A comparative study of several bio-inspired algorithms in cost optimization of cellular beams

A Tjahjono1,2, E J Wijayanti1, D Prayogo1 and F T Wong1
1 Department of Civil Engineering, Petra Christian University, Surabaya, Indonesia
2 Corresponding author: b11180019@john.petra.ac.id

Abstract. Castellated beams are commonly used in steel construction. This study will focus on castellated beams with circular-shaped openings, which are known as cellular beams. Cost optimization of cellular beams is needed to maintain cost efficiency. The optimization considers the selection of a root beam, the diameter of holes, and the total number of holes in the beam as the variables. Four metaheuristic algorithms are used to optimize the design, namely, the particle swarm optimization (PSO), differential evolution (DE), symbiotic organisms search (SOS), and artificial bee colony (ABC). A four-meter span beam with a 50 kN point live load in the middle of the beam and a 5 kN/m uniformly-distributed dead load are taken as the case study. The results indicate that the SOS algorithm yields the best optimization results in terms of the average, consistency, and converge behavior with a 30 out of 30 success rates.

1. Introduction
A castellated beam is a modification of a root beam with open sections on its web. The castellated beam is made by unifying two parts of beams which are already cut in desired patterns on its web as shown in Figure 1. This modification has a higher capacity by increasing the beam’s inertia, depth, and section modulus with no additional weight of the beam itself [1]. Castellated beams are commonly used for medium to long-span floor systems. By using castellated beams, the underfloor service ducts can be installed through the web opening without adding the floor height [2]. Normally, the openings provided for castellated beams are in hexagonal or circular shape [3]. The ones with circular-shaped openings are often called cellular beams. Cellular beams are made by cutting in a half circular pattern through the web twice [1]. Cellular beams are the modern version of castellated beams. Cellular beams could produce tapered beams with lower cost and more efficiently than castellated beams due to its geometry flexibility [1]. Therefore, this study focuses on cellular beams.

Cost efficiency in civil engineering is very important to be considered. The cost of a cellular beam is mostly affected by the area of the steel section, the diameter of holes, and the number of holes. To minimize the cost, optimization of those variables is needed. The cost optimization problem could be solved using a bio-inspired algorithm. Bio-inspired algorithms have shown better optimization results compared to the traditional one. Traditional optimization sometimes faces problems to find global optimum, while bio-inspired algorithms are more adaptive in finding global optimum [4]. However, every bio-inspired algorithm has its concepts and weaknesses. This paper will focus on comparing the performance of four bio-inspired algorithms, that is, the particle swarm optimization (PSO) [5], differential evolution (DE) [6], symbiotic organisms search (SOS) [7], and artificial bee colony (ABC) [8].
2. Review of bio-inspired algorithms

Bio-inspired algorithms are algorithms that are mostly based on animal behavior and Darwin’s natural selection theory. This theory is used to find the fittest species which can survive in the environment. This paper specifically discuss algorithms that are inspired by the behavior of birds, bees, and ants.

2.1. Particle swarm optimization (PSO)

In the year of 1995 PSO was introduced by Kennedy and Eberhart [5]. The behavior of a group of fish or birds while searching for food is the inspiration of PSO’s concept. In that case, the behavior of each bird or fish will affect the overall group’s behavior. In the beginning, the location of each particle will be generated randomly. In each iteration, every particle will move using the information of the velocity vector given in equation (1) and renews its location using equation (2):

\begin{equation}
    v_i(t+1) = Wv_i(t) + r_1C_1(x_{P_{best}}(t) - x_i(t)) + r_2C_2(x_{G_{best}}(t) - x_i(t))
\end{equation}

\begin{equation}
    x_i(t+1) = x_i(t) + v_i(t+1)
\end{equation}

where \( v_i(t+1) \) is the particle’s velocity, \( W \) is the inertia weight, \( v_i(t) \) is particle’s initial velocity, \( r_1 \) is a random number between 0 and 1, \( C_1 \) is a cognitive parameter, \( r_2 \) is a random number between 0 and 1, \( C_2 \) is a social parameter, \( x_{P_{best}}(t) \) is the location of individual best, \( x_i(t) \) is the particle’s initial location, \( x_{G_{best}}(t) \) is the location of group best, and \( x_i(t+1) \) is the particle’s new location.

2.2. Differential evolution (DE)

Storn and Price [6] introduced DE around 1997. The DE has four phases, namely, initialization, differential mutation, crossover, and selection. The initial population in the DE comes by randomizing numbers between the lower and upper bounds in the initialization phase. The mutation is the second phase of the DE. This phase takes the difference of two random vectors, named \( X_{r2,G} \) and \( X_{r3,G} \), multiplied by \( F \) then added to a third vector \( X_{r1,G} \) to become mutant vectors \( V_{i(G+1)} \). The equation can be seen in equation (3) where \( F \) is taken between 0 and 2.

\begin{equation}
    V_{i(G+1)} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})
\end{equation}
The third phase is crossover. This phase generates a trial vector $U_{j\in(G+1)}$ which is a new vector developed by combining the target vector with the mutant vector. If $\text{rand}_{j,i} \leq CR$ or $j = I_{\text{rand}}$, the mutant vectors $V_{j\in(G+1)}$ will enter the trial population as shown in equation (4), otherwise the initial vector $X_{j\in(G+1)}$ is the one that joins the trial population like shown in equation (5). $\text{rand}_{j,i}$ is taken between numbers 0 and 1 while $I_{\text{rand}}$ is obtained by randomizing integer form (1, 2, ..., $D$).

$$U_{j\in(G+1)} = V_{j\in(G+1)}$$ (4)

$$U_{j\in(G+1)} = X_{j\in(G+1)}$$ (5)

The last phase is selection. This phase is looking for vectors that will join the next generation group by taking the better value between the trial and target vector.

### 2.3. Symbiotic organisms search (SOS)

The SOS algorithm was introduced in 2014 by Cheng and Prayogo [7]. It obtains its inspiration from three types of symbiotic interactions between living things, that is, mutualism, commensalism, and parasitism. In the phase of mutualism, organisms will interact in a mutually beneficial relationship to improve their independent quality. If the results of the new organisms are better, then the organisms will be renewed. The new organisms are created using equation (6) - (8):

$$\text{new}O_i = O_i + r_1 \left[ O_{\text{best}} - O_{\text{average}}(1 + r_2) \right]$$ (6)

$$\text{new}O_j = O_j + r_2 \left[ O_{\text{best}} - O_{\text{average}}(1 + r_2) \right]$$ (7)

$$O_{\text{average}} = \frac{O_i + O_j}{2}$$ (8)

where $\text{new}O_i$ is the new candidate for $O_i$, $O_i$ is the $i$-organism in the ecosystem, $r_1$ and $r_2$ are random numbers between 0 and 1, $r_1$ and $r_2$ are rounded values of random numbers between 0 and 1, $O_{\text{best}}$ is the global best organism, $\text{new}O_i$ is the new candidate for $O_i$, and $O_i$ is the $j$-organism in the ecosystem.

In the commensalism phase, the organisms will interact in a relationship where one organism takes advantage while the other one is given neither advantage nor disadvantage. Organism $O_i$ with the randomly chosen organism $O_i$. This phase will only modify organism $O_i$. The new organism is created from the formula shown in equation (9) where $r_3$ is a random number between -1 and 1:

$$\text{new}O_i = O_i + r_3 \left[ O_{\text{best}} - O_i \right]$$ (9)

In the parasitism phase, one of the organisms will be harmed while the other one gains advantage. Organism $O_i$ will produce an artificial parasite organism where its value will then be compared to organism $O_i$. If the value of organism $O_j$ is worse, then organism $O_j$ will be replaced.

### 2.4. Artificial Bee Colony (ABC)

Karaboga and Basturk [8] in 2007 introduced a swarm-based metaheuristic algorithm called ABC. The intelligent behavior of honeybees when looking for food sources is the inspiration of this bio-inspired algorithm. ABC classified the bees into three groups, that is, employed bees which are connected to specific food sources, onlooker bees which choose a food source based on their observation on the dance of employed bees within the hive, and scout bees which look for food sources randomly.

In the initialization phase, the scout bees randomly discover food source locations. In the phase of employed bees, the bees exploit those food sources until they become exhausted. Employed bees whose food source location has been exhausted become scout bees who are looking for alternative food source
locations. The alternative food source location will be shared with the onlooker bees using equation (10):

\[
\text{newFood}_i = \text{Food}_j + r_1[\text{Food}_i - \text{Food}_j]
\]  

(10)

where \(\text{Food}_i\) is the food source at \(i\), \(r_1\) is a random number between -1 and 1, \(\text{newFood}_i\) is the modified food source at \(i\) after the onlooker bees’ phase, and \(\text{Food}_j\) is the food source at \(j\) chosen at random.

In the phase of the onlooker bees’ group, the onlooker bees will use the probability provided by employed bees to choose the new location. In the scout bees’ phase, the employed bees transform to scout bees and look for a new potential location to become an alternative in case the solution does not improve after a period of time. ABC optimization will stop the process if the number of iterations has obtained maximum iteration or the optimum value has been discovered.

3. Formulation of the cellular beam cost optimization problem

The main purpose of this study is to minimize the cost of cellular beams while considering all the constraints. When defining the cost of a cellular beam, a few factors such as the weight of the beam, cutting price, and welding price should be considered. The objective of the optimization is to obtain a vector is shown in equation (11):

\[
\text{x} = \{x_1 \ x_2 \ x_3\}
\]  

(11)

where \(x_1\) is the selection of a root beam, \(x_2\) is the diameter of hole \((D_0)\), and \(x_3\) is the total number of holes in the beam \((NH)\). Variable \(x_1\) affects the section area of the beam \((A_{\text{initial}})\). The cost of cellular beam is calculated using objective function shown in equation (12):

\[
F_{\text{cost}} = \rho A_{\text{initial}} \left( L + \frac{S}{2} \right) P_1 + L_{\text{cut}} P_2 + L_{\text{weld}} P_3
\]  

(12)

where \(\rho\) is the steel density, \(L\) is the span between two supports, \(S\) is the length from one hole’s center to the other center, \(P_1\) is the price of steel beam per unit weight, \(P_2\) is the price of cutting per unit length, and \(P_3\) is the price of welding per unit length. This paper uses dollars for all the price unit. \(L_{\text{cut}}\) is the length of cut steel, which is calculated using equation (13):

\[
L_{\text{cut}} = \pi D_0 NH + 2e(NH + 1) + \frac{\pi D_0}{2} + e
\]  

(13)

while \(L_{\text{weld}}\) is the length of welded steel, which is calculated using equation (14):

\[
L_{\text{weld}} = e(NH + 1)
\]  

(14)

where \(e\) is the clear distance between each hole (see Figure 1). The design should fulfil the following limitations (see Figure 2 and Figure 3) as shown in equation (15)-equation (26):
\[
g_7 = V_{omax} - P_{vy} \leq 0
\]
(21)
\[
g_8 = V_{supmax} - P_v \leq 0
\]
(22)
\[
g_9 = M_{amax} - M_{wmax} \leq 0
\]
(23)
\[
g_{10} = P_0 + M - M_p \leq 0
\]
(24)
\[
g_{11} = V_{tee} - 0.5P_{vy} \leq 0
\]
(25)
\[
g_{12} = Y_{max} - \frac{L}{360} \leq 0
\]
(26)

**Figure 2.** Horizontal shear in cellular beam. Adapted from [9].

**Figure 3.** Olander’s curved beam approach. Adapted from [9].

where:
- \(H_s\) is the total height of the cellular beam,
- \(M_u\) is the maximum moment,
- \(M_p\) is cellular beam’s plastic moment capacity,
- \(V_{hmax}\) is maximum horizontal shear,
**P**\textsubscript{vh} is the total horizontal shear capacity of tee section, 
**V**\textsubscript{omax} is maximum shear at the opening,
**P**\textsubscript{vy} is the total vertical shear capacity of tee section,
**V**\textsubscript{supmax} is the maximum shear at support,
**P**\textsubscript{v} is total shear capacity of tee section,
**M**\textsubscript{aamax} is the maximum moment at A-A section as shown in Figure 2,
**M**\textsubscript{wmax} is the maximum permitted web post moment,
**P**\textsubscript{o} and **M**\textsubscript{p} are the internal forces on the web as shown in Figure 3,
**V**\textsubscript{tee} is the vertical shear on the tee at \( \theta = 0^\circ \) of web opening, and
**Y**\textsubscript{max} is the maximum deflection of the cellular beam.

### 4. Cellular beam cost optimization procedures

The PSO, DE, SOS, and ABC algorithms were coded in MATLAB R2019a. Optimization runs were executed on a laptop with a 3.18 GHz Intel Core i5-7200U processor and 8GB of RAM memory. The algorithms search an optimal selection of a root beam, the diameters of holes, and the total number of holes in the beam. The program will run until it reaches the maximum iteration set.

Whenever the solution violates the constraints, $1000 penalty is added to the objective function. This study uses 50 populations and 30 repetitions for each to examine the algorithms’ consistency. This study considers the difference of function evaluation in each algorithm. Therefore, the number of iterations is taken as 200 iterations for PSO, ABC, and DE, while SOS uses 50 iterations as it has 4 function evaluations. The parameters for each algorithm are shown in Table 1.

| PSO       | DE         | SOS         | ABC         |
|-----------|------------|-------------|-------------|
| \( n = 50 \) | \( n = 50 \) | \( n = 50 \) | \( n = 50 \) |
| \( c_1 = 1 \) | \( F = 0.2 - 0.8 \) | \( p_{CR} = 0.2 \) | \( n_{Onlooker} = 50 \) |
| \( c_2 = 1 \) | \( p_{CR} = 0.2 \) | \( L = 90 \) | \( L = 90 \) |
| \( w = 0.9 \) | \( a = 1 \) | \( a = 1 \) | \( a = 1 \) |

Note: \( n \) = population size/ecosystem size/colony size; \( c_1 \) = cognitive coefficient; \( c_2 \) = social coefficient; \( w \) = inertia weight; \( F \) = scaling factor; \( p_{CR} \) = crossover probability; \( n_{Onlooker} \) = number of onlooker bees; \( L \) = trial limit; \( a \) = acceleration coefficient upper bound

The results from four algorithms were then analysed to compare their performance using statistical analysis, such as the median, the average, and the standard deviation. The optimization process flow chart of the cellular beam is shown in Figure 4.

### 5. Test Problems and Results

A four-meter span beam with simply supported idealization as shown in Figure 5 is selected for problem example [1, 9]. The problem example is taken from several previous studies as the benchmark to compare the results. The beam is loaded with a 50 kN point live load at the center of the beam and a 5 kN/m uniformly-distributed dead load including the cellular beam’s weight. The steel used for design is Grade 50 and has the modulus elasticity of 205 kN/mm\(^2\). The steel weight density is 78.5 kN/m\(^3\). \( P_1 \), \( P_2 \), and \( P_3 \) is taken as $0.85, $0.3, and $1 respectively.
To solve the problem, three variables, that are, the selection of a root beam, the diameter of holes, and the total number of holes in the beam need to be randomized. The first variable is selected from a data set that is composed of 89 Universal Beam (UB) sections from 254 x 102 x 28 to 914 x 419 x 388. The diameter of holes and the total number of holes is taken between 180 mm to 600 mm and 2 to 39 holes, respectively [1].
Table 2 shows that the four algorithms could find the optimal cost, that is $91.562, from 30 times of run in the program. The table also contains the median result of four algorithms, the statistical analysis from 30 repetitions, and the average of computational time. The success rate is counted based on the results with no constraints violated. The DE, SOS, and ABC have a 100% success rate while the PSO has only about 86%. The table shows that the SOS has the smallest standard deviation among other algorithms. It shows that the SOS has the best consistency compared to the DE, ABC, and PSO. This convergence behavior can be seen clearly through the convergence graph shown in Figure 6. The convergence graph is drawn from the best fitness from each function evaluation on the 15th trials for ABC, SOS, and DE while PSO is taken on the 12th trials. The graph only presents up to 5000th function evaluation, since there is no improvement on the cost optimization afterward. The DE is the first algorithm that found the optimal cost, followed by the SOS. On the other hand, ABC and PSO are not able to find optimal cost. Figure 7 shows the iteration process from the initialization to the final design, which in this paper PSO algorithm is taken as an example.

| Variables | PSO  | DE   | SOS  | ABC  |
|-----------|------|------|------|------|
| $x_1$     | 85   | 85   | 85   | 85   |
| $x_2$     | 249  | 249  | 249  | 249  |
| $x_3$     | 14   | 14   | 14   | 14   |
| Best ($)  | 91.562 | 91.562 | 91.562 | 91.562 |
| Average ($) | 97.633 | 92.939 | 91.562 | 92.371 |
| Worst ($) | 103.577 | 123.564 | 91.562 | 94.060 |
| Stdev ($) | 6.070 | 5.841 | 0.644 | 0.644 |
| Median ($) | 98.399 | 91.562 | 91.562 | 92.235 |
| Success Rate | 26/30 | 30/30 | 30/30 | 30/30 |
| Average Time (s) | 0.2754 | 0.4651 | 0.2445 | 1.0150 |

Figure 6. Convergence behaviour of PSO, DE, ABC, and SOS algorithms.
Figure 7. PSO results for: (a) initial design, (b) 1st iteration, (c) 3rd iteration, (d) 13th iteration, (e) final design.

6. Conclusions

This paper compares four algorithms, that is, the SOS, ABC, DE, and PSO to optimize the cost of a cellular beam using a case study. The optimization takes UB sections, hole diameter, and the total number of holes into considerations. Using 200 numbers of iterations, the optimal cost is $91.5622. Comparing all the results, PSO performed worst in this case with a success rate of 26 out of 30. On the other hand, SOS performs consistently in all 30 independent runs. The SOS also gives the best average, consistency, and convergence behavior. Therefore, the SOS is the most suitable algorithm to optimize the cost of cellular beam.

References

[1] Erdal F, Dogan E and Saka M P 2011 Optimum design of cellular beams using harmony search and particle swarm optimization J. Constr. Res. 67 (2) 237-247
[2] Soltani M R, Bouchair A and Mimoune M 2012 Nonlinear FE analysis of the ultimate behaviour of steel castellated beams J. Constr. Res. 70 101-114
[3] Mehetre A J, Talikoti R S and Sonawane P B 2020 Experimental research on equivalent rectangular opening castellated beam with fillet corner Int. J. Recent Technol. 8 (5) 5415-20
[4] Swayamsiddha S 2020 Bio-inspired algorithms: principles, implementation, and applications to wireless communication Nature-Inspired Computation and Swarm Intelligence 4 49-63
[5] Kennedy J and Eberhart R 1995 Proc. of IEEE Int. Conf. on Neural Networks (New York) 4 1942-48
[6] Storn R and Price K 1997 Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces J. Glob. Optim. 11 341–359
[7] Cheng M Y and Prayogo D 2014 Symbiotic organisms search: a new metaheuristic optimization algorithm Comput. Struct. 139 98-112
[8] Karaboga D and Basturk B A 2007 Powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm *J. Glob. Optim.* **39** (3) 459-471

[9] Kaveh A and Shokohi F 2014 Cost optimization of castellated beams using charged system search algorithm *Iran J. Sci. Technol.* **38** 235-249