Industrial Application of Genetic Algorithms to cost reduction of a Wind Turbine equipped with a Tuned Mass Damper

Jordi Pons-Prats*
International Center for Numerical Methods in Engineering (CIMNE)
Universitat Politècnica de Catalunya, Campus PMT UPC, Edifici C3, 08860 Castelldefels, Spain
Email: jpons@cimne.upc.edu

Martí Coma
International Center for Numerical Methods in Engineering (CIMNE)
Universitat Politècnica de Catalunya,Campus PMT UPC, Edifici C3, 08860 Castelldefels, Spain
Email: mcoma@cimne.upc.edu

Jaume Betran
Consultant on Multibody and Multiphysics Computation
Sant Cugat del Vallès, Spain
Email: jaume.betran.prof@gmail.com

Xavier Roca
International Center for Numerical Methods in Engineering (CIMNE)
Universitat Politècnica de Catalunya,Campus PMT UPC, Edifici C3, 08860 Castelldefels, Spain
Email: xroca@cimne.upc.edu

Gabriel Bugeda
International Center for Numerical Methods in Engineering (CIMNE)
Universitat Politècnica de Catalunya BarcelonaTech (UPC)
Universitat Politècnica de Catalunya, Campus Nord UPC, Edificio C1, Gran Capità s/n, 08034 Barcelona, Spain
Email:bugeda@cimne.upc.edu

Summary

Design optimization has already become an important tool in industry. The benefits are clear, but several drawbacks are still present, being the main one the computational cost. The numerical simulation involved in the solution of each evaluation is usually costly, but time and computational resources are limited. Computational resources can be easily increased because, nowadays, its cost is rapidly decreasing. Anyway, there is always an upper limit due to financial constraints. On the other hand, time is key in industry. Lead time must be reduced to ensure competitiveness. It means the design stage has a reduced time slot. In addition, industrial problems are multi-objective and multi-disciplinary, which increases the cost and the complexity. The present communication is focused on describing a practical application developed together with a wind energy company, who aimed to optimize the design of a Tuned Mass Damper. It is a structural device installed within the tower of a wind turbine aimed to stabilize the oscillations and reduce the tensions and the fatigue loads. The paper describes the decision process to define the optimization problem, as well as the issues and solutions applied to deal with a huge computational cost, with a multi-objective and multi-disciplinary environment including some gaps in the definition of specific points. Focusing on the optimization methodology, the communication will describe the application of RMOP, CIMNE's in-house optimization platform in comparison of the company's in-house optimization platform.

Keywords: *RMOP, Genetic Algorithms, optimization platform, Wind Energy, Wind Turbine design, tuned mass damper.*

1 Introduction

During the last few decades the Wind Energy industry has grown fast and engineers have designed wind turbines of increasing size, while seeking lower values of cost of energy (CoE). The sector started its industrialization in the late seventies and eighties and soon scaled the initial 50kW, ⊘15m rotors to 600kW, ⊘50m in the nineties, 3MW, ⊘100m in the 2000s and currently reaching 8MW, ⊘150m.[1,2] Alongside the said rotor upscale, the most convenient onshore sites were taken up and modern Wind Farm developers and contractors started exploring more remote sites. Finally, during the last decade, offshore wind resources reached competitive figures of CoE in the North Sea and some smaller sites around the world. The higher and more steady offshore winds at shallow waters allowed for taking advantage of bigger rotors of an already mature technology. Offshore wind turbines present, nevertheless, particularly complex challenges in the domain of the structure dynamics additionally to the already severe wind loading. These heavy, slender structures, built on uneven seabed, have low natural frequencies that fall well within the excitation range of wave loads. In order to damp oscillations out and therefore reduce stress in the structure, some sort of absorbers are sometimes used, allowing for significant overall cost reductions. An especially interesting kind of absorber is the so called Tuned Mass Damper (TMD), composed of a massive oscillator tuned at the target frequency and a damper system to remove the energy from the resonator. The effectiveness of TMDs highly depends on its location and mass but it may have limitations due to integration issues. The present paper describes the optimization strategy and outcomes of a pre-design study of an industrial 6MW class offshore wind turbine structure equipped with a TMD tuned at the first bending moment of the tower with a view to reduce overall structure weight and reach more competitive CoE figures. The focus is put on the strategies followed to overcome the extremely high computational time. It is directly related to the huge number of simulations accounted for the fatigue analysis for a big number of individuals dealt in a multi-objective genetic algorithms optimization scheme. The comparison of several available tools is presented. The first of them was the company's in-house optimization suite which includes gradient-based methods, plus a generic evolutionary algorithms implementation, based on NSGA-II,[3,4] and SPEA2.[5–7] The second tool was RMOP, CIMNE's in-house optimization platform, which implements genetic algorithm with Nash and Hybrid games.[8–17] The implementation of the GA algorithm in RMOP is quite standard. It was initially inspired on NSGAII, implementing additional functionality not only from the point of view of the evolutionary techniques, but also from the point of view of usability and user interface and the set-up of the internal parameters. To mention some of the implementations, standard techniques like SBX (Simulated Binary Cross-over) or tournament selection, were jointly added with I/O techniques and libraries to manage the definition of each individual evaluation. The key point in RMOP is the implementation of Game strategies, more specifically Nash Games, to enhance the convergence and accuracy of the solution. This implementation leads to an hybrid definition; the Pareto optimality criteria is enriched with the information from the Nash players, so the optimization analysis benefits from the two of them Pareto and Nash.

2 Selection of optimization strategy

There are different optimization algorithms available in the company's in-house suite. In order to choose the most effective one, a comparison is conducted. This task is motivated due to the preliminary results obtained during the initial tests and discussion with company's engineers, which suggested that there are significant differences between different Multi-Objective Genetic Algorithms (MOGA) implemented within the suite, as well as the initial reservations against RMOP. The comparison is performed using mathematical test cases commonly used for this purposes. The advantages of using these cases, among others, are that are easily implemented, fast to evaluate and designed for this purpose. A preliminary TMD test case is also presented. It corresponds to a very simplified representation of the TMD, but the main aim when analyzing this particular test case is to anticipate potential issues both from the implementation viewpoint and from the results viewpoint. The studied MOGAs are the 3 available in the suite, plus RMOP:

- Evolution: it is a generic implementation of an evolutionary algorithm. It is quite a simple implementation with a limited control over the setup parameters of the algorithm.
- NSGA2: it is a well known algorithms developed by Prof. Deb.[18] Its applicability and high performance have been documented widely.
- SPEA2: it is a well known evolutionary algorithm developed by Zitzler.[5]
- RMOP: Genetic algorithms with game theory. It is an in-house CIMNE development which combines some basic evolutionary algorithms strategies with Nash games for improved convergence and accuracy. It is possible thanks to the combination of Pareto optimality criteria with Nash Games as previously described.

The selected test cases are:

- KUR; a mathematical test case with 3 design variables and 2 objective functions. Its complexity comes from the definition of the functions.
- TNK mathematical test case with 2 design variables, 2 objective functions and 2 constraints, it is a first step into a restricted search space.

- CPT3 mathematical test case with 2 design variables, 2 objective functions and 1 constraint, its complexity is a combination between the objective functions and the restricted search space.

- OSY mathematical test case with 6 design variables, 2 objective functions and 6 constraints, which defines a very restricted search space.

- ZDT2; a mathematical test case with 30 design variables and 2 objective functions. Part of its complexity comes from the number of design variables.

- LZ09-F1 mathematical test case with 1 objective function and a variable number of design variables. This characteristics makes it interesting for constantly increase the problem complexity.

- TMD test case, a multi-objective and multi-disciplinary structural problem based on the real-world case of designing a wind turbine.

For more details about the mathematical test cases, please refer to[4, 18–21]

2.1 *Comparison of the Mathematical test cases results*

For a fair comparison, a common set-up were defined for all the algorithms and all the mathematical test cases. This common set-up defines a population size equal to 4 times the number of design variables, and a number of maximum evaluations equal to hundred times the population size. The crossover probability was defined equal to 0.9 and the mutation probability equal to 0.1. Figure 1 to 6 show a comparison between the results obtained by each algorithm. Figures 1, 3 and 5 show the convergence history of the objective functions for each test case. In all the three cases, RMOP and NSGA2 are the ones converging the faster and lower. On the other hand, figures 2, 4 and 6 show the Pareto fronts for each of the three cases. It is clear that the number of evaluations is not enough to fully capture the front shapes with enough accuracy, but, due to the fact that the aim of the analysis was to detect which algorithms performs the better with a restricted number of evaluations, then the objective was fully fulfilled.

In an overall performance analysis of the results, RMOP presents a better average performance. Evolution algorithm shows poor performance in all tests done, both in the convergence of the fitness functions and in the capture of the Pareto Front. NSGA2 shows results compatible with RMOP results in most of the problems and in most of the Pareto front regions. However, RMOP better captures the Pareto Front in all the cases. SPEA2 shows results compatible with RMOP results in some of the problems and in most of the Pareto front regions. The general performance is lower than RMOP. In some test cases (TNK and OSY mainly), Pareto front regions are not well populated when using NSGA2 and SPEA2. It is clear that this phenomena is a direct consequence of the imposed limitation on the number of evaluations and the size of the population. It was an expected drawback which was accepted for the sake of saving time. In case the company's proprietary software would be a requirement, NSGA2 method should be the most appropriate selection. In case company's software can be coupled with the RMOP optimization algorithm, then this configuration is the best choice.

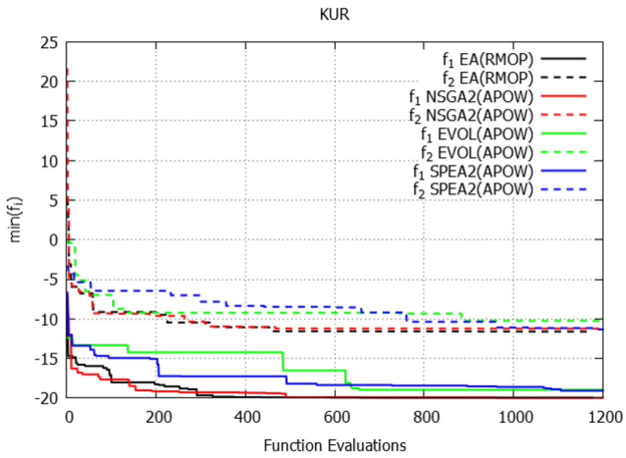

Figure 1: KUR test case convergence

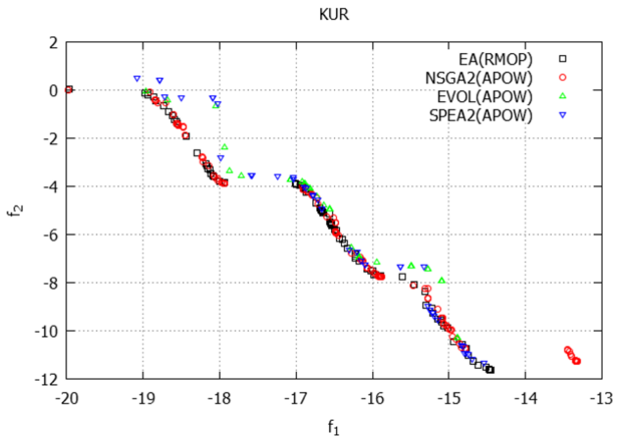

Figure 2: KUR Pareto Front

3 Industrial application: TMD Optimization

The industrial application is based on a real case, a set of wind turbine, nacelle, tower and mono-pile which the company is designing and manufacturing for an Atlantic offshore site. The addition of a TMD is under study. The wind turbine design including rotor, nacelle and tower is fixed, so the analysis will not modify any of their parameters. The main aims of the work is to optimize the mono-pile and the TMD. Two are the principal objective functions; the first of them is the structural performance of

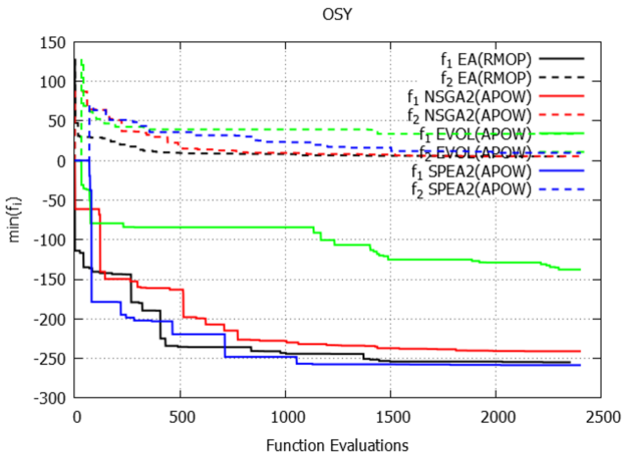

Figure 3: OSY test case convergence

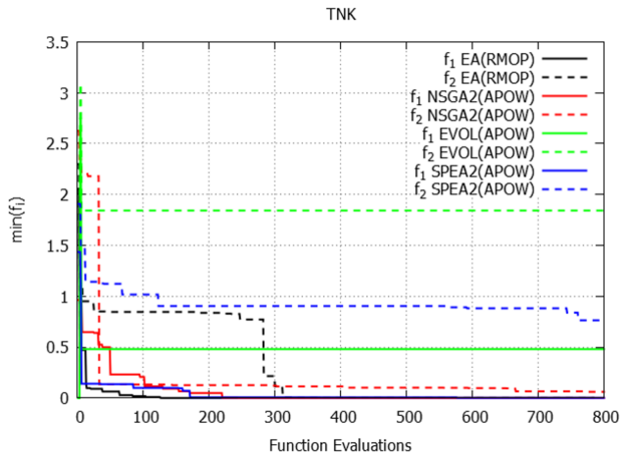

Figure 5: TNK test case convergence

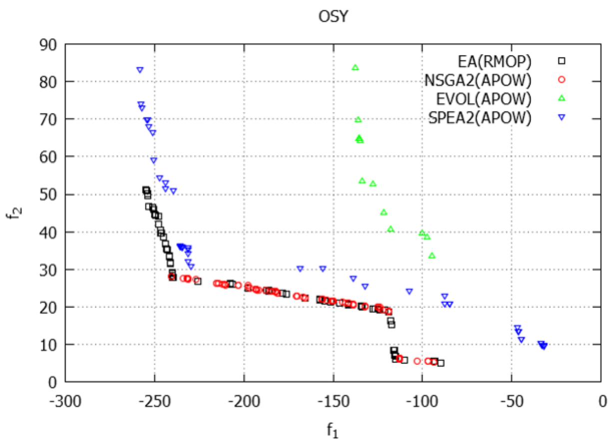

Figure 4: OSY test case Pareto Front

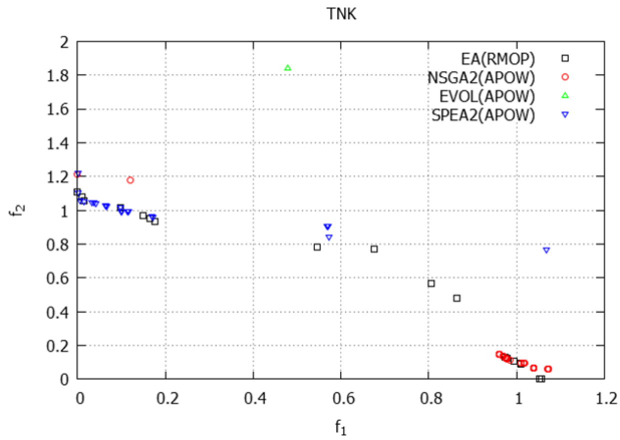

Figure 6: TNK test case Pareto Front

the structure when using or not the TMD, and according to the mass and dumping parameters, while the second one is the overall cost including the mono-pile and the TMD. The performance is split into three objective functions representing the behavior under ultimate and fatigue loads. To define the performance function, the objective functions that reflect performance in Ultimate and Fatigue loads must be defined:

$$\begin{aligned} FO_{FLS-fa} &= w_1^F \cdot M_{x,1}^F + w_2^F \cdot M_{x,2}^F + w_3^F \cdot M_{x,3}^F + \\ &\quad + w_4^F \cdot M_{x,4}^F \\ FO_{FLS-ss} &= w_1^F \cdot M_{y,1}^F + w_2^F \cdot M_{y,2}^F + w_3^F \cdot M_{y,3}^F + \\ &\quad + w_4^F \cdot M_{y,4}^F \\ FO_{ULS} &= w_1^U \cdot M_{xy,1}^U + w_2^U \cdot M_{xy,2}^U + w_3^U \cdot M_{xy,3}^U + \\ &\quad + w_4^U \cdot M_{xy,4}^U \end{aligned} \tag{1}$$

Being

- FO_{FLS-fa}:fore-aft FLS performance objective function,
- FO_{FLS-ss}: side-to-side FLS performance objective function,
- FO_{ULS}: fore-aft FLS performance objective function,
- $M_{x,i}$: moment in fore-aft direction at point i,
- $M_{y,i}$: moment in side-to-side direction at point i,
- $M_{xy,i}$: modulus of resultant moment

Where i represents the points: 1 for the Tower bottom, 2 for the Tower lower intermediate, 3 for the Tower upper intermediate, and 4 for the Tower top. Variables w_i^U and w_i^F are weights to take into account the more relevance of the moments when closer to the bottom of the tower:

$$\begin{aligned} w_i^U &= \frac{1}{1+a \cdot i} \\ w_i^F &= \frac{1}{1+a' \cdot i} \end{aligned} \tag{2}$$

Parameters a and a' are chosen according to structural criteria. For this case, in order to obtain a linear cost function and under agreement with the engineers of the company, they are chosen constants and equal to 0.1. Finally, objective functions in Ultimate and Fatigue results in:

$$
\begin{aligned}
FO_{FLS-fa} &= \frac{M^F_{x,1}}{1.1} + \frac{M^F_{x,2}}{1.2} + \frac{M^F_{x,3}}{1.3} + \frac{M^F_{x,4}}{1.4} \\
FO_{FLS-ss} &= \frac{M^F_{y,1}}{1.1} + \frac{M^F_{y,2}}{1.2} + \frac{M^F_{y,3}}{1.3} + \frac{M^F_{y,4}}{1.4} \\
FO_{ULS} &= \frac{M^U_{xy,1}}{1.1} + \frac{M^U_{xy,2}}{1.2} + \frac{M^U_{xy,3}}{1.3} + \frac{M^U_{xy,4}}{1.4}
\end{aligned}
\tag{3}
$$

The functions will be minimized in order to maximize the performance of the system.

The cost is the main objective function, because the benefits of the company is directly related to the manufacturing and installation cost (CAPEX cost) of the TMD. It can be calculated according a complete cost function, or just considering the cost of the TMD (its mass as the main contributor to the cost).

$$
\begin{aligned}
FO_{COST} = \; & 74600 + 1.415 \cdot TMDmass + \\
& +11700 + 29000 if TMDmaxEx < 0.5 \\
& 74600 + 1.415 \cdot TMDmass + \\
& +44000 \cdot TMDmaxEx + 11700 + \\
& +29000 if TMDmaxEx > 0.5
\end{aligned}
\tag{4}
$$

Due to the cost of computing a single individual, several strategies have been implemented to reduce the overall computational cost. These strategies include on one hand stopping the calculation if the individual is not fulfilling the restrictions, and on the other hand a careful selection of the load cases to be calculated, just to mention two of them. Although applying these simplifications, the evaluation work flow is quite complex involving several solvers and checkpoints.

Three different models based on three different loads' computation approaches have been studied. Based on accuracy reasons and on the possibility of customization of the tool, a FE based flexible Multibody model of a wind turbine, substructure and tuned mass damper built in SAMCEF was considered in the first place. This option was early discarded for the full optimization procedure due to the high CPU times, which were unfordable when considering the industrial cost limitation. This limitation is related, amongst others, to the limited number of software licenses available, which is not a technical issue, but an industrial issue. From the technical point of view, the license issue was limiting the parallelization of the individual evaluations. Its use was reduced to periodic verification purposes only. GH Bladed was chosen as an alternative. The CPU time per thread is similar to that of SAMCEF but the available licenses at the company allowed for multiple simulations running in parallel in different threads, which significantly speed up the overall optimization procedure. While SAMCEF is a FE solver that features mechanism modeling,[22] Bladed is a Wind Turbine dedicated multi-body simulation (MBS) software that models mechanisms with kinematic laws and features elasticity by modal condensation of its main structures.[23] This approach has an impact on accuracy of the results, mostly due to poor modeling of TMD nonlinear region, that is overcome with periodic verification with a higher standard approach. A third option is finally considered which stretches the latter approach. An ad-hoc solver is developed to compute loads of a simplified model of WT, substructure and nonlinear TMD. While the motion of the TMD remains in the linear region a fast and exact recursive closed form solution is used[24] and it swaps to Newton family solvers, HHT, when nonlinearities must be accounted for. This approach totally solves the problem of threads used in parallel and the CPU cost per load case is significantly reduced. The use of the SAMCEF model for verification guarantees accuracy of overall procedure. Finally, the selected procedure was a mix between the use of Bladed and the ad-hoc solver, which was implemented within MATLAB. Bladed was used to perform a Campbel analysis of the individual, which first determine its feasibility, and early discard those leading to a poor design.

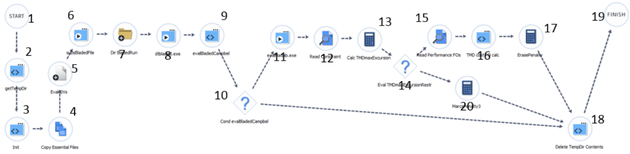

Figure 7: Evaluation workflow

Figure 7 describes the individual evaluation workflow. Step by step can be described as:

1. START: Start node for the individual evaluation.
2. getTempDir: Scripting Process (Java). Sets the local variable TempDir with the path of the current evaluation folder. It changes for each individual evaluation of the optimization.
3. Init: Scripting Process (Python). Sets initial values for several variables. The relation of initiated variables.
4. Copy Essential Files: File Copy Process. Copy the files needed for the evaluation to the temporary folder, which must be located in the optimization directory: (a) Excel file (DLCs.xlsx) containing the dynamic load cases to evaluated. (b) Bladed Campbel model file (DTBLADED.MODEL). (c) Binary Matlab file containing information about the files and folders name (Names.mat).
5. Eval.DVs: File Creation Process. Creates a file containing the value of each design variable. The file format is ASCII with one line containing:TMDmass, TMDfreq, TMDdamp, TMDnode
6. subsBladedFile: Execution Process. Runs executable file subsBladedInFile.exe. It combines the files

DTBLADED.MODEL with Eval.DVs to generate DTBLADED.IN.

7. Dir BladedRun: Directory Creation Process. Creates the directory BladedRun inside the TempDir directory so Bladed can run in it.

8. Dtbladed.exe: Execution Process. Runs executable file dtbladed.exe to perform the Campbel analysis. The call does not take parameters, it is: BladedFolder dtbladed.exe

9. evalBladedCampbel: Scripting Process (Python). Reads Campbel results from file BladedRun/modalResultsFileName and evaluates the 3P dynamic criteria. It also sets the Penalty output variable to 1 if the file does not exist or to 2 if the dynamic criteria is not met.

10. Cond evalBladedCampbel: Condition. Decision point that checks if the evaluation of the Campbel analysis is satisfactory (ValEvalBladedCampbel == 0) and the runs the node evalMatlab.exe or it skips further evaluations and goes directly to the node Delete TempDir.

11. evalMatlab.exe: Execution Process. Runs evalMatlab.exe. It calculates the dynamic load cases transformation for the current TMD, the performance objective functions, among others. Call is: ToolsFolder/evalMatlab.exe TMDmaxExcursionRestr ApplyExcursionBreak

12. Read Constraint: Parameter Reader. Reads Eval.constraint file previously written by evalMatlab.exe which contains the value of TMDExcursionRestr.

13. Calc TMDmaxExcursion: Calculator Process. Calculates the value of the maximum escursion: TMDmaxExcursion= TMDmaxExcursionRestr – TMDExcursionRestr

14. Eval TMDmaxExcursionRestr. Condition. Decision point for maximum excursion criteria.

15. Read Performance FOs: Parameter Reader. Reads Eval.2to4_individual file which contains the values of the objective functions from 2 to 4. The file format is ASCII with one line containing: FO_(FLS-fa), FO_(FLS-ss), FO_ULS

16. TMD cost FO calc: Scripting Process (Python). Calculates TMD cost objective function. See [1] for details on the function.

17. ErasePenalty: Calculator Process. Sets local variable Penalty value to 0, indicating that the evaluation is correct and have not been applied any penalty during the process.

18. Delete TempDir: Scripting Process (Python). If the evaluation has been launched from an optimization process deletes TempDir folder, if it has been launched for a single evaluation it does not erase the TempDir.

19. FINISH: Finish node for the individual evaluation. MarcaPenalty3: Calculator Process. Sets local variable Penalty value to 3, indicating that the evaluation has been stopped during the maximum excursion check.

20. MarcaPenalty3: Calculator Process. Sets local variable Penalty value to 3, indicating that the evaluation has been stopped during the maximum excursion check.

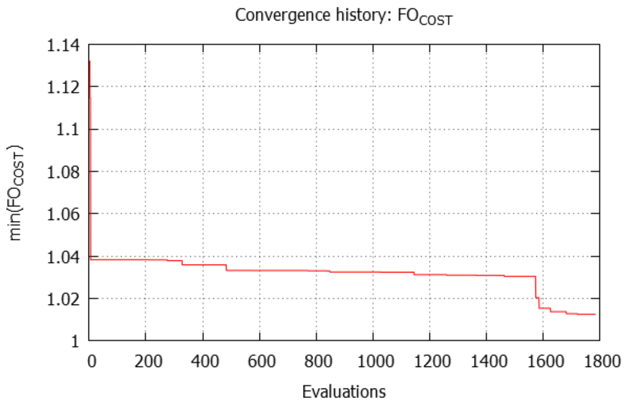

Figure 8: Cost objective function convergence

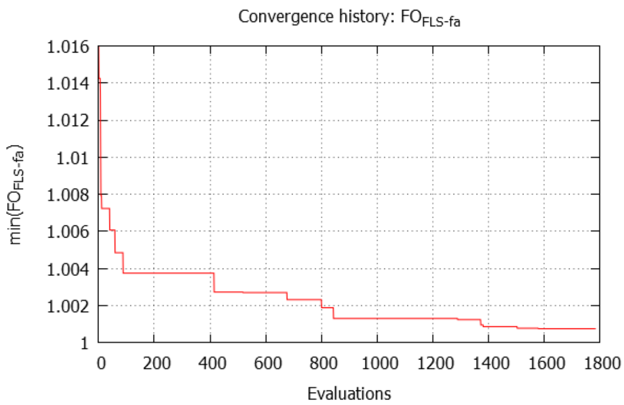

Figure 9: Fatigue loads (front-aft) objective function convergence

An initial definition of the optimization problem had 8 design variables, 4 to define the basic substructure geometry and 4 to define the TMD characteristics. The variation of substructure geometry has a direct impact on the driving substructure cost and on the dynamic behavior of the whole system. The variation of the TMD characteristics contribute

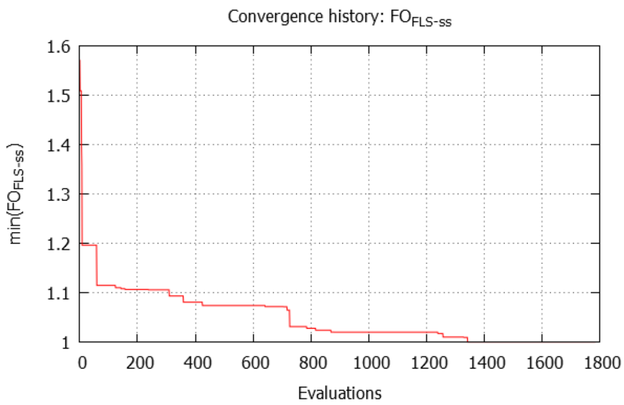

Figure 10: Fatigue loads (side to side) convergence

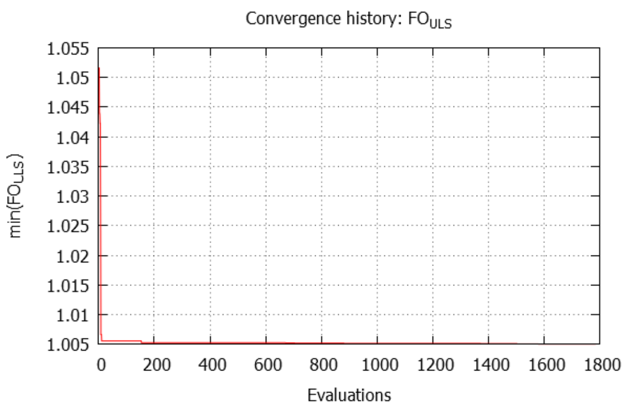

Figure 11: Ultimate loads objective function convergence

for each individual to reduce the dynamic response and has a secondary contribution to the cost. Finally, a selection of 4 design variables was made to simplify the problem, all of them related to TMD. This selection includes the mass, the frequency, the damping coefficient and the station where to install it. Objective functions has been described in 3 and 4. Both the search space and the solution space are multi-dimensional, which means the Pareto front is no longer a 2D line, nor a 3D surface. The analysis of the solution is done on the projection of this multi-dimensional space into 2D plots. Figures 12, 13 and 14 show an example of how the solutions are plotted, taking couples of the 4 objective functions. Additionally, figures 15 and 16 show the Pareto Front plotting 3 of the objective functions, one of them as a color scale. Figures 8, 9, 10, and 11 are the convergence history of each objective function. No major issues can be extracted from these plots, more than compare how fast each function is converging. Figure 8 shows the cost objective function convergence, which until the end of the analysis does not present a significant improvement. Figure 9 shows a gradual improvement of the Fatigue loads (front aft). Figure 10 shows also a constant improvement of the function, anyway, the last 200 evaluations do not provide further improvement. It shows two regions, the first half with a gradual improvement, and the second half where the improvement is less significant. Figure 11, regarding the ultimate loads, does not show improvement much improvement during the optimization. Although the scale of the y axis has been normalized, to fulfill with the NDA signed with the company, all the functions show a good improvement along the optimization analysis, compared to initial values. Later, when analyzing the Pareto Front plots, a comparison with the baseline design will be provided.

The plots for the Pareto Front show the results using Bladed plus MATLAB implementation. In all of them an improvement respect to the baseline design is obtained. There are a lot of Pareto individuals improving the values for the baseline, demonstrating the performance of the optimizer RMOP as expected. The computational cost associated to this analysis is reduced applying a preliminary selection of the load cases for each individual, without penalizing the accuracy and feasibility of the design. Figure 12 shows the cost versus the Fatigue loads. As shown in the graph both functions are opposed; to improve one function the other must get worse. This is not always the case, as happens with the two functions related to fatigue loads. Individuals that belong to the Pareto Front are marked in green. As a remark, the plots presented are a projection of the four dimensional objective functions space; this yields to a Pareto Front representation that has individuals that may appear as dominated, but are not. Those individuals appear as non-dominated in other projections. TMD cost in front of PF Fatigue side-side is shown in Figure 13. These two functions are opposed too. TMD cost in front of PF Ultimate is shown in Figure 14. The two functions are opposed, but the more expensive TMDs is, the PF Ultimate almost constant is. the Pareto Front of the two Fatigue loads (front aft and side to side) is not shown because the two functions present a strong correlation, so any improvement on one of the two also means improving the other one. Two additional Pareto Front plots, including 3 objective functions, are presented. Figure 15 and 16 do not shown the dominated points to simplify and make them more understandable. In the second one, the region where the ultimate loads keep constant while the cost or the fatigue loads increase is also there, as seen previously in figure 14. In the Pareto front plots, some individuals have been marked as "selected". Those individuals have been used by the company to evaluate the overall performance of the optimization and to compare the the four objective functions values with the baseline design. All the selected points are located near the influence area of the baseline, although they improve the baseline values. It is true that in some cases, it is not possible to simultaneously improve all the functions, but improvements of about 10 to 20% are possible. Table 3, below, describes the error of the selected individuals compared to the baseline. The individuals improving all the objective functions show lower improvement, while those with larger improvements in some functions show more variability on the error along the four objective functions.

Relative error compared to baseline			
Cost	FLS-fa	FLS-ss	ULS
-24.00	-0.55	19.03	1.18
-0.13	-0.29	-4.58	-2.31
-2.52	-0.14	-0.004	-1.25
19.54	-0.80	-23.49	-2.31
-7.56	0.09	7.15	0.04
-0.90	-0.23	-2.35	-1.84
-6.04	-0.04	3.83	-0.42

It has been mentioned that several strategies have been evaluated in order to reduce the computational cost associated to the analysis. The most relevant ones have been:

- Parallelize: it is the first one thing about, but one should bear in mind that the solver can be also parallelized. The use of smart strategy, defining how many cores is using each individual evaluation and how many are available, is of great importance. This is the first and more important startegy because it can be applied whatever analysis and solver you are going to use.

- Individual evaluation: in some cases it is not possible to interact with the individual evaluation, so its cost cannot be reduced. But it was not the case of the actual analysis. The standard procedure of the company, when validating a design includes a long list of load cases to be evaluated. Initial, the company was requesting to apply the same list to each individual on the evaluation, leading to an unfordable cost. a careful analysis determine that the relevant load cases can be restricted to only a 5%, then the computational cost has been extremely reduced.

- Constraints: one can use the constraints as restrictions, so defining a go-no go criteria. If any individual does not fulfill a constraint, the individual is penalized and the evaluation is stopped. This strategy is easy to implement if your evaluation workflow is split into several steps, otherwise it can be difficult or no sense to apply because the cost reduction is not relevant.

An important point must be highlighted in regards the restrictions applied to the individuals. There have been 137 individuals that do not meet the maximum excursion for the TMD displacement (set at 1.5 meters), with a maximum excursion of 2.1 meters. Another remark is that any individual has been penalized for the dynamic check, evaluated with the Campbel analysis and the 3P criteria. This should be analyzed further to evaluate the reason that any individual has not met the constraint; maybe is not set correctly or maybe there is some error in the implementation. If it is confirmed that the restriction was satisfactorily setup, then the use of Bladed, to perform the Campbel analysis should be removed, simplifying the workflow and reducing the computational cost of each individual evaluation.

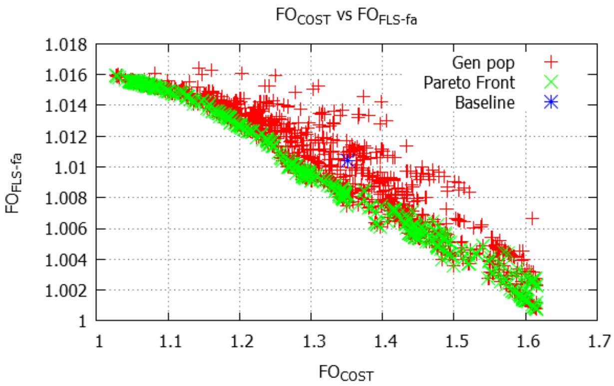

Figure 12: Cost vs Fatigue loads (front-aft)

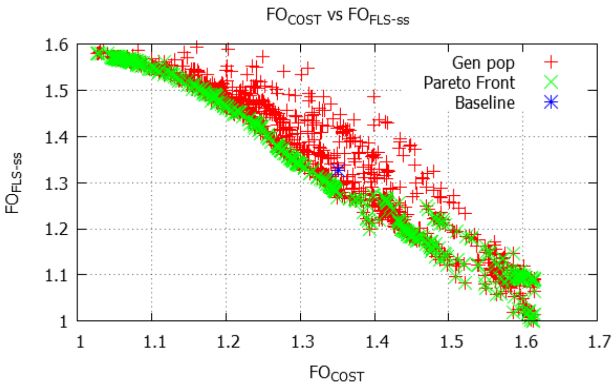

Figure 13: Cost vs Fatigue loads (side to side)

4 Conclusions and further work

Industrial applications differ from academic problems on many ways. Although academic and mathematical test case can reach a high complexity level, the multi-disciplinary involved in industrial problems, added to the complexity of the design process by itself conform the key issues. This communication is aimed to describe how the authors deal with this complexity, and how closely working with the engineers in the company face and partially solve this problem. The paper is also aimed to demonstrate the capabilities of RMOP in comparison to industrial suites. The results from both the mathematical test cases and the industrial application show how RMOP performs better, even without using additional functionalities like Nash Games. From the point of view of the industrial application, the results lead to a relevant reduction of the cost, as

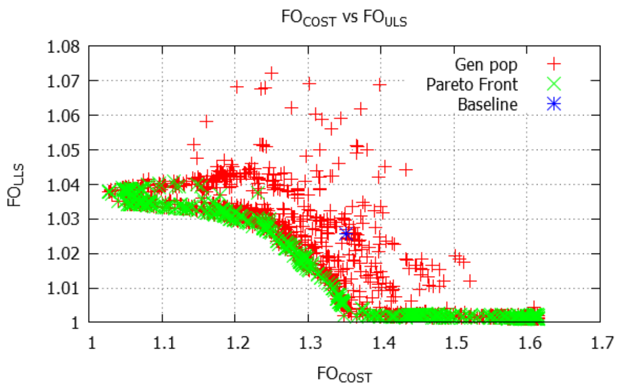

Figure 14: Cost vs Ultimate loads

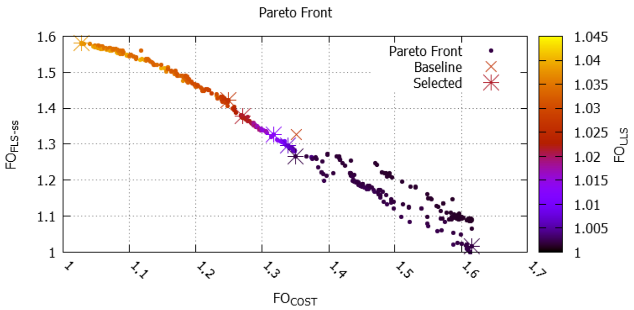

Figure 15: Cost vs Fatigue loads (side to side) vs ULS

well as the loads. The Pareto hyper-surface is defining a large number of Pareto individuals which are improving the 4 objective functions in comparison with the baseline design. Significant improvements have been obtained when improving. Although those individuals with spectacular improvements on one function do not show consistent improvements for all the four objective functions, there are many individuals improving the four functions within the range of 1 to 5%, which is more than interesting. From the point of view of the company, the cost of the TMD was the most important objective function, so it was used to identify and select those individuals and more promising configurations. Further work is two-fold. On one hand an on-going implementation of a most simplified solver in MATLAB, which can lead to a simplified solver, with a reduced computational cost but with an appropriate accuracy level. On the other hand, CIMNE is working on the continuous improvement of RMOP. It focus on the implementation of hierarchical evaluation strategies within the platform. Each of the two will mean a significant improvement on the calculation time and on the efficient use of the available solver to get a fast scan of the search space and an accurate optimal solution.

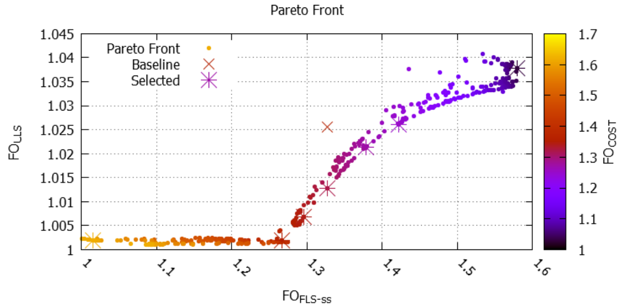

Figure 16: Fatigue loads (side to side) vs Ultimate loads vs Cost

5 Acknowledgements

The authors would like to thank the company to provide the opportunity to publish the results of the join work, although due to the non-disclosure agreement between to two parts, the name of the company and the details and values have been masked. The authors also acknowledge the engineers of the company in structural and aerodynamic loads departments for their support during the implementation of the analysis.

References

[1] Association, E. W. E. et al. *Wind energy-the facts: a guide to the technology, economics and future of wind power*. Routledge, (2012).

[2] GWEC, G. W. E. C. Global wind statistics 2015, (2016).

[3] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts, January (1989).

[4] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* **6**(2), 182–197 Apr (2002).

[5] Zitzler, E., Thiele, L., et al. An evolutionary algorithm for multiobjective optimization: The strength pareto approach, (1998).

[6] Zitzler, E. and Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation* **3**(4), 257–271 (1999).

[7] Zitzler, E., Laumanns, M., Thiele, L., et al. Spea2: Improving the strength pareto evolutionary algorithm, (2001).

[8] Lee, D., Gonzalez, L. F., Periaux, J., Srinivas, K., and Onate, E. Hybrid-game strategies for multi-objective

design optimization in engineering. *Computers and Fluids* **47**(1), 189 – 204 (2011).

[9] Lee, D. S., Gonzalez, L. F., Periaux, J., and Bugeda, G. Double-shock control bump design optimization using hybridized evolutionary algorithms. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **225**(10), 1175–1192 (2011).

[10] Lee, D., Periaux, J., Pons-Prats, J., Bugeda, G., and Oñate, E. Double shock control bump design optimization using hybridised evolutionary algorithms. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 1–8. IEEE, (2010).

[11] Lee, D.-S., Periaux, J., Onate, E., Gonzalez, L. F., and Qin, N. Active transonic aerofoil design optimization using robust multiobjective evolutionary algorithms. *Journal of Aircraft* **48**(3), 1084–1094 (2011).

[12] Lee, D., Gonzalez, L. F., Periaux, J., and Srinivas, K. Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. *IEEE Transactions on Evolutionary Computation* **15**(2), 133–150 (2011).

[13] Lee, D.-S., Gonzalez, L. F., Periaux, J., and Srinivas, K. Evolutionary optimisation methods with uncertainty for modern multidisciplinary design in aeronautical engineering. In *100 Volumes of 'Notes on Numerical Fluid Mechanics'*, 271–284. Springer (2009).

[14] Periaux, J., Lee, D. S., Gonzalez, L. F., and Srinivas, K. Fast reconstruction of aerodynamic shapes using evolutionary algorithms and virtual nash strategies in a cfd design environment. *Journal of computational and applied mathematics* **232**(1), 61–71 (2009).

[15] Lee, D., Morillo, C., Bugeda, G., Oller, S., and Onate, E. Multilayered composite structure design optimisation using distributed/parallel multi-objective evolutionary algorithms. *Composite structures* **94**(3), 1087–1096 (2012).

[16] Lee, D. S., Srinivas, K., Gonzalez, L. F., Periaux, J., and Obayashi, S. Robust multidisciplinary design optimisation using cfd and advanced evolutionary algorithms. In *Computational Fluid Dynamics Review 2010*, 469–491. World Scientific Publishing Company, Incorporated (2010).

[17] Lee, D. S., Periaux, J., Gonzalez, L., Srinivas, K., and Onate, E. Robust multidisciplinary uas design optimisation. *Structural and Multidisciplinary Optimization* **45**(3), 433–450 (2012).

[18] Deb, K., Pratap, A., and Meyarivan, T. *Constrained Test Problems for Multi-objective Evolutionary Optimization*, 284–298. Springer Berlin Heidelberg, Berlin, Heidelberg (2001).

[19] Chafekar, D., Xuan, J., and Rasheed, K. *Constrained Multi-objective Optimization Using Steady State Genetic Algorithms*, 813–824. Springer Berlin Heidelberg, Berlin, Heidelberg (2003).

[20] Deb, K. and Goel, T. *Multi-Objective Evolutionary Algorithms for Engineering Shape Design*, 147–175. Springer US, Boston, MA (2002).

[21] Deb, K., Pratap, A., and Meyarivan, T. Constrained test problems for multi-objective evolutionary optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, 284–298. Springer, (2001).

[22] Samtech, S. Samcef user manual 14.1, (2012).

[23] Bossanyi, E. Bladed theory manual version 4.0, (2010).

[24] Betran, J. and De Breuker, R. Exact method for coupled non-linear state-space simulation and its application to a flexible multibody wind turbine aeroelastic code. *Journal of Aeroelasticity and Structural Dynamics* **3**(2) (2014).