

HASHIWOKAKERO

Manchón López, Javier
(javierml85@hotmail.com)

Gutiérrez Ortega, Juan José
(juansiles16@hotmail.com)

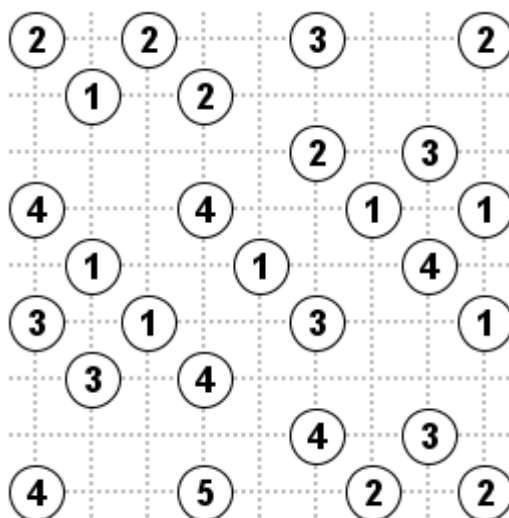
Sánchez Matas, Manuel
(manu_iznajar@hotmail.com)

Resumen

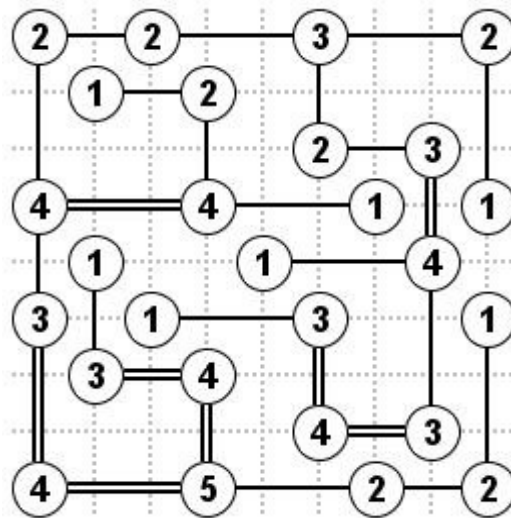
Hashiwokakero es un puzzle lógico de origen japonés, publicado por la editorial Nikoli. Se juega en un tablero rectangular de tamaño no normalizado, dispuesto como una rejilla dispuesta en celdas. Algunas de estas celdas contienen números con valores del 1 al 8, normalmente marcadas con un círculo constituyen las islas. Las celdas restantes están vacías.

El objetivo del juego es conectar todas las islas entre sí, formando un único grupo de islas conectadas mediante la creación de puentes entre las islas. Los puentes deben seguir ciertos criterios:

1. Cada puente debe comenzar y finalizar en islas distintas, uniéndolas mediante una línea recta.
2. Ningún puente puede cruzarse con otro puente o con otra isla distinta de las que une.
3. Las trayectorias deben ser perpendiculares. Es decir solo pueden ser horizontales o verticales.
4. Entre dos islas sólo puede haber como máxima dos puentes que las conecten.
5. El número total de puentes conectados a cada isla debe coincidir con el número en esa isla.



Tablero de partida, 9x9.



Tablero solución, 9x9

Planteamiento del Juego.

A la hora de plantearnos la resolución del problema, buscamos en la matriz aquellos números adyacentes por filas y columnas(horizontal y verticalmente) que sean distintos de 0, para crear todos los posibles puentes.

Numeramos las islas desde 1 hasta $N \cdot M$, siendo N el número de filas y M de columnas, utilizando la siguiente fórmula

$$\text{isla} = M * (i-1) + j \quad \text{siendo } i \text{ el n}^\circ \text{ de fila y } j \text{ el n}^\circ \text{ de la columna.}$$

Un ejemplo de una matriz de 3x3 seria:

$$\begin{bmatrix} 1 & 2 & 3; \\ 4 & 5 & 6; \\ 7 & 8 & 9 \end{bmatrix}$$

Restricciones

Solo existe un tipo de restricción, que es aquella en la que los puentes que unen dos islas no se crucen con los puentes de otras dos islas. Para que se cumpla esta restricción la suma de los puentes activos que se cruzan debe ser ≤ 1 , para que solo este activo uno de los 2. Para obtener estas ecuaciones vemos los posibles puentes que se pueden cruzar y escribimos 4 ecuaciones por cada posible cruce, es decir, uno en cada sentido posible. A continuación lo vemos con un ejemplo gráfico:

La Matriz introducida en Matlab y la matriz de islas correspondiente sería:

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} \Rightarrow \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Como vemos puede existir un puente entre la isla 2-8 y la isla 4-6 por lo cual las ecuaciones serían las siguientes:

$$X_{2_8} + X_{4_6} \leq 1;$$

$$X_{8_2} + X_{4_6} \leq 1;$$

$$X_{2_8} + X_{6_4} \leq 1;$$

$$X_{8_2} + X_{6_4} \leq 1;$$

Modelado del Juego.

Llamamos a X_{i_j} a la línea(puente) que va desde la isla i hasta las isla j.

Ejemplo simple:

La Matriz introducida en Matlab y la matriz de islas correspondiente sería:

$$\begin{array}{ccc} 1 & 2 & 0 \\ 0 & 6 & 4 \\ 3 & 4 & 2 \end{array} \Rightarrow \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

El modelado del problema seria el siguiente:

Minimizar

$$\begin{aligned} &+x_{1_2} + x_{2_1} + x_{1_3} + x_{3_1} + x_{1_4} + x_{4_1} + x_{1_7} + x_{7_1} + x_{2_3} + x_{3_2} \\ &+x_{2_5} + x_{5_2} + x_{2_8} + x_{8_2} + x_{3_6} + x_{6_3} + x_{3_9} + x_{9_3} + x_{4_5} + x_{5_4} \\ &+x_{4_6} + x_{6_4} + x_{4_7} + x_{7_4} + x_{5_6} + x_{6_5} + x_{5_8} + x_{8_5} + x_{6_9} + x_{9_6} \\ &+x_{7_8} + x_{8_7} + x_{7_9} + x_{9_7} + x_{8_9} + x_{9_8} ; \end{aligned}$$

Sujeto a:

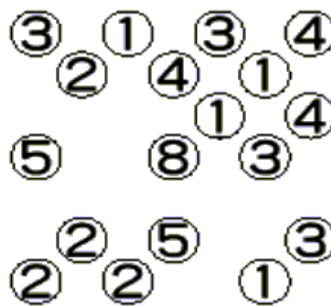
$$\begin{aligned} &x_{1_2} + x_{2_1} + x_{1_7} + x_{7_1} = 1; \\ &x_{2_1} + x_{1_2} + x_{2_5} + x_{5_2} = 2; \\ &x_{5_6} + x_{6_5} + x_{5_2} + x_{2_5} + x_{5_8} + x_{8_5} = 6; \\ &x_{6_5} + x_{5_6} + x_{6_9} + x_{9_6} = 4; \\ &x_{7_8} + x_{8_7} + x_{7_1} + x_{1_7} = 3; \\ &x_{8_7} + x_{7_8} + x_{8_9} + x_{9_8} + x_{8_5} + x_{5_8} = 4; \\ &x_{9_8} + x_{8_9} + x_{9_6} + x_{6_9} = 2; \\ &x_{2_8} + x_{4_6} \leq 1; \\ &x_{2_8} + x_{6_4} \leq 1; \end{aligned}$$

$$\begin{aligned}x_{8_2} + x_{4_6} &\leq 1; \\x_{8_2} + x_{6_4} &\leq 1;\end{aligned}$$

$$\forall x_{i_j} \in \{0, 1\}$$

Ejemplo Práctico.

Vamos a resolver el siguiente problema:



Tablero de partida, 7x7

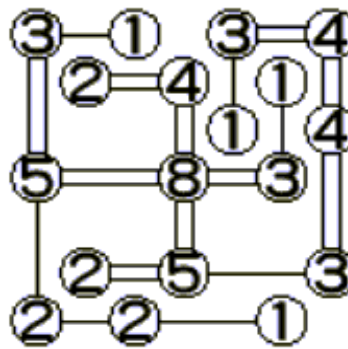
Para ello introducimos en matlab la siguiente la matriz de partida:

```
M=[3 0 1 0 3 0 4;
    0 2 0 4 0 1 0;
    0 0 0 0 1 0 4;
    5 0 0 8 0 3 0;
    0 0 0 0 0 0 0;
    0 2 0 5 0 0 3;
    2 0 2 0 0 1 0]
```

y ahora utilizamos el generador Hashi(M, 1, 'practico1.lp') para obtener el modelado del problema en formato lp.

Los formatos LP y MPS se encuentran en los archivos adjuntos.

LP_solve nos da la siguiente solución:



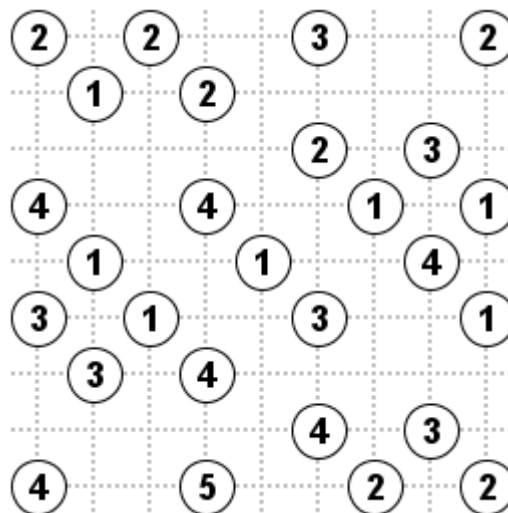
Tablero solución, 7x7

Las variables de la función objetivo que toman el valor 1 son:

X1_3, X1_22, X22_1, X5_7, X7_5, X5_19, X7_21, X21_7, X9_11, X11_9, X11_25, X25_11, X13_27, X21_42, X42_21, X22_25, X25_22, X22_43, X25_27, X27_25, X25_39, X39_25, X37_39, X39_37, X39_42, X43_45, X45_48.

Otro Ejemplo.

Vamos a resolver por último un tablero difícil, que es el siguiente:



Tablero de partida, 9x9

Para ello introducimos en matlab la siguiente la matriz de partida:

M=[2 0 2 0 0 3 0 0 2;

```

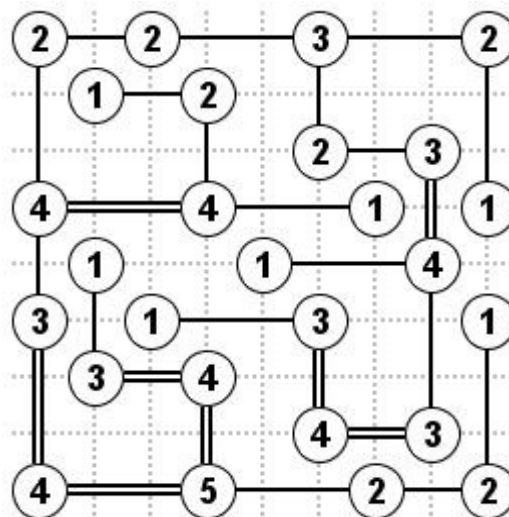
0 1 0 2 0 0 0 0 0;
0 0 0 0 0 2 0 3 0;
4 0 0 4 0 0 1 0 1;
0 1 0 0 1 0 0 4 0;
3 0 1 0 0 3 0 0 1;
0 3 0 4 0 0 0 0 0;
0 0 0 0 0 4 0 3 0;
4 0 0 5 0 0 2 0 2]

```

y ahora utilizamos el generador Hashi(M, 1, 'difícil.lp') para obtener el modelado del problema en formato lp.

Los formatos LP y MPS se encuentran en los archivos adjuntos

LP_solve nos da la siguiente solución:



Tablero solución, 9x9

Las variables de la función objetivo que toman el valor 1 son:

X1_3, X1_28, X3_6, X6_9, X6_24, X9_36, X11_13, X13_31, X24_26, X26_44, X44_26, X28_31, X31_28, X46_28, X34_31, X56_38, X41_44, X44_71, X46_73,

X73_46, X48_51, X51_69, X69_51, X54_81, X56_58, X58_56, X58_76, X76_58,
X69_71, X71_69, X73_76, X76_73, X76_79, X79_81

Finalmente mostramos el generador Hashi utilizado en Matlab:

function Hashi(matriz,lp,file)

%Autores: Juan José Gutierrez Ortega

% Javier Manchón López

% Manuel Sánchez Matas

%matriz => tablero

%lp => si lp=1 formato lp_solve sino XpressMP

%file => nombre del fichero de salida

if nargin>2

 file=fopen(file, 'w');

else

 file=1;

end;

n=size(matriz,1); %numero de filas

m=size(matriz,2); %numero de columnas

if lp

 fprintf(file,'% Objective function *\nmin:\n');

else

 fprintf(file,'Minimize\n obj: \n');

end;

%funcion objetivo

for i=1:n

 for j=1:m

 k=i;

```

for p=j-1:0
    if p>0
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'+x%d_%d ',islaA,islaE);
        fprintf(file,'+x%d_%d ',islaE,islaA);
    end;
end;

for p=j+1:m

    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    fprintf(file,'+x%d_%d ',islaA,islaE);
    fprintf(file,'+x%d_%d ',islaE,islaA);
end;

p=j;

for k=i-1:0
    if k>0
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'+x%d_%d ',islaA,islaE);
        fprintf(file,'+x%d_%d ',islaE,islaA);
    end;
end;

for k=i+1:n
    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    fprintf(file,'+x%d_%d ',islaA,islaE);
    fprintf(file,'+x%d_%d ',islaE,islaA);
end;

```



```

        end;
end;

if lp
    fprintf(file,'\n\n* Constraints */\n');
else
    fprintf(file,'\n\nSubject to\n');
end;

%RESTRICCIONES

for i=1:n
    for j=1:m
        if matriz(i,j)~=0
            valor=matriz(i,j);
            k=i;
            p=j-1;
            while p>0 && matriz(k,p)==0
                p=p-1;
            end;
            if p>0
                islaA=m*(i-1)+j;
                islaE=m*(k-1)+p;
                fprintf(file,'+ x%d_%d + x%d_%d ',islaA,islaE,islaE,islaA);
            end;

            k=i;
            p=j+1;
            while p<=m && matriz(k,p)==0
                p=p+1;
            end;
            if p<=m
                islaA=m*(i-1)+j;
                islaE=m*(k-1)+p;

```

```

        fprintf(file,'+ x%d_%d + x%d_%d ',islaA,islaE,islaE,islaA);
    end;

    k=i-1;
    p=j;
    while k>0 && matriz(k,p)==0
        k=k-1;
    end;
    if k>0
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'+ x%d_%d + x%d_%d ',islaA,islaE,islaE,islaA);
    end;

    k=i+1;
    p=j;
    while k<=n && matriz(k,p)==0
        k=k+1;
    end;
    if k<=n
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'+ x%d_%d + x%d_%d ',islaA,islaE,islaE,islaA);
    end;

    if lp
        fprintf(file,' = %d;\n', valor);
    else
        fprintf(file,' = %d\n', valor);
    end;

end;

end;

end;

```

%RESTRICCIONES PARA SABER SI EXISTEN PUENTES QUE SE CRUZAN

```
for i=2:n*m
    encontrado=0;
    fila=floor(i/m)+1;
    columna=mod(i,m);
    if columna>1
        for k=fila+2:n
            for f=fila+1:k-1
                islaA=m*(fila-1)+columna;
                islaB=m*(k-1)+columna;
                for j=1:columna-1
                    for p=columna+1:m
                        islaC=m*(f-1)+j;
                        islaD=m*(f-1)+p;
                        if lp
                            if matriz(fila,columna)~=0 && matriz(k,columna)~=0 && matriz(f,j)~=0 &&
matriz(f,p)~=0 && encontrado==0
                                fprintf(file, 'x%d_%d + x%d_%d <= 1;\n', islaA, islaB, islaC, islaD);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1;\n', islaA, islaB, islaD, islaC);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1;\n', islaB, islaA, islaC, islaD);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1;\n', islaB, islaA, islaD, islaC);
                                encontrado=1;
                            end;
                        else
                            if matriz(fila,columna)~=0 && matriz(k,columna)~=0 && matriz(f,j)~=0 &&
matriz(f,p)~=0 && encontrado==0
                                fprintf(file, 'x%d_%d + x%d_%d <= 1\n', islaA, islaB, islaC, islaD);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1\n', islaA, islaB, islaD, islaC);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1\n', islaB, islaA, islaC, islaD);
                                fprintf(file, 'x%d_%d + x%d_%d <= 1\n', islaB, islaA, islaD, islaC);
                                encontrado=1;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
```

```

        end;
    end;
end;
end;
end;
end;
end;
end;

if lp
    fprintf(file,'\n/* Variable bounds */\n');

    for i=1:n
        for j=1:m
            k=i;
            for p=j-1:0
                if p>0
                    islaA=m*(i-1)+j;
                    islaE=m*(k-1)+p;
                    fprintf(file,'0 <= x%d_%d <=1;\n',islaA,islaE);
                    fprintf(file,'0 <= x%d_%d <=1;\n',islaE,islaA);
                end;
            end;

            for p=j+1:m

                islaA=m*(i-1)+j;
                islaE=m*(k-1)+p;
                fprintf(file,'0 <= x%d_%d <=1;\n',islaA,islaE);
                fprintf(file,'0 <= x%d_%d <=1;\n',islaE,islaA);
            end;

            p=j;
        end;
    end;
end;

```

```

for k=i-1:0
    if k>0
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'0 <= x%d_%d <=1;\n',islaA,islaE);
        fprintf(file,'0 <= x%d_%d <=1;\n',islaE,islaA);
    end;
end;

for k=i+1:n
    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    fprintf(file,'0 <= x%d_%d <=1;\n',islaA,islaE);
    fprintf(file,'0 <= x%d_%d <=1;\n',islaE,islaA);
end;
end;
end;
else
    fprintf(file,'\nBinaries\n');

for i=1:n
    for j=1:m
        k=i;
        for p=j-1:0
            if p>0
                islaA=m*(i-1)+j;
                islaE=m*(k-1)+p;
                fprintf(file,'x%d_%d\n',islaA,islaE);
                fprintf(file,'x%d_%d\n',islaE,islaA);
            end;
        end;
    end;

    for p=j+1:m

```

```

        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'x%d_%d\n',islaA,islaE);
        fprintf(file,'x%d_%d\n',islaE,islaA);
    end;

    p=j;

    for k=i-1:0
        if k>0
            islaA=m*(i-1)+j;
            islaE=m*(k-1)+p;
            fprintf(file,'x%d_%d\n',islaA,islaE);
            fprintf(file,'x%d_%d\n',islaE,islaA);
        end;
    end;

    for k=i+1:n
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,'x%d_%d\n',islaA,islaE);
        fprintf(file,'x%d_%d\n',islaE,islaA);
    end;
end;

end;

end;

if lp
    fprintf(file,'% Integer definitions */\nint x1_2, x2_1');
    for i=1:n
        for j=1:m
            k=i;
            for p=j-1:0

```

```

if p>0
    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    fprintf(file,' x%d_%d',islaA,islaE);
    fprintf(file,' x%d_%d',islaE,islaA);
end;
end;

for p=j+1:m

    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    if islaA~=1 || islaE~=2
        fprintf(file,' x%d_%d',islaA,islaE);
        fprintf(file,' x%d_%d',islaE,islaA);
    end;
end;

p=j;

for k=i-1:0
    if k>0
        islaA=m*(i-1)+j;
        islaE=m*(k-1)+p;
        fprintf(file,' x%d_%d',islaA,islaE);
        fprintf(file,' x%d_%d',islaE,islaA);
    end;
end;

for k=i+1:n
    islaA=m*(i-1)+j;
    islaE=m*(k-1)+p;
    fprintf(file,' x%d_%d',islaA,islaE);
    fprintf(file,' x%d_%d',islaE,islaA);

```

```
        end;
    end;
end;
fprintf(file, '\n');
else
    fprintf(file, '\nEnd');
end;
if nargin>2, fclose(file); end;
```

Referencias Bibliográficas/Web

(Wikipedia, 2009) <http://es.wikipedia.org/wiki/Hashiwokakero>. Acceso, 25 de noviembre de 2009.

(Microsiervos) <http://www.microsiervos.com/archivo/juegos-y-diversion/hashiwokakero.html>