

## Superficies de coste anisotrópicas

C. Romo Peña<sup>(1)</sup> y J.C Torres Cantero <sup>(1)</sup>

<sup>(1)</sup> Laboratorio de Realidad Virtual de la Universidad de Granada. C/ Zamora, parcela 127-128, Polígono Tecnológico de Ogíjares, 18151 Ogíjares (Granada), [cromo@ugr.es](mailto:cromo@ugr.es), [jctorres@ugr.es](mailto:jctorres@ugr.es)

### RESUMEN

*Las superficies de coste son una solución precalculada del problema de camino mínimo en mapas raster. Normalmente los sistemas GIS utilizan un mapa para almacenar los costes unitarios, que son constantes para cada celda, lo que implica resolver un problema isotrópico. No obstante, los problemas reales no suelen ser isotrópicos.*

*En este trabajo se propone un método para calcular superficies de coste anisotrópicas. El método se ha implementado en Grass usando dos nuevas órdenes: `r.acost` y `r.adrain`.*

*El artículo explica el método de cálculo, su implementación en Grass, cómo generar automáticamente los mapas de coste unitario cuando el coste depende de la elevación y evalúa los resultados*

**Palabras clave:** *Superficies de coste anisotrópicas, camino mínimo, Grass GIS*

### ABSTRACT

*Cost surfaces are used as a precomputed solution of the shortest path problem on raster maps. Usually GIS systems use a map to store unit cost, wich implies to solve an isotropic problem. But real problems are often anisotropic. This work proposes a method to compute anisotropic cost surfaces. The method has been implemented on Grass as two new commands: `r.acost` and `r.adrain`.*

*The paper describes the approach, the GRASS implementation, how to generate automatically the unit cost map when the cost depend on an elevation map and evaluate the results.*

**Key words:** *Anisotropic cost surfaces, shortest path, Grass GIS*

### INTRODUCCIÓN

Las superficies de coste son una solución precalculada del problema de camino mínimo en mapas raster. El problema del camino mínimo se puede definir de la siguiente forma:

Dado un mapa raster **M** que representa una función de fricción (cada una de sus celdas contiene el coste de cruzarla), y dos puntos **P** y **Q** sobre el mapa, calcular el coste mínimo del desplazamiento de **P** a **Q**.

De esta forma, al modificar cualquiera de los dos puntos de entrada, se debe volver a calcular el camino mínimo (ver figura 1, izquierda). La utilización de superficies de coste permite realizar el proceso en dos pasos. En primer lugar calculamos el coste mínimo (coste de desplazarse siguiendo el camino mínimo) de uno de los puntos a cualquier lugar del territorio. El resultado de este proceso es un mapa que se conoce como superficie de coste. En un segundo paso calculamos el camino mínimo al punto destino usando la superficie de coste como entrada (ver figura 1, derecha).

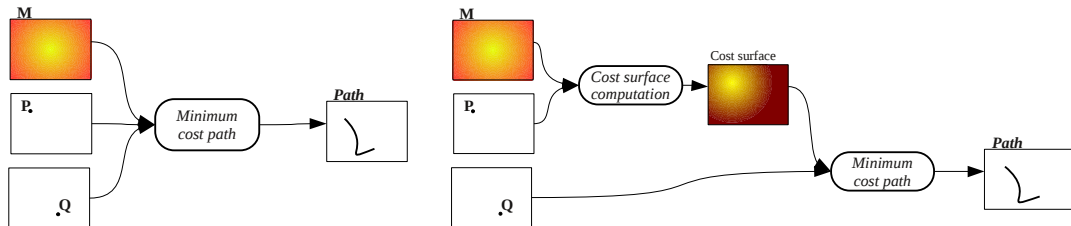


Figura 1: Cálculo del camino mínimo sin superficies de coste (izquierda) y con superficies de coste (derecha).

La ventaja de este proceso es que el coste computacional de generar la superficie de coste es similar al de calcular directamente el camino mínimo entre dos puntos, mientras que el cálculo del camino mínimo a partir de la superficie de coste es trivial. De esta forma no es necesario repetir todo el cálculo si cambia solo uno de los puntos.

Por otra parte, las superficies de coste tienen aplicaciones en sí mismas, ya que permiten mostrar información de distribución de costes sobre el territorio, para analizar tiempos de desplazamientos, optimización de emplazamientos, etc.

Para calcularlas se suele usar el algoritmo de Dijkstra [1], asumiendo una red de conexión entre celdas que conecta cada una con sus vecinas. Para cada celda podemos considerar diferentes topologías con conexiones a: cuatro vecinas, ocho vecinas o dieciséis vecinas. GRASS permite calcular superficies de coste con topologías de ocho y dieciséis vecinas.

Un aspecto importante es cómo se calcula el coste en cada celda (coste unitario o fricción). El algoritmo de Dijkstra asocia un coste a las conexiones entre celdas (arcos del grafo). Usualmente los sistemas GIS utilizan un mapa para almacenar los costes unitarios, que son constantes para cada celda, lo que implica resolver un problema isotrópico. No obstante, los problemas reales no suelen ser isotrópicos. Normalmente el coste de desplazamiento de A a B no es el mismo que el de desplazarse de B a A, es más, el coste de cruzar una celda en dirección norte-sur suele ser distinto de cruzarla en dirección este-oeste. Esto ocurre, por ejemplo, cuando se está caminando o construyendo un canal.

## Trabajos previos

Se han utilizado superficies de coste anisotrópicas para resolver problemas de distinto tipo. J. Cooper utiliza el cálculo de caminos mínimos con funciones de fricción anisotrópicas para estudiar modelos de movimiento e intercambio entre yacimientos arqueológicos precolombinos en Cuba [2]. Los cálculos se realizaron utilizando la función `r.walk` de Grass.

S. Ebener propuso utilizar superficies de coste anisotrópicas para determinar distritos de centros de asistencia médica, encontrando diferencias significativas respecto a los cálculos realizados con coste isotrópico. Para realizar el estudio se desarrolló una extensión para ArcGIS (MAPA) [3]

Michels ha utilizado superficies de coste anisotrópicas para calcular la distancia biológica entre el zooplacton en un conjunto de pozas interconectadas [4]. Por su parte.

Jobe las ha usado para estudiar la accesibilidad en zonas de montaña, para ello han utilizado la función `pathDistance` de ArcGIS [5], mientras que Bagli las ha aplicado para diseñar el trazado de líneas de alta tensión [6].

Algunos sistemas GIS incorporan funciones para calcular superficies de coste anisotrópicas. Idrisi incluye la función `varcost` para calcular superficies de coste anisotrópicas. `varcost` asume una variación continua de la fricción, que se especifica dando la dirección de máxima fricción, y la fricción en esta dirección:

$$F = F_0 \cdot \cos(\alpha - \alpha_0) \quad (1)$$

Grass dispone de la función `r.walk` para calcular superficies de coste anisotrópicas que representan tiempo de desplazamiento andando, que calcula la fricción a partir de la fricción isotrópica y del modelo de elevación, utilizando la fórmula de Langmuir:

$$T = a \cdot \Delta S + b \cdot \Delta H_+ + c \cdot \Delta H_- + d \cdot \Delta H_{--} \quad (2)$$

que expresa el tiempo de desplazamiento en función de la distancia recorrida, y la diferencia de altura, con tres coeficientes distintos en función del desnivel presente [7,8].

ArcGIS dispone de la función `PathDistance`, que permite calcular superficies de coste anisotrópicas en función de relieve y de la variación de la fricción en función de la orientación especificada para todo el territorio. Esto último permite modelar situaciones en las que la anisotropía es constante en el mapa, como la influencia del viento en el desplazamiento.

Collischonn y Pilar [9] propusieron un algoritmo de cálculo de caminos mínimos con coste anisotrópico para el diseño de carreteras y canales. Su algoritmo resuelve el problema cuando el coste es función de la elevación. El algoritmo es bastante complejo y no calcula superficies de coste, por lo que debe recalcularse cuando cambia cualquiera de los extremos del camino.

Chaoqing et al. usan el método de Collischonn mejorando la estimación de las distancias al tener en cuenta la diferencia de altura [10].

## Objetivos y estructura del trabajo

En este trabajo se propone un método para calcular superficies de coste anisotrópicas basado en la utilización de mapas de coste direccionales. El método se ha implementado en Grass usando dos nuevos órdenes: `r.acost` y `r.adrain`. El artículo explica el método de cálculo, su implementación en Grass, cómo generar automáticamente los mapas de coste unitario cuando el coste depende de la elevación y comenta resultados de la evaluación del algoritmo en un caso práctico.

La sección siguiente expone brevemente el algoritmo de cálculo de funciones de coste isotrópicas. La sección tercera describe el algoritmo propuesto. La sección cuarta muestra resultados de la evaluación del método.

## SUPERFICIES DE COSTE ISOTRÓPICAS

El cálculo de superficies de coste suele realizarse usando el algoritmo de Dijkstra basado en dividir el mapa en tres zonas:

- Celdas con coste conocido (azul en el la figura 2)

- Celdas con una cota de coste calculada. El coste mínimo es menor o igual que el temporalmente calculado (celdas rojas en la figura 2).
- Celdas no calculadas.

Las celdas con cota de coste envuelven a las celdas con coste conocido. Para las celdas que están en la frontera se ha calculado una cota de coste que es el coste mínimo de alcanzarlas desde cualquier camino que no cruce la frontera. Quedan por calcular los costes de los caminos que salen de la zona calculada, pero el coste de éstos siempre será mayor que el mínimo en la frontera. Luego la celda con coste más bajo en la frontera tiene coste correcto.

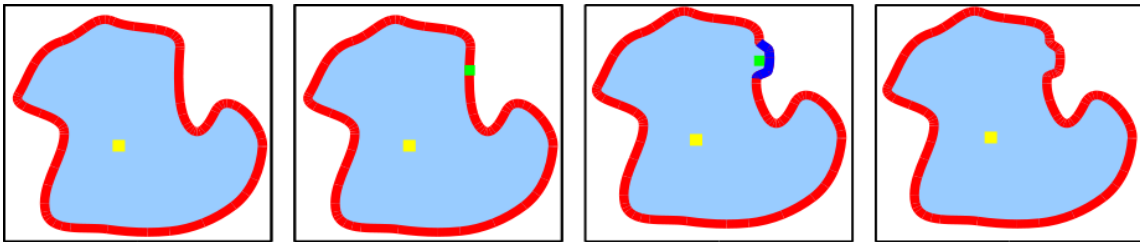


Figura 2: Proceso de cálculo de la superficie de coste.

El procedimiento de cálculo selecciona en cada iteración la celda de menor coste de la frontera (con cota de coste calculada), fija su coste y calcula una cota de coste para sus vecinas (ver figura 2), añadiéndolas a la lista de celdas con cota de coste.

Grass realiza el cálculo de la superficie de coste con la función *r.cost*, que toma como parámetros un mapa con la fricción, y el punto o mapa origen de la superficie de coste, generando un mapa raster que contiene la superficie de coste. El mapa de fricción (o coste unitario) contiene el coste de cruzar cada celda. Por tanto, el coste de desplazamiento de una celda a otra celda vecina es la mitad de la suma de los costes unitarios de ambas celdas, ponderado por la longitud del desplazamiento respecto al movimiento este-oeste (ver figura 3).

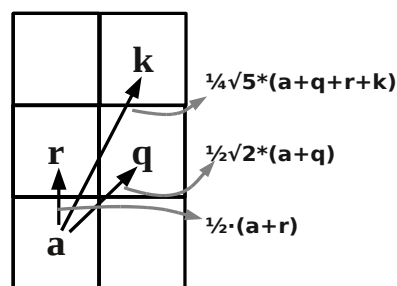


Figura 3: Cálculo del coste de desplazamiento en función del tipo de movimiento.

Para calcular el camino mínimo se sigue un algoritmo de descenso de colina, que partiendo de la información de la superficie de coste y de la posición de la celda destino, que construye el camino iterativamente, partiendo de la celda destino y añadiendo en cada iteración la celda vecina de menor coste. Este método es viable dado que al ser la fricción isotrópica, el coste de desplazamiento en todas las direcciones es el mismo, y por tanto el camino mínimo pasa por el vecino de menor coste.

## ALGORITMO PROPUESTO

La primera decisión en el diseño del algoritmo es la representación de los costes unitarios o fricciones. Hemos optado por utilizar un mapa para representar la fricción en cada dirección, de este modo se representa la fricción con un conjunto de ocho mapas. De cada celda se almacena, en cada mapa de fricción, el coste de desplazamiento a la celda vecina en esa dirección (ver figura 4). Esta estrategia tiene el inconveniente de necesitar ocho mapas para representar la fricción, pero es más general que utilizar una distribución prefijada de la fricción, o una relación prefijada entre la fricción y el modelo de elevación. De hecho, el conjunto de funciones de fricción representables de este modo incluye a las representables con las otras dos alternativas. Por otro lado, la discretización de la fricción en ocho direcciones no supone una pérdida de precisión, dado que, al trabajar con información raster, la superficie de coste se calcula sobre esas direcciones.

### Superficie de coste anisotrópica

En el caso anisotrópico tampoco es posible que una celda ya calculada tenga menor coste llegando desde una celda de la frontera.

La celda de la frontera podría tener un coste mucho menor que el temporal llegando por algún camino futuro. Pero el coste de este camino siempre será menor que el del punto de partida del camino en la frontera, y éste debe ser mayor que el de la celda calculada ya que está fue la más baja de la frontera en algún momento anterior.

Por tanto, el algoritmo de cálculo de la superficie de coste es el mismo que el usado en el caso isotrópico, con la salvedad de que en este caso se utilizan ocho mapas para representar costes de desplazamiento. Esta circunstancia se ha aprovechado para almacenar directamente el coste de desplazamiento hasta la celda vecina, en lugar del coste por unidad de longitud, ya que de esta forma el cálculo es más rápido.

Sí es necesario modificar el proceso de cálculo de la superficie de coste y el de generación del camino mínimo ya que, en este caso, el camino mínimo no pasa necesariamente por la celda vecina de menor coste.

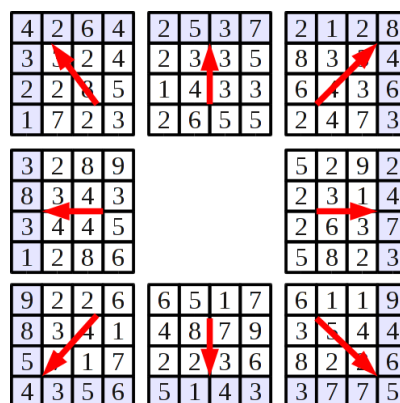


Figura 4: Mapas de fricción anisotrópica.

Tal como se ha comentado previamente, en el caso anisotrópico el camino mínimo no pasa necesariamente por la celda vecina de menor coste, por tanto, es necesario almacenar, durante el cálculo de la superficie de coste, no solo el coste de la celda, sino la dirección desde la que se ha llegado a la celda para conseguir ese coste.

Por tanto, generamos dos mapas de salida, la superficie de coste y un mapa de direcciones de mínimo coste (ver figura 5). Las direcciones se codifican como un número de dos cifras, en la que la primera cifra indica la dirección este-oeste y la

segunda la dirección sur-norte. La figura muestra el estado del proceso de cálculo al comienzo del algoritmo cuando se ha cargado el punto de partida con coste cero (amarillo en la figura) y se ha creado la banda de celdas con cota de coste alrededor de este (celdas celestes en la figura). De cada una de estas celdas se ha almacenado en el mapa de direcciones (en el centro de la parte izquierda de la imagen) la dirección para recorrer el camino de mínimo coste. En este momento todos llevan a la celda de partida. Obsérvese que las direcciones por la que se ha llegado a una celda en la dirección de mínimo coste es la opuesta a la dirección por la que se ha avanzado al generar la superficie de coste. Así, en la figura 5, a la celda con coste 6, se ha llegado desde la de partida en dirección '01', por lo que para seguir el camino mínimo se debe usar la dirección '21'.

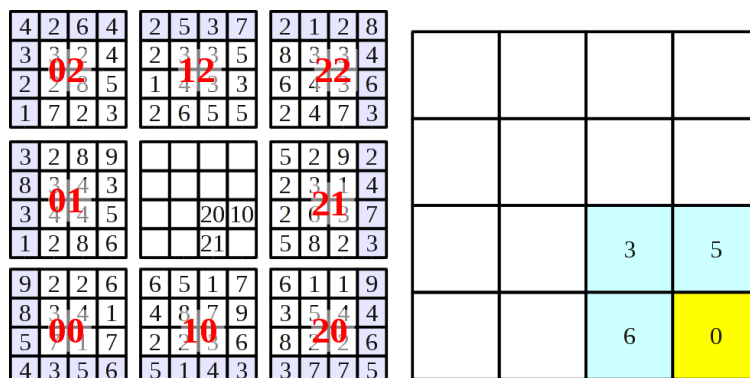


Figura 5: Cálculo de la superficie de coste. A la izquierda los mapas de fricción y en el centro de ellos el mapa de salida con direcciones de mínimo coste, a la derecha la superficie de coste. Los mapas de salida se muestran en el estado que tendrían en la segunda iteración del algoritmo.

Esta función se ha implementado generando un módulo (r.acost) que se ha integrado en las versiones 6.0 y 6.4 de Grass. Mantiene la misma interfaz que la versión isotrópica (r.cost), añadiendo parámetros para indicar los mapas de fricción y el mapa de direcciones de mínimo coste.

La generación del camino mínimo se realiza simplemente siguiendo la secuencia de pasos almacenados en la función de dirección. Se ha creado un módulo en Grass (r.adrain), con interfaz similar a la función isotrópica (r.drain), que realiza esta función. La salida de este módulo, al igual que la de r.drain, es raster.

### Movimientos con salto de caballo

La implementación actual no permite movimientos con salto de caballo. Esta sección analiza las alternativas para incluirlos. Hay dos opciones para contemplar movimientos de caballo:

- Incluir mapas de coste para las dieciséis direcciones.
- Estimar el coste a partir de los ocho mapas de coste existentes.

La primera opción tiene el inconveniente de necesitar más mapas de fricción, lo que no solo aumenta el espacio necesario, si no el tiempo de generación de éstos y el del cálculo de la superficie de coste.

Además, en la versión isotrópica basta con calcular la superficie de coste, y después descender por ella con r.drain. Ahora no es suficiente, debemos almacenar la dirección de llegada a cada celda, por lo que habría, en principio, que aumentar los códigos de dirección para incluir el salto de caballo. No obstante, como no estamos generando salida vectorial, el camino se visualizaría como una de las dos opciones indirectas. La figura 6 muestra las ocho casillas alcanzables con salto de caballo (en rosa) desde la celda central. Cada camino representa un desplazamiento

con inclinación múltiplo de 22.5°. A la derecha de la imagen se muestran las alternativas para alcanzar una de estas celdas con conectividad ocho. El camino se descompone en dos tramos, uno con inclinación 45° y otro con inclinación 0° ó 90°.

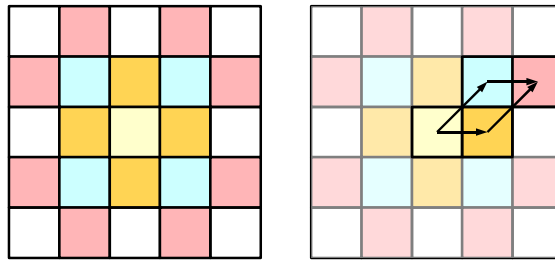


Figura 6: Caminos indirectos para realizar un desplazamiento en salto de caballo.

El camino más corto, en longitud, es el camino directo. El problema es estimar el coste de desplazarse en este camino. La longitud del camino directo es:

$$L_{k1} = \sqrt{4 \cdot R_{EW}^2 + R_{NS}^2} \quad L_{k2} = \sqrt{R_{EW}^2 + 4 \cdot R_{NS}^2} \quad (3)$$

La de los caminos indirectos

$$L_1 = R_{EW} + \sqrt{R_{EW}^2 + R_{NS}^2} \quad L_2 = R_{NS} + \sqrt{R_{EW}^2 + R_{NS}^2} \quad (4)$$

En el caso de que  $R_{EW} = R_{NS} = R$  (celdas cuadradas)

$$L_{k1} = L_{k2} = \sqrt{5} \cdot R \quad L_1 = L_2 = R + \sqrt{2} \cdot R = (1 + \sqrt{2}) \cdot R \quad (5)$$

El factor del primero es 2.24, mientras que el del segundo es 2.41, lo que supone un incremento de un 8%. Con esta diferencia no parece que el error que se cometa al estimar el coste de desplazamiento en el camino diagonal pueda ser decisivo. Cuando haya una diferencia grande de coste entre los dos caminos indirectos, el coste menor será el de uno de ellos. En caso contrario será más rentable el camino directo. Por tanto, proponemos como coste

$$C_{k1} = 0.92 \cdot \frac{C_1 + C_2}{2} \quad (6)$$

$$C_1 = C_{Horizontal} + C_{Diagonal, i+1, j} \quad C_2 = C_{Diagonal} + C_{Horizontal, i+1, j+1}$$

De esta forma podremos utilizar conectividad dieciséis (salto de caballo) sin necesidad de utilizar dieciséis mapas de coste unitario.

## EVALUACIÓN Y APLICACIONES

El método propuesto se ha implementado e integrado en Grass 6.4, y se ha evaluado sobre un ordenador con un procesador Intel Core i5 y en un sistema operativo Linux. Se han realizado pruebas calculando el desplazamiento andando en terreno de montaña en varias localizaciones y simulando procesos de difusión.

### Tiempo de desplazamiento

Para el calculo del tiempo de desplazamiento se ha utilizado la siguiente expresión:



$$\text{coste} = K_1 \cdot \text{pendiente}^2 \cdot \text{longitud} + K_2 \cdot \text{longitud} \quad (7)$$

donde *pendiente* es la pendiente en la dirección en la que se está calculando la fricción. Ésta es una versión simplificada de la fórmula de Langmuir. La expresión se calcula en un script que genera los ocho mapas de fricción en función de las constantes  $K_1$  y  $K_2$ .

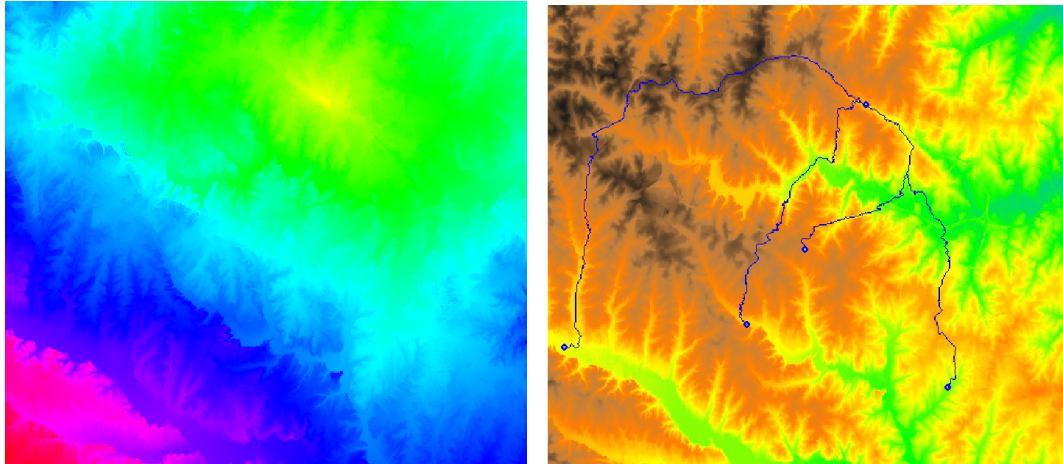


Figura 7: Superficie de coste anisotrópica (izquierda) y caminos mínimos (derecha).

El método se ha evaluado con modelos de elevación reales. La figura 7 muestra una superficie de coste anisotrópica generada para la localización nc\_spm\_08 usando como constantes 0.72 y 40000, y cuatro caminos mínimos. Puede observarse el trazado en zigzag en las zonas de pendiente, y cómo el camino aprovecha la rampa de la carretera, que tiene pendiente constante (ver detalles en figura 8).

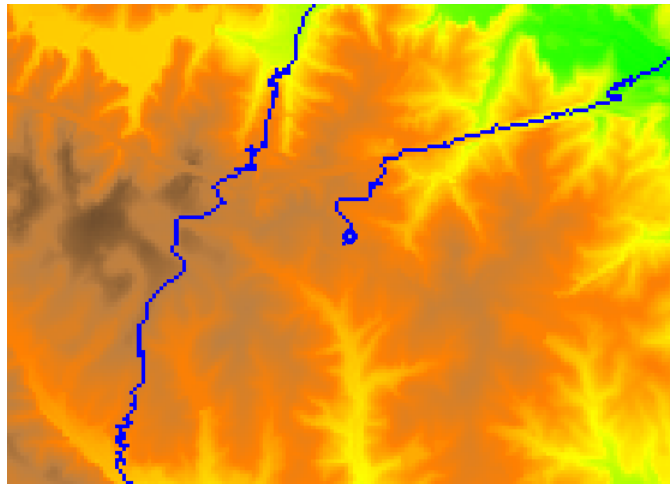


Figura 8: Detalle de los caminos mínimos mostrados en la figura 7. Puede apreciarse el trazado en zigzag (camino de la izquierda) y el trazado sobre la carretera (camino de la derecha)

La figura 9 muestra la superficie de coste como color sobre el modelo de elevación en un modelo digital de elevación de la cabecera del río Darro, usando como constantes en la expresión de la fricción (ecuación 7) 0.72 y 40. Puede apreciarse cómo el desplazamiento es más fácil manteniendo la cota a lo largo de la ladera.

La figura 10 muestra varios caminos mínimos desde el punto origen usado en la figura 9, comparando el camino anisotrópico (en rojo) con el camino isotrópico (en



azul. Para el camino isotrópico se ha usado una fricción dependiente de la pendiente en la celda, usando la misma ecuación 7.

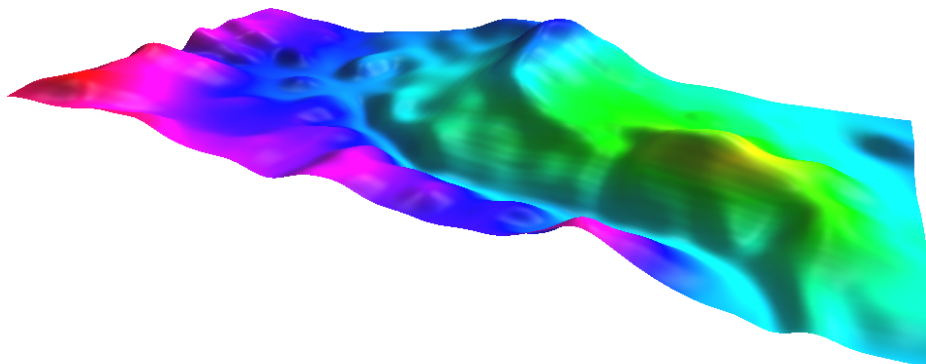


Figura 9: Superficie de coste anisotrópica mostrada sobre el modelo de elevación. Valle alto del río Darro.

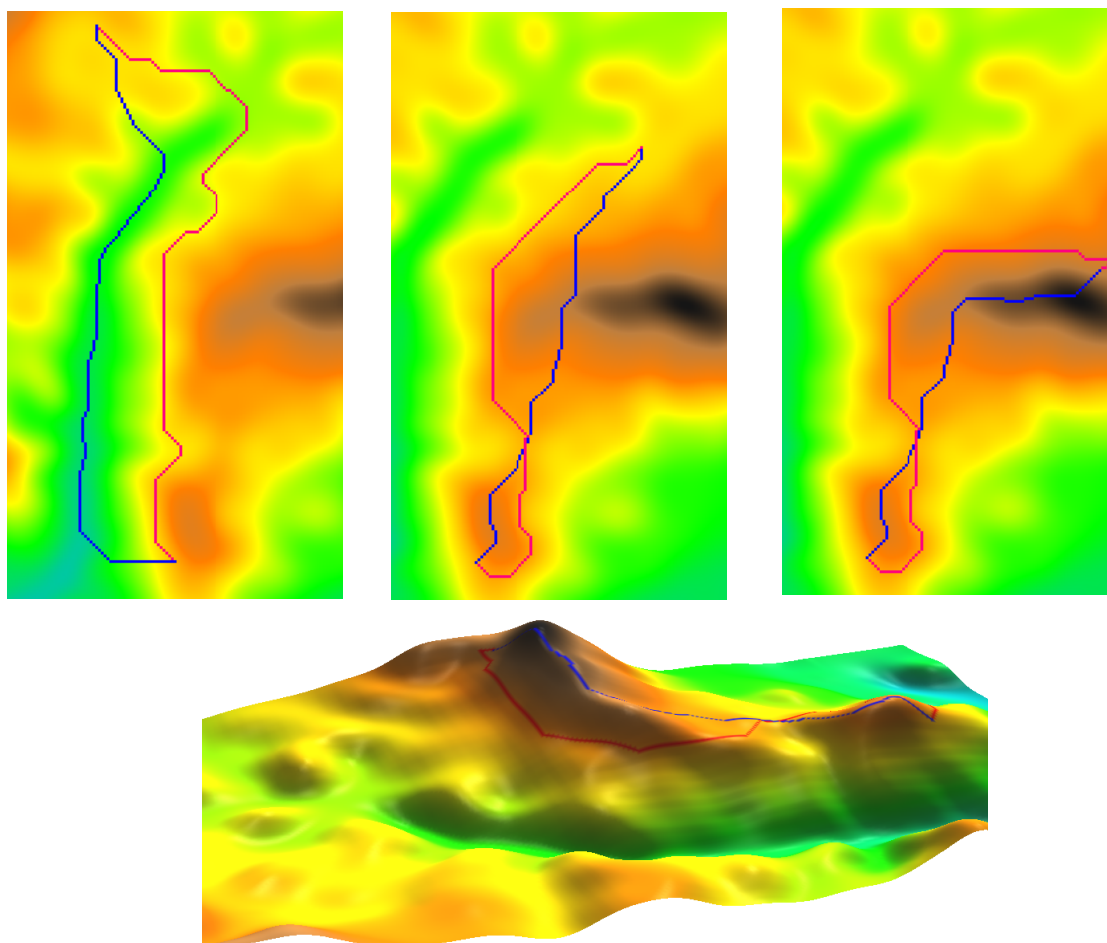


Figura 10: Comparación de caminos mínimos con superficies de coste isotrópicas (en azul) y anisotrópicas (en rojo). Valle alto del río Darro.

En estos ejemplos el coste calculado es un tiempo de desplazamiento. Con este mismo esquema se pueden calcular superficies en las que el coste represente otro tipo de magnitud, como coste económico de construcción de infraestructuras lineales (como carreteras, canales, oleoductos o tendidos eléctricos), tal como se ha comentado en la introducción.

Las superficies de coste, en sí mismas, pueden servir para analizar propiedades sobre el territorio. Típicamente se ha usado para determinar distritos o áreas de influencia. Este tipo de aplicaciones también pueden ampliarse utilizando superficies de coste anisotrópicas. La figura 11 muestra la distribución del coste de construcción de un canal de riego desde una toma fija como una superficie de coste anisotrópica. Aquí la fricción se ha calculado como una estimación del coste de construcción, dependiendo del desnivel y la distancia, haciendo que el desplazamiento siguiendo desniveles positivos sea inviable.

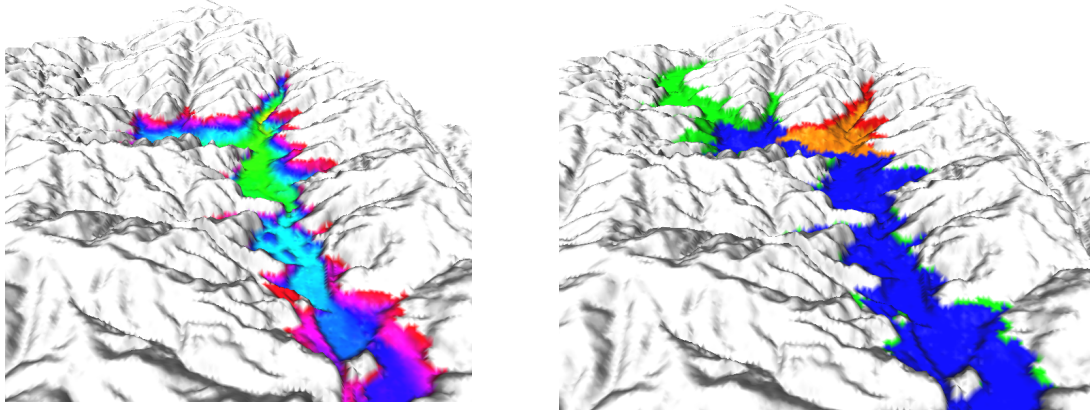


Figura 11: Coste de construcción de un canal de riego. Izquierda superficie de coste. Derecha clasificación de emplazamientos

Comparando la superficie de coste para distintos puntos de la toma podemos analizar la idoneidad de cada emplazamiento. La imagen derecha de la figura 12 muestra la clasificación del valle en función de la toma óptima. Las celdas verdes y rojas solo se irrigan desde una de las tomas. Las azules y naranjas se irrigan desde las dos, siendo en cada una de ellas más rentable irrigar desde una de las tomas.

También pueden utilizarse superficies de coste para resolver problemas que no están directamente relacionados con la resolución de problemas lineales. En general, cualquier problema que se pueda expresar como la minimización de una integral de línea:

$$\text{Coste} = \min \int_l f(x) \cdot dx \quad (8)$$

se puede resolver usando superficies de coste. Así, se pueden expresar problemas de difusión, en los que el coste calculado es el tiempo necesario para que una sustancia alcance un punto concreto.

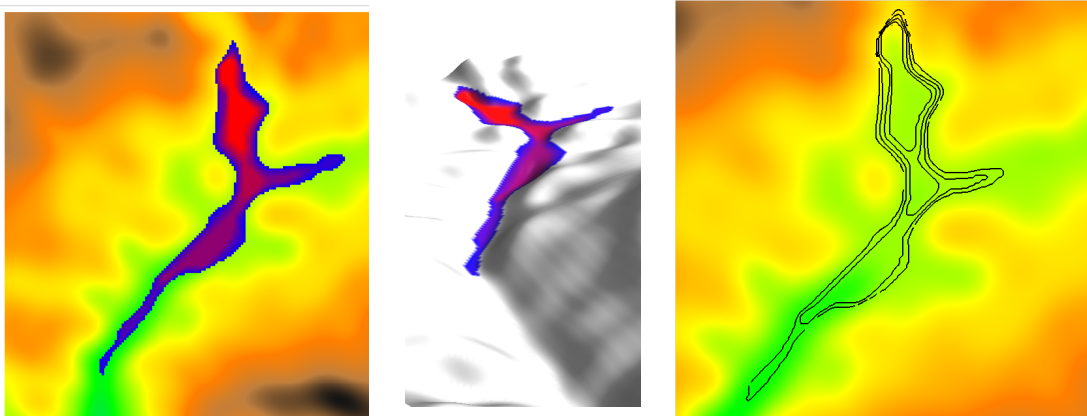


Figura 12: Simulación de un proceso de difusión.

La figura 12 muestra el resultado del proceso de simulación de un vertido en un punto del cauce del río Darro usando superficies de coste anisotrópicas. La función de fricción se ha calculado de forma que disminuye con la pendiente, cuando ésta es negativa y aumenta con ella cuando es positiva. Las curvas de nivel en la superficie de coste representan el tiempo de llegada del vertido.

El tiempo de cálculo de las superficies de coste anisotrópicas, al igual que con el caso isotrópico, crece de forma mayor que lineal (concretamente  $O(n \cdot \log(n))$ ). La tabla 1 muestra tiempos de cálculo con distintas resoluciones.

Tabla 1: Tiempo de cálculo

Resolución	Número de celdas	Tiempo de cálculo (en segundos)
207x275	56925	0,08
500x501	250500	130,45
451x601	271051	177,50

## CONCLUSIONES

Las superficies de coste constituyen una herramienta útil para resolver problemas de análisis espacial. Aunque se encuentra como una función habitual en sistemas GIS, su aplicabilidad está limitada. Por un lado las herramientas disponibles funcionan solo con problemas isotrópicos, o con problemas anisotrópicos con fuertes restricciones. Por otro lado, muchos de los problemas reales son anisotrópicos.

Se ha presentado el diseño e implementación para Grass de funciones para cálculo de superficies de coste anisotrópicas, y se han descrito varias aplicaciones de las mismas.

Como trabajos futuros nos planteamos extender la función para contemplar la conectividad a ocho vecinos, e incluir salida vectorial del módulo de cálculo de camino mínimo.

## AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por la Consejería de Economía Ciencia y Empresa de la Junta de Andalucía a través del proyecto PE-TIC5276.

## REFERENCIAS

1. DIJKSTRA, E. W. (1959), A Note on Two Problems in Connection with Graphs. *Numerische Math.* 1, 269-271
2. COOPER, J. (2010), Modelling mobility and exchange in pre-columbian cuba: gis led approaches to identifying pathways and reconstructing journeys from the archaeological record. *Journal of Caribbean Archaeology*.
3. S. EBENER, S.; EL MORJANI, Z.; RAY, N.; BLACK, M. (2005): Physical Accessibility to health care: From Isotropy to Anisotropy. *GIS Development*. <http://www.gisdevelopment.net/magazine/years/2005/jun/care2.htm>
4. MICHELS, E.; COTTENIE, K.; NEYS, L.; DEGELAS, K.; COPPIN, P.; DEMEESTER, L. (2001): Geographical and genetic distances among zooplankton populations in a set of interconnected ponds: a plea for using GIS modelling of the effective geographical distance. *Molecular Ecology* 10, 1929 –1938
5. JOBE, R.T.; WHITE, P. (2009), A new cost-distance model for human accessibility and an evaluation of accessibility bias in permanent vegetation plots in Great Smoky Mountains National Park, USA. *Journal of Vegetation Science* 20: 1099–1109.

6. BAGLI, S.; GENELETTI, D.; ORSI, F. (2010) Routeing of power lines through least-cost path analysis and multicriteria evaluation to minimise environmental impacts. *Environ Impact Asses Rev* (2010)
7. FONTANARI, S. (2001), Sviluppo di metodologie GIS per la determinazione dell'accessibilità territoriale come supporto alle decisioni nella gestione ambientale. University of Trento, Italy, Ingegneria per l'Ambiente e il Territorio.
8. FONTANARI, S.; NAPOLITANO, M.; ROBERTO, F. (2005), r.walk: GRASS GIS 6.2. [http://grass.itc.it/gdp/html\\_grass63/r.walk.html](http://grass.itc.it/gdp/html_grass63/r.walk.html)
9. COLLISCHONN, W.; PILAR, J.V. (2000), A directional dependent least cost path algorithm for roads and canal. *Int. Journal of Geographical Information Systems. Volume 14, Number 4, pp. 397-406*
10. CHAOQING YU, JAY LEE, MANDY J. MUNRO-STASIUK (2003), Extension to least-cost path algorithms for roadway planning. *Int. Journal of Geographical Information Systems. Vol. 17, N. 4, 361-376. 2003.*