Grey wolf optimizer (GWO) for Automated Offshore Crane Design

Ibrahim A. Hameed*, Robin T. Bye, and Ottar L. Osen
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
{ibib*, robin.t.bye, ottar.l.osen}@ntnu.no
http://blog.hials.no/softice/

***Abstract*— In this paper, a new meta-heuristic optimization algorithm called Grey Wolf Optimizer (GWO) is applied to offshore crane design. An offshore crane is a pedestal-mounted elevating and rotating lifting device used to transfer materials or personnel to or from marine vessels, barges and structures whereby the load can be moved horizontally in one or more directions and vertically. Designing and building offshore cranes is a very complex process. It depends on the configuration of a large set of design parameters and is characterized by increased workability and functionality for the owner and cost effectiveness in the total cost of ownership. In an attempt to reduce time and cost involved in the design process, this paper defines a best set of design parameters and uses GWO for the automatic configuration of this set of parameters in a manner that increases the maximum safe working load of the crane and reduces its total weight. Results are verified by a comparative study with other Evolutionary Algorithms (EAs). Results show that the GWO algorithm is able to provide very competitive results compared to these well-known meta-heuristics.**

Keywords—virtual prototyping; product optimization; artificial intelligence; design automation; software framework;

I. INTRODUCTION

Production of industrial goods tailored to the requirements of individual customers has become widely accepted for staying ahead of the competition when operating in highly developed markets [1]. A direct consequence of that is the extensive number of assemblies and parts needed to cover the specific needs of every customer, higher complexities, and smaller production batches, which, in turn, can lead to cost disadvantages. Since this approach is very costly for development and production, enterprises aim at optimizing their costs while still maintaining the possibility of satisfying their customers. The need to reduce the time and cost involved in taking a product from conceptualization to production and the desire to meet customers' demands have encouraged manufacturers to adopt new technologies in manufacturing such as virtual prototyping (VP) [2] and optimization algorithms.

Being a relatively new technology, VP typically involves 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 [3]. A virtual prototype, or digital mock-up, can be defined as "a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design/engineering, manufacturing, service, and recycling as if on a real physical model. The construction and testing of a virtual prototype is called virtual prototyping (VP)" [4]. Virtual prototyping is not necessarily only for design optimization; it might be used just for concept verification, presentation, and training. Based on the above definition, VP should include essentially three types of models; a computer simulation of a product, a human-product interaction model, and perspective test-related models. Depending on the application, a virtual prototype may only include a subset of these components [3].

We have previously developed a computer-automated design (CautoD) software framework for product design optimization and tested it on the design of offshore cranes [5-8]. The framework has several components. On the server-side, a crane prototyping tool (CPT) with a crane calculator 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 [5]. A client-side web graphical user interface (GUI) facilitates the process of manually selecting the design parameters in the CPT to obtain a simple visualization of the designed crane and its 2D safe working load (SWL) chart [5]. For design optimization, we have developed a client-side product optimization client, namely the Artificial Intelligence for Product Optimization (AIPO) module that uses a genetic algorithm (GA) for optimizing the design parameters in a manner that achieves the crane's desired design criteria (e.g., KPIs related to performance and cost specifications) [6]. Complementing the AIPO module, we have also implemented a Matlab® software module, the Matlab® crane optimization client (MCOC), that also used a GA for single and multi-objective optimization [7]. Subsequently, we have added other evolutionary algorithms such as Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithms to the MCOC [8].

The purpose of this paper is to further extend the MCOC with a new meta-heuristic called Grey Wolf Optimizer inspired by grey wolves and apply it to the same real-world offshore crane design problem as examined earlier to confirm the performance of GWO in practice by comparing its behavior to

other evolutionary algorithms. GWO mimics the social hierarchy and hunting behavior of grey wolves and has been successfully used for solving various benchmark optimization problems [9].

The remainder of the paper is organized as follows: A brief description of our product design optimization framework is presented in Section 2. In Section 3, an introduction to GWO algorithm and various objective functions are presented. Problem setup and results of the GWO applied to the offshore crane design problem are presented in Section 4. Finally, a comparison between various algorithms in terms of convergence time and accuracy, concluding remarks, and future work is presented in Section 5.

II. Offshore Crane Design Software Framework

This section outlines the software architecture and describes the main components of our software framework used for automation and optimization of offshore crane design. An offshore crane such as the one mounted on an offshore subsea construction (OSC) vessel in Fig. 1 is a complex system of components interacting to achieve safe movement of heavy goods, often under harsh and difficult conditions. 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 (see Fig. 2). The choice of crane components and their physical properties and interrelationships determines various measures of performance of interest to the crane designer.

Fig. 1. Boomerang-shaped knuckle-boom crane on OSC vessel (courtesy Offshore Energy Today) [10].

A. *Client*-Server Architecture

The diagram in Fig. 3 shows the client-server architecture of our software framework (adapted from [6]). On the server-side, the CPT contains a product (in this case, an offshore crane) calculator able to calculate a number of KPIs of a specified crane design based on a set of about 120 design parameters. On the client-side, a web GUI facilitates the process of either manually by trial-and-error selecting the design parameters of the crane, or automatically using an optimization algorithm (contained in a product (crane) optimization client). The GUI also provides a simple visualization of the designed crane and its 2D workspace safe working load (SWL) chart [5]. Additionally, the MCOC uses various optimization algorithms for optimizing the design parameters in a manner that achieves the crane's desired design criteria [7-8]. Each KPI is typically related to overall performance, weight and cost of the designed crane. Both WebSocket (WS) and the hypertext transfer protocol (HTTP) have been implemented as communication interfaces, using data messages conforming to the JavaScript Object Notation (JSON), which is a lightweight human-readable data-interchange format.

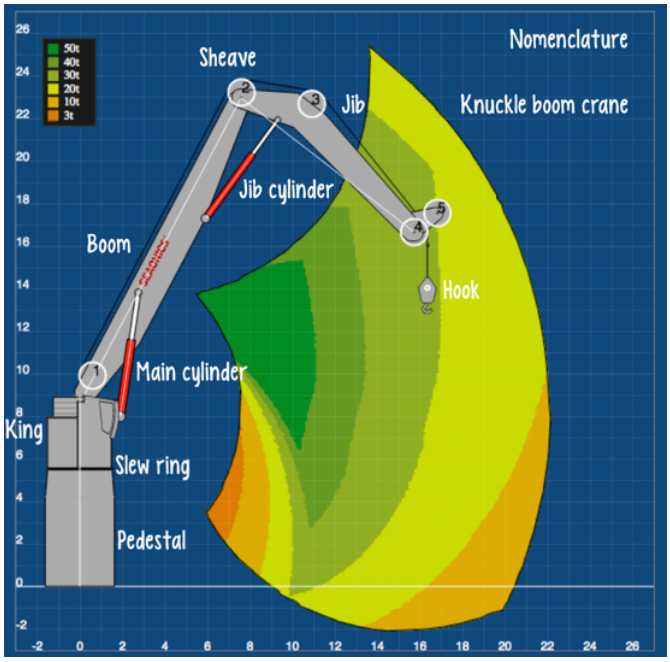


Fig. 2. Illustration of the main components of an offshore knuckle-boom crane and its 2D load chart [5-6].

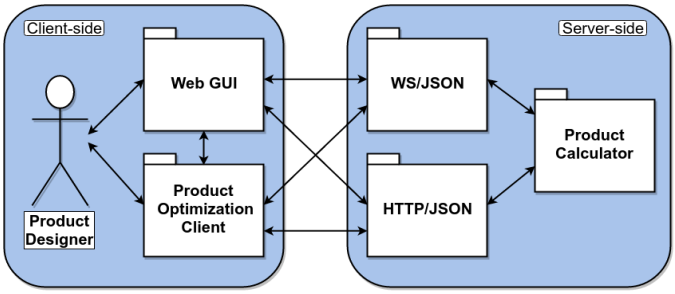


Fig. 3. Generic and modular software architecture for intelligent CautoD of offshore cranes, winches, or other products (adapted from [6]).

B. *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 Fig. 3). Here, the MCOC connect to the CPT via WS/JSON. Messages are sent as JSON objects in a standardized format that the CPT accepts, consisting of three parts (sub-objects) [6]:

(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 from "*base*" and "*mods*" and returned to the MCOC by the CPT.

This simple description is sufficient for the rapid implementation of product optimization clients in any programming language supporting either WS or HTTP for communication and JSON for data messages.

C. Crane Calculator

The components of an offshore crane may consist of several thousand parameters, however, with the help from crane designers we have been able to isolate the most important ones and reduce this number to a set of about 120 design parameters. Based on the values of these parameters, which can be set manually or by a client CautoD tool such as MCOC, the crane calculator is able to calculate a fully specified crane design and its associated KPIs [5]. The goal of the designer is therefore to determine appropriate design parameter values that achieve some desired design criteria based on KPIs, or to try to improve an existing design.

The design must simultaneously meet requirements by laws, regulations, codes and standards, as well as other constraints, such as a maximum total delivery price and total crane weight for the owner.

We have verified the accuracy of the crane calculator against other crane calculators and spreadsheets that are commonly used in the industry. The developed CPT server and web GUI client for manual crane design is now being adopted by our industrial partner and major offshore handling equipment manufacturer, Seaonics AS, located in Ålesund, Norway [10].

D. Web Graphical User Interface (GUI)

For the sake of simplicity and practicality, the previously developed web GUI is used to interact with the crane calculator via WS/JSON communication. The web GUI can be used to manually adjust the 120 design parameters in the crane calculator by a trial-and-error approach, or automatically by employing the AIPO module backend. The effect of the chosen parameter values on a number of KPIs and other design criteria can then be investigated numerically, with the possibility of exporting the resulting crane design data to text files. The GUI also provides a simple visualization of the crane's main components and its 2D SWL load chart similar to the one shown in Fig. 2. The load chart is updated in real-time when the user modifies either of the design parameters.

E. Matlab Crane Optimisation Client (MCOC)

Traditional manual trial-and-error design optimization is time-consuming and cost-inefficient, since there are more than 120 parameters that must be specified by the crane designer. This large number of parameters makes the space of all possible combinations of parameter values very large and a manual trial-and-error approach will necessarily be cumbersome and running the risk of suboptimal designs. Previously [7-8], we used the Matlab® Global Optimization Toolbox from Mathworks® [11] to implement a crane optimization client (MCOC) that used evolutionary algorithms to find the best configuration of the design parameters. Here, we extend the MCOC with the GWO algorithm.

Based on input from domain experts, four parameters are used as the best set of design parameters, namely, "*boom length*", "*jib length*", "*main cylinder max pressure*", and "*jib cylinder max pressure*". Two KPIs, namely, the maximum safe working load, SWL_{max}, and the total crane weight, W, are used as the components of our objective function for design optimization. Whilst the total crane delivery price is of great concern, we do not currently have price estimates as a function of crane design implemented in the CPT. Nevertheless, the total weight can to some extent be used as a proxy for price. This is because price will correlate to the total crane weight, and one wants to minimize both measures. Moreover, since these cranes are installed on-board vessels, it is desirable to reduce the crane weight to allow 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 lifting capacity of the crane in the entire workspace.

The goal of the crane optimizer is thus to maximize SWL_{max} while simultaneously minimizing W. To achieve this goal, the objective function can be formulated as follows:

$$f = SWL_{max}/W \tag{1}$$

where maximizing f entails maximizing SWL_{max} and minimizing W.

III. Grey Wolf Optimizer (GWO)

This section briefly describes the GWO algorithm applied to the offshore crane design problem we have presented above. GWO is a relatively new optimization algorithm proposed by Mirjalili et al. (2014) [9]. This algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [12]. GWO is similar to other meta-heuristics where the search begins with a population of randomly generated wolves (i.e., candidate solutions).

A. Background

Grey wolves are at the top of the food chain and mostly prefer to live in a pack in groups of 5–12 wolves on average with strict leadership hierarchy. Four types of grey wolves called *alpha*, *beta*, *delta*, and *omega* are employed for simulating the leadership hierarchy, shown in Fig. 4. The alphas are responsible for making decisions about hunting, sleeping place, time to wake, and so on. The alphas' decisions are dictated to the pack through betas. Alphas are not the strongest members of the pack but they are the best in terms of management skills.

The betas are subordinate wolves that help the alpha in decision-making or other pack activities. They are the best candidates to be alphas in case one of the alpha wolves passes away, becomes old, or is retired. Each beta wolf must show respect to the alpha and provide advice, but commands the other lower-level wolves as well. The beta reinforces the alpha's commands throughout the pack and gives feedback to the alpha. The lowest ranking grey wolf is omega. The omega plays the role of scapegoat. Omega wolves always have to

submit to all the other dominant wolves. They are the last wolves that are allowed to eat. Omegas in some cases are used as babysitters of the pack. If a wolf is not an alpha, beta, or omega, he/she is called subordinate, or delta.

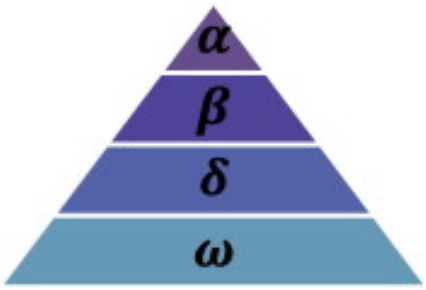


Fig. 4. Hierarchy of grey wolf (dominance decreases from top down).

Delta wolves have to submit to alphas and betas, but they dominate the omega. Scouts, sentinels, elders, hunters, and caretakers belong to this category. Scouts are responsible for watching the boundaries of the territory and warning the pack in case of any danger. Sentinels protect and guarantee the safety of the pack. Elders are the experienced wolves who used to be alpha or beta. Hunters help the alphas and betas when hunting prey and providing food for the pack. Finally, the caretakers are responsible for caring for the weak, ill, and wounded wolves in the pack [9].

In addition to the social hierarchy of wolves, group hunting is another interesting social behavior of grey wolves. The main phases of grey wolf hunting are: *searching for prey*, *encircling prey*, and *attacking prey* [13]. The hunting technique and the social hierarchy of grey wolves are mathematically modeled in order to design GWO to perform optimization.

B. Mathematical modeling of social hierarchy

In order to formulate the social hierarchy of wolves when designing GWO, in this algorithm the population is split into four groups: alpha (α), beta (β), delta (δ), and omega (ω). Over the course of iterations, the first three best solutions are called α, β, and δ, respectively. The rest of the candidate solutions are named as ω. In the GWO algorithm the hunt (optimization) is guided by α, β, and δ. The ω wolves are required to encircle α, β, and δ, so to find better solutions, they must follow these three wolve types.

C. Mathematical modeling of hunting behaviour

The encircle process could be modeled as follows [9]:

$$\begin{aligned} \vec{D} &= \left|\vec{C}.\vec{X}_p(t) - \vec{X}(t)\right|, \\ \vec{X}(t+1) &= \vec{X}(t) - \vec{A}\cdot\vec{D} \end{aligned} \tag{2}$$

where t indicates the current iteration, $\vec{C} = 2\vec{r}_2$, $\vec{A} = 2\vec{a}\cdot\vec{r}_1 - \vec{a}$, $\vec{X}_p$ is the position vector of the prey, $\vec{X}$ is the position vector of a grey wolf, $\vec{\alpha}$ is gradually decreased from 2 to 0, and r_1 and r_2 are random numbers over the range [0,1]. In order to mathematically simulate the hunting behavior of grey wolves, in the GWO algorithm we always assume that by α, β, and δ have better knowledge about the position of the prey (optimum). Therefore, the positions of the first three best solutions (α, β, δ) obtained so far are saved and other wolves (ω) are obliged to reposition with respect to α, β, and δ. The mathematical model of readjusting the positions of ω wolves is presented as follows [9]:

$$\begin{aligned} \vec{D}_\alpha &= \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|, \\ \vec{D}_\beta &= \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right|, \\ \vec{D}_\delta &= \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right|. \end{aligned} \tag{3}$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1.\left(\vec{D}_\alpha\right), \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2.\left(\vec{D}_\beta\right), \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3.\left(\vec{D}_\delta\right). \end{aligned} \tag{4}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{5}$$

where $\vec{X}_\alpha$ is the position of the alpha, $\vec{X}_\beta$ is the position of the beta, $\vec{X}_\delta$ is the position of the delta, $\vec{C}_1, \vec{C}_2, \vec{C}_3, \vec{A}_1, \vec{A}_2$, and $\vec{A}_3$ are all random vectors, $\vec{X}$ is the position of the current solution, and t is the iteration number.

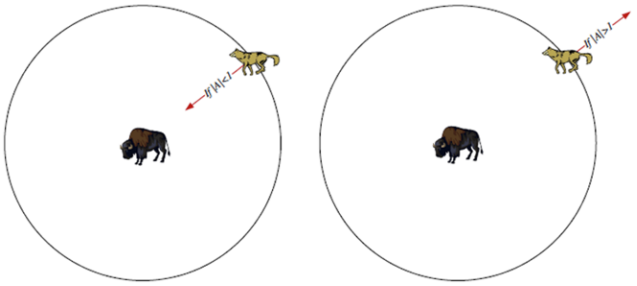


Fig. 5. Exploration versus exploitation periods depending on the value of A in GWO.

```
Initialize the grey wolf population Xi (i=1,2, …,n) and
parameters
Calculate the fitness of population
Find the first three agents Xα, Xβ, Xδ
While (t < Max number of iterations)
     Update the position of the curreent search agent by
     Eq. (5)
     Calculate the fitness of population
     Update Xα, Xβ, Xδ
End While
Return Xα
```

Fig. 6. GWO algorithm [14].

In these formulas, it may be observed that there are two vectors $\vec{A}$ and $\vec{C}$ obliging the GWO algorithm to explore and exploit the search space. With decreasing A, half of the iterations are devoted to exploration ($|A| \geq 1$) and the other half are dedicated to exploitation ($|A| < 1$), as it is shown in Fig. 5. The range of C is $2 \leq C \leq 0$, and the vector C also improves exploration when $C>1$ and the exploitation is emphasized when $C<1$. Note here that A is decreased linearly over the course of the iterations. In contrast, C is generated randomly whose aim is to emphasize exploration/exploitation at any

stage avoiding local optima. The main steps of grey wolf optimizer are given in Fig. 6.

IV. Results and Discussions

The ability of the GWO algorithm presented in this paper to perform product optimization is tested using a real world offshore crane. A real knuckle-boom crane is used as a nominal benchmark against which the optimized crane design could be compared. The nominal crane has been designed, sold, and delivered by Seaonics AS to a company in Baku, Azerbaijan [10]. The crane had a total delivery price of approximately 2.9 million EUR. Two KPIs were chosen as components of an objective function to be optimized:

(i) The maximum safe working load SWL_{max}

(ii) The total crane weight W

The total weight, W, is used as an estimate of the total delivery price of the crane. The goal of the optimization algorithm is to minimize the function given in (1). The crane design consist of 120 variables, as a best set of optimization variables, with 4 variables that greatly affect both SWL_{max} and W are chosen:

(i) The boom length, L_{boom}

(ii) The jib length, L_{jib}

(iii) The maximum pressure of the boom cylinder, P_{boom}

(iv) The maximum pressure of the jib cylinder, P_{jib}

All other design parameters were identical to those of the nominal crane. The parameter values were constrained to a range with minimum and maximum allowable limits by the manufacturer and nominal values, as it is shown in Table I.

TABLE I. Allowable Values of Decision Variables

Parameter	Lower bound	Upper bound	Nominal
L_{boom} (mm)	12000	26000	15800
L_{jib} (mm)	6000	16000	10300
P_{boom} (ton)	100	400	315
P_{jib} (ton)	50	300	215

The problem can then mathematically formulated as follows:

$$\text{Minimize} \quad f = SWL_{max}/W$$

$$\text{Subjected to} \quad 12000 \le L_{boom} \le 26000$$
$$6000 \le L_{jib} \le 16000$$
$$100 \le P_{boom} \le 400$$
$$50 \le P_{jib} \le 300$$

The algorithm is implemented in Matlab® version R2015a running on Mac OS X Version: 10.11.6 executed on 2.2 GHz Intel Core i7 Processor with a memory of 16 GB. The algorithm was independently run 20 times for 5, 50, 100, and 200 iterations for 4 search agents to guarantee stability and statistical significance of the results. Coefficient a is linearly decreased from 2 to 0 as a function of maximum iteration number. Table II shows the *average*, *best*, *worst*, and *standard deviation* values of the fitness function for GWO algorithm for 20 independent runs. Fig. 7 shows the boxplot of the resultant fitness values over 20 independent runs. Convergence curves of GWO for the best results over 20 independent runs are shown in Fig. 8.

TABLE II. GWO Fitness Values for 20 Independent Runs.

Iterations	Best	Worst	Mean	Std.	t_{mean} (m)
5	3.2674	1.8349	2.8534	0.4225	1.3908
50	3.2674	3.2123	3.2424	0.0273	18.0099
100	3.2674	3.2123	3.2428	0.0279	14.5607
200	3.2674	3.2123	3.2593	0.0198	40.2536

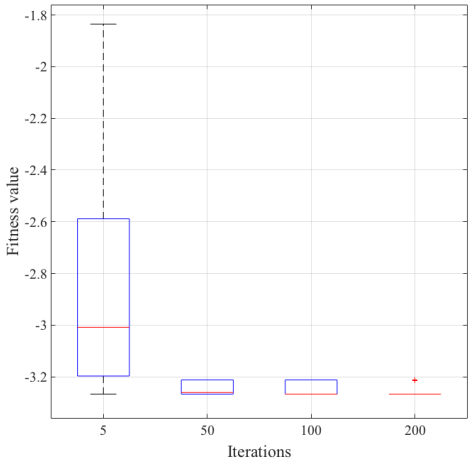
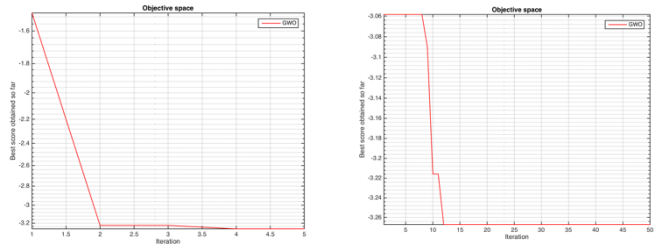


Fig. 7. Boxplot of fitness values over 20 independent runs for 5, 50, 100 and 200 iterations.



(a) 5 iterations. (b) 50 iterations.

(c) 100 iterations. (d) 200 iterations.

Fig. 8. Convergence curves for the best results over 20 independent runs.

The performance of GWO algorithm in terms of fitness value and convergence time compared to that of very well known optimization algorithms in literature such as GA, PSO, and SA for the same benchmark problem applied to the objective function given by Eq. (1) [8] is shown in Table III. Results from Table III show that the proposed GWO algorithm outperforms other algorithms in terms of stability, achieved fitness value and convergence time.

TABLE III. GWO FITNESS VALUES FOR 20 INDEPENDENT RUNS.

Algorithm	Best fitness value	Best convergence time (m)
GWO	3.27	1.3908
PSO	1.30	90.00
GA	1.26	98.40
SA	1.00	488.40

V. CONCLUSIONS

Advantages of GWO over other algorithms are as follows; easy to implement due to its simple structure, less storage and computational requirements, faster convergence due to continuous reduction of search space and fewer decision variables (i.e., α, β, and δ); its ability in avoiding local minima; and having only two control parameters (i.e., a and C) to tune the algorithm performance and hence better stability and robustness.

In this paper, two competing individual objective functions are combined into a single fitness function, which might imply that improving one objective deteriorates the other one, and vice versa. As a future work, multi-objective GWO is proposed to simultaneously optimize each objective function individually.

ACKNOWLEDGMENT

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

[1] G. Frank, D. Entner, T. Prante, V. Khachatouri, and M. Schwarz, "Towards a Generic Framework of Engineering Design Automation for Creating Complex CAD Models, " International Journal on Advances in Systems and Measurements, vol. 7, no 1 & 2, 2014, pp. 179–192.

[2] T.S. Mujber, T. Szecsi, and M.S.J. Hashmi, "Virtual reality applications in manufacturing process simulation," Journal of Materials Processing Technology, vol. 155, 2004, pp. 1834–1838.

[3] S. Gowda, S. Jayaram, and U., Jayaram, "Architectures for internet-based collaborative virtual prototyping," In: Proceedings of the 1999 ASME Design Technical Conference and Computers in Engineering Conference, DETC99/CIE-9029, Las Vegas, Nevada, 1999.

[4] G.G. Wang, "Definition and review of virtual prototyping," Journal of Computing and Information Science in engineering, 2(3), 2002, pp. 232-236.

[5] R.T. Bye, O.L. Osen, and B.S. Pedersen, "A computer-automated design tool for intelligent virtual prototyping of offshore cranes," In: Proceedings of the 29th European Conference on Modeling and Simulation (ECMS'15), Albena, Bulgaria, 2015, pp. 147–156.

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

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

[8] I.A. Hameed, R.T. Bye, and O.L. Osen, "A Comparison between Optimization Algorithms Applied to Offshore Crane Design using an Online Crane Prototyping Tool," In: Proceedings of the 2nd International Conference on Advanced Intelligent Systems and Informatics (AISI2016), Cairo, Egypt, 2016, Advances in Intelligent Systems and Computing, Springer.

[9] S. Mirjalili, S.M. Mirjalili, and A. Lewis, "A grey wolf optimizer," Advances in Engineering Software, vol. 69, 2014, pp. 46–61.

[10] Offshore Energy Today, http://www.offshoreenergytoday.com/

[11] Mathworks, Inc.: MATLAB Global Optimization Toolbox. The Mathworks, Inc., Natick, Massachusetts, 2015, http://mathworks.com/products/global-optimization/.

[12] L.D Mech, "Alpha status, dominance, and division of labor in wolf packs," *Canadian Journal of Zoology*, vol. 77, no. 8, 1999, pp. 1196–1203.

[13] C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural Processes*, vol. 88, 2011, pp. 192–197.

[14] S. Zhang, and Y. Zhou, "Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis," Discrete Dynamics in Nature and Society, Vol. 2015, 2015, Article ID 481360, pp. 1–17.