

TORRES DE HANOI

Ariza Pérez, Jesús

[\(jjisu_86@hotmail.com\)](mailto:jjisu_86@hotmail.com)



Introducción

En 1883 empezó a venderse en Francia un antiguo rompecabezas oriental, rescatado para Occidente por el profesor N. Claus (de Siam) y cuyas primeras referencias eran los escritos del ilustre mandarín Fer-Fer-Tam-Tam. Según una leyenda india, en el Templo de Benarés, bajo el domo que marca el centro del mundo, hay una placa de latón con tres agujas de diamante. Durante la creación, Dios puso sesenta y cuatro discos de oro puro de distinto tamaño en una de las agujas, formando una torre. Los bramanes llevan generaciones cambiando de lugar, uno a uno, los discos de la torre entre las tres agujas de forma que en ningún momento un disco mayor descansa sobre otro más pequeño. Cuando hayan conseguido trasladar todos los discos a otra aguja su trabajo estará terminado, y la torre y el templo se derrumbarán, y con un gran trueno, el mundo se desvanecerá. La versión simplificada que se vendía en Francia se componía de ocho discos de madera.

En realidad, la Torre de Hanoi y la leyenda india habían sido inventadas por el matemático francés [Édouard Lucas](#) (N. Claus de Siam es un anagrama de Lucas d'Amiens). Su compatriota, el escritor Henri de Parville amplió y adornó la leyenda poco tiempo después.

Memoria

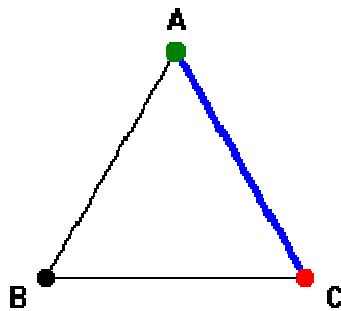
Este problema lógico consiste en cambiar una serie de discos, de una varilla inicial a una tercera varilla, pudiendo usar una segunda varilla intermedia para realizar los movimientos específicos. Eso sí, debemos limitarnos a unos requisitos que dificultan el problema.

- Un disco de mayor tamaño no puede ir encima de uno mas pequeño.
- Solo se pueden mover los discos de uno en uno.
- Se saca el disco que esta colocado mas arriba de la varilla.

Estos requisitos son las restricciones que condicionan mi problema. El objetivo es conseguir solucionar el problema en el mínimo número de movimientos posibles, que sería la solución óptima a mi problema. Para ello tengo lo siguiente:

El modelo para este puzzle matemático se puede resumir a un simple diagrama de estados, que va a variar dependiendo del número de discos que posea, eso si, va a mantener unas propiedades que lo hacen óptimo para solucionarlo.

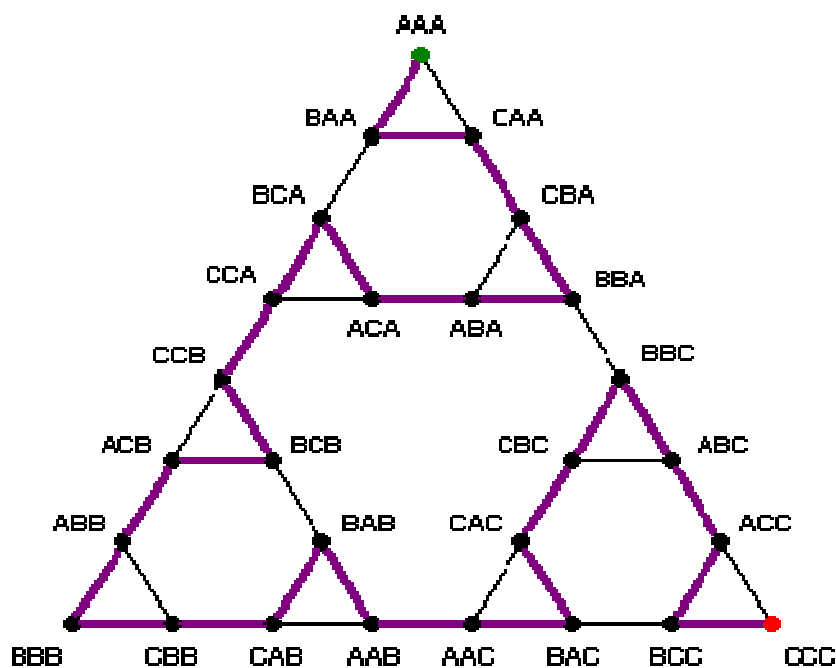
Supongamos que tenemos un solo disco, el diagrama con los posibles movimientos sería el siguiente :



Mi objetivo es llegar siempre a C y parto de una posición inicial A. Como se puede observar lo puedo realizar en 2 movimientos (pasando por B) o directamente de A a C.

¿Y si tengo 3 discos? ¿Cómo sería ese diagrama ahora?

Figura que muestra una solución pero que no es óptima.



AAA= Los 3 discos en el palo A, que sería la posición inicial.

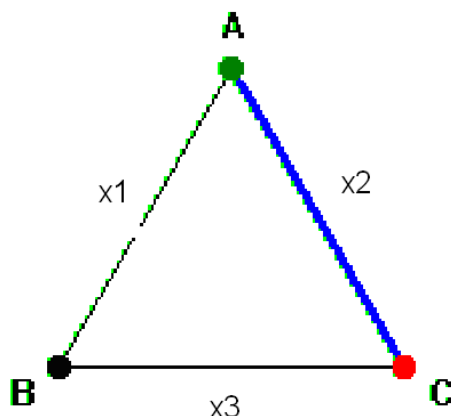
CBA= Disco uno en el palo C, disco 2 en el palo B y disco 3 en el palo A.

...

Se puede comprobar la relación que tiene el diagrama con respecto al diagrama anterior para $n-1$. Si nos fijamos solo en el primer triangulo de arriba, vemos que es el diagrama de $n=1$, pero es que el diagrama de $n=2$ también esta (los 3 triangulitos de superiores). Con esto establezco una relación para hallar un modelo general, y es que cada diagrama para n discos es 3 veces el anterior ($n-1$) unidos por unos “puentes” entre si, que serian 3 cada vez que añada un disco nuevo.

Por tanto mi modelo trata de minimizar el número de movimientos que puedo realizar para llegar del punto inicial (todos los discos en el palo A) hasta la final (todos los discos en el palo C).

El número de movimientos equivale al número de arcos que yo recorro para llegar desde A hasta C.



X1 – Arco desde A hasta B.

X2 – Arco desde A hasta C.

X3 – Arco desde B hasta C.

El valor de X_i será: 0 si no paso por ese arco, 1 si utilizo el arco.

El modelo para $n = 1$ quedaría tal que así:

Minimizar

$$X_1 + X_2 + X_3;$$

Sujeto a:

$X_0 = 1$; //es una condición inicial para empezar a funcionar.

$X_1 + X_2 = X_0$; //forzosamente si empiezo en A, o tiro por x_1 o por x_2 , por eso uno de los dos tiene que valer 1.

$$X_2 + X_3 \leq 2;$$

$X_1 + X_3 \leq 2$; //estas dos restricciones son necesarias en cada grupo de triángulos para saber que camino escojo.

//en este caso no tengo restricciones para los "puentes" porque no existen. Cada vez que añada un disco, tendré 3 nuevas restricciones equivalentes a las 3 variables puente nuevas.

Con esto tengo el modelo resuelto para 1 disco. Pero, ¿y si tengo n discos?

Pues el sistema es el mismo, teniendo en cuenta que cada grafo depende del anterior el modelo general sería:

Minimizar

$$\sum X_i \quad \text{donde } x_i \text{ es cada uno de los arcos del grafo}$$

Sujeto a:

$$X_i + X_{(i+1)} = X?$$

$$X_{(i+1)} + X_{(i+2)} \leq 2;$$

$$X_i + X_{(i+2)} \leq 2;$$

$$X_i \leq 1;$$

// X_i o valen 0 o valen 1, son variables enteras (También podrían ser binarias para ese par de valores)

//X? Será en cada triángulo la condición de entrada, 1 si entra en ese triángulo y 0 sino entra. Esta condición es un arco “puente” de ese triángulo y si es el triángulo de partida será 1 siempre (X0 en mi caso).

Las restricciones van en paquetes de 3, es decir, para $n=1$ tengo 1 paquete de 3 restricciones (estas de aquí arriba) y ningún paquete-puente.

Sabiendo que un paquete-puente no es más ni menos que los 3 arcos que unen cada uno de mis sub-triángulos, equivaldrían a 3 restricciones más de la forma siguiente:

$$X(i+9)=X_i + X(i+2)$$

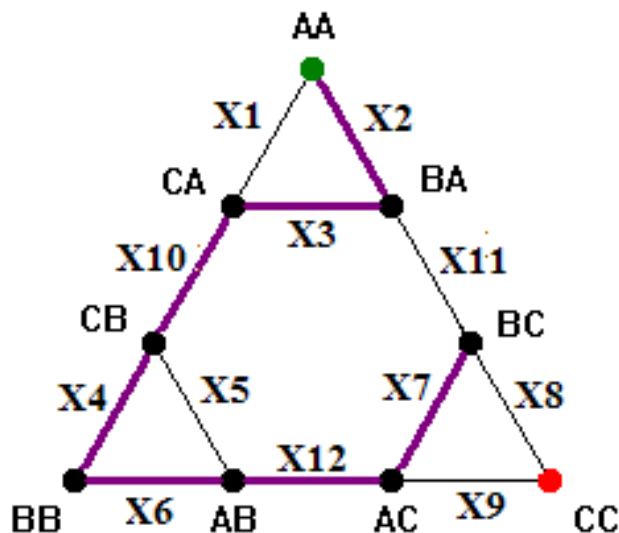
$$X(i+10)=X?-X(i+9)$$

$$X(i+11)=X(i+9)$$

//X(i+9) porque mis variables van en paquetes de 9 y 3 mas para los puentes, que serían $X(i+9)$, $X(i+10)$ y $X(i+11)$

//También sabemos que esos 3 arcos-puente tienen que ser ≤ 1 , es decir por alguno de ellos debo tirar siempre, si tiro por el de la derecha llegare mas rápidamente a la solución entonces los otros dos deberían valer 0, pero si tiro por la izquierda también se que debo usar otro puente para llegar al triangulo solución.

En el ejemplo se verá más claramente:



El verde es el punto inicial y el rojo a donde quiero ir, pues bien tenemos lo siguiente:

-X0 = 1 esto siempre va a ser así, porque si vale 0 entonces no estoy moviendo nada, será mi condición inicial para que arranque todo

Luego estoy siempre tomando como referencia el arco-puente mas situado a la izquierda en este caso el X10, y de ahí saco todas las demás restricciones puente. Por eso **$X(i+9)=X_i + X(i+2)$** donde $X(i+9)$ sería ese arco (X10) y $X_i + X(i+2)$ los arcos a los que se llega a él.

-X10, X11 y X12 representan los “puentes” a los que me refiero. La entrada al triángulo formado por X4, X5 y X6 va a depender de X10, entonces en mi restricción **$X(i+10)=X?-X(i+9)$** tenemos que $X? = X0$, Si el triángulo fuera mas grande ese valor sería el arco de donde procede para acceder a este triángulo.

-Con esto también se observa que si entro por X10, X12 tiene que valer lo mismo que X10, esa sería mi restricción **$X(i+11)=X(i+9)$** .

De esta manera es fácil saber, cuantas variables puedo tener y si en cada caso necesito puente o no, y cuantos se necesitarían:

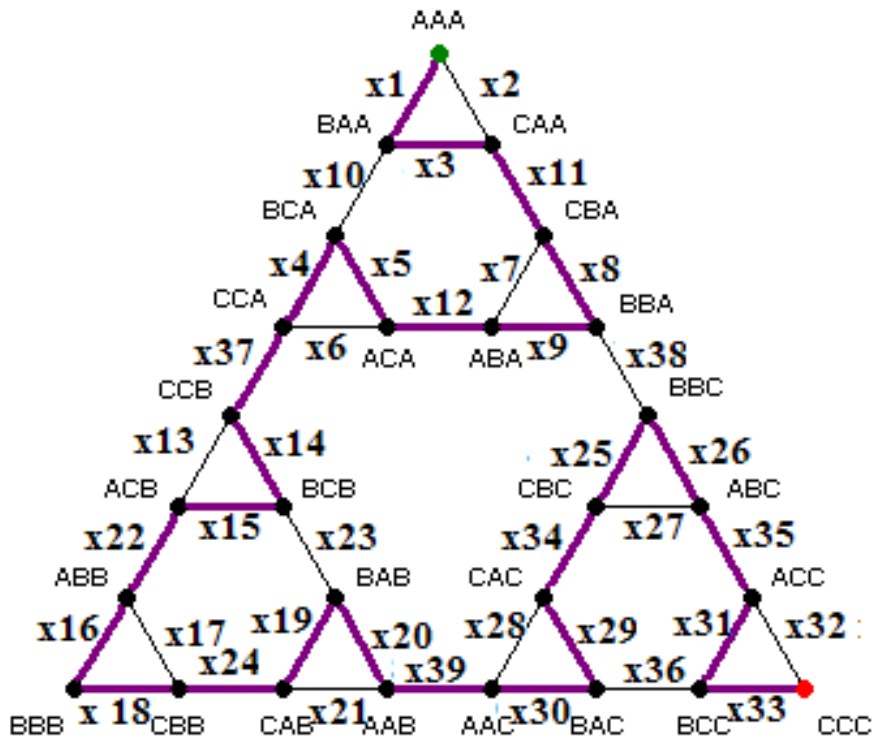
Para $n=2$ tengo 3 paquetes de 3 restricciones (o lo que es lo mismo 3 paquetes de $n=1$) y 1 puente. Para $n =3$ tendría 3 paquetes de $n=2$ y 3 puentes. Si te fijas sigue una secuencia:

n	restricciones	Paquete-puentes
1	3	0
2	9	1
3	27	3
4	81	9

Es fácil calcular la secuencia que seguiría, y el número total de variables que tendría para minimizar mi problema es equivalente a la suma de las restricciones mas los puentes.

En definitiva, para cada grafo es equivalente al anterior y se puede establecer una relación que facilita su modelado. Un ejemplo más de cómo quedarían las variables en un grafo para 3 discos ($n=3$). Las variables que teníamos para $n=2$ siguen teniendo el mismo nombre, ya que el grafo es el mismo, los siguientes dos triángulos siguen en orden las variables, para dejar las 3 últimas variables a los puentes que los unen, en este caso X37, X38 y X39.

Aquí también se puede observar claramente las condiciones de entrada para entrar a cada uno de los sub-triángulos, y es que para entrar al triángulo grande 2 (el de abajo a la izquierda) necesito que X37 sea igual a 1, si X37 es igual a 0, ese triángulo no se recorrería y ninguno de sus arcos sería recorrido de ninguna forma.



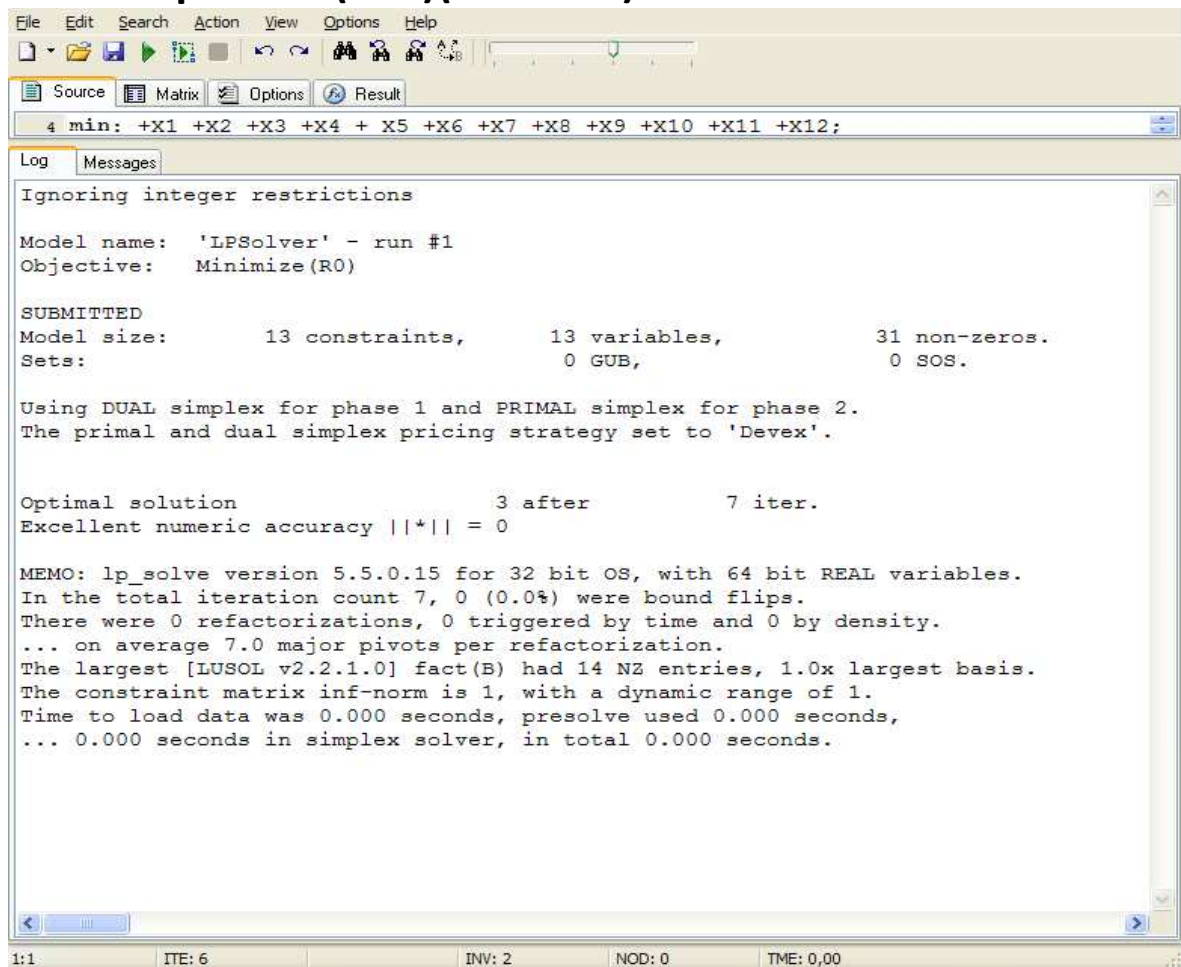
Modelo para LP (n=2)

```

1  /* Hanoi para n=2 */
2
3  /* Objective function */
4  min: +x1 +x2 +x3 +x4 + x5 +x6 +x7 +x8 +x9 +x10 +x11 +x12;
5
6  /* Constraints */
7
8  COND1: x0 =1;                      /*condicion de entrada*/
9
10 COND2: +x1 + x2 = +x0;
11 COND3: +x2 + x3 <=2;
12 COND4: +x1 +x3 <=2;
13
14
15 COND5: +x4+x5=+x10;
16 COND6: +x5+x6 <=2;
17 COND7: +x4+x6 <=2;
18
19 COND8: +x7+x8 =+x11+x12;
20 COND9: +x8+x9 <=2;
21 COND10: +x7+x9 <=2;
22
23
24 paso1: +x10=+x1+x3;
25 paso2: +x11=x0-x10;
26 paso3: +x12=+x10;
27
28
29
30 /* Variable bounds */
31 x1 <= 1;
32 x2 <= 1;
33 x3 <= 1.

```


Modelo para LP (n=2)(Solución)



The screenshot shows the LPSolver application window. The 'Source' tab is active, displaying the objective function: `4 min: +X1 +X2 +X3 +X4 + X5 +X6 +X7 +X8 +X9 +X10 +X11 +X12;`. Below this, the 'Log' tab is selected, showing the following text:

```
Ignoring integer restrictions

Model name: 'LPSolver' - run #1
Objective:  Minimize(R0)

SUBMITTED
Model size:      13 constraints,      13 variables,      31 non-zeros.
Sets:           0 GUB,                0 SOS.

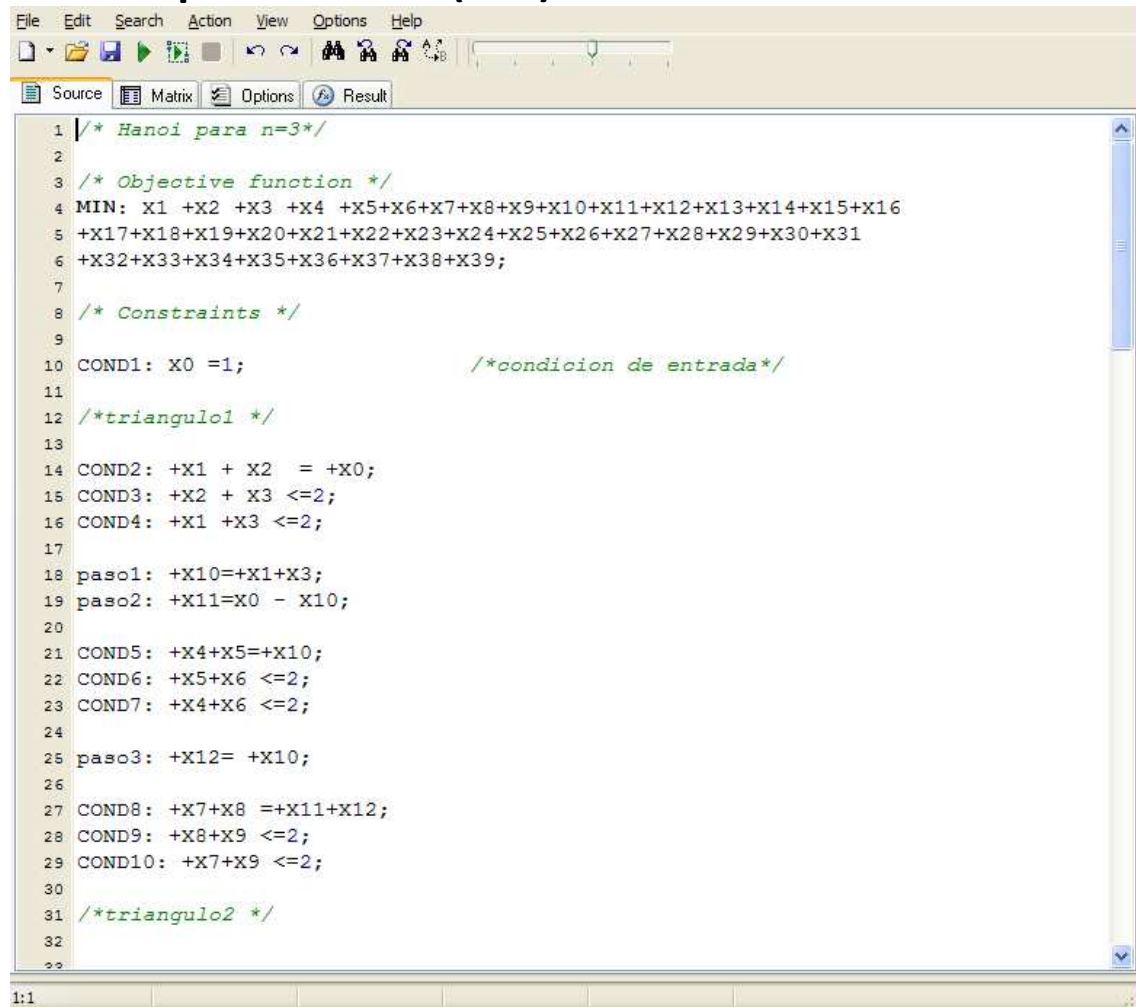
Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.

Optimal solution          3 after          7 iter.
Excellent numeric accuracy ||*|| = 0

MEMO: lp_solve version 5.5.0.15 for 32 bit OS, with 64 bit REAL variables.
In the total iteration count 7, 0 (0.0%) were bound flips.
There were 0 refactorizations, 0 triggered by time and 0 by density.
... on average 7.0 major pivots per refactorization.
The largest [LUSOL v2.2.1.0] fact(B) had 14 NZ entries, 1.0x largest basis.
The constraint matrix inf-norm is 1, with a dynamic range of 1.
Time to load data was 0.000 seconds, presolve used 0.000 seconds,
... 0.000 seconds in simplex solver, in total 0.000 seconds.
```

At the bottom of the window, a status bar displays the following information: 1:1, ITE: 6, INV: 2, NOD: 0, TME: 0,00.

Modelo para LP solve (n=3)



```
1 /* Hanoi para n=3 */
2
3 /* Objective function */
4 MIN: X1 +X2 +X3 +X4 +X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16
5 +X17+X18+X19+X20+X21+X22+X23+X24+X25+X26+X27+X28+X29+X30+X31
6 +X32+X33+X34+X35+X36+X37+X38+X39;
7
8 /* Constraints */
9
10 COND1: X0 =1; /*condicion de entrada*/
11
12 /*triangulo1 */
13
14 COND2: +X1 + X2 = +X0;
15 COND3: +X2 + X3 <=2;
16 COND4: +X1 +X3 <=2;
17
18 paso1: +X10=+X1+X3;
19 paso2: +X11=X0 - X10;
20
21 COND5: +X4+X5=+X10;
22 COND6: +X5+X6 <=2;
23 COND7: +X4+X6 <=2;
24
25 paso3: +X12= +X10;
26
27 COND8: +X7+X8 =+X11+X12;
28 COND9: +X8+X9 <=2;
29 COND10: +X7+X9 <=2;
30
31 /*triangulo2 */
32
33
```

1:1

The screenshot shows a software window with a menu bar (File, Edit, Search, Action, View, Options, Help) and a toolbar. Below the toolbar are tabs for 'Source', 'Matrix', 'Options', and 'Result'. The 'Source' tab is active, displaying a list of mathematical constraints and steps for two problems, 'triangulo2' and 'triangulo3'. The constraints are labeled COND11 through COND25, and the steps are labeled paso4 through paso8. The constraints involve variables X13 through X30 and constants 2, 37, and 39. The steps involve arithmetic operations on these variables. The interface has a light beige background and a vertical scrollbar on the right side of the text area.

```
31 /*triangulo2 */
32
33
34 COND11: +X13 + X14 = +X37;
35 COND12: +X14 + X15 <=2;
36 COND13: +X13 +X15 <=2;
37
38 paso4: +X22=+X13+X15;
39 paso5: +X23=+X37 - X22;
40
41 COND14: +X16+X17=+X22;
42 COND15: +X17+X18 <=2;
43 COND16: +X16+X18 <=2;
44
45 paso6: +X24= +X22;
46
47 COND17: +X19+X20 =+X23+X24;
48 COND18: +X20+X21 <=2;
49 COND19: +X19+X21 <=2;
50
51
52 /*triangulo3 */
53
54 COND20: +X25 + X26 = +X38+X39;
55 COND21: +X26 + X27 <=2;
56 COND22: +X25 +X27 <=2;
57
58 paso7: +X34=+X25+X27;
59 paso8: +X35=+X38+X39 - X34;
60
61 COND23: +X28+X29=+X34;
62 COND24: +X29+X30 <=2;
63 COND25: +X28+X30 <=2;
```

Referencias bibliográficas

-<http://www.psicoactiva.com/>, 1998

-(Wikipedia 2009) http://es.wikipedia.org/wiki/Torres_de_Han%C3%B3i

-<http://thales.cica.es/rd/Recursos/rd99/ed99-0222-02/matematicas1.htm>, 2006

- <http://www.kernelthread.com/projects/hanoi/> , 2006

-Juego interactivo:

<http://www.mazeworks.com/hanoi/index.htm> , 2002

