INTELLIGENT COMPUTER-AUTOMATED CRANE DESIGN USING AN ONLINE CRANE PROTOTYPING TOOL

Ibrahim A. Hameed,* Robin T. Bye,* Ottar L. Osen,*,†
Birger Skogeng Pedersen,* and Hans Georg Schaathun*
* Software and Intelligent Control Engineering Laboratory
Faculty of Engineering and Natural Sciences
Norwegian University of Science and Technology
NTNU in Ålesund, Postboks 1517, NO-6025 Ålesund, Norway
† ICD Software AS
Hundsværgata 8, NO-6008 Ålesund, Norway

KEYWORDS

Virtual Prototyping; Product Optimisation; Artificial Intelligence; Genetic Algorithm.

ABSTRACT

In an accompanying paper submitted concurrently to this conference, we present our first complete version of a generic and modular software framework for intelligent computer-automated product design. The framework has been implemented with a client-server software architecture that automates the design of offshore cranes. The framework was demonstrated by means of a case study where we used a genetic algorithm (GA) to optimise the crane design of a real and delivered knuckleboom crane. For the chosen objective function, the optimised crane design outperformed the real crane. In this paper, we augment our aforementioned case study by implementing a new crane optimisation client in Matlab that uses a GA both for optimising a set of objective functions and for multi-objective optimisation. Communicating with an online crane prototyping tool, the optimisation client and its GA are able to optimise crane designs with respect to two selected design criteria: the maximum safe working load and the total crane weight. Our work demonstrates the modularity of the software framework as well as the viability of our approach for intelligent computer-automated design, whilst the results are valuable for informing future directions of our research.

INTRODUCTION

The need to reduce the time and cost involved in taking a product from conceptualisation to production and the desire to meet customers' demands and their ability to compete have encouraged companies to turn to new and emerging technologies in the area of manufacturing. One such technology is virtual prototyping (VP) (Mujber et al., 2004). VP refers to the process of simulating the user, the product, and their combined (physical) interaction in software through the different stages of product design, and the quantitative performance analysis of the product (Song et al., 1999). Being a relatively new technology, VP typically involve the use of virtual reality (VR), virtual environments (VE), computer-automated design (CautoD) solutions, computer-aided design (CAD) tools, and other computer technologies to create digital prototypes (e.g., Gowda et al., 1999).

Together with two companies in the industrial maritime cluster of Norway, ICD Software AS (provider of industrial control systems software)[1] and Seaonics AS (designer and manufacturer of offshore equipment),[2] we have received funding from the Research Council of Norway and its Programme for Regional R&D and Innovation (VRI) for two independent but related research projects (grant nos. 241238 and 249171) for using artificial intelligence (AI) for intelligent computer-automated design (CautoD) of offshore cranes and winches, respectively. In an accompanying paper submitted concurrently (Bye et al., 2016), we present our first complete version of a generic and modular software framework for intelligent computer-automated product design. The framework has been implemented with a client-server software architecture for the design of offshore cranes and consists of several modules: a server-side crane prototyping tool (CPT); a client-side web graphical user interface (GUI); and a client-side artificial intelligence for product optimisation (AIPO) module that uses a genetic algorithm (GA) for optimisation.

The framework was demonstrated by means of a case study where we used the AIPO module and its GA to optimise the crane design of a particular real-world knuckleboom crane that has already been designed by Seaonics AS and sold to a company in Baku, Azerbaijan, for a total delivery price of approximately 2.9 million EUR. For the chosen objective function, the optimised crane design outperformed the real crane.

Motivation and Aim

For the work we present here, we will focus solely on intelligent CautoD of offshore cranes, using the software framework developed concurrently (Bye et al., 2016). In

Corresponding author: Ibrahim A. Hameed, ibib@ntnu.no.

[1] www.icdsoftware.no
[2] www.seaonics.com

Proceedings 30th European Conference on Modelling and Simulation ©ECMS Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose (Editors)
ISBN: 978-0-9932440-2-5 / ISBN: 978-0-9932440-3-2 (CD)

the concurrent paper, we tested the framework with a case study that only involved a single crane optimisation client using a single objective function. Here, we aim at complementing this work by completing the following three goals: (i) examine the modularity of the framework by developing a new crane optimisation client in Matlab (the MCOC module) to be used instead of the AIPO module; (ii) augment the abovementioned case study with a set of alternative objective functions as well as multi-objective optimisation (MOO); and (iii) interpret the results to inform the directions of future work.

To make this paper self-contained, we reproduce some of the material from our accompanying paper (Bye et al., 2016). However, much of the relevant background literature pertaining to VP, CautoD, and design of offshore cranes has been left out. The interested reader is encouraged to read the accompanying paper for further details.

METHOD

This section outlines the software architecture and describes the main components. We provide details on GAs, objective functions, and multi-objective optimisation, before we present a case study on intelligent CautoD of offshore cranes.

Software Architecture

The diagram in Figure 1 shows the client-server software architecture of the framework that we present in our accompanying paper (Bye et al., 2016). On the server-side, the CPT is able to calculate a number of key performance indicators (KPIs) of a specified crane design based on a set of about 120 design parameters. On the client-side, the web GUI facilities the process of manually selecting the design parameters of the designed CPT and providing a simple visualisation of the designed crane and its 2D workspace safe working load (SWL) chart. Additionally, the AIPO module that uses a GA for optimising the design parameters in a manner that achieves the crane's desired design criteria (that is, the level or quality of the KPIs, typically related to performance and cost).

In the work we present here, we replaced the AIPO module and its GA library with a new Matlab software module that implements a crane optimisation client for CautoD, the MCOC module. To emphasise that the framework is generic and modular, we chose to use the WebSocket (WS) communication interface instead of the hypertext transfer protocol (HTTP) that the AIPO module used (see Figure 1). WS is a protocol providing full-duplex communication channels over a single TCP connection. Because WS enables streams of messages on top of TCP, using WS for communication is advantageous for bidirectional conversations involving many small messages being sent to and from a server. JavaScript Object Notation (JSON), a lightweight human-readable data-interchange format, was used for data messages. We also kept the existing web GUI in order to obtain visualisations of load charts. The software architecture for this reduced subsystem is highlighted with white boxes and solid connections in Figure 1, whereas the remaining boxes in grey and the dashed lines indicate modules and their interconnections outside the scope of this paper.

Online Crane Prototyping Tool (CPT)

The CPT server consists of a crane calculator and two modules for handling WS/JSJON and HTTP/JSON connections (see Figure 1). Here, we let our MCOC connect via WS/JSON to the CPT (see Figure 1). Messages are sent as JSON objects in a standardised format that the CPT accepts, consisting of three parts (subobjects): (i) a "base" object with a complete set of default design parameter values; (ii) a "mods" object with a subset of design parameter values that modifies the corresponding default values; and (iii) a "kpis" object with the desired KPIs to be calculated and returned by the CPT.

Crane Calculator

The components of an offshore crane may total several thousand parameters, making it infeasible to manually pick good values for each parameter. However, through the years, crane designers have been able to reduce this number to a set of about 120 design parameters that are considered the most important. Based on the values of these parameters, which can be set manually or by a CautoD tool such as MCOC, our crane calculator is able to calculate a fully specified crane design and its associated KPIs. The goal of the designer is thus to determine appropriate design parameter values that achieve desired design criteria (based on KPIs), while simultaneously meeting requirements by laws, regulations, codes and standards.

The accuracy of our crane calculator has been verified against other crane calculators and spreadsheets currently in use in the industry, and, as a result, Seaonics AS has already adopted the CPT server and web GUI client for manual crane design.

Web Graphical User Interface (GUI)

To simplify practical use of the crane calculator, we have created a web graphical user interface (GUI) that can be used to interact with the crane calculator via WS/JSON communication. Using the web GUI to manually adjust the 120 design parameters in the crane calculator by trial-and-error, the effect of the parameters on a number of KPIs and other design criteria can be investigated numerically, with the possibility for exporting to text files, and visually, by depicting the main components of the crane and its 2D SWL load chart.

Due to space consideration, we refer to Bye et al. (2015, 2016) for a screenshot of the GUI and more information.

Matlab Crane Optimisation Client (MCOC)

The manual design process using the web GUI together with the CPT is cumbersome. Indeed, there are more than 120 parameters that must be specified by the crane designer. Clearly, this large number of parameters makes

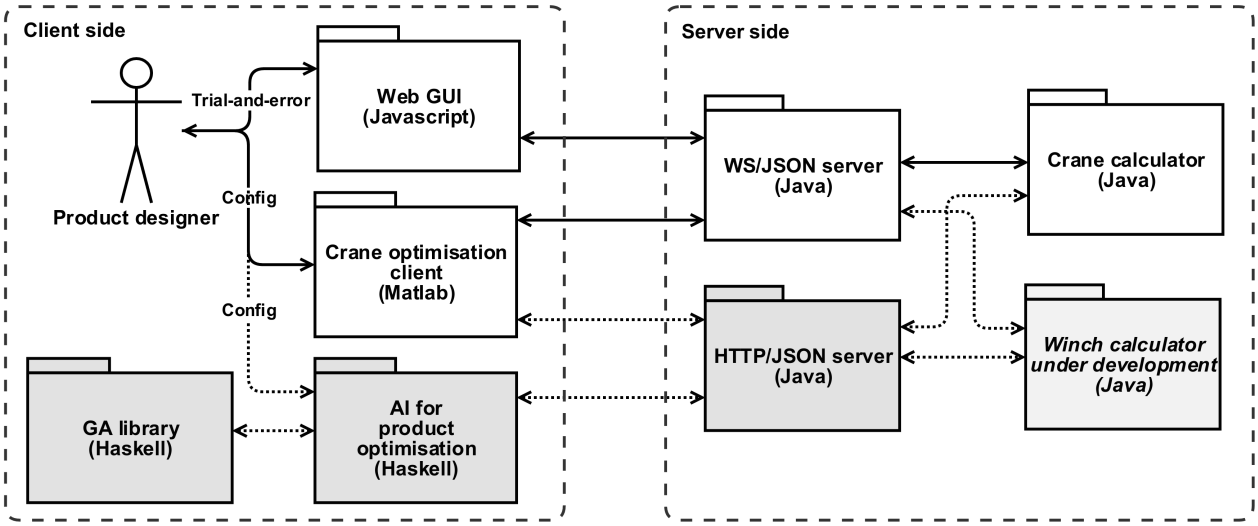

Figure 1: Generic and modular software architecture for intelligent CautoD of offshore cranes, winches, or other products. The modules in white (grey) and their solid (dashed) interconnections are inside (outside) the scope of this paper.

the search space (the space of all possible combinations of parameter values) very large and a manual trial-and-error approach will necessarily be both time-consuming and cost-inefficient and lead to suboptimal designs.

In our accompanying paper (Bye et al., 2016), we present an AIPO software module replacing the human crane designer in order to automate and optimise the design process. Here, we use Matlab to implement such a crane optimisation client, the MCOC module. Two libraries freely available from the MathWorks File Exchange[3] 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. For optimisation, we used the GA Solver and the Multiobjective GA Solver from the Global optimisation Toolbox (Mathworks, Inc., 2015). The GA solvers were used to optimise a set of objective functions that we define later.

The Genetic Algorithm (GA)

A GA is a search method based on principles of natural selection and genetics (Holland, 1975). GAs encode the decision variables of a search problem into finite-length strings of alphabets of certain cardinality. The strings, which are candidate solutions to the search problem, are referred to as *chromosomes*, the alphabets are referred to as *genes*, and the values of the genes are called *alleles*. In contrast to traditional optimisation techniques, GAs work with coding of parameters, rather than the parameters themselves. To evolve good solutions and to implement natural selection, a measure for distinguishing good solutions from bad solutions is required. This measure is usually an objective function, and is called a fitness (cost) function if the goal is to maximise (minimise) it.

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

Another important concept of GAs is the notion of population. Unlike most traditional search methods, GAs rely on a population of candidate solutions. The population size, which is usually a user-specified parameter, is one of the important factors affecting the scalability and performance of GAs. A small population size might lead to premature convergence and yield substandard solutions. On the other hand, large population sizes lead to unnecessary expenditure of valuable computational time. Once the problem is encoded in a chromosomal manner and a fitness or cost measure for discriminating good solutions from bad ones has been chosen, a GA can start to evolve solutions to the search problem using the following steps:

1) *Initialization*. The initial population of candidate solutions is usually generated randomly across the search space. However, domain-specific knowledge or other information can be easily incorporated.
2) *Evaluation*. Once the population is initialized or an offspring population is created, the fitness values of the candidate solutions are evaluated.
3) *Selection*. Selection allocates more copies of those solutions with higher fitness (lower cost) and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. The main idea of selection is to prefer better solutions to worse ones, and many selection procedures have been proposed to accomplish this idea, including roulette-wheel selection, stochastic universal selection, ranking selection and tournament selection.
4) *Recombination*. Recombination combines parts of two or more parental solutions to create new, possibly better solutions (i.e. offspring). There are many ways of accomplishing this, and good performance depends on a properly designed recombination mechanism.
5) *Mutation*. While recombination operates on two or

more parental chromosomes, mutation locally but randomly modifies a solution.

6) *Replacement*. The offspring population created by selection, recombination, and mutation replaces the original parental population. Many replacement techniques such as elitist replacement, generation-wise replacement and steady-state replacement methods are used in GAs.
7) *Repeat*. Steps 2–6 are repeated until a termination criterion is satisfied, for example, a maximum number of generations, a run-time limit, a fitness threshold, or no improvement is detected for certain number of generations or run-time.

Objective Functions

In GAs, an objective function (either a cost function or a fitness function) is used to generate an output from a set of input variables (a chromosome). The goal is to modify the output in some desirable fashion by finding the appropriate values for the input variables (Haupt and Haupt, 2004).

GAs are generally customised for solving single-objective optimisation problems (SOPs). However, many, or most, real-world engineering problems require MOO, since they have multiple, often conflicting, objectives such as minimising cost while maximising performance. GAs can be used for MOO through the aggregation of the individual objective functions into a single composite function. Determination of a single objective is possible with methods such as utility theory or the weighted sum method but the problem lies in the correct selection of the weights or utility functions to characterise the decision-makers' criteria. In practice, it can be very difficult to precisely and accurately select these weights, even for someone very familiar with the problem domain. Also, small perturbations in the weights can lead to very different solutions. For this reason and others, decision-makers often prefer a set of promising solutions given the multiple objectives (Konak et al., 2006). Such a set is called a Pareto optimal set of solutions.

Multi-Objective optimisation using a GA (MOOGA)

Combining individual objective functions into a single composite objective function is challenging and might not be realistic or even correct. The second general 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 when combining individual objectives into one composite objective function.

Case Study

We adopt the same case study as in our accompanying paper (Bye et al., 2016), where a real knuckleboom crane is used as a nominal benchmark against an optimised crane. The crane has about 120 different design parameters and a number of KPIs. Due to the large number of design parameters, the manual design process is cumbersome, time consuming and expensive. Even simple versions of such offshore cranes consist of a large number of components, including hooks, winches, slewing rings, cylinders, booms, hinges, sheaves, and pedestals. Figure 2 illustrates the main components of offshore cranes.

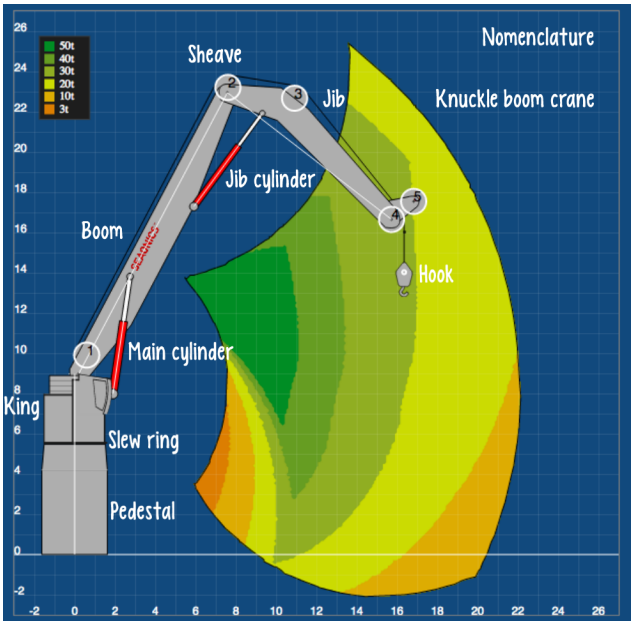

Figure 2: Illustration of the main components of an offshore knuckleboom crane and its 2D load chart. Image courtesy of ICD Software AS.

In an attempt to reduce design time, cost and satisfy customers' need, we propose a CautoD solution in which the MCOC module uses a GA to automate the process and optimise the design.

Choice of KPIs: Among many relevant KPIs, two KPIs were chosen as components of the objective functions in order to demonstrate proof-of-concept, namely the maximum safe working load SWL_{max} and the total crane weight W. Whilst the total crane delivery price is of great concern, currently there is no function implemented in the CPT that can precisely estimate the delivery price of the crane designed. Nevertheless, the total weight can, to some extent, be used as a proxy for price, because price will have some correlation to the weight, and one wants to minimise both measures. Moreover, these cranes are installed on-board vessels and a reduced crane weight allows for a higher deadweight tonnage (DWT). Hence, weight is important for both capital and operating expenditure. The maximum SWL, on the other hand, is a measure of the maximum safe lifting capacity of the crane within the workspace. The goal of the design is to maximize SWL_{max} while simultaneously minimising W. These two objectives are conflicting and competing with each other, since increasing SWL_{max} will tend to increase W and vice versa.

A number of objective functions were implemented in the MCOC and are presented below.

Objective function f_1*:* An intuitive choice for an objective function composed of SWL_{max} and W is the fitness function f_1 given by

$$f_1 = \frac{\text{SWL}_{\text{max}}}{\text{W}}, \tag{1}$$

since the evaluation of f_1 will increase when SWL_{max} increases and/or W decreases, and vice versa.

Objective function f_2*:* Another composite objective function f_2 is the weighted sum of both SWL_{max} and W given by

$$f_2 = w_1 \text{SWL}_{\text{max}} + w_2 \frac{1}{\text{W}}, \tag{2}$$

where w_1 amd w_2 are weight values used to reflect the importance or amount of contribution of SWL_{max} and W. We note that the total fitness will increase when SWL_{max} increases and/or W decreases, and vice versa.

Objective functions f_3 ***and*** f_4*:* It may be of interest to design a crane where either SWL_{max} or W is the same as for the nominal benchmark crane, while we optimise the remaining KPI. For example, it might be that the crane customer wants a crane with the same "target" weight W_{target} as the nominal crane but with a higher SWL_{max}. Likewise, the crane customer might require the optimised crane to be able to safely lift as much (but not necessarily more, since this could for example have a detrimental effect on the delivery price) as the benchmark crane, denoted as $\text{SWL}_{\text{target}}$, but with a smaller total crane weight W. The objective function must therefore "punish" deviations from the target KPI while optimising the other KPI. Thus, two possible cost functions f_3 and f_4 are given by

$$f_3 = w_1 \frac{1}{\text{SWL}_{\text{max}}} + w_2 \left| \text{W}_{\text{target}} - \text{W} \right| \tag{3}$$

and

$$f_4 = w_1 \left| \text{SWL}_{\text{target}} - \text{SWL}_{\text{max}} \right| + w_2 \text{W}, \tag{4}$$

where w_1 and w_2 are weight values as before.

Choice of Optimisation Variables*:* Among the 120 different design parameters, four design parameters that greatly affect both SWL_{max} and W were chosen as decision (i.e., optimisation) variables, namely (i) the boom length L_{boom}; (ii) the jib length L_{jib}; (iii) the maximum pressure of the boom cylinder $P_{\text{max,boom}}$; and (iv) the maximum pressure of the jib cylinder $P_{\text{max,jib}}$. The parameter values were constrained to a range with minimum and maximum limits. All other design parameters were identical to those of the nominal crane.

GA Settings*:* For GA optimisation, we used a population size (set of candidate design solutions) of 100 and let the GA run for 50 generations, giving a grand total of 5,000 evaluated designs.

RESULTS

Table 1 shows the values of the four design parameters L_{boom}, L_{jib}, $P_{\text{max,boom}}$, and $P_{\text{max,jib}}$ and the resulting maximum SWL (SWL_{max}) and total crane weight (W) for the nominal crane that we use as a benchmark with which to compare the optimisation results. During optimisation, each design parameter was constrained to a minimum and a maximum value as given by the Table 1. The table also shows the objective function evaluations of the nominal crane.

measure	units	nominal	(min, max)
L_{boom}	mm	15800	(12000, 26000)
L_{jib}	mm	10300	(6000, 16000)
$P_{\text{max,boom}}$	bar	315	(100, 400)
$P_{\text{max,jib}}$	bar	215	(50, 300)
SWL_{max}	tonne	99.978	-
W	tonne	50.856	-
objective function	evaluation	w_1	w_2
f_1	1.9659	-	-
f_2	100.00	1	1
f_2	198.29	1	5000
f_2	1098.10	10	5000
f_2	108.31	0.1	5000
f_3	0.01000	1	1
f_4	50.856	1	1

Table 1: Nominal crane, its objective function evaluations, and optimisation constraints.

Table 2 provides a summary of the results. It shows the total processing times and optimised values for SWL_{max} and W for each of the optimised cranes, the mean and standard deviation for these values, and the difference of the means when compared with the nominal crane.

objective function	SWL_{max}	W	T (min)
f_1	142.14	44.01	98.4
f_2, $w_1 = w_2 = 1$	140.63	44.22	115.21
f_2, $w_1 = 1$, $w_2 = 5000$	140.59	44.22	89.39
f_2, $w_1 = 10$, $w_2 = 5000$	140.02	44.22	106.19
f_2, $w_1 = 0.1$, $w_2 = 5000$	143.37	43.88	66.36
f_3, $w_1 = w_2 = 1$	112.54	50.81	125.82
f_4, $w_1 = w_2 = 1$	99.94	47.1	90.97
MOO	140.95	43.88	182.83
mean	132.52	45.29	109.40
standard deviation	16.60	2.47	34.68
nominal	99.98	50.86	-
difference of mean with nominal	32.54	-5.56	-

Table 2: Processing time T in minutes and optimal values of SWL_{max} and W for the set of objective functions, their mean and standard deviation, and the difference of the means from the nominal crane.

The total processing time is the total run-time from the start of the optimisation process till a result was obtained, including transfer times between the MCOC client and the CPT server.

For reference, we include the detailed results of employing f_1–f_4 and MOO for optimisation in Tables 4–11.

Maximum SWL (SWL_{max})

Table 2 shows that employing f_1, f_2, or MOO all resulted in optimised cranes with a SWL_{max} greater than 140 tonnes, or an improvement of more than 40 tonnes when compared to the $\text{SWL}_{\text{max}} = 99.98$ tonnes of the nominal crane.

Employing f_3, whose purpose is to maximimise SWL_{max} while having a W as close as possible to that of the nominal crane, resulted in an SWL_{max} of about 12 tonnes more than the nominal crane's SWL_{max}.

Finally, employing f_4 resulted in a crane with a $\text{SWL}_{\text{max}} = 99.94$, which is almost identical to the $\text{SWL}_{\text{max}} = 99.98$ tonnes of the nominal crane. This is not surprising, given that the purpose of f_4 was to minimise W while having a SWL_{max} as close as possible to that of the nominal crane.

The mean SWL_{max} for all the optimised crane designs was 132.52 tonnes, or an improvement of 32.54 tonnes when compared with the nominal crane. The standard deviation of SWL_{max} for the optimised cranes was 16.60.

Total Crane Weight (W)

Table 2 shows that employing f_1, f_2, or MOO all resulted in optimised cranes with a total W of 44.22 tonnes or less, or an improvement of about 7 tonnes when compared to the $\text{W} = 50.86$ tonnes of the nominal crane.

Employing f_4 resulted in a $\text{SWL}_{\text{max}} = 47.1$, or an improvement of nearly 4 tonnes when compared to the nominal crane.

Finally, employing f_3 resulted in a crane with a $\text{W} = 50.81$, which is almost identical to that of the nominal crane.

The mean W for all the optimised crane designs was 45.29 tonnes, or an improvement of 5.56 tonnes when compared with the nominal crane. The standard deviation of W for the optimised cranes was 2.47.

Processing Times

The total processing time for each of the optimisation processes ranged from 66.36 minutes for f_2 with $w_1 = 0.1$ and $w_2 = 5000$ to 182.83 minutes for the MOO. The mean processing time was 109.40 minutes, with a standard deviation of 34.68 minutes. The fastest processing time was more than one standard deviation lower than the mean, whereas the slowest processing time was more than two standard deviations higher than the mean. The remaining processing times were all within one standard deviation from the mean.

SWL Load Charts

The SWL load charts for nominal crane and the optimised crane designs are shown in Figures 3–6.

Each load chart shows the workspace and the SWL lifting capacity in various coloured zones of the workspace for a given crane. The legend at the top left indicates the capacity of a particular zone with colours in a spectrum from red (3 tonnes) up to blue (150 tonnes).

Comparing the charts, it is apparent that all the optimised cranes have one or several zones with a SWL capacity in the range 50–150 tonnes, whereas the zone of the nominal crane with the highest SWL capacity is 50 tonnes.

It can also be observed that the overall lifting capacity of the workspace is higher than that of the nominal crane.

However, a notable observation is that all crane designs apart from that obtained using f_3 has a smaller workspace than the nominal crane. The reason for this is that whereas the goal of using f_3 is to obtain a total crane weight W identical to the nominal crane (while maximising SWL_{max}), the other objective functions and the MOO try to minimise W. As a result, the lengths of the boom and jib are shorter than the nominal crane for these latter designs, thus making W smaller, but at the expense of a smaller workspace.

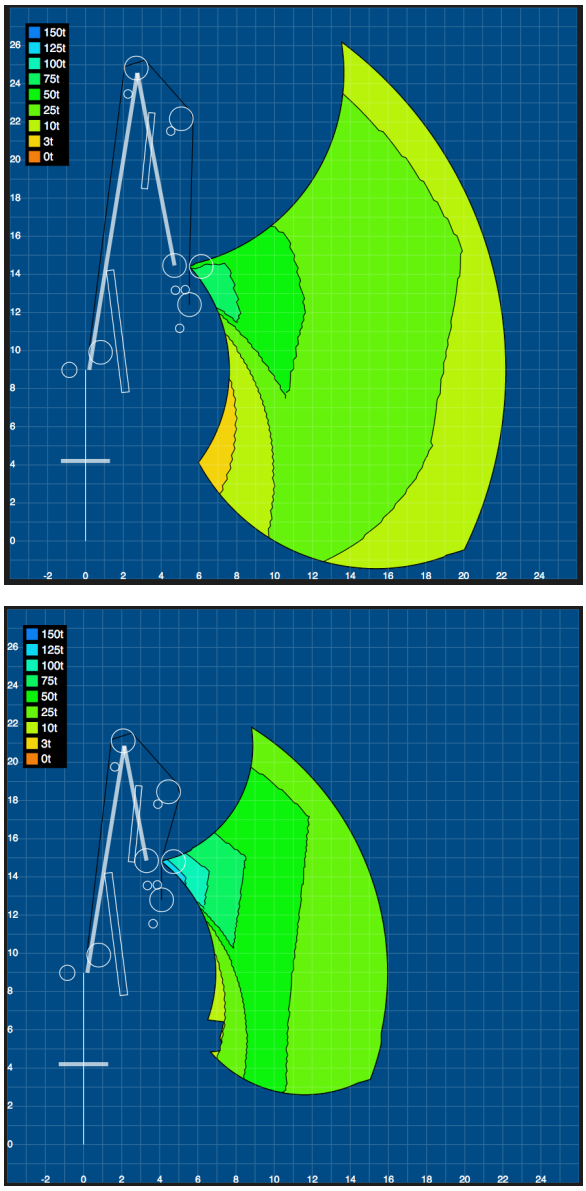

Figure 3: SWL load charts: nominal (top); f_1 (bottom).

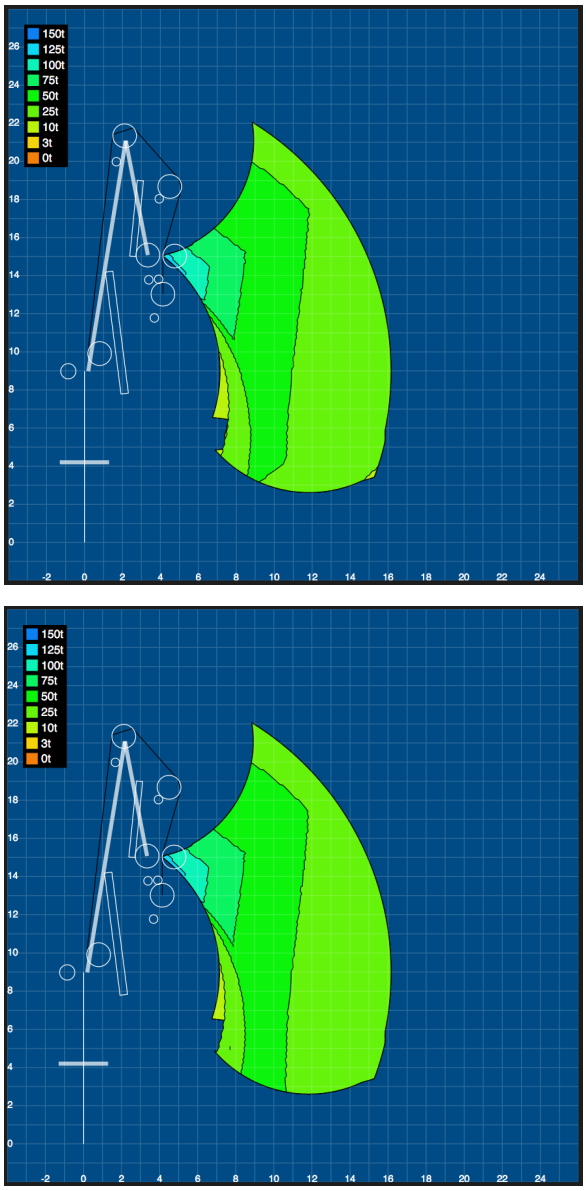

Figure 4: SWL load charts: f_2: $w_1 = w_2 = 1$ (top); f_2: $w_1 = 1$, $w_2 = 5000$ (bottom).

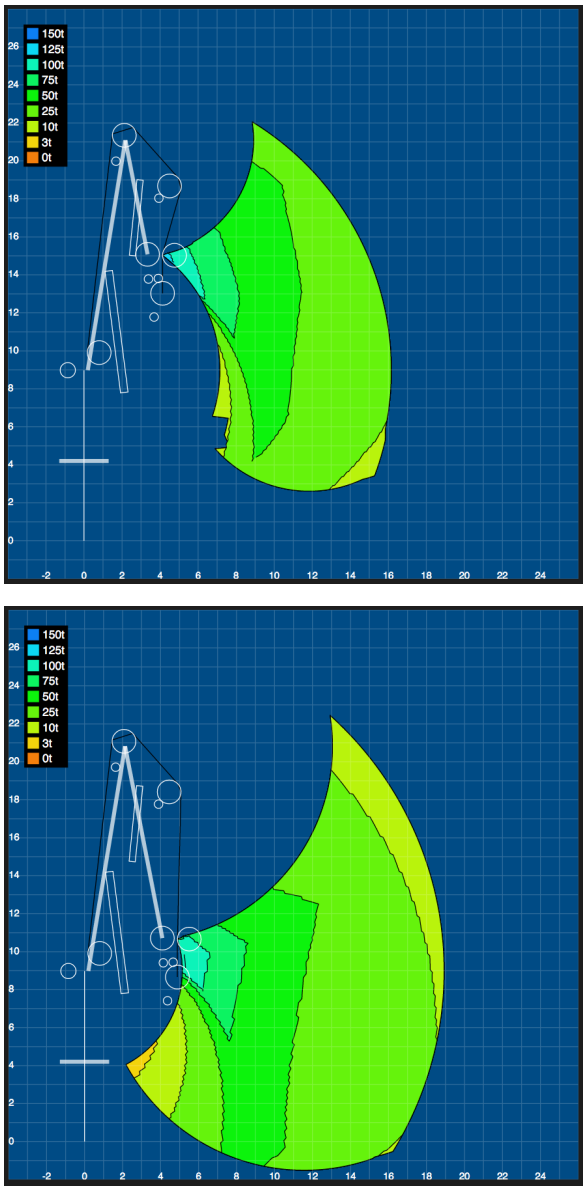

Figure 5: SWL load charts: f_2: $w_1 = 10$, $w_2 = 5000$ (top); f_2: $w_1 = 0.1$, $w_2 = 5000$ (bottom).

Multiobjective Optimisation (MOO)

For MOO, the two KPIs SWL_{max} and W were used as two individual objective functions to be respectively maximised and minimised by the Matlab MOOGA Solver. The optimal solution is provided as a set of Pareto-optimal solutions for values of the design parameters given by Table 3. Each of these solutions results in a crane design with $\text{SWL}_{\text{max}} = 140.95$ tonnes and $\text{W} = 43.88$ tonnes. A sample solution is presented in Table 11 in the Appendix.

DISCUSSION

In this paper, we have presented an intelligent computer-automated design solution for optimising offshore cranes using a genetic algorithm for single-objective or multi-objective optimisation. Candidate crane designs suggested by a GA incorporated in a Matlab crane optimisation client are sent to an online crane prototyping tool that uses a crane calculator to determine two key performance indicators, the maximum safe working load and the total crane weight, for each crane design. The CPT server sends the results back to the MCOC and the GA uses them to evolve another set of candidate solutions. The process iterates until some stopping criteria is satisfied, for example when the solutions do not improve for a prolonged number of iterations.

Case Study

To test the viability of our approach, we adopted the case study of our accompanying paper (Bye et al., 2016). Here, we used MOO and a set of four different objective functions to optimise the design of an offshore crane consisting of about 120 design parameters. Of about 120 design parameters, most were fixed to values correspond-

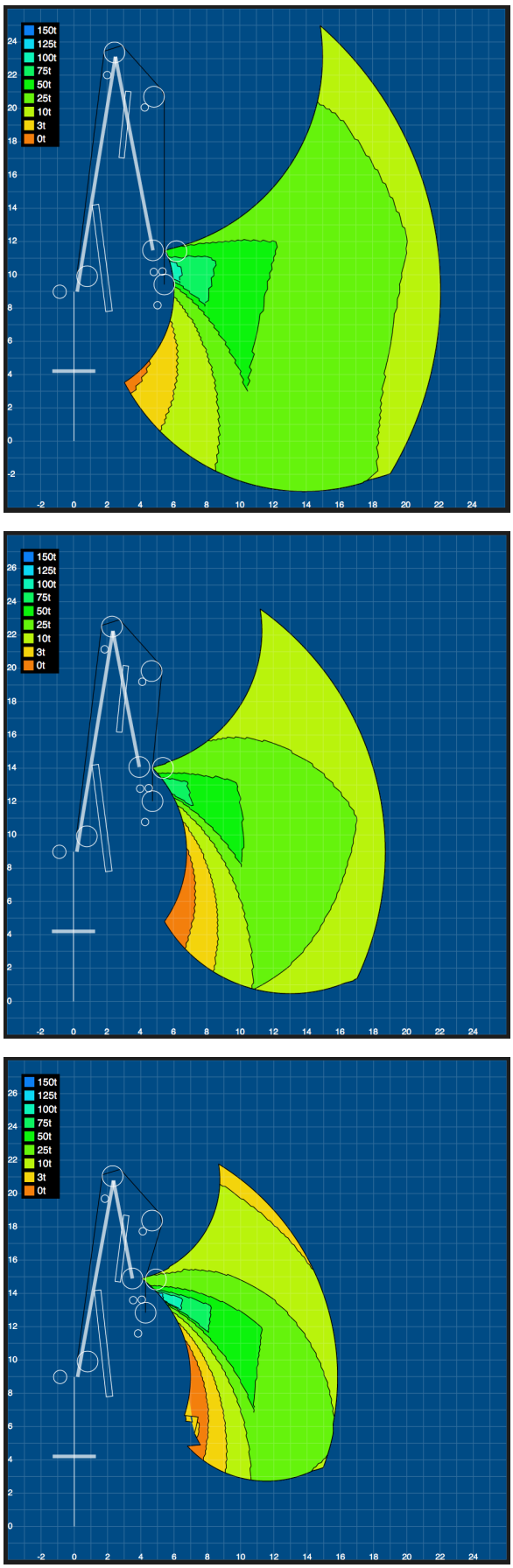

Figure 6: SWL load charts: f_3 (top); f_4 (middle); MOO (bottom).

L_{boom}	L_{jib}	$P_{\text{max,boom}}$	$P_{\text{max,jib}}$
12000.9	6000.95	331.309	67.1085
12000.8	6000.50	331.183	75.4687
12000.9	6000.39	332.326	88.6125
12000.9	6000.60	332.32	71.5644
12000.6	6000.95	331.309	67.1085
12000.6	6000.49	332.241	80.8655
12001.0	6000.69	332.101	72.0046
12000.9	6000.88	331.984	68.08
12000.9	6000.61	332.333	83.6164
12000.9	6000.98	331.788	74.0161
12001.0	6000.99	331.876	75.4702
12000.8	6000.95	331.309	67.1554
12000.9	6000.74	332.816	66.7924
12000.9	6000.55	331.308	75.4687
12000.7	6000.92	332.344	82.1006
12000.8	6000.61	332.227	82.379
12000.8	6000.82	332.414	66.8399
12000.5	6000.97	331.109	72.9711
12000.7	6000.79	331.142	87.5899
12000.6	6000.99	332.633	81.1601
12000.6	6000.95	331.151	84.4106
12000.8	6000.93	332.252	89.7784
12000.9	6000.86	331.048	82.0198
12000.6	6000.81	331.017	80.539
12000.9	6000.51	332.668	68.4269
12000.7	6000.98	331.41	79.5427
12000.9	6000.91	331.983	68.8521
12000.7	6000.99	331.75	81.9171
12000.7	6000.66	332.401	82.9699
12000.8	6000.82	331.325	72.0107
12000.9	6000.95	331.246	67.1085
12000.7	6000.49	332.264	80.928
12000.9	6000.93	331.261	68.3642
12000.8	6000.63	332.088	78.7431
12000.9	6000.51	332.668	68.4269

Table 3: Pareto set of optimised cranes using MOO that all have $\text{SWL}_{\text{max}} = 140.95$ and $\text{W} = 43.88$.

ing to the design of a real and delivered crane, whereas four design parameters, the boom length, jib length, maximum boom cylinder pressure, and maximum jib cylinder pressure, were optimised by the MCOC. The goal of the optimisation was to maximise the lifting capacity given by SWL_{max} and the total crane weight W. Two of the objective functions (f_3 and f_4) only tried to optimise one of the KPIs while keeping the other as close as possible to that of the nominal crane.

The results show that all the optimised crane designs outperform the nominal crane on the two selected KPIs. However, other KPIs not incorporated in the optimisation process will inevitably also be change when the crane design changes. This can lead to unwanted and unexpected results. For example, as can be seen from the load charts in Figures 3–6, whilst the optimised crane designs have improved the maximum SWL, most of them have reduced workspace as a sideeffect. One way to overcome this would be to incorporate another KPI, or optimisation objective, relating to the workspace area.

Future Work

Importantly, this case study was limited to optimising only a fraction of all the design parameters needed to construct an offshore crane, and only two KPIs were considered. For more realistic use, our method needs to

be expanded to involve both more design parameters and and more KPIs. The first is trivial, as it only involves minor modification to the GA; the latter is non-trivial, as many KPIs are interrelated and mutually conflicting (for example, delivery cost versus performance), and care must be taken in the choice of objective functions. We plan to work in close cooperation with our industrial partners and their crane designers to develop a set of useful KPIs and objective functions for real-world use.

Using optimisation weights for single-objective optimisation is one means for handling this problem, however, choosing the right weights can be difficult. Using MOO can, at least to some extent, handle the problem automatically without the need to determine such weights. Instead of a single design solution, one obtains a Pareto set of solutions, all with the same values for the desired KPIs. The crane designer and customer must then decide which solution in the set to implement and build for delivery.

Being able to handle many more design parameters and KPIs will likely lead to slower processing times, since objective function evaluation is the main contributor to computational load. Still, for the proof-of-concept study we do here, the mean processing time was less than 2 hours, which is many orders of magnitude smaller than what a human crane designer would require. In future work, we intend to implement several other AI algorithms for optimisation, possibly using parallel computation, and examine their performance both with respect to optimisation and processing time.

We will also use the knowledge we gain from our study on offshore cranes inform related work on optimised CautoD of offshore winches.

Concluding Remarks

The work presented here has accomplished the three goals set out in the introduction: We have successfully been able to use the software framework developed concurrently (see Bye et al., 2016) by creating a new product optimisation client customised for offshore cranes and insert it as a module in our existing framework. Moreover, we have augmented the case study we present in Bye et al. (2016) with a set of alternative objective functions and with MOO and the results are valuable for our future development. Finally, we would like to point the reader to our accompanying paper for related concurrent and future work (Bye et al., 2016).

ACKNOWLEDGEMENTS

The SoftICE lab at NTNU in Ålesund wishes to thank ICD Software AS for their contribution towards the implementation of the simulator, 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., Osen, O. L. and Pedersen, B. S. (2015), A computer-automated design tool for intelligent virtual prototyping of offshore cranes, Proceedings of the 29th European Conference on Modelling and Simulation (ECMS'15), pp. 147–156.

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) (submitted for publication).

Gowda, S., Jayaram, S. and Jayaram, U. (1999), Architectures for internet-based collaborative virtual prototyping, the 1999 ASME Design Technical Conference and Computers in Engineering Conference, DETC99/CIE-9040, Las Vegas, Nevada.

Haupt, R. L. and Haupt, S. E. (2004), *Practical Genetic Algorithms*, John Wiley & Sons, Inc.

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.

Konak, A., Coit, D. W. and Smith, A. E. (2006), Multi-objective optimization using genetic algorithms: A tutorial., *Reliability Engineering & System Safety* 91(9), 992–1007. Springer-Verlag: Berlin Heidelberg.

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

Mujber, T., Szecsi, T. and Hashmi, M. (2004), Virtual reality applications in manufacturing process simulation, *Journal of Materials Processing Technology* 155, 1834–1838.

Song, P., Krovi, V., Kumar, V. and Mahoney, R. (1999), Design and virtual prototyping of humanworn manipulation devices, the 1999 ASME Design Technical Conference and Computers in Engineering Conference, DETC99/CIE-9029, Las Vegas, Nevada.

AUTHOR BIOGRAPHIES

IBRAHIM A. HAMEED 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 in Ålesund since 2015. His research interests includes artificial Intelligence, optimization, control systems and robotics.

ROBIN T. BYE[4] graduated from the University of New South Wales, Sydney with a BE (Hons 1), MEngSc, and a PhD, all in electrical engineering. Dr. Bye began working at NTNU in Ålesund (formerly Aalesund University College) as a researcher in 2008 and has since 2010 been an associate professor in automation engineering. His research interests belong to the fields of artificial intelligence, cybernetics, and neuroengineering.

OTTAR L. OSEN is MSc in 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 in Ålesund.

BIRGER SKOGENG PEDERSEN graduated from NTNU in Ålesund with a BE in automation engineering

[4]www.robinbye.com

and is a former employee at ICD Software AS, during which time he worked on the research projects described in this paper. He is currently a MSc student of simulation and visualisation as well as a employed as a project manager in maritime technology at NTNU in Ålesund.

HANS GEORG SCHAATHUN Hans Georg Schaathun graduated from the University of Bergen with cand.mag. in 1996 (Mathematics, Economics, and Informatics), cand.scient. in 1999 (Industrial and Applied Mathematics and Informatics), and dr.scient. in 2002 (Informatics – Coding Theory), all from the University of Bergen (UiB), Norway. After a period as a lecturer and a postdoc at UiB and a lecturer and senior lecturer at the University of Surrey, England, he joined Aalesund University College (now NTNU in Ålesund) and became a professor in 2011. His current research focus is software engineering and pedagogy.

APPENDIX

All tables show the optimised values for the four design parameters, the resulting SWL_{max} and W, and the optimised objective function value. Values are compared with the nominal crane and the difference and percentage change is shown.

measure	units	optimised	difference	change
L_{boom}	mm	12038	-3762	-23.81%
L_{jib}	mm	6124	-4176	-40.54%
$P_{\text{max,boom}}$	bar	383	68	21.59%
$P_{\text{max,jib}}$	bar	262	47	21.86%
SWL_{max}	tonne	142.14	42.16	42.17%
W	tonne	44.01	-6.84	-13.45%
f_1	-	3.23	1.26	64.27%

Table 4: Optimised crane using f_1.

measure	units	optimised	difference	change
L_{boom}	mm	12266.9	-3533.10	-22.36%
L_{jib}	mm	6120.56	-4179.44	-40.58%
$P_{\text{max,boom}}$	bar	340.9	25.90	8.22%
$P_{\text{max,jib}}$	bar	266.84	51.84	24.11%
SWL_{max}	tonne	140.63	40.65	40.66%
W	tonne	44.22	-6.64	-13.05%
f_2, $w_1 = w_2 = 1$	-	140.65	40.65	40.66%

Table 5: Optimised crane using f_2, $w_1 = w_2 = 1$.

measure	units	optimised	difference	change
L_{boom}	mm	12266.9	-3533.10	-22.36%
L_{jib}	mm	6123.56	-4176.44	-40.55%
$P_{\text{max,boom}}$	bar	392.14	77.14	24.49%
$P_{\text{max,jib}}$	bar	297.62	82.62	38.43%
SWL_{max}	tonne	140.59	40.61	40.62%
W	tonne	44.22	-6.64	-13.05%
f_2, $w_1 = 1$, $w_2 = 5000$	-	253.66	55.37	27.92%

Table 6: Optimised crane using f_2, $w_1 = 1$, $w_2 = 5000$.

measure	units	optimised	difference	change
L_{boom}	mm	12269.6	-3530.40	-22.34%
L_{jib}	mm	6121.56	-4178.44	-40.57%
$P_{\text{max,boom}}$	bar	304.72	-10.28	-3.26%
$P_{\text{max,jib}}$	bar	249.9	34.90	16.23%
SWL_{max}	tonne	140.02	40.04	40.05%
W	tonne	44.22	-6.64	-13.05%
f_2, $w_1 = 10$, $w_2 = 5000$	-	1513.27	415.17	37.81%

Table 7: Optimised crane using f_2, $w_1 = 10$, $w_2 = 5000$.

measure	units	optimised	difference	change
L_{boom}	mm	12000	-3800.00	-24.05%
L_{jib}	mm	6000	-4300.00	-41.75%
$P_{\text{max,boom}}$	bar	353.42	38.42	12.20%
$P_{\text{max,jib}}$	bar	297.48	82.48	38.36%
SWL_{max}	tonne	143.37	43.39	43.40%
W	tonne	43.88	-6.98	-13.72%
f_2, $w_1 = 0.1$, $w_2 = 5000$	-	128.28	19.97	18.44%

Table 8: Optimised crane using f_2, $w_1 = 0.1$, $w_2 = 5000$.

measure	units	optimised	difference	change
L_{boom}	mm	14321.5	-1478.5	-9.36%
L_{jib}	mm	11864.1	1564.1	15.19%
$P_{\text{max,boom}}$	bar	396.89	81.89	26.00%
$P_{\text{max,jib}}$	bar	187.75	-27.25	-12.67%
SWL_{max}	tonne	112.54	12.562	12.56%
W	tonne	50.81	-0.046	-0.09%
f_3, $w_1 = w_2 = 1$	-	0.0549	0.0449	448.74%

Table 9: Optimised crane using f_3, $w_1 = w_2 = 1$.

measure	units	optimised	difference	change
L_{boom}	mm	13443.9	-2356.1	-14.91%
L_{jib}	mm	8328.99	-1971.01	-19.14%
$P_{\text{max,boom}}$	bar	273.36	-41.64	-13.22%
$P_{\text{max,jib}}$	bar	101.05	-113.95	-53.00%
SWL_{max}	tonne	99.94	-0.038	-0.04%
W	tonne	47.1	-3.756	-7.39%
f_4, $w_1 = w_2 = 1$	-	47.138	-3.718	-7.31%

Table 10: Optimised crane using f_4, $w_1 = w_2 = 1$.

measure	units	optimised	difference	change
L_{boom}	mm	12000.9	-3799.1	-24.04%
L_{jib}	mm	6000.95	-4299.05	-41.74%
$P_{\text{max,boom}}$	bar	331.31	16.31	5.18%
$P_{\text{max,jib}}$	bar	67.11	-147.89	-68.79%
SWL_{max}	tonne	140.95	40.972	40.98%
W	tonne	43.88	-6.976	-13.72%

Table 11: Optimised crane using MOO.