Matemática: Una publicación del FCNM – ESPOL
2018, Vol.16, No.1

Publicado:2018-04-02

MODELO DE PROGRAMACIÓN BINARIA A UN MODELO DE PROGRAMACIÓN NO LINEAL CON PARÁMETRO CONTINUO

(BINARY PROGRAMMING MODEL TO A NON-LINEAR PROGRAMMING MODEL WITH CONTINUOUS PARAMETER)

Bustamante Johni[1], Quito Alex[2], Pinto Félix[3]

Resumen: *En este artículo se presenta un modelo de programación entera binaria el cual se lo pasa a un modelo de programación no lineal con variables continuas parametrizadas, es decir todas la nuevas variables son funciones de variable real continuas, los ejemplos aquí detallados son modelos relativamente pequeños por cuanto este método no pretende mejorar la eficiencia del algoritmo de búsqueda, solamente representa el modelo de una forma más sencilla y visual de encontrar la solución con ayuda de la gráfica de una función continua de variable real.*

Palabras claves: Programación entera, binaria, no lineal, serie de Fourier, periodos binarios, NP- Hard; NP.

Abstract: *This paper presents a binary integer programming model which is passed to a non-linear programming model with parameterized continuous variables, that is, all the new variables are continuous real variable functions, the examples detailed here are relatively small models by when this method does not intend to improve the efficiency of the search algorithm, it only represents the model in a simpler and visual way of finding the solution with the help of the graph of a continuous function of real variable*

Keywords: Integer, binary, non-linear programming, Fourier series, binary periods, NP- Hard; NP.

Recibido: Febrero 2018
Aceptado: Marzo 2018

1.- INTRODUCCIÓN

En la vida real se presentan problemas tales como: El problema de la mochila, el mismo que consiste en determinar si un artículo (SKU) se incluye o no en la mochila de tal forma que en dicha mochila quepan la mayor cantidad de artículos (SKU).

En el momento que decimos “se incluye o no” un artículo, podemos modelar la decisión con una variable binaria, la cual representara “1” en caso que si se incluye y “0” en caso contrario, teniendo así un modelo de programación binaria.

Otro ejemplo es el problema de “la Cartera de inversiones”, en tal problema se debe determinar si se invierte o no en un “negocio” de tal forma que al final tengamos la mayor rentabilidad, nuevamente la decisión se la representa con una variable binaria, “1” en caso de realizar la inversión en dicho negocio o “0” no hacer la inversión.

[1]*Mat.E. Bustamante Johni, Ph.D., Profesor del Departamento de Matemáticas, Facultad de Ciencias Naturales y Matemáticas, Escuela Superior Politécnica del Litoral (ESPOL), (email: jobustam@espol.edu.ec).*

[2]*Ing. Quito Alex, Msc., Profesor de Investigación de Operaciones, Universidad de Guayaquil, (email: aquitoc@ug.edu.ec).*

[3]*Msc. Pinto Félix, Profesor de la carrera de Ciencias Ambientales, Universidad de Guayaquil, (email: fpintob@ ug.edu.ec).*

Además sabemos que un problema de programación binaria es un problema clasificado como NP-Hard, “No polinomial duro”, lo cual significa que su solución mediante cualquier algoritmo con una cantidad medianamente grande de variables es simplemente insoluble, por tiempo de ejecución.

En este artículo presento la forma de pasar este modelo a otro de tipo continuo,, es decir la variable es real, pero no mejora su condición de ser NPHard, por cuanto tendremos un nuevo problema de programación no lineal el cual también se encuentra clasificado como un problema NPHard.

Lo interesante de este método consiste en que con una cantidad razonable de variables la resolución es muy sencilla usando cualquier software, en nuestro caso usando “Mathematica de Wolfram”.

2.- PROBLEMA DE PROGRAMACION BINARIA:

$$z = a_1x_1 + a_2x_3 + a_3x_3 + .. + a_nx_n$$
$$s.a: \; b_1x_1 + b_2x_2 + b_3x_3 + .. + b_nx_n = c$$

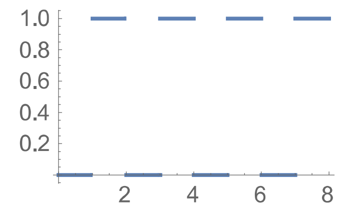
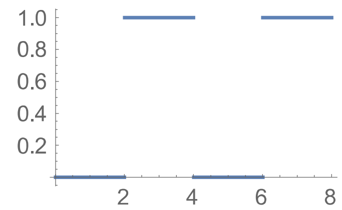
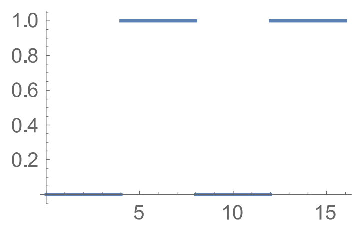
dónde:

x_k: *Variable de decisión*
a_k, b_k, c : *Constantes Reales*
Z es la función objetivos a Maximizar

2.1.- COMBINACION DE LAS VARIABLES BINARIAS Y PARAMETRIZACIÓN

Si tenemos una variable binaria, entonces tenemos solo dos posibles soluciones {0, 1}, pero si tenemos dos variables binarias entonces tendremos 4 posible soluciones, estas son:

{(0,0), (0,1),(1,0),(1,1)}, y así sucesivamente, entonces si tenemos "n" variables binarias, entonces tendremos 2^n posibles soluciones.
El método para representar a todas las posibles soluciones con un solo parámetro consiste en representar a cada variable como una función periódica de una variable real "t" de la siguiente forma:

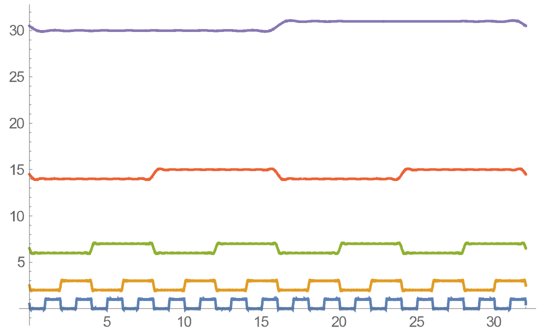
$x(t,1) = \begin{cases} 0 & 0 < t < 1 \\ 1 & 1 < t < 2 \end{cases};$ Periodo T(1)= 2^1	
$x(t,2) = \begin{cases} 0 & 0 < t < 2 \\ 1 & 2 < t < 2^2 \end{cases};$ Periodo T(2)= 2^2	
$x(t,3) = \begin{cases} 0 & 0 < t < 2^2 \\ 1 & 2^2 < t < 2^3 \end{cases};$ Periodo T(3)= 2^3	
$x(t,n) = \begin{cases} 0 & 0 < t < 2^{n-1} \\ 1 & 2^{n-1} < t < 2^{n+1} \end{cases};$ Periodo T(n)= 2^n	

2.2.-VARIABLES BINARIAS A FUNCIONES CONTÍNUAS

Para transformar las variables continuas ya parametrizadas usaremos las series de Fourier, dicho proceso lo obviaremos en este trabajo por la sencillez del mismo.
Esto es, luego de pasar las funciones x(t,k) a series de Fourier tenemos:

$$x(t,m) = \frac{1}{2} - \frac{2}{\pi}\left(\sum_{k=1}^{Nn}\left(\frac{\mathrm{Sin}\left(\frac{\pi t(2k-1)}{2^{m-1}}\right)}{2k-1}\right)\right)$$

Observamos que:
1.- Nn es la cantidad de sumando de la serie de Fourier, es decir no necesitamos calcular el valor hasta el infinito, con una cantidad no muy grande tenemos resultados bastante buenos.
2.- $x(t,m)$ es la función parametrizada correspondiente a la variable x_m,
Donde $m = 1,2,3,\ldots n$
A continuación la representación de las funciones {x(t,1), x(t,2), x(t,3), x(t,4), x(t,5)} presentadas en serie de Fourier en el dominio de $[0 \quad 2^5]$



2.3.- INTERPRETACIÓN DEL PARAMETRO Y VARIABLE DE DECISIÓN

Los valores a analizar serán los puntos intermedios es decir $\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, etc$
Ejemplo: Para $t = \frac{5}{2}$,

```
Table[x[5/2, k] + 0.0, {k, 1, 5}]
tabla

{0.0158762, 1.02235, -0.000351599, 0.0139167, -0.0317717}
```

es decir:

$$\begin{cases} x_1 = 0 \\ x_2 = 1 \\ x_3 = 0 \\ x_4 = 0 \\ x_5 = 0 \end{cases}$$

3.- PROBLEMA SENCILLO O INSTANCIA

$$Z_{max} = 3x1 + 5x2 + 4x3 + 6x4 + 2x5$$
$$s.a: \quad x1 + x2 + x3 + x4 + x5 = 2$$

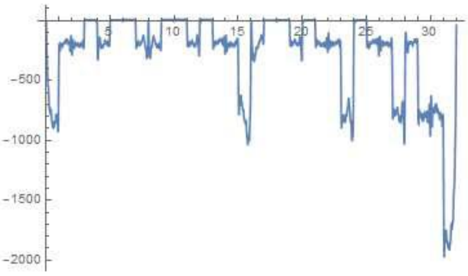
Tomamos todas las variables parametrizadas y además usamos el criterio de M-Grande para que la restricción se cumpla.

$$f(t) = 3x1 + 5x2 + 4x3 + 6x4 + 2x5 - \mathrm{M}\,(x1 + x2 + x3 + x4 + x5 - 2)^2$$

donde M es un valor positivo lo suficientemente grande, en este caso suficiente con M=200

```
a1 = 3;
a2 = 5;
a3 = 4;
a4 = 6;
a5 = 2;
M = 200;
z2[t_] := (a1 x[t, 1] + a2 x[t, 2] + a3 x[t, 3] + a4 x[t, 4] + a5 x[t, 5]) - M (x[t, 1] + x[t, 2] + x[t, 3] + x[t, 4] + x[t, 5] - 2)^2
Plot[z2[t], {t, 0, 32}]
```

representación gráfica



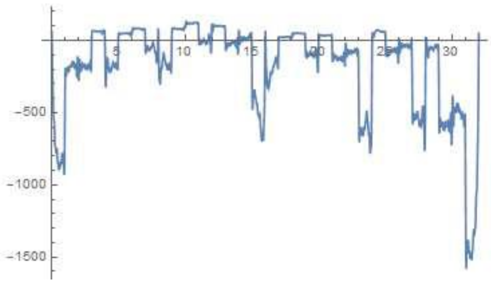
Para efectos de visualización y sin pérdida de tiempo en ejecución podemos replantear el problema:

$$f(t) = (3x1 + 5x2 + 4x3 + 6x4 + 2x5)^2 - \mathrm{M}\,(x1 + x2 + x3 + x4 + x5 - 2)^2$$

Tenemos:

```
a1 = 3;
a2 = 5;
a3 = 4;
a4 = 6;
a5 = 2;
M = 200;
z2[t_] := (a1 x[t, 1] + a2 x[t, 2] + a3 x[t, 3] + a4 x[t, 4] + a5 x[t, 5])^2 - M (x[t, 1] + x[t, 2] + x[t, 3] + x[t, 4] + x[t, 5] - 2)^2
Plot[z2[t], {t, 0, 32}]
```

representación gráfica

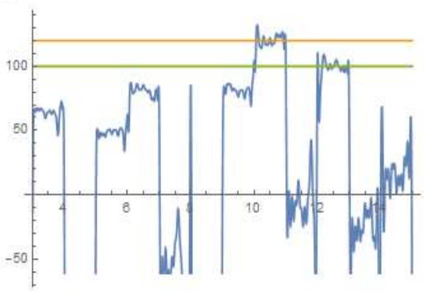


Observamos que en el intervalo de 3 a 15 se presentan valores en los cuales está el máximo.

Ampliamos la gráfica en este intervalo y tenemos:

```
Plot[{z2[t], 120, 100}, {t, 3, 15}]
```

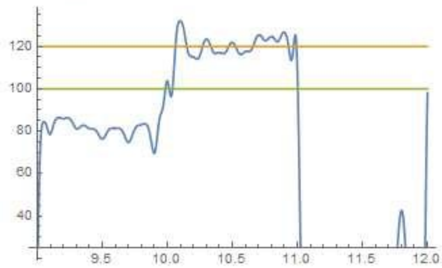
representación gráfica



Observando que el máximo esta entre los valores de 10 a 12

```
Plot[{z2[t], 120, 100}, {t, 9, 12}]
```

representación gráfica



Ahora observamos que el máximo valor se lo alcanza en cuando t= 10.5

Ahora calculamos las variables en los cuales se obtuvo el máximo:

```
val = 10.5;
xR = {x[val, 1], x[val, 2], x[val, 3], x[val, 4], x[val, 5]}
```

Obtenemos:

```
{0.0158762, 1.02235, -0.000351599, 0.986083, -0.0164271}
```

Obviamente son valores aproximados del 0 y 1, por tanto tomamos sus respectivos valores:

```
{0, 1, 0, 1, 0}
```

Y calculamos el máximo alcanzado:

```
zR = a1 xE[[1]] + a2 xE[[2]] + a3 xE[[3]] + a4 xE[[4]] + a5 xE[[5]]
18
```

Verificamos que se cumpla con la restricción:

```
xE[[1]] + xE[[2]] + xE[[3]] + xE[[4]] + xE[[5]] - 2
2
```

3.1.- VERIFICACION DEL PROBLEMA

Verificar estos resultados es muy sencillo, el problema está planteado de tal forma que se escogen los "x" que acompañan a los dos valores más grandes es decir "5 y 6", problemas que tienen resultado conocido y sirven para verificar la eficiencia del método se llaman "instancias".

4.- PROBLEMA DE LA MOCHILA (Propuesto y desarrollado por Alex Quito & Félix Pinto)

4.1.-PLANTEAMIENTO PROGRAMACION LINEAL BINARIA

EJERCICIO DE PROGRAMACIÓN LINEAL ENTERA –BINARIO.

Un excursionista debe determinar que objetos debe llevar consigo en la mochila para realizar una excursión de un día. Cada uno de los objetos tiene asociado un peso y una utilidad personal para el excursionista.

Los objetos que puede llevar, así como su peso y utilidad son los que se recogen en la tabla siguiente:

OBJETO	PESO	UTILIDAD
LINTERNA	40	40
SACO	50	80
COCINA	30	10
MANTA	10	10
COMIDA	10	4
ROPA	40	20
VARIOS	30	60

Sabiendo que el peso máximo que puede llevar en la mochila es de 100. Determinar que objetos debe llevar nuestro excursionista en la mochila para que la utilidad de los objetos sea máxima.

4.2.- PLANTEMIENTO NO-LINEAL EN GAMS

El fichero GMS es:

```
*PROBLEMA BINARIO.

VARIABLES
LINT, SACO, COCINA, MANTA, COMIDA, ROPA, VARIOS, U;

BINARY VARIABLES
LINT, SACO, COCINA, MANTA, COMIDA, ROPA, VARIOS;

EQUATIONS
OBJ, LIMITE;

OBJ..     U =E= 40*LINT + 80*SACO + 10*COCINA + 10*MANTA+
          4*COMIDA + 20*ROPA + 60*VARIOS;

LIMITE..  40*LINT + 50*SACO + 30*COCINA + 10*MANTA + 10*COMIDA +
          40*ROPA + 30*VARIOS =L= 100;

MODEL MOCHILA /ALL/;

SOLVE MOCHILA USING MIP MAXIMIZING U;
```

4.3.- RESULTADOS EN GAMS

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU OBJ	.	.	.	1.000
---- EQU LIMITE	-INF	100.000	100.000	0.333

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR LINT	.	.	1.000	26.667
---- VAR **SACO**	.	1.000	1.000	63.333
---- VAR COCINA	.	.	1.000	.
---- VAR **MANTA**	.	1.000	1.000	6.667
---- VAR **COMIDA**	.	1.000	1.000	0.667
---- VAR ROPA	.	.	1.000	6.667
---- VAR **VARIOS**	.	1.000	1.000	50.000
---- VAR U	-INF	154.000	+INF	.

4.4.- PROBLEMA POR EL MÉTODO PROPUESTO

LINT, SACO, COCINA, MANTA, COMIDA, ROPA, VARIOS

x(t,1); x(t,2); x(t,3); x(t,4); x(t,5); x(t,6), x(t,7),

respectivamente

$$Z_{max} = 40\text{ x1} + 80\text{ x2} + 10\text{ x3} + 10\text{ x4} + 4\text{ x5} + 20\text{ x6} + 60\text{ x7}$$

$$s.a: \quad 40\text{ x1} + 50\text{ x2} + 30\text{ x3} + 10\,x4 + 10\text{ x5} + 40\text{ x6} + 30\text{ x7} \leq 100$$

4.5.- PLANTEAMIENTO EL PROBLEMA BINARIO A PROBLEMA NO LINEAL PARAMETRIZADO

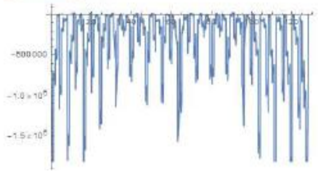
```
a1 = 40; a2 = 80; a3 = 10; a4 = 10; a5 = 4; a6 = 20; a7 = 60;
w1 = 40; w2 = 50; w3 = 30; w4 = 10; w5 = 10; w6 = 40; w7 = 30;
M = 200;
Nn = 15;
```

$$x[t\_, n\_] := \frac{1}{2} - \frac{2}{\pi}\left(\sum_{k=1}^{Nn} \frac{\text{Sin}\left[\frac{\pi t (2k-1)}{2^{n-1}}\right]}{2k-1}\right)$$

$$z2[t\_] := (a1\,x[t,1] + a2\,x[t,2] + a3\,x[t,3] + a4\,x[t,4] + a5\,x[t,5] + a6\,x[t,6] + a7\,x[t,7])^2 - M\,(w1\,x[t,1] + w2\,x[t,2] + w3\,x[t,3] + w4\,x[t,4] + w5\,x[t,5] + w6\,x[t,6] + w7\,x[t,7] - 100)^2$$

```
Plot[{z2[t]}, {t, 0, 128}]
```

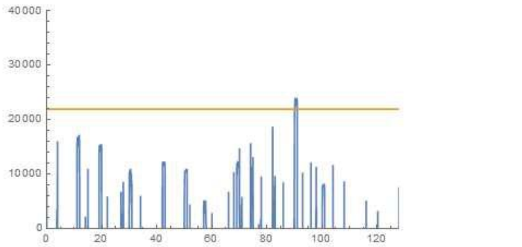
representación gráfica



4.6.- VISUALIZACIÓN E INTERPRETACIÓN DE LOS RESULTADOS

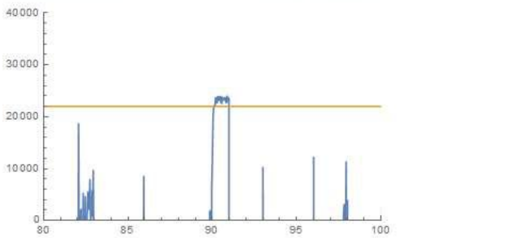
En la gráfica observamos valores relativamente altos entonces buscaremos la gráfica solo en la parte positiva y en valores altos:

```
Plot[{z2[t], 22000}, {t, 0, 128}, PlotRange → {{0, 128}, {0, 40000}}]
|representación gráfica          |rango de representación
```



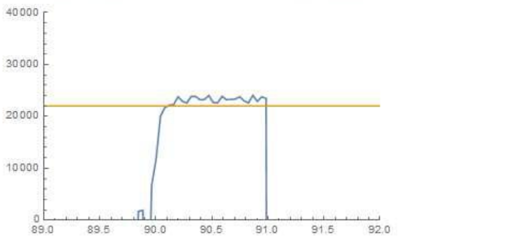
Observamos que el máximo lo alcanza en el intervalo [80 100], y son valores todavía no muy aproximados, entonces tenemos:

```
Plot[{z2[t], 22000}, {t, 0, 128}, PlotRange → {{80, 100}, {0, 40000}}]
|representación gráfica          |rango de representación
```



Observamos que el máximo lo alcanza en el intervalo [90 92], entonces tenemos:

```
Plot[{z2[t], 22000}, {t, 0, 128}, PlotRange → {{89, 92}, {0, 40000}}]
|representación gráfica          |rango de representación
```



Observamos que el máximo lo obtenemos cuanto t= 90.5

Valores reales:

```
val = 90.5;
xR = {x[val, 1], x[val, 2], x[val, 3], x[val, 4], x[val, 5], x[val, 6], x[val, 7]}
{-0.0105986, 1.00049, -0.00822016, 1.00513, 0.993151, -0.0186031, 0.99723}
```

Valores en variables binarias:

```
xE = IntegerPart[xR + 0.2]
{0, 1, 0, 1, 1, 0, 1}
```

Restricción:

```
w1 xE[[1]] + w2 xE[[2]] + w3 xE[[3]] + w4 xE[[4]] + w5 xE[[5]] + w6 xE[[6]] + w7 xE[[7]] - 100
0
```

Valor máximo:

```
zR = a1 xE[[1]] + a2 xE[[2]] + a3 xE[[3]] + a4 xE[[4]] + a5 xE[[5]] + a6 xE[[6]] + a7 xE[[7]]
154
```

5.- CONCLUSIONES

El método aquí diseñado demuestra que los problemas de programación lineal binaria se pueden pasar (equivalencia) a problemas de programación No-Lineal de variable continua, este último específicamente en un problema de optimización de una función de una sola variable, con la particularidad que dicha función está representada en serie de Fourier y la misma tiene infinitos extremos lo cual hace que el problema siga estando en la clasificación NP-Hard.

Además el método presenta una forma muy especial de representar todas las posibles combinaciones de las variables binarias en funciones que dependen de un solo parámetro.

REFERENCIAS BIBLIOGRAFICAS Y ELECTRÓNICAS

[1] **M. R. Spiegel, J. Liu, L. Abellanas** (2003): *Fórmulas y tablas de matemática aplicada*. Segunda edición. Serie Schaum. Mc Graw-Hill.

[2] **Fetter, Alexander L.; Walecka, John Dirk** (2003). *Theoretical Mechanics of Particles and Continua*. Courier Corporation. pp. 209, 210. ISBN 9780486432618.

[3] **Björner, Anders; and Stanley, Richard P**.; (2010); *A Combinatorial Miscellany*

[4] **Bóna, Miklós**; (2011); *A Walk Through Combinatorics (3rd Edition)*. ISBN 978-981-4335-23-2, ISBN 978-981-4460-00-2(pbk)

[5] **Graham, Ronald L.; Groetschel, Martin; and Lovász, László**; eds. (1996); *Handbook of Combinatorics*, Volumes 1 and 2. Amsterdam, NL, and Cambridge, MA: Elsevier (North-Holland) and MIT Press. ISBN 0-262-07169-X

[6] **Lindner, Charles C.; and Rodger, Christopher A**.; eds. (1997); *Design Theory*, CRC-Press; 1st. edition (October 31, 1997). ISBN 0-8493-3986-3.