

## ANEXO 11

### Manual de Usuario

Este manual está destinado para todos aquellos que deseen utilizar el algoritmo SCPM adaptado por el equipo de trabajo del Documento Principal (DP) titulado *Diseño de Modelo de Ruteo Eficiente para Búsqueda de Minas Antipersona en Colombia*. Se explicarán una serie de pasos para poder obtener finalmente una ruta acorde con lo establecido en el trabajo anteriormente mencionado.

#### REQUISITOS

Instalar la última versión disponible de la API AMPL C++ (al momento de redacción de este manual, la última versión es 2.0.4.0).

Contar con un compilador moderno del lenguaje C++.

Sugerencia: para mayor claridad en la descarga de la AMPL API para C++, habilitación de la librería y ubicación de programas necesarios ver este video: [https://www.youtube.com/watch?v=KO\\_H1ME1NSA](https://www.youtube.com/watch?v=KO_H1ME1NSA). Tenga en cuenta que la información está dirigida a usuarios con sistema operativo Windows.

Cada instancia, debe poseer ciertos archivos necesarios para poder trabajar con ella. En la Fig. 34 se presentan los archivos que deben estar contenidos en una instancia llamada '10x10\_n'. A continuación, se procede a describir el cómo obtener estos archivos.

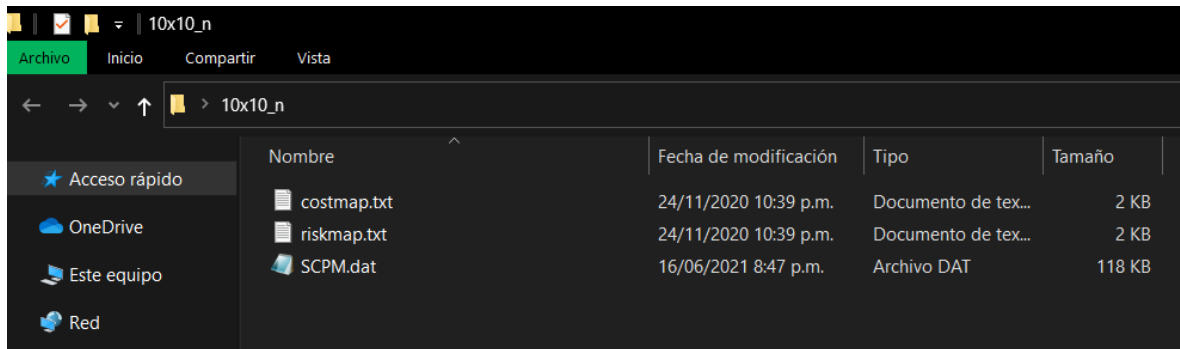


Fig. 34. Archivos de cada instancia

#### A. Estructura de las instancias

En este apartado, se procede a describir los elementos que componen los archivos de texto (.txt) que contienen las instancias que representan el Área Peligrosa.

Inicialmente, se requiere que el usuario cree una carpeta con el nombre de la instancia. Las instancias usadas en el DP se caracterizan por nombrarse con sus dimensiones y un indicador único que las diferencia de las demás. Por ejemplo, si la instancia es de diez nodos de ancho por diez nodos de alto, entonces el nombre de la carpeta sería '10x10\_n' donde n es el identificador único.

Una vez creada la carpeta, en ella deben existir dos archivos de texto llamados 'costmap.txt' y 'riskmap.txt' que contienen los valores de costo y riesgo, respectivamente, asociados a cada nodo de una instancia en específico. En la Fig. 35 a) está la representación gráfica de una instancia genérica con todos los obstáculos que la integran. Internamente, el modelo representa los nodos como un conjunto de identificadores numéricos que corresponden a una enumeración consecutiva iniciada en la celda superior izquierda del mapa de la instancia, y

continúa de manera horizontal, correspondiendo a cada nodo un único identificador, como se ilustra en Fig. 35 b).

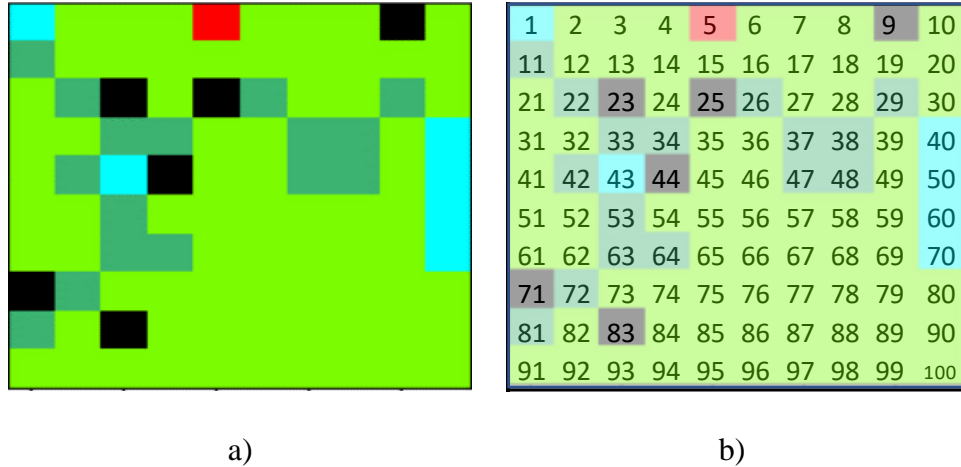


Fig. 35. Instancia genérica (gráfica)

En la Fig. 36 se encuentran el contenido de los archivos de texto 'costmap.txt' y 'riskmap.txt' en los incisos a) y b) respectivamente.

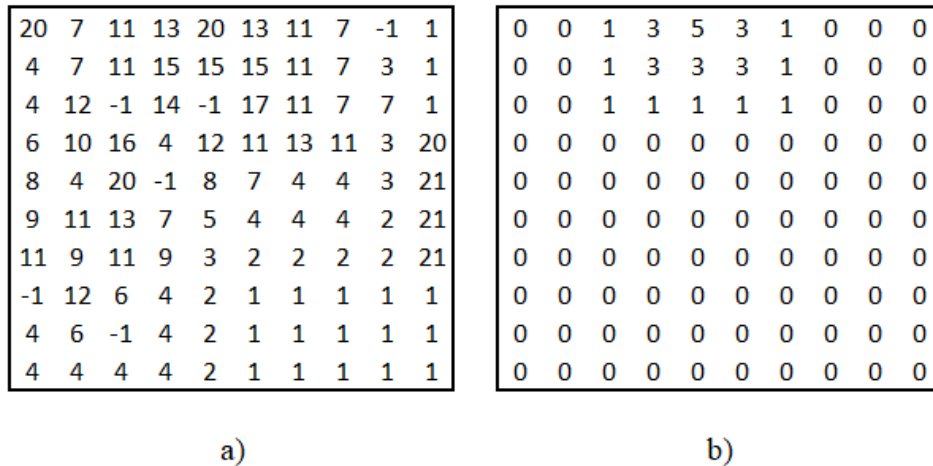


Fig. 36. Estructura de los archivos. a) costmap; b) riskmap

Para la instancia en consideración, se observa que existen patrones de distintos colores que corresponden a distintos tipos de obstáculos, como está explicado en el DP. Los valores tanto de a) como de b) en la Fig. 36 son asignados por el usuario. La ventaja de este modelo es que permite el uso de valores esquemáticos que se adapten a las necesidades del usuario para obtener soluciones bajo diferentes contextos. Sin embargo, siempre que un nodo particular represente algún tipo de obstáculo intransitable (como, por ejemplo, rocas o muros), en el archivo de texto 'costmap.txt', debe asignarle un costo equivalente a menos uno (-1). De esta forma, el

algoritmo SCPM interpretará que tal nodo no puede ser usado en la ruta. Es importante remarcar que este valor es diferente a indicar la presencia de una mina, pues en ese caso es posible desplazarse, pero no es adecuado para preservar la integridad del Agente Desminador.

De esa manera, los archivos ‘costmap.txt’ y ‘riskmap.txt’ se visualizarían como en la Fig. 37 a) y b), respectivamente.

costmap.txt: Bloc de notas									
Archivo	Edición	Formato	Ver	Ayuda					
20	7	11	13	20	13	11	7	-1	1
4	7	11	15	15	15	11	7	3	1
4	12	-1	14	-1	17	11	7	7	1
6	10	16	4	12	11	13	11	3	20
8	4	20	-1	8	7	4	4	3	21
9	11	13	7	5	4	4	4	2	21
11	9	11	9	3	2	2	2	2	21
-1	12	6	4	2	1	1	1	1	1
4	6	-1	4	2	1	1	1	1	1
4	4	4	4	2	1	1	1	1	1

a)

riskmap.txt: Bloc de notas									
Archivo	Edición	Formato	Ver	Ayuda					
0	0	1	3	5	3	1	0	0	0
0	0	1	3	3	3	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

b)

Fig. 37. Apariencia de los archivos de texto. a) costmap; b) riskmap

### B. Creación del archivo SCPM.dat

Para poder utilizar el modelo SCPM con el fin de obtener la ruta de cubrimiento de una instancia, se hace necesario traducir las características de costo y riesgo de esta en parámetros expresados en sintaxis válida de AMPL. Esta información se consolida en un archivo .dat, que debe poseer el nombre ‘SCPM.dat’. La construcción de este fichero se realiza en dos etapas haciendo uso de dos utilidades o ejecutables, como se expone a continuación.

### 1) Primera etapa

Haciendo uso del ejecutable *datparser*, se extrae información referente a la instancia de sus archivos ‘costmap.txt’ y ‘riskmap.txt’, y se crean, en sintaxis AMPL, los siguientes parámetros:

- Conjunto *rocks*: compuesto por los identificadores de aquellos nodos sobre los cuales se encuentra localizado un obstáculo intransitable.
- Parámetro *N*: constante escalar que indica el número total de nodos de la instancia.
- Parámetro *COV*: Matriz binaria, cuyo valor  $COV_{ij}$  indica si desde el nodo *i*, se puede cubrir con el alcance el nodo *j*. Se asume que el nodo *i*, al ser accedido, queda cubierto. Esta es la razón por la que  $COV_{ij}$  equivale a uno (1), cuando  $i = j$ . El nombre del parámetro son las tres primeras letras de cubrimiento en inglés (covering) en mayúscula.
- Parámetro *A*: Matriz binaria, cuyo valor  $A_{ij}$  indica si es posible acceder al nodo *j* desde el nodo *i*. Cuando  $i = j$ ,  $A_{ij}$  es equivalente a cero, pues no es posible acceder a un lugar donde actualmente se está ubicado. El nombre del parámetro se debe a la primera letra de la palabra adyacencia en mayúscula.
- Parámetro *C*: Vector escalar, donde el valor  $C_i$  indica el costo de acceder al nodo *i*. Los nodos intransitables no poseen un costo asociado ya que no pueden ser accedidos.
- Parámetro *R*: Vector escalar, donde el valor  $R_i$  indica el riesgo de acceder al nodo *i*.
- Parámetro *NADIR\_COST*: Punto nadir para la dimensión de costo.
- Parámetro *NADIR\_RISK*: Punto nadir para la dimensión de riesgo.

Para utilizar el ejecutable, se debe abrir una terminal y digitar el siguiente comando:

```
$ datparser instance_name
```

instance\_name: la ruta al directorio de la instancia, donde estén sus archivos.

En caso de que el ‘datparser’ no esté incluido en la variable de entorno *PATH*, el directorio activo de la consola deberá ser aquel donde está el ejecutable.

### 2) Segunda etapa

Si el lector observa en la anterior etapa, existen dos parámetros denominados *NADIR\_COST* y *NADIR\_RISK*. Estos son necesarios para la estandarización de la función objetivo del modelo. Sin embargo, en este punto, aún resta por consolidar en el archivo .dat los valores objetivos utópicos para las dimensiones de costo y riesgo. Esto se consigue usando la utilidad *findutopic*, la cual obtiene el valor utópico para una dimensión específica, y la adjunta al archivo ‘SCPM.dat’. Esto se logra usando el siguiente comando:

```
$ findutopic instance_name file_mod
```

instance\_name: la ruta al directorio de la instancia, donde esté sus archivo .dat.

file\_mod: se debe especificar el archivo .mod correspondiente a la dimensión (costo o riesgo) a la cual se le hallará el valor utópico. Este proceso consta de resolver la instancia específica con un modelo relajado que retorne un valor objetivo muy favorable, el cual no esté contenido en la región factible. La función objetivo dependerá del argumento especificado. Los valores pueden ser: ‘SCPMC.mod’ para la dimensión de costo; ‘SCPMR.mod’ para la dimensión de riesgo.

En caso de que el 'findutopic' no esté incluido en la variable de entorno PATH, el directorio activo de la consola deberá ser aquel donde está el ejecutable.

Ejemplo:

Se halla el valor utópico de costo para la instancia '10x10\_2'

```
$ findutopic 10x10_2 SCPMC.mod
```

Se halla el valor utópico de riesgo para la instancia '10x10\_2'

```
$ findutopic 10x10_2 SCPMR.mod
```

Los valores utópicos quedarán expresados en el archivo 'SCPM.dat' como los parámetros UTOPIC\_COST, UTOPIC\_RISK para la dimensión de costo y riesgo, respectivamente.

### *C. Resolución de instancia*

Ya obtenidos los archivos necesarios listados en la Fig. , se puede solucionar la instancia para obtener la ruta de cubrimiento. Esto se logra usando la utilidad 'SCPM', como se describirá a continuación. Se abre una terminal, y se ejecuta el siguiente comando:

```
$ SCPM instance_name
```

instance\_name: la ruta al directorio de la instancia, donde estén sus archivos.

En caso de que 'SCPM' no esté incluido en la variable de entorno PATH, el directorio activo de la consola deberá ser aquel donde está el ejecutable.

Finalmente, los resultados que se obtienen al solucionar una instancia son: nombre de la instancia; KPI's definidos en el proyecto (CTP y RAP); factor de preferencia entre las dimensiones de costo y riesgo (ver definiciones del modelo en sección V); función objetivo; MIPGAP (brecha de optimalidad); Estado final de resolución ('solve\_result\_num' de AMPL); Número de restricciones EAST añadidas; Número iteraciones; Tiempo de ejecución. Dichas estadísticas se registran en el documento de texto "Resultados.txt".

Para consultar los códigos fuente realizados por los integrantes del equipo, consulte el siguiente repositorio: <https://github.com/sserna1231/SCPM>