Optimization of the box-girder of overhead crane with constrained new bat algorithm

Optimisation de poutre caisson de pont roulant avec un nouvel algorithme de chauve souris sous contrainte

Asma Chakri*, Rabia Khelif & Mohamed Benouaret

Industrial Mechanics Laboratory, Department of Mechanical Engineering, University Badji Mokhtar of Annaba (UBMA), BP12-23000, Annaba, Algeria.

Soumis le : 27/10/2015 *Révisé le : 15/05/2017* *Accepté le : 23/06/2017*

ملخص

إن التصميم الأمثل لعارضة رافعة بأقل وزن ممكن من شأنه أن يخفض من تكاليف التصنيع والتشغيل . وعلى غرار استعمال عارضة جوفاء بدعم ارتكاز بسيطة، وهذا التصميم مقيد بمساحة مقطع العارضة كدالة موضوعية وكذلك بالقيود المفروضة للصفائح المجهدة والكلل والتشوه المقبول الناشئ عن الانبعاج، يمكن حل المسألة المحصل عليها بالخوارزمية الجديدة □- المقيدة. بحيث تم إدماج أربع تعديلات على الخوارزمية الجديدة لزيادة كفاءتها ورفع مستوى □ للمقارنة لتمكين الخوارزمية الأخذ بعين الاعتبار الاجهادات على العارضة. لقد برهنت الخوارزمية الجديدة عن فعالية وقوة ونجاعة، وكذلك الأبعاد المثلى لتجويف العارضة الملحومة التي تم الحصول عليها هي أفضل من الموجودة في الدراسات النظرية واستعمال الفولاذ العالي المقاومة يؤدي إلى اقتصاد في الوزن يصل إلى 30%.

كلمات مفتاحيه:

رافعة علوية، عارضة جوفاء، خوارزمية جديدة ، مستوى المقارنة ، فولاذ عالي المقاومة

Abstract

An optimal design for minimum weight crane girder can reduce the manufacturing and operating costs. The box-girder is modeled as a simply supported beam, and the constrained optimization problem was formulated with cross-section area of the box-girder as objective function, and restrictions on plates' stress, fatigue, buckling and allowable deflection. The resulted problem is solved with □ □constrained new bat algorithm. Four modifications have been embedded to the standard bat algorithm to increase its performances, and the □ □level of comparison was introduced so that the new bat algorithm can handle constraints. Results show that the □ □constrained new bat algorithm is efficient, robust and reliable and the obtained optimal crane dimensions are better than those they exist in the literature. The use of higher strength steel leads to a mass saving up to 30%.

Keywords: *Overhead crane - Box-girder - New bat algorithm -□ □ □ level of comparison- Higher strength steel.*

Résumé

Une conception optimale pour un poids minimal de la poutre principale d'un pont roulant peut réduire les coûts de fabrication et de fonctionnement. La poutre en caisson est modélisée comme une poutre simplement appuyée et le problème d'optimisation sous contrainte est formulé avec la considération de l'aire de la section transversale du caisson comme fonction objective, et des restrictions sur la contrainte des plaques, la fatigue, le flambement et la déformation admissible. Le problème résultant est résolu par le nouvel algorithme de chauve-souris sous contrainte□□. Quatre modifications ont été intégrées à l'algorithme de chauve-souris standard pour augmenter ses performances et le niveau de comparaison e est introduit afin de permettre au nouvel algorithme de chauve-souris de prendre en considération les contraintes. Les résultats montrent que le nouvel algorithme de chauve-souris est efficace, robuste et fiable ainsi que les dimensions optimales du caisson soudé obtenues sont meilleures que celles qui existent dans la littérature. L'utilisation des aciers à haute résistance peut conduire à des économies en poids de plus de 30%.

Mots clés: *Pont roulant - poutre en caisson - Le nouvel algorithme de chauve-souris - Niveau de comparions □ - acier à haut résistance*.

* *Corresponding author : chakri.as623@gmail.com*

INTRODUCTION

For moving and handling materials, cranes are widely used in modern industries. The archaeological record shows that the cranes were invented by the ancient Greeks 500 years BC [1]. The first modern crane was designed by William George Armstrong in 1838 [2] using water powered hydraulic jigger. In these days, three main types of cranes that are used in industries, fixed cranes such as the tower crane, mobile cranes such as truck-mounted crane and the overhead cranes, also known as a bridge crane.
Electrical overhead traveling cranes (EOT crane) are very popular due to their operation simplicity and the free floor space. Since the main girders are the principle structure that maintains most of the load, careful attention should be devoted to their design. They exist several types of girder, such as pipe girder, plate girder, truss girder and the typical box-girder which has many advantages such as the reduced overall weight, the high torsional rigidity and low manufacturing time [3].
Several studies have been carried out on the numerical optimization of the crane girder. Rao [4] formulated the crane girder problem as a minimum weight design problem with restrictions on the maximum allowable deflection, stress, overall stability and rigidity as well as the shock absorbing capacity during accidental collision. The problem was solved using the interior penalty function method. Farkas [5] used the combinatorial discrete backtrack programming method to solve the crane girder optimization problem. He showed that the use of higher strength steel can produce lighter girder. In addition the weight minimization, Jarmai [6] add other criterion to be optimized such as fabrication cost, welding and painting costs. The resulting multi-objective problem was solved with a decision support system which contains several optimization algorithms such as the min-max method. Pavlovic et *al*. [7] optimized the box-section of the main girder with the rail placed at the middle of the upper flange, with consideration of the lateral stability and local stability, using the Lagrange multipliers' method. With the same method, Savkovic et al. [8] did the same work but the rail is placed above the web plate.
Recently, some important optimization algorithms were inspired by observing animal or insect groups carrying out a collaborative work, such as particle swarm optimization method (PSO) [9] which was inspired from the social and cognitive behavior of birds or fishes. Ant colony optimization algorithm (ACO) is based on the collaborative work of ants when they look for food [10]. The cuckoo search algorithm (CS) introduced by Yang and Deb [11] was inspired from the parasitic breeding behavior of cuckoo birds.
The bat algorithm (BA) proposed by Yang [12] was inspired from the echolocation behavior of micro-bats. When flying, bats emit a pulse of sound to the environment. The sound waves travel across the air until they hit an object or a surface and come back as an echo. Bats hear the echo, and by analyzing the time delay they can build a precise mental image of their surrounding and determine precisely the distance, shapes and objects' direction. The capability of echolocation of micro-bats is fascinating, as these bats can find their prey and discriminate different types of insects even in complete darkness [12]. The earlier studies showed that BA can solve constrained and unconstrained optimization problems with much more efficiency and robustness compared to the genetic algorithm (GA) and PSO [12-14].
Despite the fact that BA is very powerful algorithm and can produce robust solutions on low dimensional problems, its performance diminishes significantly when the problem dimension increases due to the phenomenon of premature convergence problem. In this study, A new modifications and idealized rules have been embedded to BA to improve its performance and exploitation / exploration abilities. The proposed new bat algorithm (NBA) has been tested on several benchmark problems and compared with varieties of swarm and evolutionary algorithms.

Algorithm 1. The standard bat algorithm.

1. Define the objective function
2. Initialize the bat population $-Lb_i \le x_i \le Ub_i$ (i=1,2,..,n) and v_i
3. Define frequencies f_i at x_i
4. Initialize pulse rates r_i and loudness A_i
5. *While* ($t \le t_{\max}$)
6. $f_i = f_{\min} + (f_{\max} - f_{\min})rand$ %Adjust frequency

7. $v_i^{t+1} = v_i^t + \left(x^* - x_i^t\right) f_i$ %Update velocities
8. $x_i^{t+1} = x_i^t + v_i^{t+1}$ %Update locations/solutions
9. *if* ($rand > r_i$)
10. $x_{new} = x_{old} + \varepsilon < A^{t+1} >$ %Generate a local solution around % the selected best solution
11. *end if*
12. *if* ($rand < A_i$ & $F(x_i) < F(x^*)$)
13. Accept the new solutions
14. $r_i^{t+1} = r_i^0 \left[1 - \exp(-\gamma t)\right]$ %Increase r_i
15. $A_i^{t+1} = \alpha A_i^t$ %Reduce A_i
16. *end if*
17. Rank the bats and find the current best x^*
28. *end while*
29. Results processing

The ε–level of comparison is introduced the NBA as a constraint handling technique so that the ε–NBA can solve constrained optimization problem. We will use the ε–NBA to solve the crane girder optimization problem.

In the next section we present a brief overview of the standard bat algorithm. In section three, we introduce the new bat algorithm. The ε–level of comparison is described in section four. the crane girder optimization problem is detailed in section five, and finally results and discussions are in section six.

2. THE STANDARD BAT ALGORITHM

The standard Bat algorithm was inspired from the echolocation process of bats. By observing the behavior and characteristics of the micro-bats, Yang [12] proposed the standard BA in accordance to three major characteristics of the echolocation process of the micro-bats. The used idealized rules in BA are:

a) All bats use echolocation to sense distance and they also know the difference between food/ prey and barriers in some magical way [12].
b) Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and the rate of pulse emission r depending on the proximity of the target [12].
c) Loudness varies from a large positive A_0 to a minimum constant value A_{min} [12].

Algorithm 1 presents the pseudocode of the standard bat algorithm. For each bat (i), its position (x_i) and velocity (v_i) in a N-dimensional search space should be defined. x_i and v_i should be subsequently updated during the iterations. The rules for updating the position and velocities of a virtual bat (i) are presented in Lines 6-8, where $rand \in [0, 1]$ is a random vector drawn from a uniform distribution.

Here x^* is the current global best location (solution) which is located after comparing all solution among all the n bats. A new solution for each bat is generated locally using random walk presented in Line 10, where $\varepsilon \in [-1,1]$ is a random number while $< A_i^{t+1} >$ is the average loudness of all the bats at this time step.

The loudness A_i and the rate of pulses emission r_i are updated as the iteration proceed. The loudness decrease and the pulse rate increase as the bat get closer to its prey. The equation for updating the pulse rate and the loudness are defined in Line 14 and 15 respectively, where $0 < \alpha < 1$ and $\gamma > 0$ are constants. As $t \rightarrow t_{max}$, we have $A_i^t \rightarrow 0$ and $r_i^t \rightarrow r_i^0$. The initial loudness A_0 can typically be $A_0 \in [1, 2]$, while the initial emission rate $r^0 \in [0, 1]$.

3. THE NEW BAT ALGORITHM

The new bat algorithm has the same flowchart as the standard bat algorithm. Four modifications have been introduced to the standard BA with aim to enhance its exploitation and exploration abilities.

3.1. The 1st modification

We suppose that the bats can know their surroundings. Therefore, we introduction the position of randomly selected bat to the formulas of the bats movements as its is shown in Algorithm 2 Line 6-12 where x_k^t is the location of randomly selected bat ($k \neq i$) and x^* is the best solution. $F(.)$ is fitness function. f_1 and f_2 are the frequencies. *rand*1 and *rand*2 are two random vectors drawn from a uniform distribution between 0 and 1.

3.2. The 2nd modification

The second modification concerns the local search part. We permit to the bats to move from their current position to a new random position with equations described in Algorithm 2 Line 14 where $<A^t>$ is the average loudness of all bats and $\varepsilon \in [-1,1]$ is a random vector. w_i is a parameter applied to reduce the space search while the iterative process proceed. It starts from a large value about a quarter of the space length and it decrease to around 1% of the quarter of the space length (Algorithm 2 Line 15). w_{i0} and $w_{i\infty}$ are the initial and final value that w_i can take over the iteration procedure. In general we set w_0 and w_∞ as follow:

$$w_{i0} = (Ub_i - Lb_i)/4 \qquad (1)$$

$$w_{i\infty} = w_{i0}/100 \qquad (2)$$

t is the current iteration and $t_{\max}$ is the maximum number of iterations. Ub_i and Lb_i are the upper and lower bounds.

3.3. The 3rd modification

Equations proposed by Yang [12] (Algorithm 1, Line 14 &15) to update the pulse rate and loudness reach their final value during the iterative process very quickly, thus reducing the possibility of the auto-switch from the random walk to the local search due to a higher pulse rate, and the acceptance of a new solution (low loudness). Therefore we propose to use these monotonically increasing, decreasing, pulse rate and loudness (Algorithm 2, Line 19 & 20), where the index 0 and ∞ stand for the initial and final value.

3.4. The 4th modification

The final improvement we made to the original BA is to allow the bats to update the pulse rate and loudness, and to accept a new solution if their movement produces a solution better than the old one instead of the global best solution as it's in the original algorithm. This modification was also suggested by [15]. In addition, the acceptance of a new solution requires the fulfilling of two conditions. First, the solution has to produce a fitness value lower than the actual. Second, a randomly generated number has to be lower than

Algorithm 2. The new bat algorithm.

1.	Define the objective function	
2.	Initialize the bat population $-Lb_i \leq x_i \leq Ub_i$ (i=1,2,..,n)	
3.	Evaluate fitness $F_i(x_i)$	
4.	Initialize pulse rates r_i loudness A_i and w_i	
5.	*While* ($t \leq t_{\max}$)	
6.	Select a random bat ($k \neq i$)	
7.	$\begin{cases} f_1 = f_{\min} + (f_{\max} - f_{\min})rand1 \\ f_2 = f_{\min} + (f_{\max} - f_{\min})rand2 \end{cases}$	%Generate frequencies
8.	*If* $F(x_k^t) < F(x_i^t)$	%Update locations/solutions
9.	$x_i^{t+1} = x_i^t + (x^* - x_i^t)f_1 + (x_k^t - x_i^t)f_2$	

else
$x_i^{t+1} = x_i^t + (x^* - x_i^t) f_1$
endif
if (*rand* > r_i)
$x_i^{t+1} = x_i^{t+1} + < A^t > \varepsilon \, w_i^t$ %Generate a local solution around the %selected solution
$w_i^t = \left(\dfrac{w_{i0} - w_{i\infty}}{1 - t_{\max}} \right) \left(t - t_{\max} \right) + w_{i\infty}$ %Update w_i
end if
if (*rand* < A_i & $F(x_i^{t+1}) < F(x_i^t)$)
Accept the new solutions
$r^t = \left(\dfrac{r_0 - r_\infty}{1 - t_{\max}} \right) \left(t - t_{\max} \right) + r_\infty$ % Increase r_i
$A^t = \left(\dfrac{A_0 - A_\infty}{1 - t_{\max}} \right) \left(t - t_{\max} \right) + A_\infty$ %Reduce A_i Eq. (14)
end if
if ($F(x_i^{t+1}) < F(x^*)$)
Update the best solution x^*
end
end while
Results processing

The current corresponding loudness. There exists a probability that the movement of the bat produces a solution better even to the global best solution and cannot be accepted because the randomly generated number is higher than the current loudness, especially at the end of the iterative process where the value of loudness is lower. Therefore, we allow to the algorithm to update the global best position whenever the bat's walk produce a solution with better fitness value even if it was not accepted to update the bat's position. The pseudo-code of the new bat algorithm is illustrated in Algorithm 2.

4. CONSTRAINT HANDLING METHOD

Consider the following constrained optimization problem:

$$\begin{aligned} & \textit{Minimize } F(x) \\ & \textit{subject to } g_j(x) \le 0, \; j = 1, ..., q \\ & \qquad h_j(x) = 0, \; j = q+1, ..., m, \\ & \qquad l_i \le x_i \le u_i, \; i = 1, ..., n \end{aligned} \tag{3}$$

where $F(x)$ is an objective function (the fitness function), $h(x)$ and $g(x)$ are the equality and inequality constraints, $x = (x_1, x_2, \ldots, x_n)$ is an n dimensional vector of decision variables. u_i and l_i are the upper and lower bounds of x_i, respectively. The upper and lower bound define the search space Σ, while the equality and inequality constraints define the feasible region Φ.

To solve the upper constrained optimization problem, the ε-constraint method[16] was adopted to handle the equality and inequality constraint. The main idea of this method is to define an ε–level comparison as an order relation on the set of ($F(x)$, $\phi(x)$) where $\phi(x)$ is the constraint violation function which is defined as the following:

$$\phi(x) = \sum_{j=1}^{q} \max\{0, g_j(x)\}p + \sum_{j=q+1}^{m} \left|h_j(x)\right|^p \quad (4)$$

where p is a positive number. $\phi(x)$ indicates by how much a point x violates the constraint. The constraint violation function $\phi(x)$ has the following property:

$$\begin{cases} \phi(x) = 0 \ (x \in \Phi) \\ \phi(x) > 0 \ (x \notin \Phi) \end{cases} (5)$$

The e level comparisons are defined by a lexicographic order in which $\phi(x)$ precedes $F(x)$, because the feasibility of x is more important the minimization of $F(x)$[16]

Consider two point x_1 and x_2 with their corresponding fitness and constraint violations values F_1,F_2 and $\phi_1(\phi_2)$. Then, for any ε ($\varepsilon \geq 0$), the ε–level comparisons $<_\varepsilon$ between (F_1,ϕ_1) and (F_2,ϕ_2) is defined as follows:

$$(F_1,\phi_1) <_\varepsilon (F_2,\phi_1) \Leftrightarrow \begin{cases} F_1 < F_2, \text{ if } \phi_1,\phi_2 < \varepsilon \\ F_1 < F_2, \text{ if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, \text{ otherwise} \end{cases} \quad (6)$$

The ε–level is updated until the iteration counter t reaches the control iteration T_c. After the iteration counter exceeds T_c, the ε level is set to zero to obtain a solution with no constraint violation.

$$\varepsilon^0 = \phi(x_\theta)$$

$$\varepsilon^t = \begin{cases} \varepsilon^0 \left(1-(t/T_c)\right)^{cp}, & 0 < t < T_c \\ 0, \ t \geq T_c \end{cases} (7)$$

where x_θ is the top θ–th individual and $cp\in[2,10]$. In this study $cp = 5$.

Therefore the ε–constraint new bat algorithm (ε–NBA) is built with the replacement of the ordinal comparison with the ε–level comparisons which is introduced in Line 17 and 22 of Algorithm 2 as follows respectively:

17.*if* ($rand < A_i$) &

$(F(x_i^{t+1}),\phi(x_i^{t+1})) <_\varepsilon (F(x_i^t),\phi(x_i^t))$ (8)

22.*if* $(F(x_i^{t+1}),\phi(x_i^{t+1})) <_\varepsilon (F(x^*),\phi(x^*))$ (9)

5. MATHEMATICAL FORMULATION OF THE CRANE GIRDER OPTIMIZATION PROBLEM

5.1 Objective function

The cross-sectional area is taken as the objective function to be minimized (Fig. 1):

$$A = h(t_{w1} + t_{w2}) + 2b_1 t_f \quad (10)$$

5.2 Constraint functions

The following constraints are in accordance with the BS2573[17], BS5400[18], and the works of Farkas[5] and Jarmai[6].

5.2.1. Constraint on the static stress in the lower flange

The normalized constraint on the static stress in the lower flange at mid-span due to biaxial bending is:

$$g_1 = 1 - \left(\frac{M_x}{\alpha_d P_s W_x} + \frac{M_y}{\alpha_d P_s W_y} \right) \geq 0 \quad (11)$$

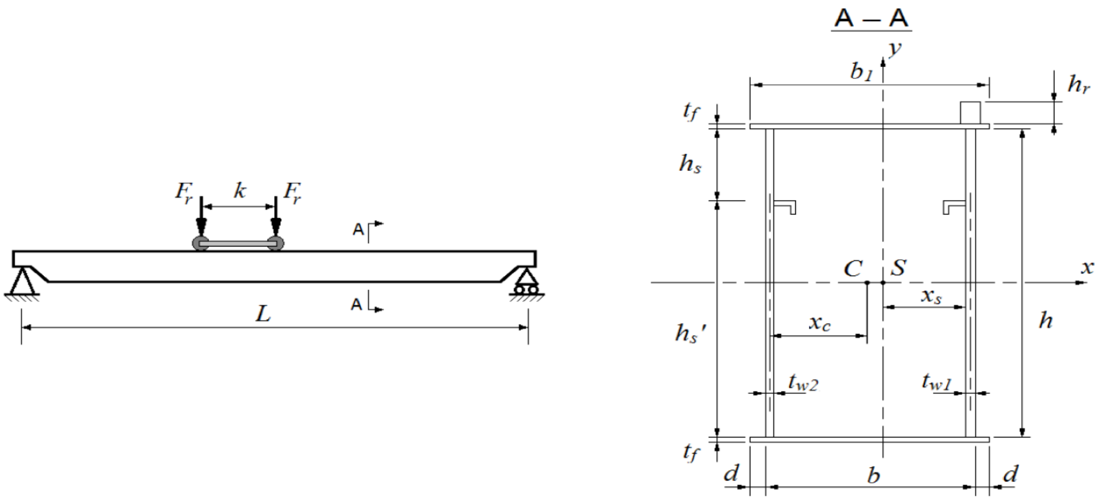

Figure 1. Crane girder configuration and dimensions of the cross-section.

where M_x and M_y are the bending moment, W_x and W_y are the section moduli, Y_s is the yield stress and α_d is the duty factor. P_s is the allowable stress which is P_s=0.59Y_s where Y_s is the yield stress. The approximate formulas for moments of inertia are:

$$I_x = \frac{h^3\left(t_{w1}+t_{w2}\right)}{12} + \frac{b_1 t_f}{2}\left(h+t_f\right)^2 \qquad (12)$$

$$I_y = h t_{w1} x_s^2 + h t_{w2}\left(b-x_s\right)^2 + \frac{b_1^3 t_f}{6} + \frac{b_1 t_f}{2}\left(b-2x_s\right)^2 \qquad (13)$$

where

$$x_s = \frac{1}{2}\left(b - t_{w1} + hb\left(\frac{t_{w1}-t_{w2}}{A}\right)\right) \qquad (14)$$

The section moduli are:

$$W_x = \frac{2I_x}{\left(h+t_f\right)};\ \ W_y = \frac{I_y}{\left(b-x_s+d\right)} \qquad (15)$$

and the bending moment:

$$M_x = \frac{L^2}{8}\left(1\cdot 05A\rho + p_r + p_s\right)g + \frac{F}{2L}\left(L-\frac{k}{2}\right)^2 \qquad (16)$$

$$M_y = 0\cdot 15\left(\frac{L^2}{8}\left(1\cdot 05A\rho + p_r + p_s\right)g + \frac{G_t}{8L}\left(L-\frac{k}{2}\right)^2\right) \qquad (17)$$

where $F = \left(\psi_d H + G_t\right)/4$ is the wheel load, ψ_d is the impact factor. The factor of 1.05 expresses the mass of diaphragms. p_r+p_s is the linear weight of the rail and the sidewalk

5.2.2. Constraint on fatigue stress

the restriction on the

$$g_2 = 1 - \left(\frac{M_{xf}}{P_{ft}W_x} + \frac{M_y}{P_{ft}W_y}\right) \geq 0 \qquad (18)$$

M_{xf} is the moment due to fatigue expressed as follow:

$$M_{xf} = \frac{L^2}{8}\left(1.05A\rho + p_r + p_s\right)g + \frac{K_p \psi_d H + G_f}{8L}\left(L - \frac{h}{2}\right)^2 \quad (19)$$

where K_p is the spectrum factor.

P_{ft} is the permissible fatigue stress. The approximate formulas to calculate P_{ft} are presented in section 6.2.

5.2.3. Constraint on local flange buckling

The constraint on the local buckling of the upper flange is as follows:

$$g_3 = 1 - \left(\frac{\sigma_{lf}}{Y_s K_{lf}} + \left(\frac{\sigma_{bf}}{Y_s K_{bf}}\right)^2\right) \geq 0 \qquad (20)$$

where

$$\sigma_{lf} = \frac{M_x}{W_x};\ \sigma_{bf} = \frac{M_y}{W_y} \qquad (21)$$

The calculation of the coefficients K_{1f} and K_{bf} depends on the slenderness ratio of the upper flange:

$$\lambda_f = \frac{b}{t_f}\sqrt{\frac{P_s}{355}} \qquad (22)$$

$$\left|\begin{array}{l} \text{if } \lambda_f \leq 24 \Rightarrow K_{1f} = 1 \\ \text{if } 24 < \lambda_f \leq 47 \Rightarrow K_{1f} = \left(\dfrac{24}{\lambda_f}\right)^{0.75} \\ \text{if } 47 < \lambda_f \leq 130 \Rightarrow K_{1f} = \left(\dfrac{26}{\lambda_f}\right)^{0.85} \\ \text{if } 130 < \lambda_f \leq 300 \Rightarrow K_{1f} = 0.274 - \dfrac{\lambda_f}{7000} \end{array}\right. \qquad (23)$$

$$K_{bf} = 1.3 - 0.0027\lambda_f \qquad (24)$$

5.2.4. Constraint on the local buckling of the main web

The local buckling constraint of the main web is:

$$g_4 = 1 - \left(\sqrt{\left(\frac{((h-h_s)/h)\sigma_{lf} + \sigma_{bf}}{Y_s K_{1w}}\right)^2 + \left(\frac{\sigma_{cw}}{Y_s K_{2w}}\right)^2 + \left(\frac{(h_s/h)\sigma_{lf}}{Y_s K_{bw}}\right)^2 + 3\left(\frac{\tau}{Y_s K_{qw}}\right)^2}\right) \geq 0 \qquad (25)$$

The approximate formula for the compressive stress may be written as:

$$\sigma_{cw} = \frac{F}{t_{w1} a_w} \qquad (26)$$

$$a_w = 50 + 2\left(h_r + t_f - 5\right) \qquad (27)$$

The superposition of shear stresses is:

$$\tau = \tau_Q + \tau_t = \frac{FS_x}{I_x\left(t_{w1}+t_{w2}\right)} + \frac{\left(b-x_c\right)F}{2A_k t_{w1}} \qquad (28)$$

where

$$S_x = \frac{bt_f h}{4} \qquad (29)$$

$$A_k = h \cdot b \qquad (30)$$

$$h_w = 1,9\sqrt{\frac{a_w h}{5K_w}} \qquad (31)$$

$$K_w = \left(3,4 + \frac{2,2h}{5a}\right)\left(0,4 + \frac{a_w}{2a_d}\right) \qquad (32)$$

a is the distance between diaphragms.

The slenderness ratio of the upper part of web for local compression is as follows:

$$\lambda_e = \frac{h_w}{t_{w1}}\sqrt{\frac{R_{yw}}{355}} \qquad (33)$$

Therefore, the coefficient K_{2w} is calculated as the following:

$$\left|\begin{array}{l} \text{if } \ \lambda_e \le 24 \Rightarrow K_{2w} = \left(\dfrac{24}{\lambda_e}\right)^{0.5} \\ \text{if } \ 43 < \lambda_e \le 59 \Rightarrow K_{2w} = \left(\dfrac{28}{\lambda_e}\right)^{0.68} \\ \text{if } \ 59 < \lambda_e \le 90 \Rightarrow K_{2w} = \left(\dfrac{30}{\lambda_e}\right)^{0.75} \\ \text{if } \ 90 < \lambda_e \le 130 \Rightarrow K_{2w} = \left(\dfrac{36}{\lambda_e}\right)^{0.9} \\ \text{if } \ 130 < \lambda_e \le 200 \Rightarrow K_{2w} = 0.38 - \dfrac{\lambda_e}{2000} \end{array}\right. \qquad (34)$$

The slenderness of the upper part of web for bending is as follows:

$$\lambda_w = \frac{0.2h}{t_{w1}}\sqrt{\frac{R_{yw}}{355}} \qquad (35)$$

The coefficients K_{bw}, K_{1w} and K_q are:

$$K_{bw} = 1.3 - 0.0027\lambda_w \qquad (36)$$

$$\begin{cases} \text{if} \quad \lambda_w \le 24 \Rightarrow K_{1w} = 1 \\ \text{if} \quad 24 < \lambda_w \le 47 \Rightarrow K_{1w} = \left(\dfrac{24}{\lambda_w}\right)^{0.75} \\ \text{if} \quad 47 < \lambda_w \le 130 \Rightarrow K_{1w} = \left(\dfrac{26}{\lambda_w}\right)^{0.85} \\ \text{if} \quad 30 < \lambda_w \le 300 \Rightarrow K_{1w} = 0.274 - \dfrac{\lambda_w}{7000} \end{cases} \qquad (37)$$

and

$$\begin{cases} \text{if} \ \ \lambda_w \le 40 \Rightarrow K_q = 1 \\ \text{if} \ \ 40 < \lambda_w \le 112 \Rightarrow K_q = 1 - 0.385\left(\dfrac{\lambda_w - 40}{60}\right)^{0.743} \\ \text{if} \ \ 112 < \lambda_w \le 200 \Rightarrow K_q = 1 - 0.660\left(\dfrac{\lambda_w - 40}{160}\right)^{0.505} \\ \text{if} \ \ 200 < \lambda_w \le 300 \Rightarrow K_q = 0.34 - 0.07\left(\dfrac{\lambda_w - 200}{100}\right)^{0.8} \end{cases} \qquad (38)$$

The ratio $\phi = a_d / h$ is approximately 2.

5.2.5. Constraint on the local buckling of the secondary web

Limit state on the local buckling of the secondary web

$$g_5 = 1 - \left(\left(\frac{((h-h_s)/h)\sigma_{lf} + \sigma_{bf}}{Y_s K_{1w}} \right) + \left(\frac{(h_s/h)\sigma_{lf}}{Y_s K_{bw}} \right)^2 + 3\left(\frac{\tau}{Y_s K_{qw}} \right)^2 \right) \ge 0 \qquad (39)$$

To evaluate g_5, we use the same equations and coefficients used in g_4 calculation (Eqs. (26-38)), but we have to use t_{w2} instead of t_{w1} and we neglect the local compression, so σ_{cw}=0.

5.2.6. Constraint on static deflection

Restriction on the static deflection due to the wheel loads:

$$g_6 = 1 - \left(\frac{H\left(L-k\right)\left(3L^2 - \left(L-k\right)^2\right)}{192 E I_x w_P} \right) \ge 0 \qquad (40)$$

where w_p is the permissible deflection.

6. RESULTS AND DISCUSSIONS

6.1. Minimization of benchmark functions

In the this section, we examine our algorithm (NBA) in running benchmark functions and comparing the results with those obtained with some standard algorithms, namely, particle swarm optimization (PSO)[9], genetic algorithm (GA)[19], differential evolution (DE)[20], harmony search (HS)[21] and in addition to the standard bat algorithm (BA)[12]. The considered benchmark functions are:

Spherical function

$$\begin{cases} F_1 = \sum_{i=1}^{d} x_i^2 \\ -100 \le x_i \le 100 \end{cases} \tag{41}$$

Griewank's function

$$\begin{cases} F_2 = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ -600 \le x_i \le 600 \end{cases} \tag{42}$$

Rastrigin's function

$$\begin{cases} F_3 = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i) \right] \\ -5.12 \le x_i \le 5.12 \end{cases} \tag{43}$$

Ackley's function

$$\begin{cases} F_4 = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) \\ \quad - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) \\ \quad + 20 + \exp(1) \\ -32 \le x_i \le 32 \end{cases} \tag{44}$$

The optimization aim for all these test function is to minimize the outcome. The parameters settings of each algorithm are:

- **NBA:** An extensive analysis was performed to carryout parameter settings of NBA, for best practice we recommend the following settings $r_0 = 0.1$, $r_\infty = 0.7$, $A_0 = 0.9$, $A_\infty = 0.6$, $f_{min} = 0$ and $f_{max} = 2$.
- **BA:** The standard bat algorithm was implemented as it is described in [12] with $r_0 = 0.1$, $A_0 = 0.9$, $\alpha = \gamma = 0.9$, $f_{min} = 0$ and $f_{max} = 2$.
- **PSO:** A classical particle swarm optimization model has been considered [9]. The parameters setting are $c_1 = 1.5$, $c_2 = 1.2$ and the inertia coefficient w is a monotonically decreasing function from 0.9 to 0.4.

Table 1. Comparison between algorithm on benchmark functions

Function	NBA	BA	PSO	HS	GA	DE
Sphere	**2.256E-01**	4.920E+04	2.852E+03	9.618E+03	1.678E+03	4.411E+01
Griewank	**1.405E-01**	5.816E+02	7.481E+01	8.040E+01	1.900E+01	2.303E-01
Rastrigin	**1.193E+02**	3.086E+02	2.599E+02	1.580E+02	5.746E+01	1.551E+02
Ackley	**3.191E+00**	1.996E+01	1.474E+01	1.540E+01	5.920E+00	5.839E+00

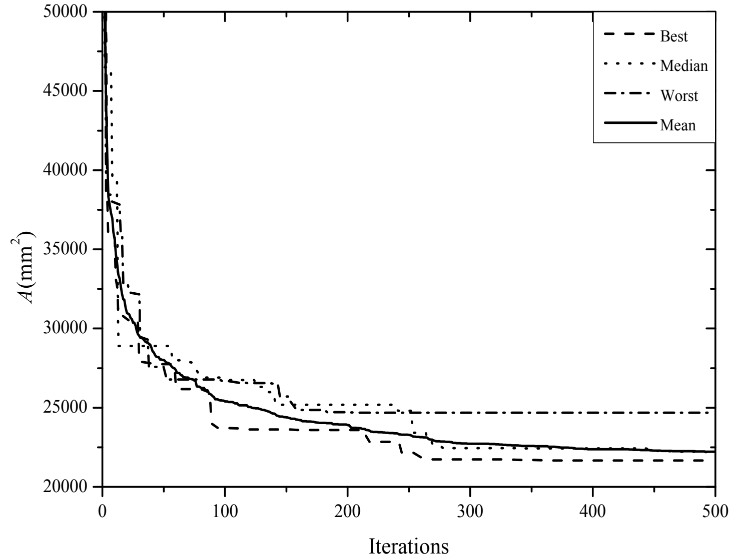

Figure 2. Convergence rate of the best run, the median and the worst run as well as the mean of 25 runs.

- **GA:** standard genetic algorithm [19] with Crossover probability = 0.95 and Mutation probability = 0.05.
- **DE:** The classical differential evolution as described in [20] with "DE/rand/1/bin" strategy is considered. The parameters setting are $CR = rand[0.2, 0.9]$ and $F = rand[0.4, 1]$.
- **HS:** The considered harmony search algorithm is the standard one described in [21] with the following setting $BW = 0.2$, $HMCR = 0.95$, $PAR = 0.3$.

For the common parameters, each algorithm was run 25 times, the population was fixed to $N = 30$, and the maximum number of iteration $t_{max} = 500$ and the dimension of each benchmark functions is $D = 30$. The mean of the minimum obtained after each run are presented in Table 1. As the results show, the optimization performances of the NBA are better than the other algorithm.

6.2. Optimization of the crane girder

To illustrate the performance of the ε–NBA in the optimization of the crane girder, we performed the computations with the following data proposed by Farkas [5]: $H = 200$kN, $L = 22.5$m, $G_t = 42.25$kN, $k = 1.9$m, $h_r = 70$mm, $a = 2.25$m, $p_r + p_s = 190$kg/m, $E = 210$GPa.
Three states of loading are considered, namely, light, moderate and heavy where their characteristics are the following:

- Light: $K_p = 0.50$, $\alpha_d = 1$, $\psi_d = 1.1$, $w_p = \mathrm{L}/500$ and $P_{ft} = 169+145(R–0.1)$;
- Moderate: $K_p = 0.63$, $\alpha_d = 0.95$, $\psi_d = 1.3$, $w_p = \mathrm{L}/600$ and $P_{ft} = 155+135(R–0.1)$;
- Moderate: $K_p = 0.80$, $\alpha_d = 0.90$, $\psi_d = 1.4$, $w_p = \mathrm{L}/700$ and $P_{ft} = 142+125(R–0.1)$.

R represents the ratio between the minimum and the maximum stress. It can be calculated with the following formulas:

$$R = \left(\sigma_{\min} / \sigma_{\max}\right) = \left(M_{x1} / M_x\right) \quad (45)$$

$$M_x = \frac{L^2}{8}\left(1 \cdot 05 A\rho + p_r + p_s\right) g \quad (46)$$

The list of discrete value of the variables is as follows:

- h: from 500 up to 2500 in steps of 1mm;
- b: from 100 up to 1500 in steps of 1mm;
- t_{w1}, t_{w2} & t_f: 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 18, 20, 22 and 25mm.

The parameters setting of ε–NBA are the same as those of NBA listed in section 6.1.

In the first test, we apply the ε–NBA to search for the values of h, b, t_{w1}, t_{w2} and t_f which minimize the area of the cross-section of the

Table 2. Statistical results for different pairs (N, t_{max}).

t_{max}		N 10	30	50	75	100
500	Best	21666	21666	21666	21666	21666
	Medain	21692	21666	21666	21666	21666
	Worst	22620	22360	22226	21670	21669
	Mean	21931	21803	21690	21666	21666
	SD	324.12	233.77	109.49	0.7838	0.5879
	Mean(G)	0	0	0	0	0
1000	Best	21666	21666	21666	21666	21666
	Medain	22054	21666	21666	21666	21666
	Worst	22478	22194	22194	21669	21666
	Mean	21932	21687	21691	21666	21666
	SD	265.29	103.40	103.77	0.8139	0
	Mean(G)	0	0	0	0	0
1500	Best	21666	21666	21666	21666	21666
	Medain	21668	21666	21666	21666	21666
	Worst	22450	21670	21669	21668	21666
	Mean	21744	21666	21666	21666	21666
	SD	205.63	1.0151	0.8139	0.3919	0
	Mean(G)	0	0	0	0	0
2000	Best	21666	21666	21666	21666	21666
	Medain	21666	21666	21666	21666	21666
	Worst	22441	21666	21666	21666	21666
	Mean	21858	21666	21666	21666	21666
	SD	265.77	0	0	0	0
	Mean(G)	0	0	0	0	0

Table 2. Success rate for different pairs (N, t_{max}).

t_{max}	N 10	30	50	75	100
500	16%	52%	76%	81%	85%
1000	43%	72%	89%	98%	96%
1500	52%	86%	98%	97%	100%
2000	64%	93%	99%	99%	100%

Crane girder with consideration of light loading state and steel grade $Y_s = 230$MPa. The bat population was set to $N = 25$ and the maximum number of iterations was fixed to $t_{max} = 500$. The algorithm was run 25 times. At each run, the minimum of the objective function was recorded every single iteration. The best run was selected as the run where the algorithm achieved the best minimum of the objective function among the 25 runs. The same analogy was used to select the worst and the median. The convergence rate of the objective function of the best run the median and the worst run are depicted in Figure 2, in addition to the mean which represents the mean of the objective function at each iteration for 25 runs. As it can be seen, each run converge to a different solution because ε–NBA is a stochastic algorithm, which depends on random generation of solution. To increase the reliability and the robustness of the algorithm, an analysis of the effect of the population size and number of iterations is conducted.

Table 4. Optimization results of the crane girder for various steel grade and state of loading.

N	Y_s	h	t_{w1}	t_{w2}	b	t_f	A	Saving	Static Stress	Fatigue stress	Deflection	Ref
Light $w_p=L/500$	230	1050	6	6	400	14	23800		130< 136	95< 193	29.8< 45.0	[5]
		1206	6	5	420	10	21666	8.97%	136< 136	99< 191	23.7< 45.0	ε–NBA
	355	950	5	5	375	14	20000		159< 209	116< 191	40.0< 45.0	[5]
		914	5	4	448	10	17186	14.07%	187< 209	134< 188	45.0< 45.0	ε–NBA
	450	1050	5	5	375	10	18000		174< 266	125< 189	43.2< 45.0	[5]
		1014	5	4	417	8	15798	12.23%	202< 266	144< 187	44.3< 45.0	ε–NBA
Moderate $w_p=L/600$	230	1150	7	7	375	14	26600		129< 129	102< 175	25.2< 37.5	[5]
		1194	6	5	466	12	24318	8.58%	126< 129	99< 174	19.9< 37.5	ε–NBA
	355	1050	6	6	325	14	21700		165< 199	130< 173	35.5< 37.5	[5]
		1023	6	5	375	10	18753	13.58%	196< 199	153< 171	37.5< 37.5	ε–NBA
	450	1000	5	5	325	16	20400		170< 253	133< 172	36.1< 37.5	[5]
		1022	5	4	412	10	17438	14.52%	194< 252	151< 170	37.3< 37.5	ε–NBA
Heavy $w_p=L/700$	230	1150	7	7	450	14	28700		119< 122	105< 160	21.7< 32.1	[5]
		1292	7	5	519	10	25884	9.81%	121< 122	107< 158	17.1< 32.1	ε–NBA
	355	1000	6	6	325	18	23700		157< 188	139< 158	31.7< 32.1	[5]
		1125	6	5	393	10	20235	14.62%	177< 189	156< 156	29.1< 32.1	ε–NBA
	450	1050	5	5	425	14	22400		149< 239	131< 157	29.0< 32.1	[5]
		1111	5	4	437	10	18739	16.34%	176< 239	155< 155	29.6< 32.1	ε–NBA

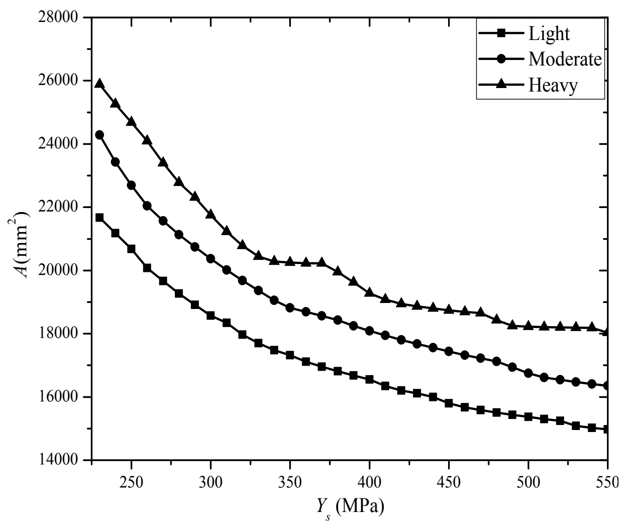

Figure 3. Optimal cross-section area according to the steel yield stress.

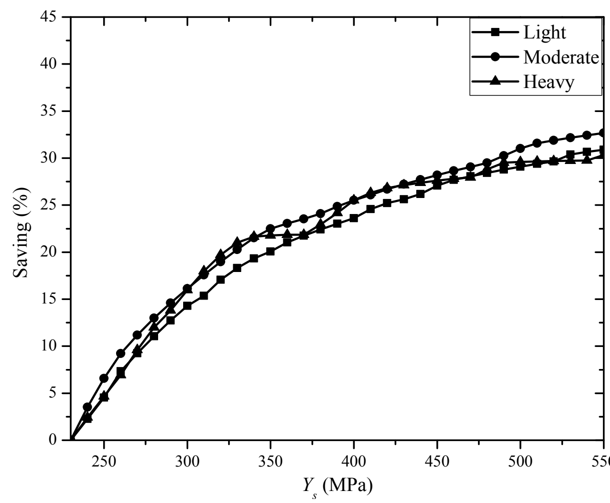

Figure 4. Percentage of weight saving according to the steel yield stress.

Table 2 presents the statistical results of 25 runs of the ε–NBA with different population sizes and maximum number of iterations. The results present the best, the median and the worst solution in addition to the mean and the standard deviation. As it can be seen, by increasing the bat population and the number of iterations, the best, the median, the worst and the mean have a tendency to converge to the same solution and the standard deviation tends to zero (i.e. N = 100 and t_{max} = 1500). That means when the values of the pair (N, t_{max}) increase, the robustness and reliability of the results increases. In addition, the analysis of the constraint violation shows that the use of the ε–level of comparison for constraint handling leads the algorithm to find feasible solutions with no constraint violation.

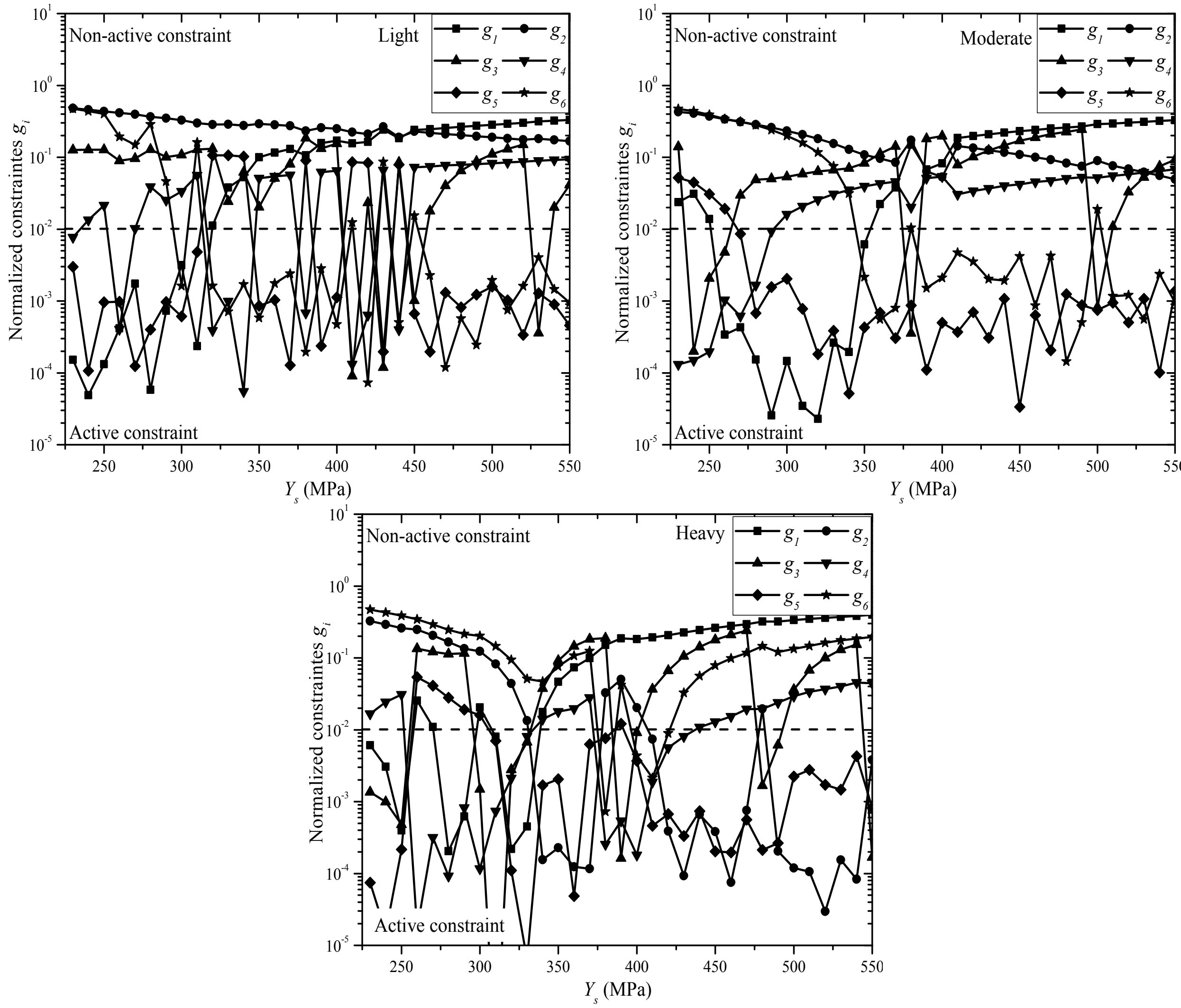

Figure 5. Behavior of the normalized constraints according to the steel yield stress.

To analysis of the success rate so that ε–NBA converge to the optimal solution, several pairs of (N, t_{max}), were considered. The algorithm was run 100 times, and each time the run was successful it was recorded. A run is considered successful if the obtained minimum is A = 21666mm² (which is from Table 1 is the optimal solution). Each times the algorithm achieve this value it was recorded and the success rate represent the ratio of the number of the successful runs over the total number of runs, which is in this case 100. As it can be seen, the success rate increase as the value of the pair (N, t_{max}) increases, and for $N \geq 100$ and $t_{max} \geq 1500$, the success rate is 100%. That means with the precedent parameters' setting, there is a probability of 100% that the algorithm converge toward the optimum.

We consider three states of loading (light, moderate and heavy) and three steel grade (Y_s = 230, 355, 450). We fix the bat population to N = 100 and the maximum number of iterations to t_{max} = 1500. Table 4 presents a comparisons between the optimal solutions obtained with ε–NBA and the results of Farkas [5] obtained using the combinatorial discrete backtrack programming method. As it can be seen, the solution obtained with ε–NBA are much better to those of Farkas [5] with economization in the girder weight between 8% and 16% according to the loading state and the steel grade.

The use of higher strength steels may result in saving in mass, manufacturing cost and operating energy. Figure 3 presents the optimal cross-section area of the box-girder for different steel grades (the yield stress varies between 230 and 550MPa). As the yield stress increases the optimal area decreases. In case of heavy loading, we observe some steadiness in the optimal area evolution around 350 and 375MPa, this is

an effect of the discrete optimization. The Figure 4 presents the saving percentage according to the yield stress. We observe that the loading state has an effect on the saving but not as much as the yield stress.
Figure 5 present the constraints status of the optimal configuration according the yield stress. Three cases of loading states are considered. The constraints have been normalized and if the value of a constraint is low then 0.01, it is considered to be active, and if it is negative, the constraint is violated. From the results, we observe that constraint of the fatigue stress in the lower flange at mid-span (g_2) is passive for light and moderate loading, and it become active in case of heavy loading for higher yield stress. In most cases the constraint on local buckling of the main and secondary (g_4 & g_5) web are active. The constraint on the local buckling of the upper flange (g_3) varies between passive and active due to the discrete values of the design parameters. For the three loading cases, we observe that the constraint on static stress of the lower flange (g_1) is active for low yield stress values and active for high yield stress value, the same remark for the static deflection constraint (g_6).

7. CONCLUSION

In this study, we presented a methodology for the optimization of the main girder of an overhead travelling crane with double box-girder, based on a new bat algorithm. Due to the premature convergence problem, four modifications have been embedded to the standard bat algorithm in the aim to increase its exploitation / exploration abilities. The resulted new bat algorithm was tested on unimodal / multimodal benchmark functions and compared with other classical algorithms. The results show that NBA can surpass some standard algorithms such as the genetic algorithm and particle swarm optimization. To solve constrained optimization problems, ε–level of comparison was introduced to NBA to formulate the ε–NBA which can handle constrained problems.
The crane-girder problem was formulated with the area of box-section as the objective function. Constraints have been imposed on the local buckling of the upper flange and main and secondary web, the static and fatigue stress of the lower flange, in addition to the permissible maximum deflection. Three states of loading were considered. The crane-girder optimization problem was solved with ε–NBA. Results show that the proposed algorithm can achieve better results to those exist on the literature. By varying the steel yield stress, we observe that the use of higher strength steel leads to a much lighter crane girder, which can reduce the cost of manufacturing and operation.

REFERENCES

[1] J. J. Coulton, "Lifting in early Greek Architecture," *The Journal of Hellenic Studies,* vol. 94, pp. 1-19, 1974.

[2] http://www.machine-history.com/Armstrong%20Hydraulic%20Crane. (04/07/2015).

[3] S. W. Cho and B. M. Kwak, "Optimal Design of Electric Overhead Crane Girders," *Journal of Mechanisms, Transmissions, and Automation in Design,* vol. 106, pp. 203-208, 1984.

[4] S. S. Rao, "Optimum Design of Bridge Girders for Electric Overhead Traveling Cranes," *Journal of Engineering for Industry,* vol. 100, pp. 375-382, 1978.

[5] J. Farkas, "Economy of higher-strength steels in overhead travelling cranes with double-box girders," *Journal of Constructional Steel Research,* vol. 6, pp. 285-301, 1986.

[6] K. Jarmai, "Decision support system on IBM PC for design of economic steel structures applied to crane girders," *Thin-Walled Structures,* vol. 10, pp. 143-159, 1990.

[7] G. Pavlovic, M. Savkovic, M. Gasic, R. Bulatovic, and N. Zdravkovic, "Optimization of the box section of the main girder of the double beam bridge crane according to the criteria of lateral stability and local stability of plates," *Machine Design,* vol. 4, pp. 197-204, 2012.

[8] M. Savković, M. Gašić, D. Ćatić, R. Nikolić, and G. Pavlović, "Optimization of the box section of the main girder of the bridge crane with the rail placed above the web plate," *Structural and Multidisciplinary Optimization,* vol. 47, pp. 273-288, 2013/02/01 2013.

[9] R. C. Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 81-86 vol. 1.

[10] M. Dorigo, V. Maziezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating ants," *Systems, Man and Sybernetics B. IEEE Trans.,* vol. 26, pp. 29-41, 1996.

[11] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210-214.

[12] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. vol. 284, J. González, D. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 65-74.

[13] X. S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations,* vol. 29, pp. 464-483, 2012.

[14] A. Gandomi, X.-S. Yang, A. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications,* vol. 22, pp. 1239-1255, 2013.

[15] O. Hasançebi, T. Teke, and O. Pekcan, "A bat-inspired algorithm for structural optimization," *Computers & Structures,* vol. 128, pp. 77-90, 2013.

[16] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1-8.

[17] I. British-Standards, "BS 2573: Part 1, Rules for design of cranes, specification for classification, stress calculation and design criteria for structures," ed. London: BSI, 1983.

[18] I. British-Standards, "BS 5400: Part 3, Code of practice for design of steel bridges," ed. London: BSI, 1982.

[19] L. Davis, *Handbook of genetic algorithms* vol. 115: Van Nostrand Reinhold New York, 1991.

[20] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on,* vol. 15, pp. 4-31, 2011.

[21] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *SIMULATION,* vol. 76, pp. 60-68, 2001.