Beetle Antennae Search without Parameter Tuning (BAS-WPT) for Multi-objective Optimization

Xiangyuan Jiang\textsuperscript{a,b}, Shuai Li\textsuperscript{b}

\textsuperscript{a}Institute of Marine Science and Technology, Shandong University, Qingdao, China.
\textsuperscript{b}the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China.

Abstract. Beetle antennae search (BAS) is an efficient meta-heuristic algorithm inspired by foraging behaviors of beetles. This algorithm includes several parameters for tuning and the existing results are limited to solve single objective optimization. This work pushes forward the research on BAS by providing one variant that releases the tuning parameters and is able to handle multi-objective optimization. This new approach applies normalization to simplify the original algorithm and uses a penalty function to exploit infeasible solutions with low constraint violation to solve the constraint optimization problem. Extensive experimental studies are carried out and the results reveal efficacy of the proposed approach to constraint handling.

1. Introduction

Nature-inspired algorithms have giant potential to solve optimization problems and have been successfully implemented in various scientific and engineering domains [1]. Main challenges of meta-heuristic algorithms lie in how to handle various constraints imposed on variables and how to simplify the parameter tuning of algorithms.

Because of the presence of constraints, the feasible space may be largely reduced, making the searching process to be a challenging task compared with non-constrained optimization to a single objective. To solve the aforementioned optimization problem, Tessema and Yen [2] propose an adaptive penalty function to exploit infeasible solutions with appropriate fitness value and low constraint violation. And then, they extend the constraint-handling results to multi-objective evolutionary optimization problem based on adaptive penalty function and distance measure [3]. Using a multi-objective formulation, Runarsson and Yao [4] propose a common approach to apply a penalty function to bias the search towards a feasible solution.

In terms of parameter tuning, we further develop our previous work, named as the beetle antennae search (BAS) algorithm, which is inspired by the searching and detecting behavior of longhorn beetles. In this paper, we improve the result in [5] to be a simple implementation and need no parameter tuning. As the original BAS in [5] does not consider constraint, we also modify the original BAS to be capable of...
solving constrained optimization problem simultaneously. The main contributions of the paper are stated as below:

1. Normalization method is used to extend the original BAS algorithm to be a general form without parameter tuning, which makes the algorithm implement simply.
2. Based on the improved BAS algorithm without parameter tuning (BAS-WPT), constraint optimization problems are formulated as a multi-objective optimization problem and further handled by penalty function method.

The rest of the paper is organized as follows. In Section 2, improved BAS algorithm is proposed. In Section 3, BAS-WPT are used in constraint optimization problem. In Section 4, numerical results are presented and compared. In Section 5, a concluding remark is drawn.

2. Optimization Design of The Proposed Approach

In this section, by considering the original BAS, an improved BAS is presented without parameter tuning to simply the application for user. The algorithm is basically an original implementation except for the normalization of input data.

2.1. the Original BAS

For clear illustration, the original BAS is included in Algorithm 1, which is capable of searching global optimum of both convex and non-convex problem in a general function:

\[
\begin{align*}
\text{Minimize} & \quad f(x), x = [x_1, x_2, \cdots, x_N]^T \\
\text{Maximize} & \quad f(x), x = [x_1, x_2, \cdots, x_N]^T
\end{align*}
\]

where \( f \) is the fitness function and \( x \in \mathbb{R}^N \) denotes the input data in \( N \) dimensions. The main formula of the natural-inspired BAS consist of two aspects: searching behavior and detecting behavior. The searching behavior is used to explore the optimal point of the fitness function by introducing a normalized random unit vector \( \vec{b} \) to enhance the searching ability,

\[
\begin{align*}
x_r &= x' + d \vec{b}, \\
x_l &= x' - d \vec{b},
\end{align*}
\]

(1)

and the detecting behavior is used to exploit in an iterative form,

\[
x' = x'^{-1} + \delta \vec{b} \text{sign}(f(x_r) - f(x_l)),
\]

(2)

where \( d \) represents the distance between two antennae of a longhorn beetle and \( \delta \) represents the step size of each iteration.

Evidently, the presetting of parameters such as \( d \) and \( \delta \) influences performance of BSA seriously. Thus, we attempt to develop a much effective and robust improvement.

2.2. the Proposed BAS-WPT Approach

Fig. 1 demonstrates the iterative optimization process of BAS-WPT which can be seen as a variable scale algorithm from the figure obviously.

For the sake of simplicity, we use normalization method to tune parameters of BAS adaptively. Assume that \( x_i \) is the \( i \)-th element of \( x \), lies in the range from \( x_i^{\min} \) to \( x_i^{\max} \), and then \( x \) satisfied \( x_i \in [x_i^{\min}, x_i^{\max}] \), where \( x_i^{\min} \) is lower bound and \( x_i^{\max} \) is the upper bound. The input data used in fitness function could be formulated in the following expression at each iteration:

\[
x_i = x_i(x_i^{\max} - x_i^{\min}) + x_i^{\min}.
\]

(3)
Figure 1: The iterative optimization process of BAS-WPT in 5 steps. The dotted line in red color denotes the trajectory of fitness function. The triangle represents a beetle, on both sides of which two solid circles denote antennae of the beetle, \( d \) is the distance between two antennae and \( \delta \), corresponding to the dotted line in black color, is the step size of searching.

Finally, we obtain the global optimum \( f(\tilde{x})_{\text{bst}} \) corresponding to the position \( \tilde{x}_{\text{bst}} \) by BAS-WPT algorithm with normalized variant \( x_{\text{bst}} \).

To simplify the parameter tuning further more, we also construct the relationship between searching distance \( d \) and step size \( \delta \) as follows:

\[
\delta^t = c_1 \delta^{t-1} + \delta^0, \quad d^t = \delta^t / c_2,
\]

where \( c_1 \) and \( c_2 \) are constants to be adjusted by designers.

3. Constraint Handling by BAS-WPT

In this section, we extend the BAS-WPT algorithm into constrained optimization problem with penalty function method.

3.1. Problem Formulation

A constraint optimization problem can be formulated as

\[
\begin{align*}
\text{Minimize} & \quad f(x), \\
\text{Maximize} & \quad f(x), \\
\text{s.t.} & \quad g_j(x) \leq 0, \quad j = 1, \ldots, K, \\
& \quad x_{\text{max}}^i \leq x_i \leq x_{\text{min}}^i, \quad i = 1, \ldots, N,
\end{align*}
\]

There are \( K \) inequality function constraints required to be satisfied by the optimal solution. The presence of constraints of both inequality functions and variants restrict the searching area to be a interest region, where suitable solution could be found.
3.2. Penalty Function Method

To solve the constrained optimization problem, we present penalty function method to deal with inequality function constraint.

In penalty functions, infeasible solutions are penalized for the violation of the inequality constraint by putting penalty terms on the original fitness function, which will reduce the probability of selecting an infeasible solution. Specially, penalty function in our study is formulated in the following form:

$$F(x) = f(x) + \lambda \sum_{j=1}^{K} h_j(x) g_j(x),$$  \hspace{1cm} (6)

where $F(x)$ is the improved fitness function, $f(x)$ is the original fitness function, $\lambda$ is the penalty parameter usually predefined as a large enough value (e.g. $10^{10}$), and the constraint violation $h_j(x)$ is defined as

$$h_j(x) = \begin{cases} 1, & g_j(x) > 0 \\ 0, & g_j(x) \leq 0 \end{cases}$$ \hspace{1cm} (7)

When anyone of the inequality constraints $g_j(x) > 0$ satisfies accompanying with a large value $\lambda$, the second term of (6) dominates the fitness function, which makes $F(x) \rightarrow \infty$. Otherwise, all $h_j(x) = 0$ are satisfied, and thus $F(x) = f(x)$.

Algorithm 1 corresponding to BAS-WPT algorithm demonstrates the improvement based on BAS adopted in the research to design a more feasible approach to solve the constraint optimization problem.

**Algorithm 1: BAS-WPT algorithm for constrained optimization**

**Input:** Initialize the input data $x^0$ at 0 time instance in standard normalization form
$$x_i^0 = (\text{rand}() - x_i^{\text{min}})/(x_i^{\text{max}} - x_i^{\text{min}})$$ for each element , and initialize the parameters $c_1, c_2, \delta^0$.

**Output:** $x_{bst}$, $f_{bst}$.

while ($t < T_{max}$ or (stop criterion)) do

- Search in variable space with two kinds of antennae according to (1);
- Update the state variable $x'$ according to (2);
- Generate the normalized vector $\bar{x}$ according to (4);
- Construct the improved fitness function according to (6) and (7);
- if $F(x')$ satisfies optimum condition then
  - $F_{bst} = F(x')$, $x_{bst} = x'$.
- Update parameters according to (4). Calculate the best potion $\bar{x}_{bst}$ by $x_{bst}$ similarly to (3).

return $x_{bst}$, $f_{bst}$.

4. Experimental Study

In this section, we present two examples from optimization literatures to demonstrate the performance of the proposed BAS-WPT algorithm for constrained optimization.
There are four variables in pressure vessel problem which aims at minimizing the fitness function below:

\[
\begin{align*}
\text{minimize } f(x) &= 0.6224x_1x_3x_4 + 1.7781x_2^2x_3^2 \\
&\quad + 3.1661x_1^2x_4 + 19.84x_1^2x_3,
\end{align*}
\]

s.t. \[ g_1(x) = -x_1 + 0.0193x_3 \leq 0, \]
\[ g_2(x) = -x_2 + 0.00954x_3 \leq 0, \]
\[ g_3(x) = -\pi x_3^3 - \frac{4}{3}x_3^3 + 1296000 \leq 0, \]
\[ g_4(x) = x_4 - 240 \leq 0, \]
\[ x_1 \in \{1, 2, 3, \ldots, 99\} \times 0.0625, \]
\[ x_2 \in \{1, 2, 3, \ldots, 99\} \times 0.0625, \]
\[ x_3 \in [10, 200], \]
\[ x_4 \in [10, 200]. \]

Table 1 illustrates the best results obtained by the proposed BAS-WPT algorithm using only 150 iterations and other various existing algorithms need much more iterations to solve the pressure vessel optimization problem. It is worth pointing out that the best result from the proposed BAS-WPT algorithm is better than most of the existing ones and has the fastest convergence simultaneously.

### Table 1: Comparisons of results for Pressure Vessel Function

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$g_1(x)$</th>
<th>$g_2(x)$</th>
<th>$g_3(x)$</th>
<th>$g_4(x)$</th>
<th>$f_*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>0.8125</td>
<td>0.4375</td>
<td>42.0984</td>
<td>176.6378</td>
<td>-8.8000e-7</td>
<td>-0.0359</td>
<td>-3.5586</td>
<td>-63.3622</td>
<td>6059.7258</td>
</tr>
<tr>
<td>[7]</td>
<td>1</td>
<td>0.625</td>
<td>51.2519</td>
<td>90.9913</td>
<td>-1.011</td>
<td>-0.136</td>
<td>-18759.75</td>
<td>-149.009</td>
<td>7172.300</td>
</tr>
<tr>
<td>[8]</td>
<td>0.8125</td>
<td>0.4375</td>
<td>42.0870</td>
<td>176.7791</td>
<td>-2.10e-4</td>
<td>-0.0359</td>
<td>-5.1084</td>
<td>-63.2208</td>
<td>6061.129</td>
</tr>
<tr>
<td>[9]</td>
<td>1.000</td>
<td>0.625</td>
<td>51.000</td>
<td>91.000</td>
<td>-0.0157</td>
<td>-0.1385</td>
<td>-3233.916</td>
<td>-149</td>
<td>7079.037</td>
</tr>
<tr>
<td>[10]</td>
<td>0.8125</td>
<td>0.4375</td>
<td>41.9768</td>
<td>182.2845</td>
<td>-0.0023</td>
<td>-0.0370</td>
<td>-22888.07</td>
<td>-57.7155</td>
<td>6171.000</td>
</tr>
<tr>
<td>[11]</td>
<td>1.125</td>
<td>0.625</td>
<td>58.291</td>
<td>43.690</td>
<td>0.000016</td>
<td>-0.0689</td>
<td>-21.2201</td>
<td>-196.3100</td>
<td>7198.0428</td>
</tr>
<tr>
<td>[12]</td>
<td>0.9375</td>
<td>0.5000</td>
<td>48.3290</td>
<td>112.6790</td>
<td>-0.0048</td>
<td>-0.0389</td>
<td>-3652.877</td>
<td>-127.3210</td>
<td>6410.3811</td>
</tr>
<tr>
<td>[13]</td>
<td>0.8125</td>
<td>0.4375</td>
<td>40.3239</td>
<td>200.0000</td>
<td>-0.034234</td>
<td>-0.05285</td>
<td>-27.10585</td>
<td>-40.0000</td>
<td>6288.7445</td>
</tr>
<tr>
<td>[14]</td>
<td>1.125</td>
<td>0.625</td>
<td>58.1978</td>
<td>44.2930</td>
<td>-0.00178</td>
<td>-0.06979</td>
<td>-97.760</td>
<td>-132.28</td>
<td>7980.894</td>
</tr>
<tr>
<td>[15]</td>
<td>1.125</td>
<td>0.625</td>
<td>48.97</td>
<td>106.72</td>
<td>-0.1799</td>
<td>-0.1578</td>
<td>97.760</td>
<td>-132.28</td>
<td>7980.894</td>
</tr>
<tr>
<td>[16]</td>
<td>1.125</td>
<td>0.625</td>
<td>58.2789</td>
<td>43.7549</td>
<td>-0.0002</td>
<td>-0.06902</td>
<td>-3.71629</td>
<td>-196.245</td>
<td>7198.433</td>
</tr>
<tr>
<td>[17]</td>
<td>0.7782</td>
<td>0.3846</td>
<td>40.3196</td>
<td>200.000</td>
<td>-3.172e-5</td>
<td>4.8984e-5</td>
<td>1.3312</td>
<td>-40</td>
<td>5885.33</td>
</tr>
<tr>
<td>BAS-WPT</td>
<td>0.8125</td>
<td>0.4375</td>
<td>42.09355</td>
<td>176.7715</td>
<td>9.43E-05</td>
<td>0.03592</td>
<td>-413.6252</td>
<td>-63.2285</td>
<td>6062.04676</td>
</tr>
</tbody>
</table>

* denotes the violation of the corresponding constraint.

### 4.2. Himmelblau Function

We also consider the Himmelblau’ nonlinear optimization problem which is a famous benchmark used used for several evolutionary algorithm before. The problem consists of 5 variables, 6 inequality constraints and 10 boundary conditions and could be further stated as follows:

\[
\begin{align*}
\text{minimize } f(x) &= (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\
&\quad + (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,
\end{align*}
\]

s.t. \[ x_1 \leq 10, \]
\[ x_2 \leq 10, \]
\[ x_3 \leq 10, \]
\[ x_4 \leq 10, \]
\[ x_5 \leq 10. \]
Table 2: Comparisons of results for Himmelblau function

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$g_1(x)$</th>
<th>$g_2(x)$</th>
<th>$g_3(x)$</th>
<th>$L_*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>78.00</td>
<td>33.00</td>
<td>29.995256</td>
<td>45.00</td>
<td>36.775813</td>
<td>92</td>
<td>98.8405</td>
<td>20</td>
<td>-30665.54</td>
</tr>
<tr>
<td>[19]</td>
<td>78.6200</td>
<td>33.4400</td>
<td>31.0700</td>
<td>44.1800</td>
<td>35.2200</td>
<td>90.5208</td>
<td>98.8929</td>
<td>20.1316</td>
<td>-30373.949</td>
</tr>
<tr>
<td>[21]</td>
<td>78.00</td>
<td>33.00</td>
<td>29.995</td>
<td>45.00</td>
<td>36.776</td>
<td>90.7147</td>
<td>98.8405</td>
<td>19.9999</td>
<td>-30665.6088</td>
</tr>
<tr>
<td>BAS-WPT</td>
<td>78.00</td>
<td>33.00</td>
<td>27.1131</td>
<td>45.00</td>
<td>45.00</td>
<td>91.9997</td>
<td>100.4170</td>
<td>20.02056</td>
<td>-31011.3244</td>
</tr>
</tbody>
</table>

$\diamond$ denotes the violation of the corresponding constraint.

**minimize** $f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.29329x_1 - 40792.141$,

**s.t.**

$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.002053x_3x_5,$

$g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_3 + 0.0021813x_3^2,$

$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4,$

$0 \leq g_1(x) \leq 92,$

$90 \leq g_2(x) \leq 110,$

$20 \leq g_3(x) \leq 25,$

$78 \leq x_1 \leq 102,$

$33 \leq x_2 \leq 45,$

$27 \leq x_3 \leq 45,$

$27 \leq x_4 \leq 45,$

$27 \leq x_5 \leq 45,$

The results are listed in Table 2 whose corresponding experiments for the BAS-WPT algorithm just need only one beetle to run 200 instance. Evidently, the best result generated from the BAS-WPT shows the most excellent performance among all the results listed in Table 2. The above experiments justify that the proposed BAS-WPT algorithm is effective to handle constraint optimum problem and could achieve a good performance with high convergence rate.

5. Conclusion

This paper extends nature-inspired BAS algorithm to solve multi-objective optimization problem and releases it to be a new version without parameter tuning. Two typical benchmarks are considered to validate performances of the algorithm, whose numerical results justify the efficacy of the proposed algorithm.

References