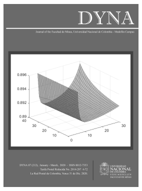
UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS



DYNA
http://dyna.medellin.unal.edu.co/

Coupling SAP 2000 with ABC algorithm for truss optimization

Rafael Baldissera dos Santos & Jorge Luis Palomino-Tamayo

Engineering School, Federal University of Rio Grande do Sul. Porto Alegre, Brasil. rafael_baldissera@hotmail.com, jorge.tamayo@ufrgs.br

Received: May 9th, 2019. Received in revised form: December 3rd, 2019. Accepted: January 14th, 2020.

Abstract
In this work, a numerical tool is implemented for size optimization of spatial truss structures. To achieve this goal, the metaheuristic artificial bee colony (ABC) algorithm based on the vivid behavior of honey bees is combined with the commercial software package SAP 2000 via its Application Programming Interface (API). Linking the optimization algorithm to SAP 2000 will allow access to the complete finite element library and potential of the commercial software. In order to do so, the numerical tool is developed within the Excel environment by using macros with the Visual Basic Programming Language. A complete picture is then presented to practitioners of the elaborated numerical tool, which can be used in daily design applications. The correct functionality is demonstrated by means of two challenging practical examples. Extension to other optimization applications can be easily adapted.

Keywords: size optimization; truss; ABC algorithm; SAP 2000.

Acoplamiento de SAP 2000 con el algoritmo ABC para la optimización de armaduras

Resumen
En este trabajo, se implementa una herramienta numérica para la optimización de tamaño de armaduras espaciales. Para este propósito, el algoritmo metaheurístico de abejas (ABC) basado en el comportamiento real de abejas se combina con el programa comercial SAP 2000 por medio de su interface de programación (API). Enlazar el algoritmo de optimización a SAP 2000 permitirá tener acceso a la biblioteca completa de elementos finitos e potencialidad del programa comercial. La herramienta numérica se desarrolla dentro del ambiente de macros de Excel utilizando el lenguaje de programación Visual Basic. Un marco completo de la herramienta elaborada se presenta a los diseñadores, la cual puede ser utilizada en aplicaciones diarias de diseño. El correcto funcionamiento se demuestra por medio de dos ejemplos prácticos desafiantes. Extensión a otras aplicaciones de optimización pueden ser fácilmente adaptadas.

Palabras clave: optimización de tamaño; armadura; algoritmo ABC; SAP 2000.

1. Introduction

Currently, automation of the design tasks in engineering projects is required to meet project schedules. In this context, the design of trusses constitutes an active field of interest in research. Automation can be accomplished by elaborating customized numerical tools, in which commercial softwares such as SAP 2000, ETABS, ANSYS, etc., among others, may play an important role. SAP 2000, in particular, is a well-known program for structural analysis and design, used in most design offices due to its versatility and friendly output interface. Moreover, it allows users to customize their design tasks by means of its Application Programming Interface (API). That is, the user can access the program by scripts written in Visual Basic programming language without accessing the graphical interface directly. In turn, the graphical interface enables better visualization of results in real time and also exporting these results to Excel spreadsheets. Combining SAP 2000 with optimization algorithms to minimize the weight of truss structures, can lead to more feasible solutions in relation to manual solutions. For this purpose, metaheuristic algorithms, which do not require the evaluation of function derivatives and surpass local minimums, can be used.

How to cite: Baldissera dos Santos, R. and Palomino-Tamayo, J.L, Coupling SAP 2000 with ABC algorithm for truss optimization. DYNA, 87(212), pp. 102-111, January - March, 2020.

© The author; licensee Universidad Nacional de Colombia.

Revista DYNA, 87(212), pp. 102-111, January - March, 2020, ISSN 0012-7353
DOI: http://doi.org/10.15446/dyna.v87n212.79614

To optimize truss structures the well-known Artificial Bee Colony algorithm (ABC) proposed by Karaboga [1], based on the productive behavior of honey bees around a given hive, is selected here. This algorithm has proved to perform well for truss optimization problems in other works [2-5] by attaining global minimums, even for complex functions, and also due to its competences compared to other important algorithms, such the cultural [6], ray [7], hybrid harmony search [8], and surrogate model [9], all of which have also been applied successfully to truss optimization problems. Meanwhile, other applications of ABC to other problems can be found in references [10,11]. Furthermore, in [12] a genetic algorithm procedure is combined with SAP2000 for optimizing truss structures by using parallel computing. To minimize the computational time, several processors and computers are used in the computation.

The aim of this work is to give a complete picture of the construction of a numerical tool for the size optimization of spatial trusses using two computer programs commonly encountered in most design offices, i.e., SAP 2000 and Excel, rather than discussing improvements for the ABC algorithm itself, all of which can be found in other works [4,5]. The coupling is developed by means of the Visual Basic programming language of macros in Excel, in which the ABC algorithm and SAP 2000 sentences are coded. Then, the necessary steps to perform this linking procedure are presented. The resulting tool can be further extended to spatial frames and shell optimization problems with minor modifications of the developed code. The complete listing code then can be requested from the authors by mailing the second author.

Finally, some conclusions are then given in the discussion section in order to enhance the performance of the combined tool because of the studied examples. It should be highlighted that this tool is able to cope with imposed constraints expressed in terms of limiting displacements and stresses. Shortcomings due to excessive computational times are also commented. Because the original ABC deals only with unconstrained problems, a commonly static penalty approach is used to convert the constrained problem into an unconstrained one as suggested in other references [3-5].

2. The ABC algorithm coupled with SAP 2000

In the following, subsections comments are made regarding the original ABC algorithm proposed by Karaboga [1], about the definition of the unconstrained weight optimization function, and to the steps needed to couple SAP 2000 with ABC via macros of Excel.

2.1. The original ABC algorithm

The ABC algorithm proposed by Karaboga [1] was initially applied to find global minimums of some complex functions. Since then, it has been applied to truss optimization problems with the inclusion of some modifications to enhance its performance [2-5]. The approach is based on the intelligent foraging behavior of honey bees in a given hive and surroundings. The best solution is accomplished by dividing the bee objects into different tasks. For this purpose, three groups of bees are used: employed, onlooker and scout bees. The location of each food source and the amount of nectar refer here to the design variables and fitness function of the problem at hand, respectively. Each food source corresponds to a possible solution of the problem and the best solution corresponds to the position of the most profitable source. In the truss optimization context, the bar areas and the self-weight of the truss are, respectively, the design variables and fitness function. In this paper, the fitness function is taken as the inverse of the weight of the truss [5] as will be shown later.

Employed bees leave the hive to search for promising food sources. Thereafter, they return to the hive to share their knowledge about a particular food source with their nest mates. Then, the profitability of the source is shared with a certain probability with the onlooker bees.

The exchange of information is guided by a probability based on the amount of nectar (fitness function) that an unemployed bee (onlookers or scout) will determine in a new food source in the neighborhood of its choice. Each bee that finds a solution becomes an employed bee, and when its supply is exhausted, it becomes a scout bee. The parameters of the ABC algorithm are the number of food sources (*NS*), which here is considered to be equal to the sum of the number of employed and onlooker bees, and the dimension of each solution vector D, which represents the number of groups of bar elements to be optimized. The steps of the algorithm are briefly described next. For further details, see reference [5].

Step 1: A initial random population is generated$(X_1, \ldots . X_{Ns})$, with $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ established as follows:

$$x_{ij} = x_{\min\_j} + rand[0,1](x_{\max\_j} - x_{\min\_j}) \qquad (1)$$

where $j = 1,2, \ldots D$, $i = 1,2 \ldots NS$ and $x_{\max\_j}$, x_{min_j} are the upper and lower bounds of the design variable j, and $rand[0,1]$ represent a random number between 0 and 1. After initialization of the population, the fitness of each food is evaluated.

Step 2: Beginning of the employed bee phase. Each employed bee searches the neighborhood of its current food source to determine a new food source in the following manner:

$$v_{ij} = x_{ij} + rand[-1,1](x_{ij} - x_{kj}) \qquad (2)$$

where $k = 1,2, \ldots NS$ and $j = 1,2 \ldots D$ are randomly chosen indexes. It must be noted that subscript, k, in the last term of eq. (2) must be different from subscript i and $rand[-1,1]$ represents a random number between -1 and 1. Values of v_{ij}, which exceed their limit values, are set to their limit values.

Step 3: After generating the new food source (design variable), the amount of nectar it contains (fitness function) will be evaluated together with a greedy selection process.

When, the quality of the new source is better than the current position, the employed bee leaves its position and moves to the new food source, i.e., if the fitness of the new food source is better than that of X_i, the new food source replaces X_i in the population and becomes a new member.

Step 4: Beginning of the onlooker bee phase. First an onlooker bee selects a food source according to the information provided by the employed bees. The probability of selection of each food source, p_i is determined by means of eq. (3).

$$p_i = \frac{f_i}{\sum_{i=1}^{NS} f_i} \quad (3)$$

where f_i is the fitness value of the food source X_i. After selecting a food source, the onlooker bee generates a new food source by means of eq. (2). Once the new source is generated, it will be evaluated besides a greedy selection as in the case of the employed bee phase.

Step 5: Beginning of scout bee phase. If a candidate solution of food source cannot be further improved after a number of trials, this is abandoned and the associated employed bee becomes a scout, whose new food source is generated by means of eq. (4).

$$v_{ij} = x_{\min\_j} + rand[0,1](x_{\max\_j} - x_{min\_j}) \quad (4)$$

with $j = 1,2 \ldots D$. The abandoned food source is replaced by a randomly generated food source by evaluating the fitness function. In the ABC algorithm, the predetermined number of trials for abandoning a food source is named $limit$. At most one employed bee in each cycle becomes a scout.

Step 6: If a termination condition is met, the process is stopped and the best food source is reported. Otherwise, the algorithm goes to Step 2, and an iteration cycle is established. The termination condition is defined here when a maximum number of iterations or cycles called $MITER$ (predefined by the user) is reached. Often $limit$ is taken as $MITER/3$. For the studied examples $MITER$ is arbitrarily chosen as 2000 iterations, while $limit$ is 667. A flowchart of the ABC algorithm showing all the aforementioned steps is depicted in Fig. 1. The fitness function is evaluated by means of the subroutine RODAR, which will be explained in more detail in section 2.3, and where the structural analysis of the truss is carried out by SAP 2000.

2.2. *Size optimization of trusses*

The most feasible solution is defined as a solution which resists all external actions with the lightest weight, without incurring in any violation of the imposed constraints. Indeed, the weight optimization process represents the continuous modification of cross section areas, x_{ij}, of the members that compose a given truss, which in turn are subjected to constraints in terms of displacements and stresses. The corresponding optimization is formulated based on the following solution vector of areas X.

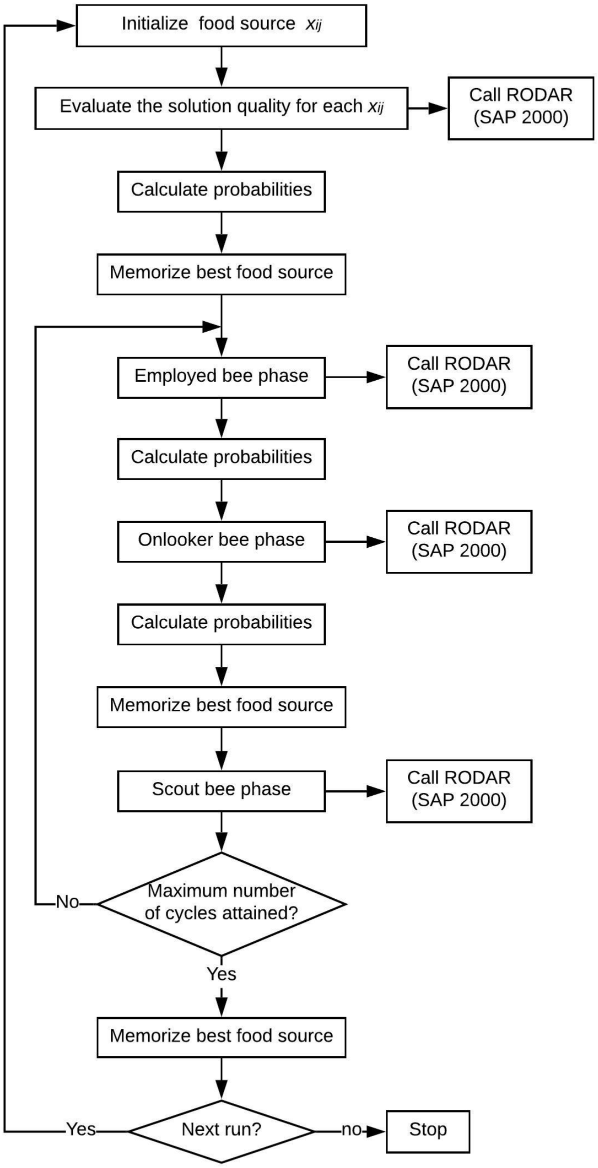


Figure 1. Flowchart for ABC algorithm coupled with SAP 2000.
Source: The Authors.

$$X = (x_1, x_2, x_3 \ldots x_D)^T \quad (5)$$

$$x_{\min\_n} \leq x_n \leq x_{\max\_n} \quad (6)$$

with $n = 1,2,3 \ldots D$, subjected to the following constraints:

$$g_i(X) = \left|\frac{\sigma_i}{\sigma_{ai}}\right| - 1 \leq 0 \quad (7)$$

$$g_j(X) = \left|\frac{\delta_j}{\delta_{aj}}\right| - 1 \leq 0 \quad (8)$$

$$g_k(X) = \left|\frac{\sigma_{kb}}{\sigma_{akb}}\right| - 1 \leq 0 \quad (9)$$

where subscripts are $i,k = 1,2,3 \ldots NM$, $j = 1,2,3 \ldots ND$, and other parameters such as NM and ND represent the number of elements in the truss and the number of degrees of freedoms, respectively. In eq. (5), X is a candidate solution, $x_{\min\_n}$ and $x_{\max\_n}$ are the upper and lower limits of the design variable x_n, and D is the total number of design variables. That is, x_n represents the cross section area A_n of the nth element, whereas g_i, g_j and g_k are the normalized constrained functions due to yield stresses, nodal displacements, and buckling stresses, respectively. Here, σ_i is the current stress in the ith element, σ_{ai} is the allowable stress in the same element, δ_j is the displacement in the direction of the degree of freedom j and δ_{aj} is the allowable displacement in that direction. Finally σ_{kb} is the compressive stress in the kth element and σ_{akb} is the associated allowable buckling stress. In this manner, the penalty function is defined by means of eq. (10)

$$P(X) = W(X).K.\left\{\sum_{i=1}^{NM} g_i(X) + \sum_{j=1}^{ND} g_j(X) + \sum_{k=1}^{NM} g_k(X)\right\} \tag{10}$$

with,

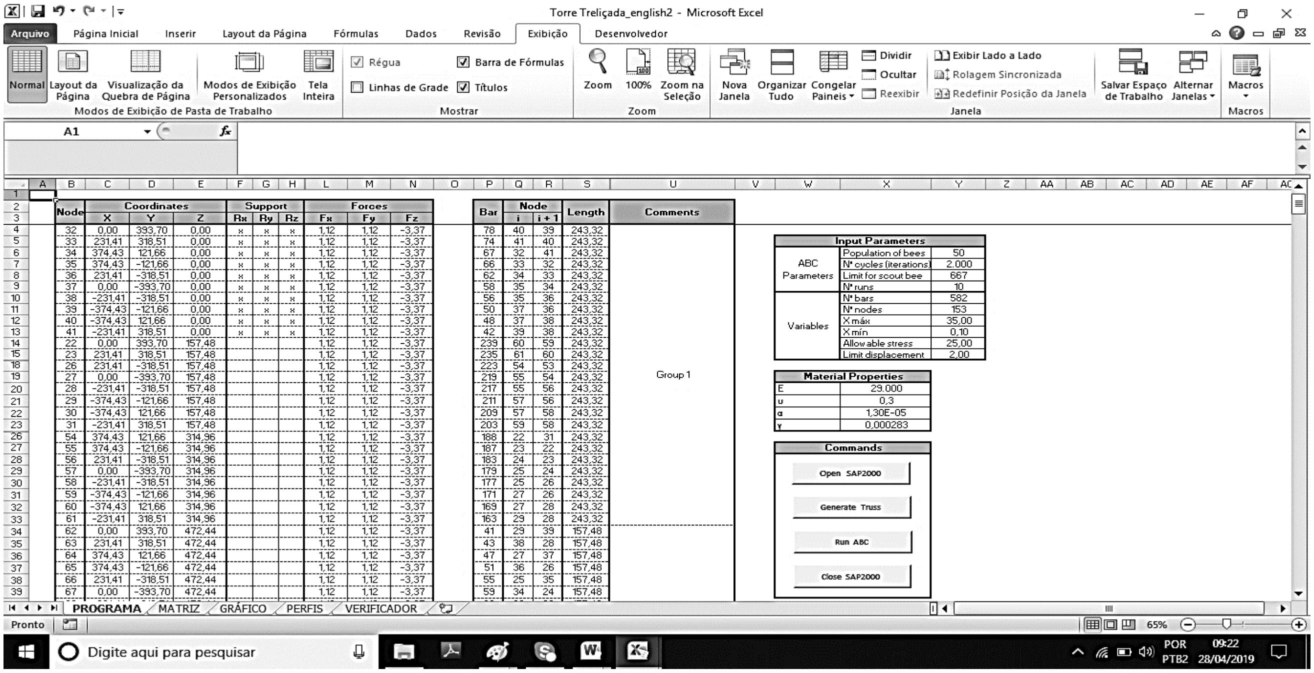
$$W(X = A) = \sum_{j=1}^{NM} A_j L_j \rho_j \tag{11}$$

where A_j, L_j and ρ_j are the cross section area, length and material density of the jth element, respectively. The variable $W(A)$ is the weight of the truss, K is a penalty constant taken as 1.0, and g_m with $m = i,j,k$ represent the amount of violation of the corresponding constraint. Each time that a constraint is violated, the value of g_m is activated, otherwise it is zero. Eq. (12) presents the objective function to be minimized in the process. Indeed, the constraint optimization problem becomes an unconstrained one by using the penalty approach [5]. The fitness function used in eq. (3) is then determined by $1/W(X)$.

$$W_c = W(X) + P(X) \tag{12}$$

2.3. Interface between SAP 2000 and ABC

During the size optimization, process the ABC algorithm evaluates a series of possible combinations until the best feasible solution is finally attained. During this process, the cross section areas of each bar change at each iteration. As a result, stresses and displacements in the truss are evaluated at each new combination. The Application Programming Interface (API) of SAP 2000 is therefore used to control the program, i.e., API allows users to write external applications in Visual Basic Application (VBA) to manage the graphical interface of the program. In the installation directory of SAP 2000, there exists a file named CSI_OAPI_Documentation [13], in which a list of commands is provided for this purpose. In this paper, a computer program is written in VBA via macros of Excel to couple SAP 2000 with the ABC algorithm. Input geometric and material property datasets of a given truss are read by means of Excel spreadsheets as depicted in Fig. 2. As may be observed, various tables are defined to read the complete input data. They are better visualized and explained in the following Figs. 3-6. For instance, the nodal coordinates of the truss, prescribed displacements and forces are defined in the table shown in Fig. 3.



Node	Coordinates			Support			Forces		
	X	Y	Z	Rx	Ry	Rz	Fx	Fy	Fz
32	0,00	393,70	0,00	x	x	x	1,12	1,12	-3,37
33	231,41	318,51	0,00	x	x	x	1,12	1,12	-3,37
34	374,43	121,66	0,00	x	x	x	1,12	1,12	-3,37
35	374,43	-121,66	0,00	x	x	x	1,12	1,12	-3,37
36	231,41	-318,51	0,00	x	x	x	1,12	1,12	-3,37
37	0,00	-393,70	0,00	x	x	x	1,12	1,12	-3,37
38	-231,41	-318,51	0,00	x	x	x	1,12	1,12	-3,37
39	-374,43	-121,66	0,00	x	x	x	1,12	1,12	-3,37
40	-374,43	121,66	0,00	x	x	x	1,12	1,12	-3,37
41	-231,41	318,51	0,00	x	x	x	1,12	1,12	-3,37
22	0,00	393,70	157,48				1,12	1,12	-3,37
23	231,41	318,51	157,48				1,12	1,12	-3,37
26	231,41	-318,51	157,48				1,12	1,12	-3,37
27	0,00	-393,70	157,48				1,12	1,12	-3,37
28	-231,41	-318,51	157,48				1,12	1,12	-3,37
29	-374,43	-121,66	157,48				1,12	1,12	-3,37
30	-374,43	121,66	157,48				1,12	1,12	-3,37
31	-231,41	318,51	157,48				1,12	1,12	-3,37
54	374,43	121,66	314,96				1,12	1,12	-3,37
55	374,43	-121,66	314,96				1,12	1,12	-3,37
56	231,41	-318,51	314,96				1,12	1,12	-3,37
57	0,00	-393,70	314,96				1,12	1,12	-3,37
58	-231,41	-318,51	314,96				1,12	1,12	-3,37
59	-374,43	-121,66	314,96				1,12	1,12	-3,37
60	-374,43	121,66	314,96				1,12	1,12	-3,37
61	-231,41	318,51	314,96				1,12	1,12	-3,37
62	0,00	393,70	472,44				1,12	1,12	-3,37
63	231,41	318,51	472,44				1,12	1,12	-3,37
64	374,43	121,66	472,44				1,12	1,12	-3,37
65	374,43	-121,66	472,44				1,12	1,12	-3,37
66	231,41	-318,51	472,44				1,12	1,12	-3,37
67	0,00	-393,70	472,44				1,12	1,12	-3,37

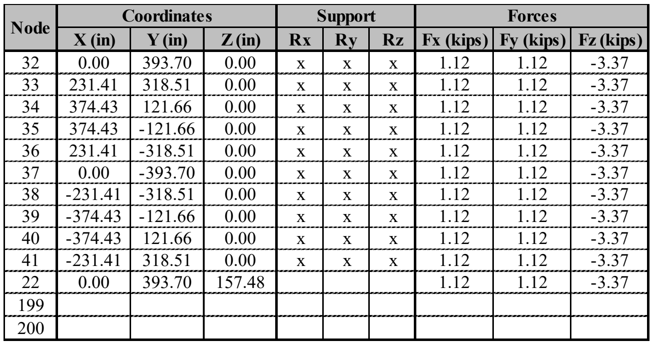
Bar	Node		Length	Comments
	i	i + 1		
78	40	39	243,32	Group 1
74	41	40	243,32	
67	32	41	243,32	
66	33	32	243,32	
62	34	33	243,32	
58	35	34	243,32	
56	35	36	243,32	
50	37	36	243,32	
48	37	38	243,32	
42	39	38	243,32	
239	60	59	243,32	
235	61	60	243,32	
223	54	53	243,32	
219	55	54	243,32	
217	55	56	243,32	
211	57	56	243,32	
209	57	58	243,32	
203	59	58	243,32	
188	22	31	243,32	
187	23	22	243,32	
183	24	23	243,32	
179	25	24	243,32	
177	25	26	243,32	
171	27	26	243,32	
169	27	28	243,32	
163	29	28	243,32	
41	29	39	157,48	
43	38	28	157,48	
47	27	37	157,48	
51	36	26	157,48	
55	25	35	157,48	
59	34	24	157,48	

Input Parameters		
ABC Parameters	Population of bees	50
	N° cycles (iterations)	2.000
	Limit for scout bee	667
	N° runs	10
Variables	N° bars	582
	N° nodes	153
	X máx	35,00
	X mín	0,10
	Allow able stress	25,00
	Limit displacement	2,00

Material Properties	
E	29.000
υ	0,3
α	1,30E-05
γ	0,000283

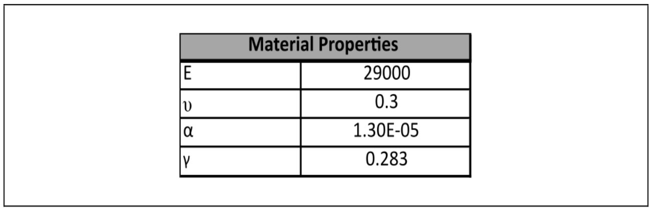
Commands

Figure 2. Excel spreadsheet for defining input geometry of the truss.
Source: The Authors.



Node	Coordinates			Support			Forces		
	X (in)	Y (in)	Z (in)	Rx	Ry	Rz	Fx (kips)	Fy (kips)	Fz (kips)
32	0.00	393.70	0.00	x	x	x	1.12	1.12	-3.37
33	231.41	318.51	0.00	x	x	x	1.12	1.12	-3.37
34	374.43	121.66	0.00	x	x	x	1.12	1.12	-3.37
35	374.43	-121.66	0.00	x	x	x	1.12	1.12	-3.37
36	231.41	-318.51	0.00	x	x	x	1.12	1.12	-3.37
37	0.00	-393.70	0.00	x	x	x	1.12	1.12	-3.37
38	-231.41	-318.51	0.00	x	x	x	1.12	1.12	-3.37
39	-374.43	-121.66	0.00	x	x	x	1.12	1.12	-3.37
40	-374.43	121.66	0.00	x	x	x	1.12	1.12	-3.37
41	-231.41	318.51	0.00	x	x	x	1.12	1.12	-3.37
22	0.00	393.70	157.48				1.12	1.12	-3.37
199									
200									

Figure 3. Nodal coordinates, prescribed displacements and forces.
Source: The Authors.

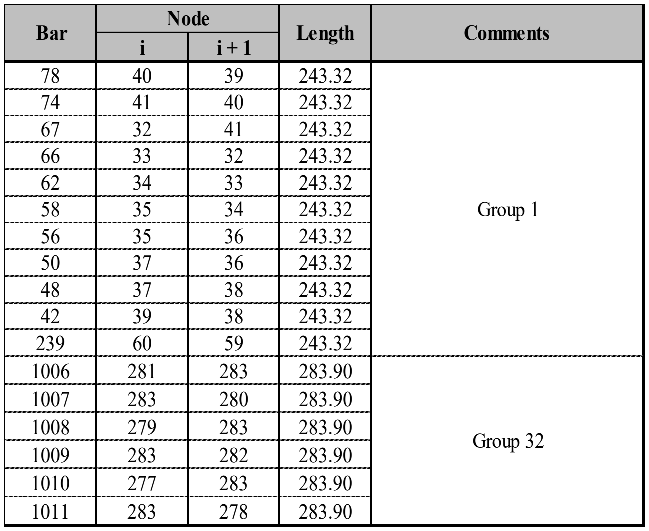


Material Properties	
E	29000
υ	0.3
α	1.30E-05
γ	0.283

Figure 4. Material properties.
Source: The Authors.

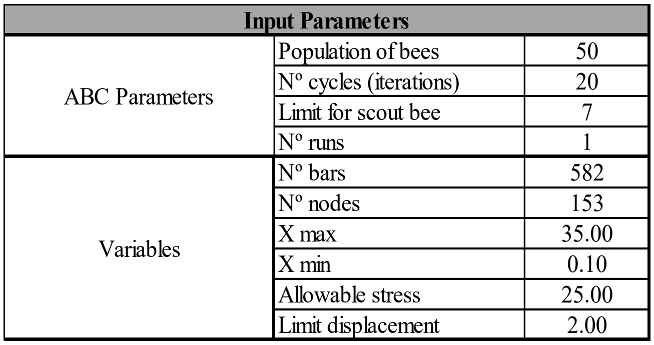
Nodal coordinates are defined in columns C, D and E of the current Excel spreadsheet for global coordinates along X, Y and Z directions, respectively. Meanwhile, corresponding restrictions are identified by a symbol “x” in columns F, G and H. Concentrated forces are specified in columns L, M and N for the loaded nodes. Furthermore, truss material properties are input into the table shown in Fig. 4, where E is the elasticity modulus of the material, v is the Poisson coefficient, α is the thermal coefficient and γ is the material density, whereas element connectivity is registered in columns Q and R in the table shown in Fig. 5. Finally, input parameters for the ABC algorithm are displayed in Fig. 6. For all the aforementioned data, units are input consistently.

After the entire dataset is defined, the coupled algorithm is executed in sequence by means of the buttons displayed in Fig. 7. Firstly, the Open SAP 2000 button calls and opens the SAP 2000 program via macros, secondly the Generate Truss button is pressed to generate the geometry of the truss and the corresponding material properties are assigned to each element, see e.g., the created dome truss shown in Fig. 8. Next, the Run ABC button is pressed, and the ABC algorithm is executed following the flowchart depicted in Fig. 1. Finally, after all runs are executed, the SAP 2000 program is closed by pressing the Close SAP 2000 button. As an example, some sentences of the computer subroutine called RODAR used in Fig. 1 are listed next. This subroutine is permanently accessed by the ABC algorithm to evaluate the fitness function of the problem. Firstly, the solution vector *sol*(), which accesses as a double type variable, contains the randomly generated areas of the members according to eq. (1). This candidate solution *sol*() is assigned to the truss members during the first loop in the current iteration. Thereafter the SAP 2000 program is executed, and the corresponding displacements are computed for a given load case. Finally, internal forces are recovered and corresponding axial stresses are computed at each member in the final loop. Both displacements and stresses are then used to evaluate the constraints established in eq. (7), eq. (8) and eq. (9).



Bar	Node		Length	Comments
	i	i + 1		
78	40	39	243.32	Group 1
74	41	40	243.32	
67	32	41	243.32	
66	33	32	243.32	
62	34	33	243.32	
58	35	34	243.32	
56	35	36	243.32	
50	37	36	243.32	
48	37	38	243.32	
42	39	38	243.32	
239	60	59	243.32	
1006	281	283	283.90	Group 32
1007	283	280	283.90	
1008	279	283	283.90	
1009	283	282	283.90	
1010	277	283	283.90	
1011	283	278	283.90	

Figure 5. Element connectivity.
Source: The Authors.



Input Parameters		
ABC Parameters	Population of bees	50
	Nº cycles (iterations)	20
	Limit for scout bee	7
	Nº runs	1
Variables	Nº bars	582
	Nº nodes	153
	X max	35.00
	X min	0.10
	Allowable stress	25.00
	Limit displacement	2.00

Figure 6. Input parameters for the ABC algorithm.
Source: The Authors.

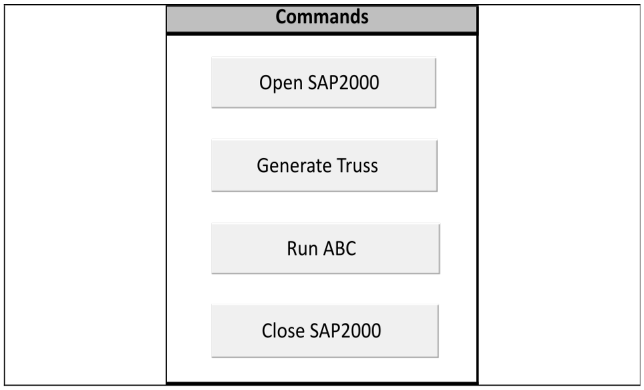


Figure 7. SAP 2000 model.
Source: The Authors.

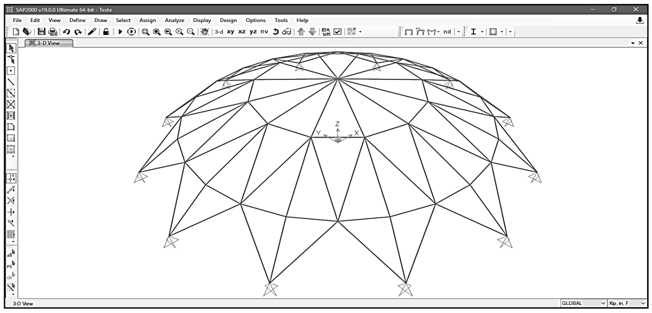

SAP2000 v19.0.0 Ultimate 64-bit - Teste
File Edit View Define Draw Select Assign Analyze Display Design Options Tools Help
3-D View
GLOBAL
Kip, in, F

Figure 8. Dome truss in SAP 2000.
Source: The Authors.

```
Public Sub RODAR(sol() As Double)
.......
'Zero self-weight
ModValue(7) = 0

'Define section and materials
Dim Area(D - 1) As Double
Dim SectionName(D - 1) As String
For j = 0 To D - 1
      Area(j) = sol(j)
      SectionName(j) = sol(j)
      ret=mySapObject.SapModel.PropFrame.SetGeneral
            (SectionName(j), "MATERIAL", 1, 1, Area(j), 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, -1)
       ret=mySapObject.SapModel.FrameObj.SetSection
           (MyName(j), SectionName(j))
       ret=mySapObject.SapModel.PropFrame.SetModifiers
           (SectionName(j), ModValue)
Next j
!Run analysis
ret = mySapObject.SapModel.Analyze.RunAnalysis

'Select load combination and get displacements
ret=mySapObject.SapModel.Results.Setup.SetCaseSelectedFor
     Output("CargasPontuais")
ret=mySapObject.SapModel.Results.JointDispl("ALL",eItemTy
     peElm_GroupElm, numberResults, obj, elm, loadCase,
     stepType, stepNum, u1, u2, u3, r1, r2, r3)
For k = 0 To numberResults - 1
       displacement1(k) = u1(k)
       displacement2(k) = u2(k)
       displacement3(k) = u3(k)
Next k

'Generate stresses
ret=mySapObject.SapModel.Results.FrameForce("ALL",eItemT
     ypeElm_GroupElm, numberResults, obj, objSta, elm,
      elmSta, loadCase, stepType, stepNum, P, V2, V3, t, M2,
      M3)
stress(0) = P(0) / Area(0)
k = 1
For j = 1 To numberResults - 1
       If obj(j) <> obj(j - 1) Then
           stress(k) = P(j) / Area(k)
           k = k + 1
       Else
       End If
Next j
……
End Sub
```

It should be highlighted that during the iterative process, the algorithm will call subroutine RODAR, n times at each iteration or cycle, with n being the total number of bees in the colony (*NS*). Thus, this subroutine commands the structural analysis of the truss by means of SAP 2000. For instance, the number of times that a given truss is analyzed in one computer run can be estimated by the following expression.

$$N = \frac{NS}{2} + MITER * NS \qquad (13)$$

where N is the total number of analyses or evaluations of the fitness function in a given computer run for a predefined number of $MITER$ iterations. Commonly, a given example is executed at least ten times [2-6] in order to verify the robustness of the algorithm, i.e., the obtained results should not change considerably from run to run. Furthermore, the total number of analyses in eq. (13) should be multiplied by the number of computer runs in order to assess the overall number of evaluations in a given optimization problem. During the analysis process, the coupled algorithm VBA-SAP 2000 proved to be time consuming, e.g., the optimization of a simple planar truss conformed by 10 bars with 400 iterations, and for one computer run took almost 4.5 hours on an average desktop computer. The excessive computational time reported by SAP 2000 is linked to the execution of other functions such as the creation of result tables, reports, display of deformed configurations, etc., all of which cannot be disabled during execution and are time consuming. This truly constitutes a limitation, which can be surpassed by means of parallel computing. Indeed, a similar conclusion is reported by Ghozi and co-workers [12], in which SAP 2000 was coupled to genetic algorithms for truss optimization. The advantage of the coupling is better highlighted when various PCs are used. It is inferred that the use of one single computer can still be allowed for some examples, but it can be forbidden for others. To speed up the validation process of the optimization tool, another conventional computer subroutine to solve truss structures was also coded together with the ABC algorithm. Corresponding comparisons and execution times for these two options are properly discussed in section 4.

3. Applications

In this section two examples with other numerical results for truss size optimization are analyzed to show the capability of the implemented tool. The first and second examples consider continuous and discrete variables, respectively, with the latter variables being defined within the internal dataset of SAP 2000. For all examples, the bee population is defined as 50 with a number of iterations equal to 2000 ($MITER$), and a scout limit of $MITER$ /3.

3.1. One hundred twenty-bar dome truss

The previous dome structure depicted in Fig. 8 is optimized in this section via continuous variables. The dome geometry can be described by three rings in the following manner. The first ring, which passes by all supports at the base of the dome, has a diameter of 1251.02 in (31.77 m) with zero elevation from ground surface. The second ring, which is located at 118.11 in (3 m) from ground surface, has a diameter of 984.252 in (25 m). The third ring has a diameter of 546.61 in (13.88 m) and is located at a height of 230.315 in (5.85 m) from the ground surface. The last node in the dome is the highest one with 275.591 in (7 m) in elevation and is submitted to a point load of -13.49 kips (-60 kN). Other point loads of -6.744 kips (-30 kN) and -2.248 kips (-10 kN) are applied at nodes passing by the third and second rings, respectively.

In total the dome has 37 nodes and 120 bar elements, all of which have been categorized in seven different groups. Bar elements of group one are defined by the connectivity of the highest node with those passing by the third ring. Groups two and five are defined by members located in the plane of the third and second rings, respectively. Members of groups three and four join elements from the second and third rings, respectively. Finally, elements from group six and seven join the supported nodes with those of the second ring. The minimum allowable cross section area of each member is limited to 0.775 in^2 (5.0 cm^2). The allowable tensile stress is 0.6 f_y and the compressive stress is constrained by means of eq. (14).

$$\sigma_i = \begin{cases} \dfrac{(1-\lambda_i).f_y}{\left(\dfrac{5}{3}+\dfrac{3\lambda_i}{8C_c}-\dfrac{\lambda_i^3}{8C_c^3}\right)} & \lambda_i < C_c \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & \lambda_i \geq C_c \end{cases} \qquad (14)$$

where f_y is the yield stress of steel, E is the modulus of elasticity, λ_i is the slenderness ratio ($\lambda_i = kL_i/r_i$), k is the effective length factor, L_i is the length of the member, r_i is the radius of gyration and $C = \sqrt{2\pi^2 E/f_y}$. The material density is 0.288 lb/in^2 (7971.81 kg/m^3), $f_y = 58$ ksi (400 MPa), $E = 30450$ ksi (210000 MPa) and $r_i = 0.499A_i^{0.677}$ for pipe sections, in which A_i is the cross section of the ith member. In this example, the displacement of all nodes is limited to ±0.1969 in (±0.5 cm) for all directions. The results of the optimization are presented in Table 1, in which the best and worst results from ten independent runs are presented.

Table 1.
Comparisons of results for the 120-bar dome truss.

Group	Haididi and co-workers [2]		Present study	
	ABC	MABC	Best	Worst
1	3.2984	3.2985	3.2984	3.2982
2	2.7894	2.7928	2.7929	2.7908
3	3.8743	3.8748	3.8748	3.8745
4	2.5719	2.5719	2.5719	2.5716
5	1.1549	1.1501	1.1501	1.1533
6	3.3341	3.3328	3.3329	3.3327
7	2.7860	2.7838	2.7837	2.7842
Weight (W) (lb)	19908.03	19901.42	19901.30	19902.80
Violation (V)	-	-	-	-

Source: The Authors.

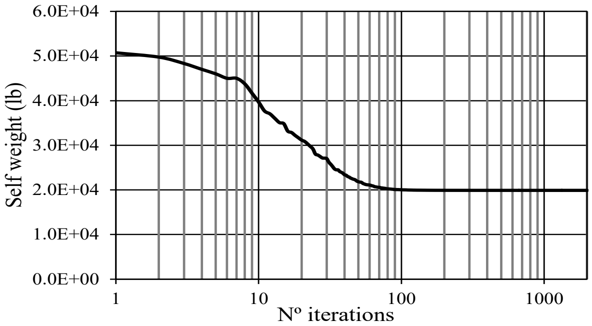


Figure 9.Penalized self-weight versus number of iterations.
Source: The Authors.

In addition, these results are also compared with other results [2], and then a good match is found for the final weight. The optimization history of the weight is depicted in Fig. 9. As may be observed, only 100 iterations are needed to practically obtain an optimized weight of 19901.3 lb (88.52 kN) without any violation of constraints.

3.2. *Five hundred eighty-two-bar tower truss*

In [5] the size discrete optimization of a tower truss of 582 bars is proposed. The geometry of this truss as created in a SAP 2000 environment is shown in Fig. 10. In total 32 groups are considered for size optimization. The tower has a total height of 3148.8 in (80 m), and its elliptical base defines two semi-axes of dimensions 10 m and 9.5 m. The steel has an elasticity modulus of $E = 29000$ ksi (200 GPa) and material density of $\rho = 0.283$ lb/in^2 (7833.4 kg/m^2). The maximum allowable stress in tension and compression is ±21.6 ksi (148.93 MPa), together with a limiting maximum displacement of ±3.15 in (8 cm) for all directions. Horizontal forces of 1.12 kips (5 kN) along the X and Y directions are applied to all nodes, meanwhile vertical forces of -6.74 (30 kN) and -3.37 kips (15 kN) are applied to all nodes in the upper and bottom part of the tower, respectively, along the Z direction. The solution is restricted to truss elements with slenderness ratios of 300 for tensile members, and 200 for compressive members.

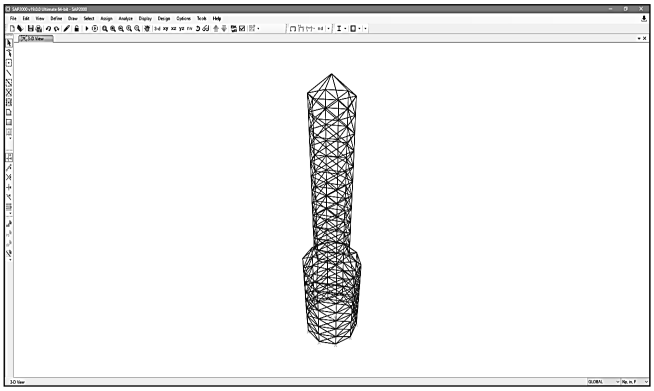


Figure 10. Tower truss in SAP 2000.
Source: The Authors.

Table 2.
Profile list from AISC code for the tower truss.

W8x21	W10x54	W14x99	W12x152	W24x229	W30x292
W10x22	W12x58	W10x100	W12x152	W36x230	W40x297
W8x24	W10x60	W16x100	W14x159	W44x230	W36x300
W6x25	W14x61	W21x101	W27x161	W12x230	W14x311
W12x26	W21x62	W24x104	W24x162	W14x233	W33x318
W8x28	W12x65	W12x106	W12x170	W30x235	W30x326
W12x30	W16x67	W14x109	W30x173	W27x235	W36x328
W14x30	W10x68	W21x111	W24x176	W33x241	W44x335
W8x31	W12x72	W10x112	W14x176	W36x245	W14x342
W10x33	W14x74	W27x114	W27x178	W40x249	W33x354
W14x34	W18x76	W30x116	W21x182	W24x250	W36x359
W8x35	W10x77	W24x117	W40x183	W12x252	W14x370
W16x36	W12x79	W33x118	W12x190	W14x257	W14x398
W14x38	W14x82	W18x119	W30x191	W27x258	W14x426
W10x39	W27x84	W14x120	W24x192	W36x260	W14x455
W8x40	W18x86	W21x122	W14x193	W30x261	W14x500
W12x40	W12x87	W24x131	W27x194	W44x262	W14x550
W14x43	W10x88	W14x132	W40x199	W33x263	W14x605
W12x45	W16x89	W12x136	W33x201	W40x277	W14x665
W10x45	W14x90	W14x145	W14x211	W12x279	W14x730
W14x48	W21x93	W24x146	W30x211	W24x279	
W10x49	W27x94	W27x146	W40x215	W36x280	
W12x50	W12x96	W21x147	W27x217	W14x283	
W12x53	W18x97	W36x150	W33x221	W33x291	

Source: The Authors.

Table 3.
Comparison of results for the 582-bar tower truss.

G	PSO [5]		MABC [5]		Present study	
1	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
2	W12x79	23.20	W18x86	25.30	**W21x93**	**27.30**
3	W8x24	7.08	W8x24	7.08	**W8x24**	**7.08**
4	W10x60	17.60	W10x60	17.60	**W10x60**	**17.60**
5	W8x24	7.08	W8x24	7.08	**W8x24**	**7.08**
6	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
7	W8x48	14.10	W10x49	14.40	**W10x45**	**13.30**
8	W8x24	7.08	W8x24	7.08	**W12x26**	**7.65**
9	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
10	W10x45	13.30	W12x53	15.60	**W10x22**	**6.49**
11	W8x24	7.08	W8x24	7.08	**W8x24**	**7.08**
12	W10x68	20.00	W21x62	18.30	**W16x67**	**20.00**
13	W14x74	21.80	W27x84	24.80	**W10x77**	**22.60**
14	W8x48	14.10	W10x45	13.30	**W16x67**	**20.00**
15	W18x76	22.30	W27x84	24.80	**W12x58**	**17.00**
16	W8x31	9.12	W8x31	9.12	**W8x24**	**7.08**
17	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
18	W16x67	20.00	W12x53	15.60	**W10x77**	**22.60**
19	W8x24	7.08	W8x24	7.08	**W6x25**	**7.34**
20	W8x21	6.16	W10x22	6.49	**W8x21**	**6.16**
21	W8x40	11.70	W16x36	10.60	**W12x58**	**17.00**
22	W8x24	7.08	W8x24	7.08	**W8x24**	**7.08**
23	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
24	W10x22	6.49	W10x22	6.49	**W12x26**	**7.65**
25	W8x24	7.08	W6x25	7.34	**W8x24**	**7.08**
26	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
27	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
28	W8x24	7.08	W8x24	7.08	**W8x24**	**7.08**
29	W8x21	6.16	W8x21	6.16	**W8x21**	**6.16**
30	W8x21	6.16	W10x22	6.49	**W8x21**	**6.16**
31	W8x24	7.08	W8x24	7.08	**W12x26**	**7.65**
32	W8x24	7.08	W6x25	7.34	**W8x24**	**7.08**
W	365124.90 lb		370178.60 lb		**373482.36 lb**	
V	1.01E-02		0.0		**8.10E-04**	

Source: The Authors.

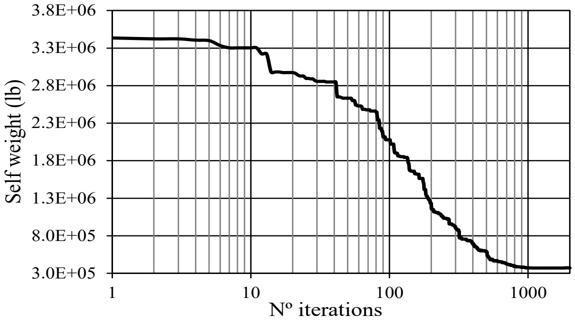


Figure 11. Penalized self-weight versus number of iterations.
Source: The Authors.

Table 2 displays 140 I-shape steel commercial sections obtained from the AISC database considered for the current optimization. The considered cross section areas are within the established minimum and maximum values of 6.16 in^2 (39.74 cm^2) and 215 in^2 (1387.09 cm^2), respectively. Fig. 11 displays the weight evolution of the tower with the increment of iterations. As may be observed, the convergence ratio to the optimal solution demands at least 2000 iterations to attain a stable solution. Meanwhile, the final optimized steel sections according to each group are listed in Table 3. Also, results of reference [5], in which a modified version of the ABC algorithm called MABC along with the results obtained from the Particle Swarm Optimization (PSO) algorithm, are presented for comparison. As expected, the current result of 373482.36 lb (1661.33 kN) is very close to the results reported in reference [5]. However, an excessive computational time is reported for the current example, in which only a single execution is performed. This resulted in a small violation of the imposed constraints of 8.1E-04. It is further believed that by using more iterations in the computation, this violation will be eliminated.

4. Discussion

In this section, execution times for the coupled algorithm ABC-SAP 2000 are discussed for various benchmarks. For all examples a maximum population of 50 bees with a maximum number of 2000 iterations at each computer run is adopted. The efficiency of the numerical tool in terms of computational times is displayed in Table 4. All the analyses have been carried out in sequential executions by using a single computer with the following characteristics: Intel Core i5 processor with four cores of 3.10 GHz and 8GB RAM memory. The results correspond to the average times required for the evaluation of one objective or fitness function for various well-known examples found in the specialized literature [2], [4]. As can be seen, SAP 2000 computational times are greater than those from the conventional subroutine used to solve the truss problem. Nevertheless, SAP 2000 computational times are nearly constant independently of problem complexity. Conversely, the execution time increases for the conventional subroutine. This is explained because as the complexity of the problem increases, i.e., the number of unknowns augments, the SAP 2000 solver of linear equations is more efficient than the solver of the conventional subroutine. But, in general it is clearly stated that to take complete advantage of the implemented tool, parallel computing is required to speed up the process. For instance, the required computational time for the optimization of the dome truss for one single run in SAP 2000 is approximately (50/2+2000×50)×0.835/3600 = 23.2 hours (almost a day) for 2000 iterations and 50 bees according to eq. (13).

Table 4.
Average computational times for one-single evaluation of the objective function for various benchmarks.

Truss example [2]-[4]	**Conventional truss solver Time (s)**	**SAP 2000 Time (s)**
Planar 10-bar truss	0.00307	0.790
Planar 18-bar truss	0.00658	0.796
Planar 45-bar truss	0.03270	0.876
120-bar dome truss	0.34100	0.835

Source: The Authors.

5. Conclusions

In this paper, a procedure is presented to combine the well-known optimization algorithm ABC with SAP 2000 commercial software via Visual Basic of macros in Excel. The resulting tool is then applied to size optimization problems of spatial trusses subjected to imposed constraints, which are expressed in terms of limiting stresses and displacements. The size optimization can be successfully applied to either continuous or discrete variables. The following conclusions can be drawn from this study.

In this initial stage of the study, the coupling between SAP 2000 and ABC proved to work well and seems to be promising because a real advantage of the commercial software potential can be exploited. Then, applications involving dynamic and plastic analysis can be further studied.

In order to improve execution times for challenger applications, parallel computing via several PCs certainly will speed up the optimization process. Thus, this task constitutes the natural step to be accomplished in future studies.

In its current form, the resulting tool certainly can be used in design offices by practitioners to optimize moderate trusses, thus, minimizing the costs of real projects.

References

[1] Karaboga, D., Technical report TR-06, Computer Engineering Department, Erciyes University, Turkey, 2005, 10 P.

[2] Hadidi, A., Azad, S.K. and Azad, S.K., Structural optimization using artificial bee colony algorithm, Second International Conference on Engineering Optimization, 2010, pp. 1-10.

[3] Lemonge, A., Duarte, G. and Fonseca, L., An algorithm inspired by bee colonies coupled to an adaptive penalty method for truss structural optimization problems. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 41(126), pp. 1-19, 2019. DOI: 10.1007/s40430-019-1629-7

[4] Sonmez, M., Artificial bee colony algorithm for optimization of

trusses. Applied Soft Computing, 11(2), pp. 2406-2418, 2011. DOI: 10.1016/j.asoc.2010.09.003

[5] Sonmez, M., Discrete optimum design of truss structures using artificial bee colony algorithm. Structural and Multidisciplinary Optimization, 43(1), pp. 85-97, 2011. DOI: 10.1007/s00158-010-0551-5

[6] Jalili, S. and Hosseinzadeh, S., A cultural algorithm for optimal design of truss structures. Latin American Journal of Solids and Structures, 12(9), pp. 1721-1747, 2015. DOI: 10.1590/1679-78251547

[7] Kaveh, A., Ghazaan, M. and Bakhshpoori, T., An improved ray optimization algorithm for design of truss structures. Periodica Polytechnica, 57(2), pp. 97-112, 2013. DOI: 10.3311/PPci.7166

[8] Cheng, M., Prayogo, D., Wu, Y. and Lukito, M., A hybrid harmony search algorithm for discrete sizing optimization of truss structure. Automation in Construction, 69, pp. 21-33, 2016. DOI: 10.1016/j.autcon.2016.05.023

[9] Kremper, E., Bernardino, H., Barbosa, H. and Lemonge, A., Performance evaluation of local surrogate models in differential evolution-based optimum design of truss structures. Engineering Computations, 34(2), pp. 499-547, 2017. DOI: 10.1108/EC-06-2015-0176

[10] Mernik, M., Liu, S., Karaboga, D. and Crepinsek, M., On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. Information Sciences, 291(10), pp. 115-127, 2015.DOI: 10.1016/j.ins.2014.08.040

[11] Mansouri, I., Soori, S., Amraie, H., Hu, J. and Shahbazi, S., Performance based design optimum of CBFs using bee colony algorithm. Steel and Composite Structures, 27(5), pp. 613-622, 2018. DOI: 10.12989/scs.2018.27.5.613

[12] Ghozi, M., Pujo, A. and Suprobo, P., Evolutionary parallel SAP 2000 for truss structure optimization. International Journal of Academic Research. [online]. 3(2), pp. 822-827, 2011. [date of reference May 4th of 2019]. Available at: https://www.researchgate.net/publication/273832628_Evolutionary_Parallel_SAP2000_for_truss_structure_optimization.

[13] CSI, SAP 2000 Integrated Software for Structural Analysis and Design. Computers and Structures Inc., Berkeley, California, USA, s.a.

R. Baldissera dos Santos, received the BSc. Eng. in Civil Engineering in 2017 from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil. His research includes simulation and modeling of optimization problems with computational intelligence techniques. He is currently enrolled in a Master degree Program at University of Porto, Portugal.
ORCID: 0000-0002-5043-9762

J.L. Palomino-Tamayo, received BSc. in Civil Engineering in 2004 from the National University of Engineering. Lima, Peru. He also received the MSc. in 2011 and the PhD in 2015 in Structural Engineering, all of them from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil. He has experience in computational mechanics with focus on structural and solid mechanics, soil dynamics and soil-structure interaction problems. Currently, he is a full-time professor in the Civil Engineering Department at Federal University of Rio Grande do Sul, Porto Alegre, Brazil
ORCID: 0000-0002-6301-7008

UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
Eléctrica e Ingeniería de Control

Oferta de Posgrados

Maestría en Ingeniería - Ingeniería Eléctrica
Maestría en Ingeniería - Automatización industrial
Especialización en Eco-eficiencia Industrial

Mayor información:

E-mail: ingelcontro_med@unal.edu.co
Teléfono: (57-4) 425 52 64