time history response and exceeds 98% for fatigue damage calculation performed during a reanalysis, and if the fatigue damage calculation results in a value larger than 5.

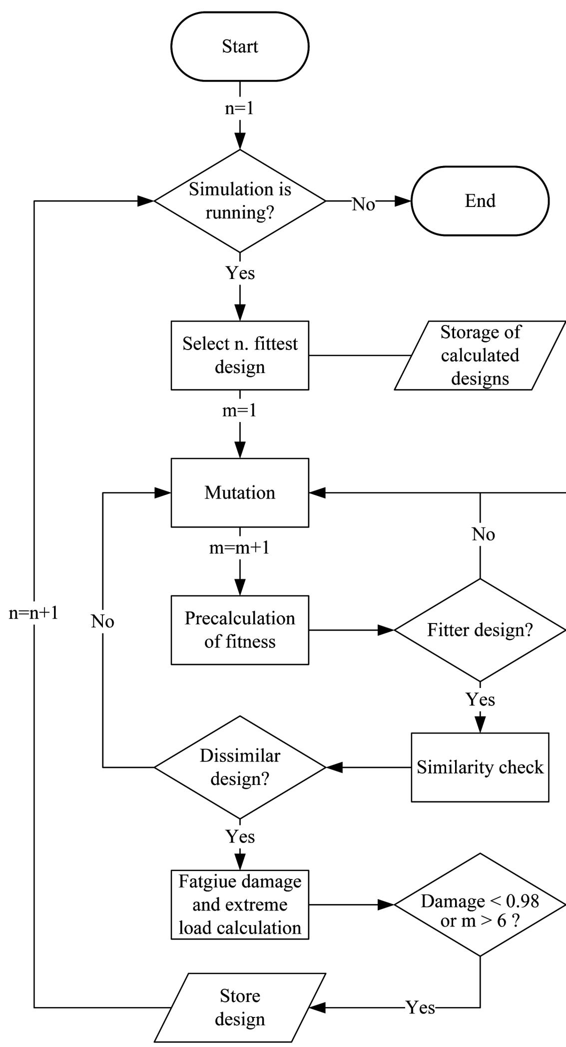

Fig. 5 Flowchart of reanalysis process

RESULTS

One way to monitor the progress of the GA is to display the fittest individual and the average fitness over the population from each generation. Figure 6 shows the evolution of the maximal fitness for several runs. The raw fitness shown in this figure is normalized (for display purposes) to the fitness calculated when using the minimum possible dimensions for legs and braces. The mathematically highest possible fitness can never be reached due to the fatigue constraints. It can be seen that the maximal fitness improves rapidly at the beginning and that the improvements slow down after the 40th generation. At that point, the values begin to converge. The mean fitness increases continuously until the last generation, which can be observed in Fig. 7. Figure 7 shows the origin of the children used in each generation. The average fitness improves only if the new generation includes children from crossover or reanalysis, since reproduction does not increase the fitness of an individual. Crossover contributes around five children to the next generation until about generation 60 and has no influence on the next generation after generation 70. Individuals used as children for the last 30 generations are generated only by mutation or reproduction, with reproduction being the dominant mechanism.

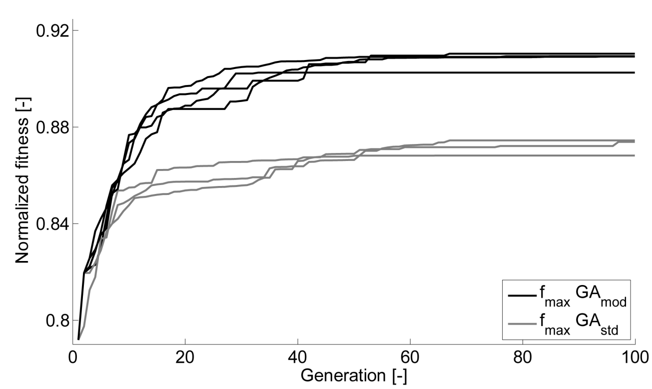

Fig. 6 Normalized maximal fitness (f_{max}) for modified GA (GA_{mod}) and standard GA (GA_{std}) over generations

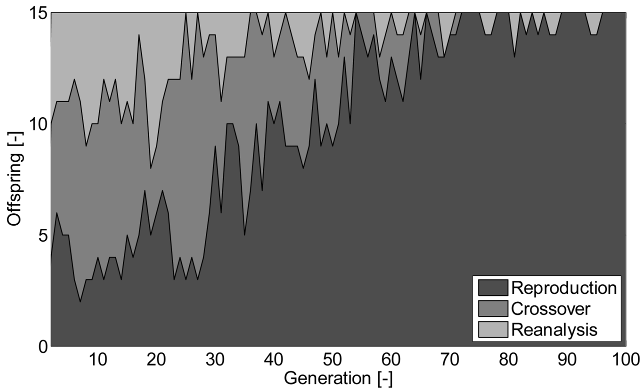

Fig. 7 Origin of offspring over generations

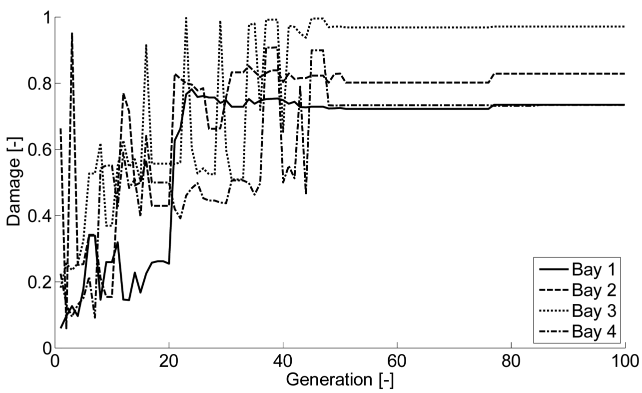

Fig. 8 Fatigue damage of the fittest design plotted over a number of generations separately for each of the four bays

An interesting outcome of the optimization, which was already observed by Pasamontes et al. (2014), is the fact that the structural constraints defined by fatigue damage are not active in each bay. Figure 8 shows the utilization of fatigue damage of the fittest design in each generation. It can be observed that the fatigue constraint is only limiting for the third bay.

A direct comparison of runs performed with and without the use of reanalysis can also be seen in Fig. 6. All runs start with the same initial population. However, the runs performed without the use of reanalysis converge much more slowly and actually reach a much lower fitness during the first 100 generations. The runs have to continue for an additional 200 generations to achieve a fitness similar to the runs with the improved algorithm (not shown).

DISCUSSION

We have shown that the use of reanalysis in parallel with time-domain simulations significantly reduces the number of generations required to improve a jacket design. Several runs of the algorithm were performed with similar results. This is necessary due to the stochastic nature of the algorithm. Even if the same initial population is used, the progress of the optimization varies for different runs (see Fig. 6). This is a drawback of the GA, which provides no guarantee of obtaining the global optimal point. Arora (2012) recommends executing the algorithm several times and allowing it to run longer to overcome this drawback. This enhances the probability of obtaining the global optimum but requires additional computational effort. Reeves and Rowe (2003) examined the premature convergence of GAs in more detail and identified the loss of population diversity as a reason for it. Such a phenomenon is known as genetic drift (Bäck, 1996) and can easily occur when the GA is used with a small population size. A number of techniques have been proposed to limit the effect of genetic drift and maintain population diversity, including preselection (Fogel et al., 1966), crowding (Chambers, 2000; Dawkins, 2011), and fitness sharing (Davidor, 1990). However, none of the runs that were performed for this study were seemingly affected by this phenomenon, and there was no need to implement one of the aforementioned techniques. The extensive use of mutation in our algorithm is credited for this, since mutation changes the designs in the neighborhood of the current design and leads, therefore, to higher population diversity.

The fitness calculation of mutated individuals, based on response time histories of the nonmutated designs, uses the principle of locality (weak coupling). This assumption for the reanalysis is reasonable, since it has been observed in a previous work that member forces do not significantly change under small structural modifications of the cross-section (Zwick et al., 2012). The reanalysis performs a fatigue damage assessment. The algorithm considers only mutated individuals as valid if the estimated damage value of the mutated design does not exceed 98% of the possible fatigue damage, as the calculation is approximate. This safeguards the algorithm against the consideration of invalid designs (which would be eliminated during the next iteration, thereby slowing the algorithm unnecessarily). It is also conceivable to use the damage value of the actual design for such an approximate assessment. A mutated individual would be neglected if the damage value calculated for the nonmutated parent individuals comes too close to the limit.

One modification of the mutation process within the reanalysis could be the option to mutate an entire gene, which describes a single design variable (e.g., the leg diameter in the second bay). Hence, the mutation of an entire gene would change the binary code for the design variable completely and lead to even more diversity, while the current approach changes only single bits inside the entire chromosome.

The precalculation of the fitness led to a continuous improvement of the average fitness over each generation, since only individuals with a precalculated fitness higher than the minimal fitness of the current generation were considered as potential offspring. In addition, studies were performed in which only individuals with fitness higher than the maximal fitness of the current generation were selected for each new generation. The algorithm converged rather more quickly than it would have if only individuals with fitness above the fitness of the worst individual had been taken, but it tended to converge to a suboptimal design due to the loss of diversity, and therefore this strategy cannot be recommended. This modification also shows the different effects of fitness precalculation and reanalysis on the performance of the GA. The fitness precalculation does not generate new individuals and will not lead to a better design. However, it speeds up the algorithm by avoiding unnecessary analyses of newly generated individuals. This modification saves computational resources, while the reanalysis generates potential children for a new generation. A higher population diversity and a larger number of evaluated individuals are the results of this modification. In our case, typically around 15 individuals were reanalyzed in addition to the 15 individuals from the time-domain simulations. Hence, it is reasonable to assume that the reanalysis reduced the number of generations to almost one half.

Since the reanalysis improved the GA significantly, it would be reasonable to extend the process further. Of course, a longer simulation time and more load cases are necessary for a realistic assessment of the design, which would also allow for mutating and reanalyzing significantly more individuals. With respect to the implementation, it is also possible to perform the reanalysis process continuously alongside the actual algorithm as a separate parallel process. Mutation and reanalysis would then be performed even during the selection and evaluation process, and as soon as a fitter individual is found, it would be injected into the population.

An additional improvement regarding the fitness calculation can be achieved by tuning the fitness function. Parameters such as the utilization of damage can be used to calculate the raw fitness, instead of just being used as a constraint. In fact, it was observed that designs with similar damage values per bay were more likely to produce fitter offspring. The distribution of the weight per bay is an interesting factor as well. Designs with heavier bays on the top usually failed the fatigue damage calculation. A fitness function that takes this observation (and further domain-specific knowledge) into account might further reduce the necessary iteration steps. However, this does require structure-specific knowledge or parameters and is therefore not straightforward to use. Furthermore, the precalculation of fitness might not be applicable anymore due to parameters that are available only after the numerical analysis. This would account for more complex fitness functions in which the utilization of damage is used to evaluate the individual, for example. Another possibility connected to the fitness evaluation is to use an enhanced scaling function. As already mentioned, ranked, power law, sigma truncation, exponential, or top scaling functions exist and were used as approaches in the past. A different scaling function might be able to further improve the convergence.

The similarity check of individuals generated by crossover or mutation also contributed to a reduction in iteration steps to obtain an improved design. Time-consuming simulations in the time domain were not performed for individuals who were, to a certain degree, similar to individuals who had been already calculated. Similar to the fatigue damage calculation during the reanalysis, a time-domain simulation for an individual was not performed if the fatigue constraints were significantly exceeded (damage values > 5). This is a conservative approach and gives space for further improvement, but has already led to a reduction in time-domain simulations.

CONCLUSIONS

Optimization of jacket structures was performed with a GA, and the aim was to demonstrate the feasibility of this approach and introduce various improvements in the algorithm. The effect of these improvements was studied with a simple yet complete example. Only one load case was used, and the considered structural model was not completely realistic. Nevertheless, the complexity was basically the same as in a realistic application, and the results showed the suitability of the approach and the importance and influence of the proposed modifications of the basic algorithm.

In general, it is difficult to assess the influence of modifications to the GA (such as including reanalysis) and the impact of tuning the parameters (such as changing the mutation probability), since a GA is a stochastic method. The influence of modifications can be determined only if the algorithm is executed several times and consistent patterns are visible. The GA, therefore, is not considered a mathematically guided algorithm. The optimum design evolves from generation to generation without a stringent mathematical formulation such as those typical of traditional gradient-following optimizing algorithms. However, the advantage of this method lies in finding nonobvious solutions and in the simplicity of the algorithm, which allows for quickly adapting the optimization process, e.g., optimizing the topology of the structure in parallel with the dimensions without significant changes to the algorithm (cf. Pasamontes et al., 2014).

The modifications proposed in this paper show a significant reduction in iteration steps required to obtain improved design solutions. Evaluating the algorithm several times showed that a reduction in the runtime to one third of the number of iteration steps is possible compared to running without the aforementioned modifications. In particular, the use of reanalysis during the time-domain simulation sped up the algorithm and, moreover, safeguarded it from premature convergence. Hence, a major disadvantage could be overcome, but it remains to be seen whether the algorithm is ready for use as a design tool for industrial engineers.

FURTHER WORK

The model used in this study is not representative of a commercial OWT. The method, therefore, has to be adapted for use with commercial designs. Studies with a more detailed jacket model, including further parameters such as the footprint, deck width, or even the number of bays as design variables, should be considered. It is also necessary to include a more realistic soil model to consider more load cases and a larger simulation length for each load case, in order to obtain more realistic results and to validate this optimization method. Moreover, costs should be included in the objective function, since the lightest design is not necessarily the most cost-efficient.

The algorithm can be further improved by including an eigenfrequency check prior to the time-domain simulation to refuse nonsuitable designs before a time-consuming time-domain simulation is carried out. In addition, detailed parameter studies to investigate the influence of algorithm parameters and to monitor the performance of the algorithm would be valuable. The number of cutpoints used in a multipoint crossover, the mutation probability, and the implementation of additional evolution processes such as immigration could be the subjects of future investigations. Furthermore, the algorithm parameters, such as the mutation probability and location, could depend on damage values or degree of diversity instead of being constant or randomly selected. Finally, the implementation of a termination criterion is worth studying further. Possible criteria are, for example, the difference between the maximal and average fitness, the degree of diversity, or the source of offspring.

ACKNOWLEDGEMENTS

Support by the Norwegian Research Centre for Offshore Wind Technology (NOWITECH FME, Research Council of Norway, contract no. 193823) is gratefully acknowledged. In particular, we thank FEDEM Technology AS for its support. The first author would like to thank Kam Sripada (NTNU, Trondheim) for her valuable comments on a previous draft of this paper.

REFERENCES

Arora, JS (2012). *Introduction to Optimum Design*, Academic Press, 896 pp.

Ashuri, T (2012). *Beyond Classical Upscaling: Integrated Aeroservoelastic Design and Optimization of Large Offshore Wind Turbines*, PhD Thesis, Delft University of Technology, Delft, Netherlands.

Azau, S, and Casey, Z (2011). *Wind in Our Sails—The Coming of Europe's Offshore Wind Energy Industry*, Report by the European Wind Energy Association (EWEA), Retrieved from www.ewea.org.

Bäck, TH (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, 328 pp.

Baker, JE (1987). "Reducing Bias and Inefficiency in the Selection Algorithm," *Proc 2nd Int Conf Genet Algorithms*, Hillsdale, NJ, Lawrence Erlbaum Associates, 14–21.

Chambers, L (1995). *Practical Handbook of Genetic Algorithms: New Frontiers, Volume II*, CRC Press, 448 pp.

Chambers, L (2000). *Practical Handbook of Genetic Algorithms: Application*, CRC Press, 544 pp.

Davey, E, Ewing, F, Loughhead, J, and Simpson, M (2012). *EAEM Guide to the UK Offshore Wind Industry*, Lloyd's Register, Retrieved from www.eaem.co.uk.

Davidor, Y (1990). "Epistasis Variance: Suitability of a Representation to Genetic Algorithms," *Complex Syst*, 4(4), 369–383.

Dawkins, R (2011). *The Blind Watchmaker*, Penguin, 368 pp.

Det Norske Veritas (2010). *Fatigue Design of Offshore Steel Structures*, Recommended Practice DNV-RP-C203, Det Norske Veritas, Høvik, Norway.

Fischer, T, de Vries, W, and Schmidt, B (2010). *UpWind Design Basis (WP4: Offshore Foundations and Support Structures)*, Technical Report by the Endowed Chair of Wind Energy (SWE), Institute of Aircraft Design, Universität Stuttgart, Germany.

Fogel, LJ, Owens, AJ, and Walsh, MJ (1966). *Artificial Intelligence Through Simulated Evolution*, John Wiley, 170 pp.

Gen, M, and Cheng, R (2007). *Genetic Algorithms and Engineering Optimization*, John Wiley, 512 pp.

GL Garrad Hassan (2011). *Bladed Theory Manual Version 4.2*, 115 pp.

Goldberg, DE (1985). *Optimal Initial Population Size for Binary-coded Genetic Algorithms*, TCGA Report 85001, University of Alabama, Tuscaloosa, AL.

Goldberg, DE (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 432 pp.

Goldberg, DE, and Deb, K (1991). "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," *Foundations of Genetic Algorithms*, Morgan Kaufmann, 69–93.

Gosavi, A (2003). *Simulation-based Optimization—Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic, 554 pp.

Grefenstette, JJ (1986). "Optimization of Control Parameters for Genetic Algorithms," *IEEE Trans Syst Man Cybern*, SMC-16(1), 122–128.

Holland, JH (1975). *Adaptation in Natural and Artificial Systems: An Introduction Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, 183 pp.

Jonkman, J, Butterfield, S, Musial, W, and Scott, G (2009). *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*, Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO, 75 pp.

King, J, Cordle, A, and McCann, G (2013). "Cost Reduction in Offshore Wind Turbine Jacket Design Using Integrated Analysis Methods and Advanced Control," *Proc 23rd Int Offshore Polar Eng Conf*, Anchorage, AK, USA, ISOPE, 1, 233–241.

Kirsch, U (2010). *Reanalysis of Structures*, Springer, 310 pp.

Lohr, SL (1999). *Sampling: Design and Analysis*, Thomson Learning, 494 pp.

Man, KF, Tang, KS, and Kwong, S (1996). "Genetic Algorithms: Concepts and Applications," *IEEE Trans Ind Electron*, 43(5), 519–534.

Muskulus, M, and Schafhirt, S (2014). "Design Optimization of Wind Turbine Support Structures—A Review," *J Ocean Wind Energy*, ISOPE, 1(1), 12–22.

Pasamontes, LB, Torres, FG, Zwick, D, Schafhirt, S, and Muskulus, M (2014). "Support Structure Optimization for Offshore Wind Turbines with Genetic Algorithm," *Proc ASME 2014 33rd Int Conf Ocean Offshore Arct Eng*, San Francisco, CA, USA, OMAE, 9B, V09BT09A033:1–7.

Popko, W, et al. (2012). "Offshore Code Comparison Collaboration Continuation (OC4) Phase I—Results of Coupled Simulations of an Offshore Wind Turbine with Jacket Support Structure," *Proc 22nd Int Offshore Polar Eng Conf*, Rhodes, Greece, ISOPE, 1, 337–346.

Reeves, CR, and Rowe, JE (2003). *Genetic Algorithms: Principles and Perspectives—A Guide to GA Theory*, Kluwer Academic Publisher, 332 pp.

Sadjadi, FA (2004). "Comparison of Fitness Scaling Functions in Genetic Algorithms with Applications to Optical Processing," *Proc SPIE*, Bellingham, WA, USA, SPIE, 5557, 356–364.

Schaffer, JD, Caruana, RA, Eshelman, LJ, and Das, R (1989). "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization," *Proc 3rd Int Conf Genet Algorithms*, San Francisco, CA, USA, Morgan Kaufman, 51–60.

Surry, PD, and Radcliffe, NJ (1996). "Inoculation to Initialize Evolutionary Search," *Evol Comput: AISB Workshop*, Brighton, UK, Springer-Verlag, 269–285.

Vermula, NK (2010). *Design Solution for the UpWind Reference Offshore Support Structure—Deliverable D4.2.5 (WP4: Offshore Foundations and Support Structures)*, Technical Report, Rambøll Wind Energy, Esbjerg, Denmark, 90 pp.

Vorpahl, F, Popko, W, and Kaufer, D (2011). *Description of a Basic Model of the "UpWind Reference Jacket" for Code Comparison in the OC4 Project Under IEA Wind Annex XXX*, Technical Report, Fraunhofer Institute for Wind Energy and Energy System Technology IWES, Bremerhaven, Germany.

Wood, D (2011). *Small Wind Turbines—Analysis, Design, and Application*, Springer, 272 pp.

Yoshida, S (2006). "Wind Turbine Tower Optimization Method Using a Genetic Algorithm," *Wind Eng*, 30(6), 453–470.

Zwick, D, Muskulus, M, and Moe, G (2012). "Iterative Optimization Approach for the Design of Full-height Lattice Towers for Offshore Wind Turbines," *Energy Procedia*, 24, 297–304.