EVOLUTIONARY WINCH DESIGN USING AN ONLINE WINCH PROTOTYPING TOOL

Ibrahim A. Hameed[*], Robin T. Bye[*], Birger Skogeng Pedersen[**] and Ottar L. Osen[*,†]
[*] Software and Intelligent Control Engineering Laboratory
[*] Department of ICT and Natural Sciences
[**] Mechatronics Laboratory
[**] Department of Ocean Operations and Civil Engineering
[*,**] NTNU, Norwegian University of Science and Technology
[*,**] Postboks 1517, NO-6025 Ålesund, Norway
[†] ICD Software AS
[†] Hundsværgata 8, NO-6008 Ålesund, Norway

KEYWORDS

Virtual prototyping; Product optimisation; Computer-automated design; Maritime winch design; Genetic algorithm; Particle swarm optimisation; Simulated annealing; Multi-objective optimisation using genetic algorithm; Artificial intelligence.

ABSTRACT

This paper extends the work of a concurrent paper on an intelligent winch prototyping tool (WPT) that is part of a generic and modular software framework for intelligent computer-automated product design. Within this framework, we have implemented a Matlab winch optimisation client (MWOC) that connects to the WPT and employs four evolutionary optimisation algorithms to optimise winch design. The four algorithms we employ are (i) a genetic algorithm (GA), (ii) particle swarm optimisation (PSO), (iii) simulated annealing (SA), and (iv) a multi-objective optimisation genetic algorithm (MOOGA). Here, we explore the capabilities of MWOC in a case study where we show that given a set of design guidelines and a suitable objective function based on these guidelines, we are able to optimise a particular winch design with respect to some desired design criteria. Our research has taken place in close cooperation with two maritime industrial partners, Seaonics AS and ICD Software AS, through two innovation and research projects on applying artificial intelligence for intelligent computer-automated design of maritime equipment such as offshore cranes and maritime winches.

INTRODUCTION

At the Software and Intelligent Control Engineering (SoftICE) Laboratory[1] at NTNU in Ålesund, virtual prototyping (VP) has been a key focus of research for several years. In particular, together with two industrial partners from the world-leading maritime industrial cluster in Norway, namely ICD Software AS[2] and Seaonics AS,[3] the SoftICE lab has recently completed two independent but related research projects on using artificial intelligence (AI) for intelligent computer-automated design (CautoD) of offshore cranes and maritime winches. In these projects, we have developed a generic and modular software framework for intelligent computer-automated design (CAutoD) of maritime cranes and winches (Bye et al., 2016) and examined how various intelligent evolutionary algorithms can be applied to optimise the design (Hameed, Bye, Osen, Pedersen and Schaathun, 2016; Hameed, Bye and Osen, 2016*a*,*b*). Here, we have implemented a Matlab winch optimisation client (MWOC) that connects with an online winch prototyping tool (WPT) that we present in an accompanying paper submitted concurrently to this conference (Bye et al., 2017).

In the remainder of this paper, we first provide some background on product design optimisation in general and VP of maritime winch systems in particular. We proceed with presenting our method, including a short overview of our product optimisation system, the MWOC, and the evolutionary algorithms that we have used. Then, we provide a case study of a winch design optimisation problem and present its results. Finally, we discuss our findings and directions for future work.

BACKGROUND

Product Design Optimisation

CAutoD revolves around the concept of *optimisation*, where the design problem is formulated as an optimisation problem where the best solution from all feasible solutions is determined by the optimisation of an objective function. In case of multiple criteria decision-making where there is a set of competing and conflicting objective functions, multi-objective optimisation (MOO) can also employed. To be able to perform such an optimisation, the design has to be broken down into a set of design parameters, and the goal of the designer is to determine suitable values for the design parameters such that the objective function is optimised. When the optimisation problem is difficult or impossible to solve using analytical or exact methods, a common solution is to apply population-based

Corresponding author: Ibrahim A. Hameed, ibib@ntnu.no.
[1] http://blog.hials.no/softice
[2] http://www.icdsoftware.no
[3] http://www.seaonics.com

Proceedings 31st European Conference on Modelling and Simulation ©ECMS Zita Zoltay Paprika, Péter Horák, Kata Váradi, Péter Tamás Zwierczyk, Ágnes Vidovics-Dancs, János Péter Rádics (Editors)
ISBN: 978-0-9932440-4-9/ ISBN: 978-0-9932440-5-6 (CD)

evolutionary algorithms to get a satisfactory solution in a feasible time (e.g., see Zhang et al., 2011, for a survey).

Virtual Prototyping of Maritime Winch Systems

Maritime winches come in many shapes and uses, including trawling, anchor handling, mooring, towing, cranes of various sizes, and launch and recovery of remotely operated vehicles (ROVs). An example is depicted in Figure 1, which shows the Seaonics SCM-LARS, a new super compact launch and recovery system (LARS) that comes with a 3000 m umbilical on an electric winch with permanent magnet (PM) motors.

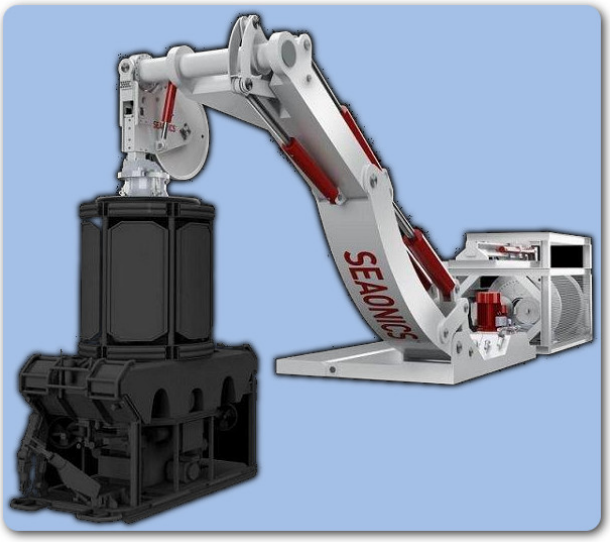


Figure 1: SCM-LARS with electric winch PM motors. Adapted from image courtesy of Seaonics AS.

As in our accompanying paper (Bye et al., 2017), we focus here on four important winch design components, namely the drum, electric motors, hydraulic motors, and the wire. Choosing appropriate values for design parameters related to these four components is necessary to yield winch designs that comply with desired design requirements and performance measures, or key performance indicators (KPIs). Typically, because many of these parameters are dependent on each other, human winch designers will have to iteratively try a large set of combinations of parameter settings to arrive at a satisfactory design (Pearlman et al., 2017). For more details about the design process of winches and the motivation for our work, please see Bye et al. (2017).

In this paper, we employ evolutionary algorithms in an attempt to automate this design optimisation process. We extend the case study we present in Bye et al. (2017) and are mainly concerned with torque performance as a KPI. Notably, however, many other concerns should be taken into account by the winch designer, such as manufacturing and operating costs, equipment weight, and adhering to laws, regulations, design codes, and standards.

METHOD

Product Optimisation System

In Bye et al. (2017), we present an overview of the client-server software architecture of our product optimisation system, reproduced in Figure 2 for convenience. The

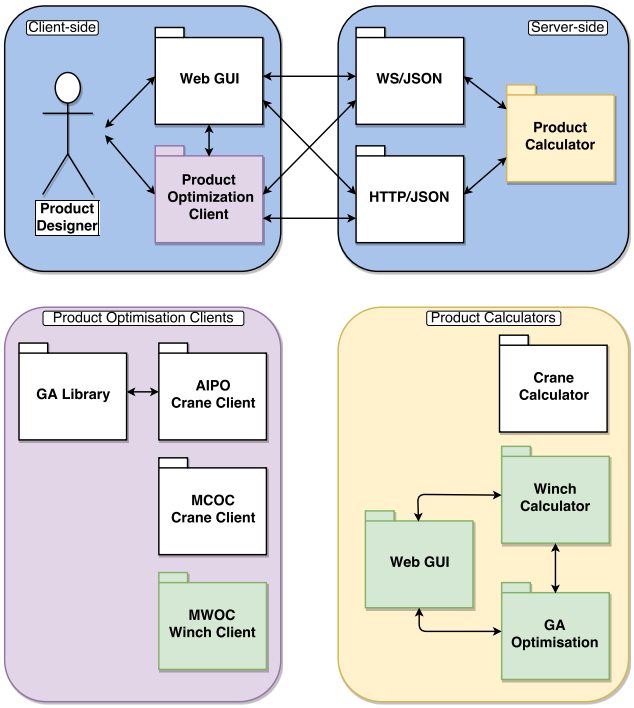


Figure 2: Software architecture for intelligent CautoD of offshore cranes, winches, or other products. Green boxes indicate work not presented previously. Adapted from Bye et al. (2017).

main component of the server is a product calculator, for example for offshore cranes or winches, that typically contains a large number of different and interdependent product design parameters with complex and nonlinear interactions. By setting different combinations of parameter values, the product calculator calculates a number of KPIs for each combination, which are effectively equivalent to different designs of the same product. Given a product design with desirable KPIs, the challenge is to determine the parameter values such that those KPIs goals being met.

On the client-side, product designers can opt to use a web graphical user interface (GUI) to manually obtain suitable parameter values in the product calculator by trial-and-error, or implement a custom-made product optimisation client (POC) to automate the design process. Both the web GUI and POCs connect to the product calculator via the Hypertext Transfer Protocol (HTTP) or the WebSocket (WS) protocol and use JavaScript Object Notation (JSON). Consequently, the client can obtain KPIs, as well as other measures of interest, that result from setting various combinations of parameter values in the product calculator.

Here, we extend the case study presented in Bye et al. (2017) and implement a *winch* optimisation client in Matlab, the MWOC, that employs four different optimisation methods, namely GA, PSO, SA, and MOOGA.

Matlab Winch Optimisation Client (MWOC)

The MWOC module makes use of two libraries freely available from the MathWorks File Exchange[4] that were used for the WS/JSON interface, namely MatlabWebSocket, which is a simple library consisting of a websocket server and client for Matlab, and JSONlab, which is a toolbox to encode/decode JSON files in Matlab (Hameed, Bye and Osen, 2016*a*). For optimisation, we used a set of solvers for evolutionary optimisation algorithms available in the Global Optimization Toolbox (Mathworks, Inc., 2015), namely the GA Solver, the PSO Solver, the SA Solver, and the Multiobjective GA Solver.

The Genetic Algorithm (GA)

The GA (Holland, 1975) is the earliest, most well known, and probably most widely used evolutionary algorithm. Since its popularisation by Goldberg (1989), GAs have consistently served as an effective optimisation tool and therefore have become a very popular choice for optimisation problems in a variety of disciplines (e.g., see Haupt and Haupt, 2004; Simon, 2013).

A GA aims to exploit random search to solve optimisation problems and uses some genetic operators to direct the search into the region of better performance within the search space. Specifically, GAs have at least the following elements in common: a *population* of chromosomes (candidate solutions), *selection* according to fitness, *crossover* to produce new offspring, and random *mutation* of new offspring. Each iteration of this process is called a *generation*. A GA is iterated for a number of generations until a satisfactory solution is found. The entire set of generations is called a *run*.

Table 1 shows a summary of the GA parameter settings we used in our accompanying paper (Bye et al., 2017) and that we also use in this study.

setting	typical
candidates, N_c	100
parents, N_p	50
elites, N_e	10
mutations, N_m	10
generations, N_g	500

Table 1: GA settings with typical values.

Particle Swarm Optimisation (PSO)

PSO is a computational method that optimises a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO is a population based stochastic optimisation technique inspired by social behaviour of bird flocking or fish schooling (Kennedy and Eberhart, 1995). Similar to GAs, PSO optimises a problem by having a population of candidate solutions, called *particles*, that constitutes a swarm moving around in the search space looking for the best solution according to a simple mathematical formula for the particles' position and velocity. Each particle's movement is influenced by its inertia, where each particle tends to maintain its velocity and direction; its local best known position (due to *personal influence/experience*), where a particle tends to return to its previous position if it is better than the current one; and finally by the global best known positions of its neighbour particles in the search-space (due to *social influence by neighbours*). Neighbours are defined topologically and are not based on the solution space. By following these simple rules, the swarm of particles will move toward the best solutions.

Table 2 shows a summary of the PSO settings we use in this paper.

setting	typical
swarm/population size, N_s	200
inertia range, W	[0.1 1.1]
min neighborhood fraction, σ	0.25
max iterations/generations, N_g	2000
max cognition learning rate, $\phi_{1,\max}$	1.49
max social learning rate , $\phi_{2,\max}$	1.49

Table 2: PSO settings with typical values.

Simulated Annealing (SA)

SA is an optimisation algorithm that mimics the cooling and crystallising behaviour of chemical substances (Kirkpatrick et al., 1983). The algorithm is initialised with a candidate solution x_0 to some minimisation problem and its *temperature* parameter, T_0, set to a high value so that the candidate solution is likely to change to some other configuration. It then randomly generates an alternative candidate solution x and measures its cost. If the cost of x is less than of x_0, the algorithm updates the candidate solution accordingly. Otherwise, it updates the candidate solution with a probability less than or equal to one. For every iteration, the temperature decreases, resulting in a tendency of the candidate solution to settle in a low-cost state. At the beginning of the algorithm, exploration is high and exploitation is low and vice versa at the end. A cooling schedule is used to control the rate of convergence.

Table 3 shows a summary of the SA settings used in this paper. Here, x_0 is row vector with the default values of the five optimisation parameters: (i) the drum inner radius, (ii) the gear ratio and (iii) quantity of electric motors, and (iv) the gear ratio and (v) quantity of hydraulic motors (more details in the Case Study section).

setting	typical
start point, x_0	[2.927 189.0 3 159.16 4]
reannealing interval, L	50
initial temperature, T_0	100
max iterations/generations, N_g	15000
cooling rate, α	0.95

Table 3: SA settings with typical values.

[4]http://www.mathworks.com/matlabcentral/fileexchange

Multi-Objective Optimisation GA (MOOGA)

Most realistic engineering problems have conflicting and competing multiple-objectives (MOs), for example, simultaneously minimising cost and maximising performance of a car engine. Combining individual objective functions into a single composite objective function, where each individual objective is weighted, is challenging and might not be realistic or even correct. An alternative approach is to determine an entire Pareto optimal solution set or a representative subset. A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective to achieve a certain amount of gain in the other. Determining a set of Pareto solutions overcomes the problem of weight selection often used in the aforementioned composite objective functions used in traditional single objective GA.

CASE STUDY

We extend the case study in Bye et al. (2017), where we investigate a winch KPI consisting of the three torque profiles that result from a given winch design, namely the S1 continuous duty cycle[5] torque T_{S1}, the maximum torque T_{max}, and the required torque T_{req}. These torque profiles are all functions of the wire velocity v, which has a resolution of N_{r} sample points between zero and the maximum wire velocity v_{max}. As in Bye et al. (2017), we employ a resolution of $N_{\mathrm{r}} = 20$.

The S1 torque can be defined as the maximum continuous duty cycle with constant load that the electric motors can safely operate under; the maximum torque as an upper threshold at which the electric motors can safely operate under but only for shorter periods of time; and the required torque as the minimum torque required for safe operation for a given constant load.

As explained in Bye et al. (2017), the torque requirements of the winch increase with winch layers, and we therefore conservatively optimise with respect to the outermost layer 15.

Design Parameters

As for our work in Bye et al. (2017), we consider the 28 design parameters given in Table 4. The default values for each parameter yield the torque profiles depicted in Figure 3 and correspond to a baseline winch design obtained by a human operator at Seaonics AS and has not been subject to any optimisation.

As argued in Bye et al. (2017), we limit the winch design optimisation to only five parameters that influence the torque profiles, namely the inner diameter of the drum, and the gear ratios and quantities of the electric and hydraulic motors (shown in bold in Table 4).

For each design parameter to be optimised, we add constraints, that is, minimum and maximum allowed values. Table 4 summarises the parameter settings, including default, minimum, maximum, and optimised parameter values for each parameter p.

[5]One of eight duty cycle classifications (S1–S8) provided by the International Electrotechnical Commission in the IEC 60034-1 standard.

Objective Function

Our KPI of interest relates to the torque profiles of T_{req}, T_{max}, and T_{S1}. We adopt the same guidelines[6] as in Bye et al. (2017):

- T_{req} should be lower than T_{max} for all wire velocities v_k, except for standstill where $v_0 = 0$, where it could be allowed to be higher.
- T_{req} should be lower than T_{S1} at wire velocities used for continuous operation, that is, typically from half the maximum wire velocity v_{max} and higher.
- conversely, T_{req} should preferably lie between T_{max} and T_{S1} for wire velocities *not* used for continuous operation, that is, typically from half the maximum wire velocity v_{max} and lower.

These guidelines are intended to prevent ending up with a winch design with bigger and more expensive motors than necessary, leading to T_{req} being below T_{S1} for all velocities, including low velocities that are not suitable for continuous operation. This is not necessary, because it is possible to operate above the nominal S1 duty rating for shorter periods of time. Thus, In addition to reducing costs, this choice can improve weight and performance, since bigger motors will have higher mass and inertia. However, it must be kept in mind that after operating above the S1 duty cycle for a short period of time, the motors must be allowed to operate *below* S1 to avoid overheating.

We adopt the same cost function as in Bye et al. (2017) to achieve winch designs that adhere to our guidelines:

$$f_{\mathrm{cost}} = \sum_{k=1}^{N_{\mathrm{r}}} a \cdot \mathrm{R}_k + (1-a) \cdot \mathrm{S}_k \tag{1}$$

where

$$\mathrm{R}_k = \begin{cases} \delta \mathrm{R}_k^2 \text{ for } \delta \mathrm{R}_k > 0 \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

$$\mathrm{S}_k = \delta \mathrm{S}_k^2 \tag{3}$$

$$\delta \mathrm{R}_k = T_{\mathrm{req}}(v_k) - T_{\mathrm{max}}(v_k) \tag{4}$$

$$\delta \mathrm{S}_k = T_{\mathrm{req}}(v_k) - T_{\mathrm{S1}}(v_k) \tag{5}$$

$$a = 0.5 \tag{6}$$

and v_k is the kth sample of the wire velocity. For algorithms implemented using fitness functions such as the GA described previously, we simply negate the cost function:

$$f_{\mathrm{fitness}} = -f_{\mathrm{cost}} \tag{7}$$

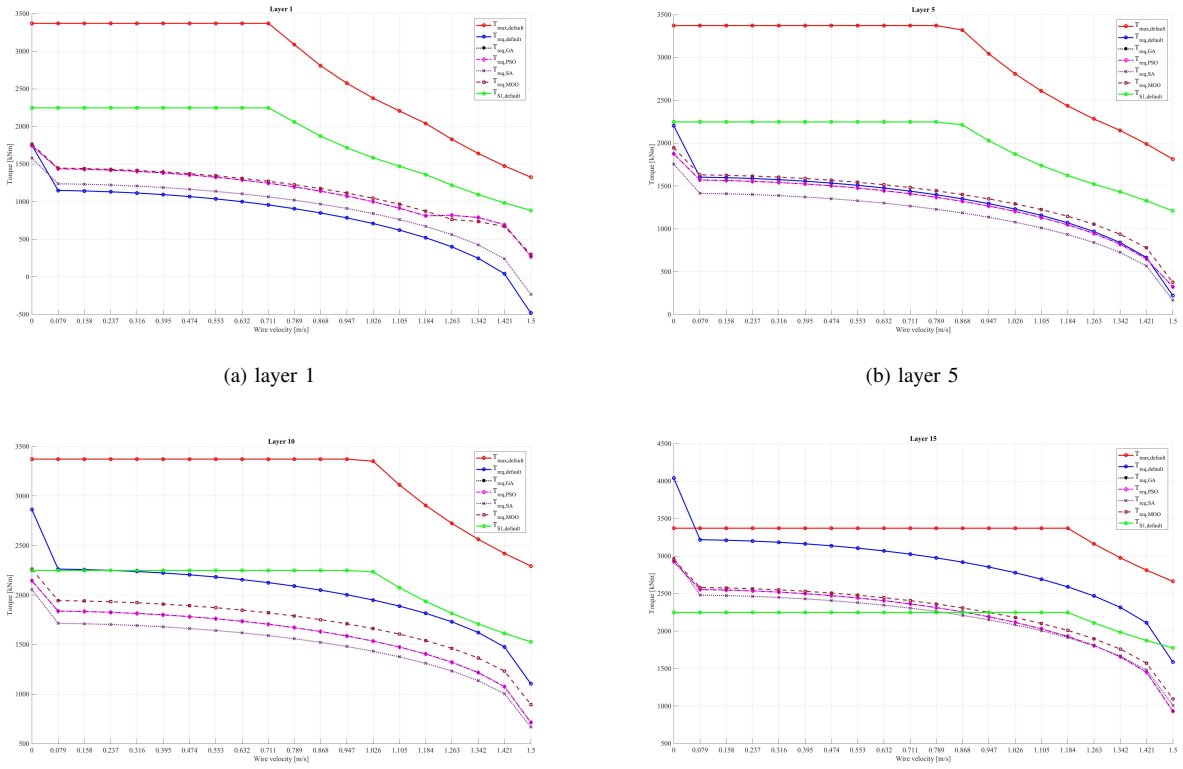
Results

Figure 3 shows the torque profiles for T_{req} (blue), T_{S1} (green), and T_{max} (red) for winch layers 1, 5, 10, and 15, before optimisation (i.e., default design values) and after optimisation using GA ($T_{req,GA}$), PSO ($T_{req,PSO}$), SA ($T_{req,SA}$), and MOOGA ($T_{req,MOOGA}$), respectively.

[6]As per information provided by Seaonics AS.

subset	p	name	units	default	min	max	GA	PSO	SA	MOOGA
general	1	wavePeriodTime	s	10	-	-				
	2	load	kN	1500	-	-				
	3	iPlanetary	-	0.50	-	-				
	4	desiredWireVelocity	m/s	1.50	-	-				
drum	5	r	m	2.20	-	-				
	6	**innerDiameter**	**m**	**2.927**	**2.70**	**2.99**	**2.7007**	**2.7000**	**2.9702**	**2.8792**
	7	mass	kg	130000	-	-				
electric motor	**8**	**gearRatio**	-	**189.00**	**170**	**200**	**199.90**	**200.00**	**200.00**	**192.53**
	9	**motorQuantity**	-	**3**	**1**	**5**	**5**	**5**	**5**	**5**
	10	fieldWeakeningSpeed	rev/min	1440	-	-				
	11	maxSpeed	rev/min	2000	-	-				
	12	maxSpeedPower	kW	157.0	-	-				
	13	nominalPower	kW	210	-	-				
	14	nominalSpeed	rad/s	93.410	-	-				
	15	i	kg·m^2	9.00	-	-				
	16	nominalTorque	N·m	2248.15	-	-				
hydraulic motor	**17**	**gearRatio**	-	**159.16**	**150**	**190**	**189.99**	**190.00**	**158.58**	**188.83**
	18	**motorQuantity**	-	**4**	**1**	**5**	**3**	**3**	**4**	**3**
	19	friction	-	0.950	-	-				
	20	staticEfficiency	-	0.789	-	-				
	21	dynamicEfficiency	-	0.916	-	-				
	22	maxSpeed	rev/min	1600	-	-				
	23	displ	cm^3	1000	-	-				
	24	pressureDrop	bar	280	-	-				
	25	i	kg·m^2	0.5500	-	-				
	26	nominalTorque	N·m	4457.71	-	-				
wire	27	diameter	N·m	77.0	-	-				
	28	diameterReductionFactor	-	0.8660	-	-				
cost value	-	-	-	876900	-	-	784123	783757	805600	{0, 876900}

Table 4: Default and optimised winch designs using GA, PSO, SA, and MOOGA. Empty fields signify default values.



(a) layer 1

(b) layer 5

(c) layer 10

(d) layer 15

Figure 3: Torque profiles for T_{req} (blue), T_{S1} (green), and T_{max} (red) *before* optimisation and required torque after optimisation using GA ($T_{req,GA}$), PSO ($T_{req,PSO}$), SA ($T_{req,SA}$), and MOOGA ($T_{req,MOOGA}$), respectively, for winch layers 1, 5, 10, and 15.

Compared with the unsatisfactory default design in Figure 3, all optimisation methods result in very similar torque profiles for winch layer 15 that closely adhere to the guidelines presented previously, where T_{req} should lie below T_{max} and above T_{S1} for low wire velocities, and below T_{S1} (and T_{max}) for high velocities. Moreover, for winch layer 10, all optimised torque profiles are highly satisfactory, with T_{req} below T_{S1} (and T_{max}) for all wire velocities, which is a significant improvement from the default design. Finally, for winch layers 1 and 5, the torque profiles are similar to those of the default design, except that T_{req} is lower than T_{S1} for zero velocity. This is an improvement but not a requirement, since T_{req} is allowed to be higher than T_{S1} for short periods of time.

Whilst the qualitative examination of the torque profiles show that the desired relationships between T_{req}, T_{S1}, and T_{max} are met in a highly similar manner for all the optimisation methods, investigation of Table 4 indicate that there are several combinations of optimised parameter values that achieve the same desired performance. For example, using GA and PSO, the inner diameter of the drum is optimised to a value very close to the lower boundary value of 2.70 m, whereas using SA and MOOGA results in values of 2.97 (close to the upper boundary of 2.99) and 2.88, respectively.

Similarly, the gear ratio of the electric motors is optimised to a value very close to the upper boundary of 200 for GA, PSO, and SA, whereas MOOGA found the value of 192.53. For hydraulic motors, on the other hand, the gear ratio was optimised towards a value very close to the upper boundary of 190 for GA, PSO, and MOOGA, whereas the optimised value found by SA was 158.58, which is actually quite close to the default value of 159.16. This latter result may be due to the SA optimising the quantity of hydraulic motors to 4, whereas GA, PSO and MOOGA only used 3 (the default design used 4). Thus, the optimised design found by SA had one extra hydraulic motor compared with GA, PSO, and MOOGA, and therefore had less need for a high gear ratio.

Finally, all optimisation methods optimised the quantity of electric motors to 5, two more than the default design that only used 3 (which is probably a major contributor to the default design being unsatisfactory).

It is unsurprising that having extra electric motors help reducing the required torque T_{req} relative to the S1 duty cycle torque T_{S1}. Notably, however, all optimisation methods apart from SA simultaneously reduced the number of hydraulic motors to 3 from the default quantity of 4, to avoid excess torque capabilities as laid out in the guidelines.

Both GA and PSO found an optimised inner diameter of the drum very close to the lower constraint of 2.70 m. This makes intuitive sense, since a smaller diameter has lower torque requirements. But why does the SA and MOOGA have higher optimised diameter values towards the upper constraint then? We believe the answer may be twofold. First, increasing the inner diameter of the drum may ensure that the default speed requirements of the motors are satisfied, because the wire velocity increases with the diameter of the drum for the same rotational speed. Second, SA uses one extra hydraulic motor, whereas MOOGA has a very high gear ratio for the hydraulic motors. Both parameter values will improve the torque profiles, thus possibly dominating, or voiding the need for a low inner diameter of the drum.

Finally, we observe that GA and PSO has the greatest reduction in cost function evaluation when compared with the default design (bottom line in Table 4. Moreover, we did not get a Pareto optimal set because the objective functions are not competing.

We conclude from these results that the evolutionary optimisation algorithms have found different combinations of optimised parameter values (winch designs) that all yield very similar torque profiles. The optimised parameter values improve an unsatisfactory default winch design to yield designs that are satisfactory, but not excessively good, and adhere to the design guidelines we set out previously.

DISCUSSION

This paper extends our case study on winch optimisation (Bye et al., 2017) with a set of four evolutionary optimisation algorithms; the GA, PSO, SA, and MOOGA. The algorithms are implemented in a Matlab winch optimisation client, the MWOC, that connects to an online winch prototyping tool, the WPT, thus demonstrating practical use of our software framework for intelligent product design.

For the case study, we set out to improve a default winch design that did not satisfy the desired performance, or KPI, as described qualitatively in a set of design guidelines. These guidelines describe the desired relationship between the required torque T_{req}, the S1 duty cycle torque T_{S1}, and the maximum torque T_{max}, which all are functions of the wire velocity. In Bye et al. (2017), we devised a cost function derived from these guidelines with the intention of using the aforementioned evolutionary algorithms to determine optimised parameter values that should result in satisfactory winch designs. Obviously, simply adding a large number of motors would result in highly proficient torque profiles. However, doing so would result in winch designs that are better than needed and also are very costly. Intelligently, the evolutionary optimisation algorithms we have employed here are able to strike a sweet spot between the weak default winch design and too powerful winch designs that massively exceed the desired torque performance.

Our work demonstrates that employing evolutionary algorithms in our product optimisation system can be useful for CAutoD of maritime equipment such as offshore winches and cranes but also for any other product for which the design can be parameterised and the design goal is to determine a suitable set of parameter values for which the resulting design satisfy some desired design criteria, or KPIs.

Future Work

A limitation of our work is that we have assumed that the parameters chosen for optimisation can take on any value within some pre-defined boundaries (minimum and maximum values). However, when our industrial partner Seaonics AS are doing their winch design, they need to pick suitable *combinations* of components (sets of fixed parameter values) that are commercially available and meet the clients' needs. Seaonics AS is currently in the process of providing us with an extensive set of real-world manufactured components that we could use to build up this library. After this, we will update our system and let Seaonics's professional winch designers test our software. During this process, we will try to identify and correct possible shortcomings with the potential of subsequently adopting the software for common use, such as developing new objective functions for MOO, adding user's guides, developing new application-dependent design guidelines with corresponding objective functions for optimisation, improving the web graphical user interface (GUI) of the WPT, and more. For more details about our work and future research, we refer to our accompanying paper submitted in parallel (Bye et al., 2017).

ACKNOWLEDGEMENTS

The SoftICE lab at NTNU in Ålesund wishes to thank ICD Software AS for their contribution in the software development process, and Seaonics AS for providing documentation and insight into the design and manufacturing process of offshore cranes. We are also grateful for the support provided by Regionalt Forskningsfond (RFF) Midt-Norge and the Research Council of Norway through the VRI research projects *Artificial Intelligence for Crane Design (Kunstig intelligens for krandesign (KIK))*, grant no. 241238, and *Artificial Intelligence for Winch Design (Kunstig intelligens for vinsjdesign (KIV))*, grant no. 249171.

REFERENCES

Bye, R. T., Hameed, I. A., Pedersen, B. S. and Osen, O. L. (2017), An intelligent winch prototyping tool, Proceedings of the 31st European Conference on Modelling and Simulation (ECMS '17). Under review.

Bye, R. T., Osen, O. L., Pedersen, B. S., Hameed, I. A. and Schaathun, H. G. (2016), A software framework for intelligent computer-automated product design, Proceedings of the 30th European Conference on Modelling and Simulation (ECMS '16), pp. 534–543.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional.

Hameed, I. A., Bye, R. T. and Osen, O. L. (2016*a*), A Comparison between Optimization Algorithms Applied to Offshore Crane Design using an Online Crane Prototyping Tool, Proceedings of the 2nd International Conference on Advanced Intelligent Systems and Informatics (AISI '16), Vol. 533 of *Advances in Intelligent Systems and Computing (AISC)*, pp. 266–276.

Hameed, I. A., Bye, R. T. and Osen, O. L. (2016*b*), Grey wolf optimizer (GWO) for automated offshore crane design, Proceedings of the IEEE Symposium Series on Computational Intelligence (IEEE SSCI '16), pp. 1–6.

Hameed, I. A., Bye, R. T., Osen, O. L., Pedersen, B. S. and Schaathun, H. G. (2016), Intelligent computer-automated crane design using an online crane prototyping tool, Proceedings of the 30th European Conference on Modelling and Simulation (ECMS '16), pp. 564–573.

Haupt, R. L. and Haupt, S. E. (2004), *Practical Genetic Algorithms*, 2nd edn, Wiley.

Holland, J. H. (1975), *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Oxford, England.

Kennedy, J. and Eberhart, R. C. (1995), Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, Vol. IV, IEEE, IEEE service center, Piscataway, NJ, pp. 1942–1948.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), Optimization by Simulated Annealing, *Science* 220(4598), 671–680.

Mathworks, Inc. (2015), *MATLAB Global Optimization Toolbox, R2015b*, The Mathworks, Inc., Natick, Massachusetts.

Pearlman, S. M., Gordon, D. R. and Pearlman, M. D. (2017), Winch Technology — Past Present and Future: A Summary of Winch Design Principles and Developments, Technical report, InterOcean Systems LLC. Accessed on 14 February 2017 from http://www.interoceansystems.com/winch_article.htm.

Simon, D. (2013), *Evolutionary Optimization Algorithms: Biologically Inspired and Population-Based Approaches to Computer Intelligence*, John Wiley & Sons, Inc., Hoboken, New Jersey.

Zhang, J., Zhan, Z., Lin, Y., Chen, N., jiao Gong, Y., hui Zhong, J., Chung, H., Li, Y. and Shi, Y. (2011), Evolutionary computation meets machine learning: A survey, *IEEE Computational Intelligence Magazine* 6(4), 68–75.

AUTHOR BIOGRAPHIES

IBRAHIM A. HAMEED is an IEEE Senior Member and has a BSc and a MSc in Industrial Electronics and Control Engineering, Menofia University, Egypt, a PhD in Industrial Systems and Information Engineering from Korea University, S. Korea, and a PhD in Mechanical Engineering, Aarhus University, Denmark. He has been working as an associate professor at NTNU Ålesund since 2015. His research interests includes artificial intelligence, optimisation, control systems, and robotics.

ROBIN T. BYE[7] is an IEEE Senior Member who graduated from the University of New South Wales, Sydney with a BE (Hons 1), MEngSc, and a PhD, all in electrical engineering. Working at NTNU in Ålesund since 2008, Dr. Bye is an associate professor in automation engineering. His research interests belong to the fields of artificial intelligence, cybernetics, neuroengineering, and education.

BIRGER SKOGENG PEDERSEN graduated from NTNU Ålesund with a BE in automation engineering and is a former employee at ICD Software AS. He is currently a MSc student of simulation and visualisation and employed as a research assistant in the Mechatronics Laboratory at NTNU Ålesund.

OTTAR L. OSEN is MSc in Engineering Cybernetics from the Norwegian Institute of Technology in 1991. He is the head of R&D at ICD Software AS and an assistant professor at NTNU Ålesund.

[7] www.robinbye.com