Virtual bat algorithm for the computation of Duhamel's Integral applied to structural systems with one degree of freedom

Algoritmo del murciélago virtual en el desarrollo de la Integral de Duhamel para sistemas estructurales con un grado de libertad

Sergio Vanegas*, Iván Amaya^{1**}, Rodrigo Correa***

* Estudiante de Maestría en Ingeniería Civil, Escuela de Ingeniería Civil. Universidad Industrial de Santander, Bucaramanga. COLOMBIA

** Estudiante de Doctorado en Ingeniería, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Universidad Industrial de Santander, Bucaramanga. COLOMBIA

*** Profesor Titular, Escuela de Ingeniería Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Universidad Industrial de Santander, Bucaramanga. COLOMBIA

Fecha de Recepción: 12/12/2012

Fecha de Aceptación: 06/06/2013

PAG 278 - 289

Abstract

The current article presents the use of the virtual bat optimization algorithm, for solving Duhamel's Integral in dynamic systems with one degree of freedom. Some situations are solved, as a demonstrative example, calculating global maxima. The results found were comparable (both, in precision and in accuracy) to those achieved with other calculation approaches. In a similar fashion, acceloremeter data of a Chilean seism (2010) was used to calculate the dynamic response and its maximum displacement. In all cases, the number of virtual bats were varied, noting an influence over the run time.

Keywords: Bat algorithm, Duhamel's Integral, dynamic structure, impulsive excitation, earthquake accelerograms

Resumen

En este artículo se propone el uso del algoritmo de optimización de murciélagos virtuales para resolver la Integral de Duhamel en sistemas dinámicos con un grado de libertad. Se desarrollaron ejercicios a título demostrativo, para el cálculo de máximos globales. Los resultados fueron muy comparables, tanto en precisión, como en exactitud frente a aquellos obtenidos por otros métodos de cálculo. Se utilizó igualmente el acelerograma del sismo de Chile (2010), para el que se reporta la respuesta dinámica y el desplazamiento máximo. Se variaron para estos ejemplos la cantidad de murciélagos virtuales notándose su influencia sobre el tiempo de computación.

Palabras Clave: Algoritmo de murciélagos, Integral de Duhamel, dinámica estructural, excitaciones impulsivas, acelerograma de sismos

1. Introduction

When performing a dynamic analysis of civil structures, it is often found that they are subject to arbitrary excitations, which, in a general case, are not harmonic. The dynamic answer to these excitations (that can be provoked by impulsive excitations or by accelerations of the ground due to seismic activity), can be predicted through the calculation of an integral that can be analytically evaluated when the excitation signal is a simple expression. This integral is known as Duhamel's Integral (DI), and it allows to theoretically calculate the response of a linear system (arbitrary structure) with a single degree of freedom and due to impulsive excitations. The latter, by definition, is an excitation applied for a short time interval. The impulse belonging to it can be defined as the product of a force by its duration (time).

A seism, which can also be considered an impulsive excitation, can be analyzed through the aforementioned integral. In this case, information of the telluric movement is required, which is gathered through an accelerogram, and which basically is a data series (accelerations) separated by short time intervals.

¹ Autor de correspondencia / Corresponding author:
E-mail: ivan.amaya2@correo.his.edu.co

This article shows, in an exemplary way, how it is possible to analyze (evaluate) Duhamel's Integral through a numeric optimization strategy known as Virtual Bat Algorithm (VBA), to obtain the maximum and minimum displacements of the dynamic response of a system with a single degree of freedom.

Metaheuristic optimization algorithms have become popular because of their simplicity and easiness of programming. Moreover, they provide good results (i.e. precision and accuracy) in optimization problems with a relative degree of complexity, and they can operate on systems with restrictions. Among them, Particle Swarm Optimization (PSO) (Parsopoulos & Vrahatis 2010), as well as one of its more recent modifications, i.e. Binary Particle Swarm Optimization (BPSO) (Kennedy & Eberhart 1997), Harmony Search (HS) (Ramos et al., 2011), and the Firefly algorithm (Xing-She Yang 2010), are some of the more outstanding ones. There is also another important approach, i.e. the bat algorithm (BA), which was proposed in 2010. This mimics the behavior of microbats when looking for prey, with the impressive ability of being able to differentiate between obstacles and their target. The current article deals with the use of this algorithm for solving Duhamel's Integral, and focuses on the effect of varying the population size on the capability of finding an appropriate answer.

2. Discussion and development

2.1 Theoretical foundation

Duhamel's Integral is a theoretical equation that allows calculation of the response of a linear system with a single degree of freedom, for arbitrary structures, which are under an external and brief excitation. For the general case of a damped system, DI has the following form (Paz 1992; Maldonado & Chio 2004):

$$u(t) = \frac{1}{m\omega_D} \int_0^t P(\tau) e^{-\xi\omega(t-\tau)} \text{sen} [\omega_D(t-\tau)] d\tau \quad (1a)$$

m = mass of the system (Masa del sistema)

ω_D = damped angular frequency = $\omega\sqrt{1-\xi^2}$ (Frecuencia angular amortiguada)

ω = natural frequency of the system = $\sqrt{k/m}$ (Frecuencia natural del sistema)

k = system stiffness (Rigidez del sistema)

ξ = damping coefficient (Coeficiente de amortiguación)

$P(\tau)$ = external force at instant τ (function that

simulates the external impulse) (Fuerza externa en el momento τ – función que simula el impulso externo)

$u(t)$ = time displacement (Tiempo de desplazamiento)

When dealing with a system with no damping ($\xi=0$), DI becomes:

$$u(t) = \frac{1}{m\omega_n} \int_0^t P(\tau) \text{sen} [\omega_D(t - \tau)] d\tau \quad (1b)$$

And after a trigonometric change, it yields:

$$u(t) = [A(t) \text{sen } \omega_D t - B(t) \cos \omega_D t] \frac{e^{-\xi\omega t}}{m\omega_n} \quad (1c)$$

$$\begin{aligned} A(t) &= \int_0^t P(\tau) e^{\xi\omega t} \cos \omega_D \tau d\tau \\ B(t) &= \int_0^t P(\tau) e^{\xi\omega t} \text{sen } \omega_D \tau d\tau \end{aligned} \quad (1d)$$

2.1.1 Spectral response for an excitation at the support (seismic movement)

An important problem, from the point of view of the structure, is the analysis of a system under the effect of an excitation at the base (foundation or support) of the structure. The movement equation of the damped oscillator under excitation is (Maldonado & Chio 2004):

$$m\ddot{u} + c(\dot{u} - \dot{u}_s) + k(u - u_s) = 0 \quad (1e)$$

m = mass of the system (Masa del sistema)

$\ddot{u}$ = acceleration of the system (Aceleración del sistema)

$\dot{u}$ = speed of the system (Velocidad del sistema)

u = displacement of the system at the highest point of the structure (Desplazamiento del sistema en el punto más alto de la estructura)

u_s = displacement of the system at its base (Desplazamiento del sistema en su base)

This equation can also be written as a function of the relative movement of the mass with respect to the movement of the support, obtaining:

$$y + 2\xi\omega\dot{y} + \omega^2 y = -\ddot{u}_s(t) \quad (1f)$$

where y is the relative movement, that can be defined as:

$$y = u - u_s \quad (1g)$$

The solution of this equation to the response of the deformation $\mathbf{y}(\mathbf{t})$, depends, basically, on the acceleration of ground's movement, $\mathbf{u}_s''(\mathbf{t})$, the natural vibrating frequency $\omega=\sqrt{\mathbf{K}/\mathbf{m}}$, and the damping ratio, ξ , of the structure. The analytical solution of this differential equation is:

$$\mathbf{y}(\mathbf{t}) = -\frac{1}{\omega_n} \int_0^t \ddot{\mathbf{u}}_s(\tau) e^{-\xi\omega(t-\tau)} \text{sen} [\omega_D(t-\tau)] d\tau \quad (1h)$$

Due to the fact the seismic acceleration of the terrain varies irregularly with time, an analytical solution of the problem is just impossible. However, numerical procedures, supported on the constantly increasing processing velocity of modern computers, pose an effective alternative for dealing with these situations. Seismic accelerograms are already digitalized and filtered to control and debug initial errors, thus obtaining an appropriate accelerogram of the terrain. The function represents $\mathbf{u}_s''(\mathbf{t})$ the accelerogram defined by the corrected numerical coordinates in some time intervals close enough to provide an appropriate approximation.

Having solved the movement equation, a response throughout the time of the accelerogram, is obtained. With it, it is possible to define the maximum response achieved by the system, which represents an important variable for predicting its behavior when facing a seism with similar characteristics.

2.1.2 Virtual bat algorithm (VBA)

During this article it is proposed to transform the calculation of the maximum displacement through DI, into a maximization problem. Moreover, it was decided to use BA for this task. This algorithm was initially proposed by Yang, and it is based on the approximate description of some bat species given their natural echolocation features, which can be used to locate objects and prey using the echo produced by their ultrasonic squeal, that bounces between the surrounding objects. This system is so developed that they can avoid obstacles as small as a human hair. It is believed that they also generate a 3D scenario using this technique, as well as the time difference between the perception at their ears, and the variation in the intensity of the echoes (X. Yang 2010).

In spite of being proposed in 2010, there is already evidence of some variants that strive to improve its behavior as an optimization approach (Xing-She Yang 2010; X.-S. Yang & Gandomi 2012). In order to simplify the algorithm, some ground rules were established regarding the behavior of bats (X. Yang 2010; Gandomi et al., 2012; X. Yang 2011):

1. Virtual bats use echolocation to differentiate between obstacles and prey.
2. Virtual bats fly randomly with velocity v_i at a position x_i emitting a frequency between f_{min} and f_{max} , varying the wavelength λ and the intensity A_0 when locating their prey.

They can adjust the wavelength (or the frequency) as well as the emission ratio r , $r \in [0, 1]$ depending on the proximity to their target.

3. The intensity varies from an initial value A_0 to a minimum of A_{min} .

Summarizing, a pseudocode of the algorithm can be expressed as the original proposal of Yang (X. Yang 2010; Gandomi et al., 2012; X. Yang 2011; Nakamura et al., 2012; Ponnappa & Pupong 2012):

Define the objective function $f(x)$, $x = (x_1, \dots, x_d)^T$.

Initialize the virtual bats population x_i ($i = 1, 2, \dots, n$) and v_i

Define frequencies f_i at x_i .

Initialize the pulse rate r_i and their intensity A_i .

while ($t < \text{Max. iteration number}$)

Generate new solutions by adjusting the frequencies, and update each bat's velocity and position, using equation. (1) to equation (3).

if ($\text{rand} > r_i$)

Select a solution between the best ones.

Generate a local solution around the best solution selected, using a random walk.

end if

if ($\text{rand} < A_i$ & $f(x_i) < f(x_*)$)

Accept the new solutions.

Increase r_i and reduce A_i .

end if

Classify bats and find the actual best x_* .

end while

During simulations, the initialization of the frequencies, as well as the position and velocity of each bat, is defined using Expressions (2) to (4):

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (2)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (3)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (4)$$

Donde/ where:

β = Valor aleatorio entre 0 y 1/Random value between 0 and 1.

f_i = Frecuencia para el murciélago virtual i -ésimo/ Frequency for the i -th virtual bat.

f_{min}, f_{max} = Frecuencias mínimas y máximas. Valores recomendados entre 0 y 100 (X. Yang 2010)/Minimum and maximum frequencies. Values between 0 and 100 are recommended (X. Yang 2010).

v_i^t, v_i^{t-1} = Velocidad en el paso de tiempo t y $t-1$, respectivamente, para el murciélago i -ésimo/ Velocity at time step t and $t-1$, respectively, for the i -th bat.

x_i^t, x_i^{t-1} = Posición en el paso de tiempo t y $t-1$, respectivamente, para el murciélago i -ésimo/ Position at time step t and $t-1$, respectively, for the i -th bat.

x_* = Mejor posición en el actual paso de tiempo/ Best position at the current time step.

The local search in the algorithm is carried out using a random walk, that is given in Equation (5):

$$x_{new} = x_{old} + \epsilon \bar{A}^t \quad (5)$$

Donde/ where:

ϵ = Número aleatorio entre -1 y 1 / Random number between -1 and 1.

$\bar{A}^t$ = Promedio de A_i en el actual paso de tiempo / Average of A_i at the current time step

In order to increase r_i and A_i , Expressions (6) and (7) are used:

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (7)$$

Donde/ where:

α = Valor aleatorio entre 0 y 1. En esta investigación, se escogió el valor 0.9 (X. Yang 2010) / Random value between 0 and 1.

During this research, a value of 0.9 was chosen (X. Yang 2010).

γ = Valor constante mayor que cero. Esta investigación asumió el valor 0.9 (X. Yang 2010) / Constant value bigger than zero.

During this research, 0.9 was assumed (X. Yang 2010).

2.1.3 Virtual bat for Duhamel's Integral

In order to implement VBA for solving DI, it was necessary to modify the demo provided by (Matlab Central n.d.), with the following one:

Objective Function (Expression (1))

Initialize virtual bats population x_i ($i = 1, 2, \dots, n$) and v_i in a given domain (defined between 0 and t-max).

Define frequencies f_i at x_i (defined between $f_{min}=0$ and $f_{max}=2$).

Initialize pulse rate $r_i=0.5$ and their intensity $A_i=0.5$ (constants for each time step).

Define an error $E = \infty$.

Define a tolerance for stop criteria, $Tol=0.01$

while ($t < \text{Max iteration number}$ && $E > Tol$)

Evaluate the function (Equation (1)) and calculate $f(x')$ (for this case, $u(t)$).

Look for the maximum or minimum (x_s) on the previously evaluated values.

Generate new solutions by adjusting frequencies, and update velocities and position [Expressions (1) to (3)]. If the new solutions,

x'_p are out of bounds, then bring them back.

if ($\text{rand} > r_i$)

Generate a new local solution around the best solution selected, x_s , through a random walk:

$$x_{new} = x_s + 0.01 * \text{rand}(0, 1) \quad (8)$$

end if

Evaluate x_{new} in the function, $f(x_{new})$.

if ($rand < A_i$ && $f(x_{new}) < f(x_i^l) \rightarrow \text{find min}$)
 $f(x_{new}) > f(x_i^l) \rightarrow \text{find max}$)

Update solutions x_{new} to x_i^{t+1}

if [$f(x_{new}) < f(x_*) \rightarrow \text{find min}$]
 $f(x_{new}) > f(x_*) \rightarrow \text{find max}$]
 x_{new} becomes x_*

Calculate error E .

end while

Some of the changes performed on the algorithm are related to the random walk defined by equation (5), which becomes equation (8). It was observed that by using this new equation, the search process is enhanced.

2.2 Experimental results

For analyzing VBA applied to DI, several tests were run, but only some of them are reported due to space limitations. A couple examples shown in (Maldonado & Chio 2004) are used, in order to observe the algorithm's ability to find an appropriate answer. Following that, an analysis on a real-life seism is performed, using data of an accelerometer set in Chile, which registered a seism from 2010 (Facultad de Ingeniería Civil - Universidad de Chile n.d.). For each situation, the run time is reported, using a computer with the following specifications: AMD Phenom II X6 @ 2.8 GHz, 8 GB RAM, working under Windows 7 Professional. For each experiment, the number of virtual bats were varied, in order to see its effect on the answer.

2.2.1 Situation N°1. Harmonic function.

The excitation produced into the system with a single degree of freedom, is presented through Expression (9).

$$P(t) = 1500 \sin(12.566t) \quad (9)$$

Los datos entregados al software son los siguientes/The data given to the software is as follows:

Mass= 7.6 Ton
 $K= 30000 \text{ KN/m}$
 $\xi=5\%$

As a control group, the analytical answer found in (Maldonado & Chio 2004) is used. In this case, the best displacement is $u(t)=5.2 \text{ cm}$ manifested at a time of 0.085 seconds. The last column of Table 1 shows the time required to finish the process. The plot of the dynamic response is generated using 200 points. Table 1 summarizes the results for simulations with different number of bats.

Figure 1 shows the behavior of displacement through time, provided by the algorithm.

Tabla 1 Resumen de la cantidad de simulaciones para la situación N°1

Table 1. Summary of the number of simulations for the first situation

$p(t) = 1500\text{sen}(12.566t)$								
(Mejor desplazamiento 5,2 cm para $t=0.085$ /The best displacement 5,2 cm for $t=0.085$)								
Para 200 puntos en el gráfico/For 200 points in the graph								
Murciélagos/Bat's		Desplazamiento [cm]/Displacement [cm]					Error [%]	Tiempo/time [s]
		1	2	3	4	5 Promedio/Average		
10	U(t)	5.070	4.894	4.969	5.151	4.989	5.015	3.70
	t	0.160	0.170	0.148	0.091	0.095	0.133	
	T	4.656	4.656	4.902	4.655	4.701	4.714	
50	U(t)	5.061	5.213	5.213	5.173	5.075	5.147	1.03
	t	0.162	0.079	0.068	0.077	0.161	0.109	
	T	4.765	4.758	4.753	4.724	4.752	4.750	
100	U(t)	5.160	5.214	5.196	5.196	5.180	5.189	0.21
	t	0.065	0.081	0.076	0.084	0.085	0.078	
	T	4.868	4.797	4.864	4.778	4.771	4.816	
150	U(t)	5.215	5.212	5.132	5.206	5.205	5.194	0.12
	t	0.076	0.068	0.894	0.088	0.084	0.242	
	T	4.902	4.896	4.887	4.902	4.911	4.900	
200	U(t)	5.198	5.209	5.214	5.206	5.215	5.208	0.16
	t	0.089	0.088	0.081	0.093	0.082	0.086	
	T	4.886	4.982	4.979	4.976	4.975	4.980	
Máximo Local/ Local Maximum								
Máximo global/ Global Maximum

Figura 1. Situación N°1 con 100 murciélagos. Desplazamiento en [m] y tiempo en [s]

Figure 1. Situation 1 with 100 bats. Displacement in [m] and time in [s]

2.2.2 Situation N° 2. Triangle function

In this case, a function defined by sections is used to excite the system. Its definition can be seen in equations (10), and it is plotted in Figure 2.

$$P(t) = \begin{cases} P(t)_1 = \frac{150}{0.6} t \rightarrow 0 \leq t \leq 0.6 \text{ s} \\ P(t)_2 = -\frac{150}{0.6} t + 300 \rightarrow 0.6 < t \leq 1.2 \text{ seg.} \end{cases} \quad (10)$$

Figura 2. Señal de excitación para la segunda situación
Figure 2. Excitation signal for the second situation

Once again, answers from (Maldonado & Chio 2004) were used as a control group. In this case, the best displacement was $u(t)=0.293$ cm for a time of 0.6 seconds, approximately. Table 2 summarizes the results achieved with the algorithm, finding in the last column the time it took to converge to an answer. A plot of the response, with 100 dots is shown in Figure 3. The data fed to the algorithm was the following:

Mass= 12 Ton
K= 50000 KN/m
 $\xi=0\%$ (Sin amortiguación)/(no damping).

Tabla 2 Resumen de la cantidad de simulaciones para la situación N°2
Table 2. Summary of the results for Situation N° 2

Ejercicio 2/Exercise N°. 2									
Mejor desplazamiento 0,293 cm para t=0.6/The best displacement 0,293 cm for t= 0,06									
Para 100 puntos en el gráfico/For 100 points un the graph									
Murciélagos/Bat's		Desplazamiento [cm]/Displacement [cm]					Error [%]		Tiempo/time [s]
		1	2	3	4	5Promedio/Average			
10	U(t)	0.293	0.256	0.248	0.242	0.298	9.52	33.079	
	t	0.631	0.662	0.508	0.483	0.620			
	T	32.354	33.051	32.729	33.591	33.670			
50	U(t)	0.289	0.296	0.298	0.298	0.293	0.63	33.212	
	t	0.636	0.597	0.632	0.614	0.597			
	T	33.204	33.075	33.392	33.207	33.183			
100	U(t)	0.299	0.295	0.299	0.297	0.299	1.50	33.549	
	t	0.617	0.629	0.616	0.594	0.608			
	T	33.815	33.321	33.539	33.573	33.499			
150	U(t)	0.299	0.298	0.298	0.299	0.298	1.80	33.497	
	t	0.618	0.640	0.617	0.618	0.620			
	T	33.541	33.695	33.482	33.465	33.304			
200	U(t)	0.299	0.298	0.298	0.299	0.299	1.82	33.963	
	t	0.623	0.621	0.621	0.618	0.628			
	T	34.010	33.725	34.358	34.069	33.652

Figura 3. Situación N°2 con 100 murciélagos virtuales. Desplazamiento en [m] y tiempo en [s]

Figure 3. Situation No. 2 with 100 bats. Displacement in [m] and time in [s]

2.2.3 Situation N°3. Earthquake accelerogram. Seism in Chile (Maule's earthquake)

The excitation used in this situation is taken from data of an 8.8 quake, known as Maule's earthquake, which happened on February 27th, 2010. Some of the specifications of this seismic event are: peak acceleration of 465.484 cm/s², peak velocity of 27.707 cm/s and a peak ground displacement of 7.001 cm. The algorithm is fed with a file containing the accelerations reported by (Facultad de Ingeniería Civil - Universidad de Chile n.d.), and the following structural values:

$Mass = 7.6 \text{ Ton}$

$K = 30000 \text{ KN/m}$

$\zeta = 5\%$

Since this situation does not have comparative data, BA was used to observe the response and to check the run time required for calculating the maximum displacement. Figure 4 shows the seismic data, as well as the algorithm's response for time displacement as a function of time. Once again, Table 3 summarizes the results for different number of bats.

Tabla 3. Resumen de los resultados para la situación N° 3

Table 3. Summary of the results for situation N° 3

Resultados con el terremoto de Chile. Terremoto del Maule/Result with Chile earthquake, Maule Earthquake						
Para 100 puntos en el gráfico/For 100 points in the graph						
Murciélagos/Bat's	Desplazamiento [cm]/Displacement [cm]				Tiempo/time [s]	
	1	2	3	4	Promedio/Average	
100	U(t) Max	0.709	0.762	0.716	0.736	0.673
	t Max	6.802	6.867	6.910	6.898	7.091
	U(t) Min	-0.395	-0.397	-0.415	-0.408	-0.401
	t Min	3.658	7.466	7.436	3.648	5.912
	T	42.741	42.647	42.764	42.828	42.690
						42.734
200	U(t) Max	0.580	0.762	0.762	0.602	0.755
	t Max	6.755	6.819	6.866	6.770	6.832
	U(t) Min	-0.477	-0.411	-0.470	-0.478	-0.423
	t Min	7.405	3.641	7.392	7.404	7.430
	T	43.462	43.443	43.140	43.162	43.391
						43.320
500	U(t) Max	0.735	0.755	0.761	0.765	0.718
	t Max	6.907	6.854	6.873	6.825	6.800
	U(t) Min	-0.437	-0.398	-0.437	-0.479	-0.459
	t Min	7.423	7.466	7.375	7.403	7.387
	T	46.002	45.631	45.721	45.355	45.696
						45.681
1000	U(t) Max	0.765	0.759	0.766	0.762	0.764
	t Max	6.842	6.865	6.819	6.834	6.829
	U(t) Min	-0.469	-0.479	-0.473	-0.442	-0.479
	t Min	7.409	7.397	7.393	7.379	7.397
	T	48.763	48.598	48.468	48.371	48.683
						48.577
1500	U(t) Max	0.766	0.763	0.761	0.762	0.764
	t Max	6.834	6.831	6.872	6.864	6.829
	U(t) Min	-0.466	-0.443	-0.480	-0.477	-0.469
	t Min	7.390	7.380	7.402	7.405	7.396
	T	52.170	52.326	52.727	52.264	52.364
						52.370
2000	U(t) Max	0.766	0.766	0.762	0.762	0.766
	t Max	6.825	6.823	6.865	6.831	6.833
	U(t) Min	-0.460	-0.479	-0.465	-0.478	-0.459
	t Min	7.388	7.397	7.390	7.396	7.414
	T					
						55.601
Máximo Local/Local Maximum						
Máximo global/Global Maximum						
Local Minimum

Figura 4. Situación N°3 con 2000 murciélagos virtuales. Desplazamiento en [m] y tiempo en [s]

Figure 4. Situation N° 3 with 2000 virtual bats. Displacement in [m] and time in [s]

3. Conclusions

During this research, it was possible to observe that for all of the three exemplary cases with Duhamel's Integral, highly satisfactory results were achieved when calculating the maximum displacement of a structure with a single degree of freedom. The virtual bat algorithm is quite precise when finding the maximum and minimum of a function, so it can be used as a particular problem requires it. The run times of the algorithm were relatively short for the number of bats used during the tests.

It was observed that the algorithm tends to remain in local extremes when the search domain is big in comparison to the number of bats used (i.e. the bat algorithm works better, at least for this application, when searching in packs). In any case, and thanks to it being a stochastic metaheuristic, the algorithm finds the optimum, in this case Duhamel's Integral, very quickly. The results summarized in the tables show that the answer reported by the algorithm does not differ too much, and that it stabilizes around an answer in a short time. Because of this, it is recommended to keep working on it, using it for solving problems with a relatively high mathematical complexity.

4. Referencias/References

- Facultad de Ingeniería Civil - Universidad de Chile (2012)**, Base de Datos Sismo de Chile. Available at: <http://terremotos.ing.uchile.cl/> [Accessed November 1].
- Gandomi A.H. et al. (2012)**, Bat algorithm for constrained optimization tasks. Neural Computing and Applications. Available at: <http://www.springerlink.com/index/10.1007/s00521-012-1028-9> [Accessed November 24, 2012].
- Kennedy J. and Eberhart R.C. (1997)**, A Discrete Binary Version of the Particle Swarm Algorithm. In 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. IEEE, pp. 4104–4108. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=637339> [Accessed December 11, 2012].
- Maldonado E. and Chio C.G. (2004)**, Análisis Sísmico de Edificaciones 1st ed., Ediciones Universidad Industrial de Santander.
- Matlab Central, Bat Algorithm (demo) (2012)**, Available at: <http://www.mathworks.com/matlabcentral/fileexchange/37582-bat-algorithm-demo> [Accessed November 1].
- Nakamura R.Y.M. et al. (2012)**, BBA: A Binary Bat Algorithm for Feature Selection. In XXV SIBGRAPI - Conference on Graphics, Patterns and Images. Ouro Preto, pp. 291–297. Available at: <http://www.decom.ufop.br/sibgrapi2012/eproceedings>.
- Parsopoulos K.E. and Vrahatis M.N. (2010)**, Particle Swarm Optimization and Intelligence: Advances and Applications,
- Paz M. (1992)**, Dinámica Estructural 3rd ed., Editorial Reverté, S.A.
- Ponnappa M. and Pupong P. (2012)**, Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem Using Bat Algorithm. In 2nd International Conference on Management and Artificial Intelligence IPEDR. Singapore: IACSIT Press, pp. 98–102.
- Ramos C.C.O. et al. (2011)**, A Novel Algorithm for Feature Selection Using Harmony Search and its Application for Non-Technical Losses Detection. Computers & Electrical Engineering, 37(6), pp.886–894. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0045790611001479> [Accessed December 11, 2012].
- Yang X. (2010)**, A New Metaheuristic Bat-Inspired Algorithm. In International Workshop on Nature Inspired cooperative Strategies for Optimization (NICSO 2010). Granada: Springer, pp. 65–74.
- Yang X. (2011)**, Bat Algorithm for Multiobjective Optimization. International Journal of Bio-Inspired Computation, 3(5), pp.267–274.
- Yang X.-S. and Gandomi, A.H. (2012)**, Bat Algorithm: A Novel Approach for Global Engineering Optimization. Engineering Computations, 29(5), pp.464–483.
- Yang Xing-She. (2010)**, Nature-Inspired Metaheuristic Algorithms 2nd ed., United Kingdom: Luniver Press.