

Triángulos Ortogonales en el plano ^{*}

M. A. Garrido, A. Márquez, J. R. Portillo y P. Reyes

Departamento de Matemática Aplicada I

Universidad de Sevilla

{vizuete,almar,josera,preyes}@cica.es

Resumen

Como extensión natural del problema *Single Bend Wiring*, estudiamos la cuestión de unir en el plano los vértices de triángulos dados, mediante aristas verticales y horizontales sin intersección entre ellas. El número de codos permitidos en los trazados ortogonales de cada triángulo determina la naturaleza computacional del problema. Probamos la equivalencia del problema permitiendo un codo con el de intersección de rectángulos, por lo que se obtiene tiempo óptimo de resolución $O(n \log n)$, suponiendo una mejora respecto al *Single Bend Wiring*. El problema para dos codos es una variante de los conocidos problemas de etiquetado, para el que presentamos un algoritmo incremental cuadrático.

Si el número de codos es mayor obtenemos la NP-completitud de una variante del problema, en la que se fijan los trazados permitidos para cada triángulo.

Palabras Clave: Trazados ortogonales. Intersección de rectángulos. Etiquetado.

1 Introducción

Es bastante frecuente, en aplicaciones de Electrónica-Infornática, encontrarse con la necesidad de acudir a trazados ortogonales para unir grupos de puntos, sin intersecciones entre ellos, como por ejemplo en el diseño de circuitos VLSI. Igualmente nos encontramos con el uso de tales trazados en la construcción de diagramas relacionales y en algunas aplicaciones de Software, como diagramas de flujo y diseños de bases de datos [1, 2, 3]. Este hecho ha llamado la atención de muchos investigadores, y a ello es debido la gran cantidad de resultados obtenidos sobre la materia [4, 5, 6, 7, 8, 9, 10].

El problema original, conocido con el nombre de *Single Bend Wiring*, fue estudiado por Raghavan *et al.* [9]. En general, se trata de conectar pares de puntos situados en el plano, llamados *wires*, mediante trazados ortogonales, es decir una arista uniendo un par de vértices se representa por una sucesión de segmentos verticales y horizontales. A cada cambio de dirección en las aristas se denominará *codo*. Así, el problema *Single Bend Wiring*

^{*}Parcialmente financiado por los grupos de investigación PB96-1374, incluido en el proyecto DGICYT y FQM-0164, adscrito al proyecto PAI

se formularía como la determinación de aristas ortogonales sobre la malla rectangular del plano, uniendo los pares de puntos y utilizando a lo más un codo en cada una de ellas, sin que se produzcan intersecciones. En este estudio ofrecen un algoritmo que nos permitirá determinar, en tiempo cuadrático, si un conjunto de pares de puntos en el plano pueden o no ser conectados de esta forma. Debido a que los caminos no pueden cruzarse, en la gran mayoría de los casos el problema no tiene solución. En este caso prueban, que encontrar el subconjunto de *wires* de mayor cardinalidad para el que existe solución es un problema de carácter NP-duro. Para ello relacionan el problema *Single Bend Wiring* con conocidos problemas de satisfacibilidad (véase [11]).

Naturalmente, aunque el problema original es importante, desde un punto de vista práctico, las relaciones no suelen ser tan simples que permitan modelizarlas mediante pares de puntos. Así, presentamos en este trabajo algunas extensiones naturales del problema original *Single Bend Wiring*, considerando tríos en lugar de pares de puntos. La especial característica de esta variante es que, en vez de relacionarlo con enunciados conocidos dentro del campo de la Lógica, hemos encontrado la solución aquí propuesta estableciendo la equivalencia con problemas de la Geometría Computacional, como son la Intersección de Rectángulos y Etiquetado.

Formalmente podríamos enunciar el problema general estudiado en este trabajo de la siguiente forma:

Triángulos Ortogonales con k codos $OT(k)$

ENTRADA: Un conjunto de triángulos $\{T_1, T_2, \dots, T_n\}$, donde cada T_i es un conjunto de tres puntos en el plano, y un número entero no negativo k .

PREGUNTA: ¿Pueden unirse simultáneamente todos los puntos de cada conjunto T_i , mediante una inmersión ortogonal de K_3 en la malla del plano, con, a lo sumo, k codos y sin intersecciones?

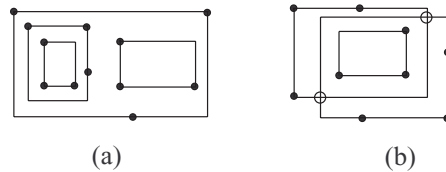


Figura 1: Dos entradas del problema $OT(2)$, (a) con solución, (b) sin solución

En primer lugar estudiaremos los problemas $OT(0)$ y $OT(1)$. El primero de ellos evidentemente no tiene solución y el segundo probamos, mediante su relación con el problema de Intersección de Rectángulos, que se puede resolver en tiempo óptimo $O(n \log n)$. Pasamos a continuación al estudio del caso de trazados con dos codos, $OT(2)$, y probamos, haciendo uso de una estructura de árbol, que es resoluble en tiempo $O(n^2)$. Por último estudiamos una variante del problema cuando se permite un número de codos superior a 2, obteniendo la naturaleza NP-completa del mismo. Concluiremos el trabajo con el planteamiento

de algunos problemas abiertos cuyo estudio posterior puede resultar de interés. A lo largo del trabajo, cada vez que nos refiramos a un rectángulo, entenderemos que sus lados son paralelos a los ejes coordenados.

2 Los problemas $OT(0)$ y $OT(1)$

Es evidente que resulta totalmente imposible unir los vértices de un triángulo, con trazados ortogonales en la malla del plano, si para ello no está permitido la utilización de ningún codo. Por lo tanto el problema $OT(0)$ no tiene solución en ningún caso.

Pasemos al estudio del problema $OT(1)$. Hemos de tener en cuenta el siguiente lema:

Lema 1 *Tres puntos del plano podrán unirse con aristas verticales y horizontales y a lo sumo un codo si, y sólo si, los tres puntos están situados en los vértices de un rectángulo.*

Si tenemos por tanto una entrada del problema $OT(1)$ estudiaremos en primer lugar si todos los triángulos se encuentran en las condiciones del anterior lema (este estudio previo puede realizarse en tiempo lineal). En caso negativo no será necesario continuar, ya que el problema no tendría solución, y en caso afirmativo el problema lo podremos trasladar al estudio de la intersección de los n rectángulos correspondientes, para el que existe un algoritmo que lo resuelve en tiempo óptimo $O(n \log n)$ (véase [12, 13]). Podemos por tanto enunciar el siguiente resultado:

Teorema 2 $OT(1)$ se resuelve en tiempo óptimo $O(n \log n)$.

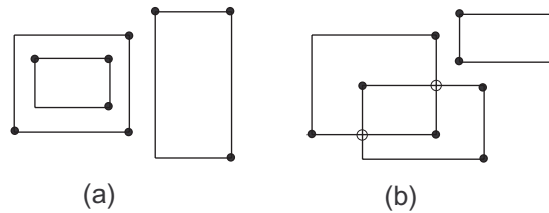


Figura 2: Dos entradas del problema $OT(1)$

La Figura 2 muestra dos entradas del problema $OT(1)$, la primera de ellas tiene solución, mientras que la segunda no la tiene ya que si bien se pueden trazar sin intersecciones dos de los tres triángulos, no pueden trazarse simultáneamente los tres.

Podemos resaltar que la solución para tríos de puntos, que se obtiene a través de un conocido algoritmo de la Geometría Computacional, es de mejor comportamiento que el correspondiente para pares de puntos, *Single Bend Wiring*, obtenido a través de problemas de satisfacibilidad por Raghavan et al. [9], ya que la complejidad de este último (del cual no está probada su optimalidad) era de $O(n^2)$.

3 El problema $OT(2)$

Consideremos ahora el problema en el caso de que podamos utilizar, a lo sumo, 2 codos. Cada uno de los triángulos de la entrada del problema puede encontrarse en alguna de las situaciones siguientes:

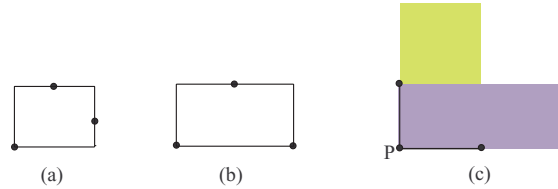


Figura 3: Posibilidades de $OT(2)$

Lema 3 *Para poder conectar los tres vértices de un triángulo, utilizando trazados ortogonales en la retícula del plano con, a lo sumo, 2 codos, será necesario que al menos dos de los tres vértices estén situados en la misma vertical u horizontal*

Si no ocurre ésto (Figura 3(a)), se necesitarían al menos 3 codos para poder conectarlos.

Si dos de los vértices están por ejemplo en la misma horizontal, puede ocurrir que el tercer vértice no se encuentre en la misma vertical que ninguno de ellos (Figura 3(b)), siendo en este caso obligatorio trazar el rectángulo.

En cambio si dos de los vértices están en la misma horizontal y dos de los vértices en la misma vertical (Figura 3(c)), para poder unirlos con, a lo sumo 2 codos, existen infinitas posibilidades, ya que podríamos trazar cualquiera de los rectángulos que teniendo un vértice en el punto P está contenido en alguna de las franjas (horizontal o vertical) infinitas que se aprecian en el gráfico.

Obsérvese, por tanto, el parecido de este problema con el problema de etiquetado [14], consistente en colocar sin solapamientos, etiquetas informativas sobre mapas. Está probado [15] que este problema, en forma general, es NP-duro. Es más, aun en algunos casos particulares, poniendo limitaciones al mismo, el problema sigue siendo NP-duro [16]. A pesar de ello, el problema que tratamos sí puede ser resuelto en tiempo polinomial como demuestra el siguiente:

Teorema 4 $OT(2)$ se resuelve en tiempo $O(n^2)$.

Demostración:

En primer lugar, a partir de la posición de los vértices, construimos (si es posible) los rectángulos obligatorios (tipo (b)) o extensibles (tipo (c)). Si existe alguna tripleta de puntos del tipo (a), el problema no tiene solución.

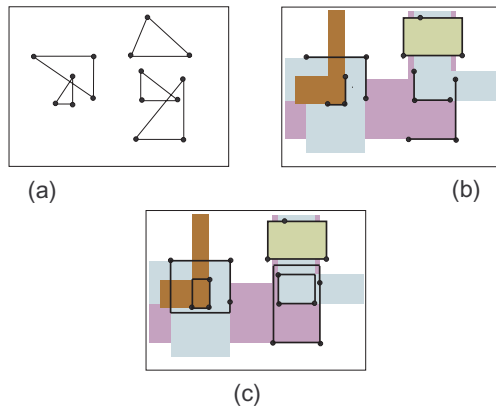


Figura 4: (a) Una entrada de $OT(2)$. (b) Bandas que define. (c) Una solución.

Con el objeto de codificar la extensión a lo largo de las bandas infinitas mostrada en la Figura 3(c), asociamos a cada rectángulo un vector con las coordenadas de sus cuatro aristas (Superior $\equiv N$, Inferior $\equiv S$, Izquierda $\equiv W$ y Derecha $\equiv E$), las cuales llevan asociados banderas que indican la posibilidad de ampliación del rectángulo a través de las susodichas franjas. Así, el triángulo dibujado en la Figura 3(b) tiene las banderas de sus cuatro aristas en 0, pues sus vértices no pueden ser modificados. En cambio, el triángulo mostrado en la Figura 3(c) tiene habilitadas a 1 (posibilidad de cambio) las banderas correspondientes a las aristas N y E , pues, si fuera necesario, podemos prolongar los trazados ortogonales a lo largo de una de esas bandas. Por supuesto, las banderas correspondientes a las aristas W y S serán 0. El proceso de creación de estas estructuras de datos *vectores con banderas* se hace en tiempo lineal en el número de triángulos ortogonales n .

A continuación, usamos el conocido algoritmo de Edelsbrunner [12] para estudiar la intersección de los rectángulos. Este algoritmo tiene un coste $O(n \log n + s)$, siendo s el número de intersecciones.

Nótese que la intersección de dos de los rectángulos determinado por los triángulos ortogonales es de uno de los siguientes tipos :

1. No hay intersección (Figura 5(a)).
2. Hay intersección entre aristas con bandera 0 (aristas “fijas”): El problema no tiene solución (Figura 5(b)).
3. Hay intersección de “esquina”, es decir, dos aristas de un triángulo cortan respectivamente a dos aristas de otro. El problema no tiene solución (Figura 5(c)).
4. Una arista con bandera 1 corta a dos aristas de otro triángulo (Figura 5(d₁)). Para evitar la intersección, el primer triángulo debe “engullir” al segundo, extendiéndose a lo largo de la banda correspondiente a la arista marcada con bandera 1. Todas las

demás banderas del primer triángulo se marcarán con ceros, si no lo estuvieran ya, puesto que ya sólo puede prolongarse el rectángulo en la dirección en que lo ha hecho (Figura 5(d₂)).

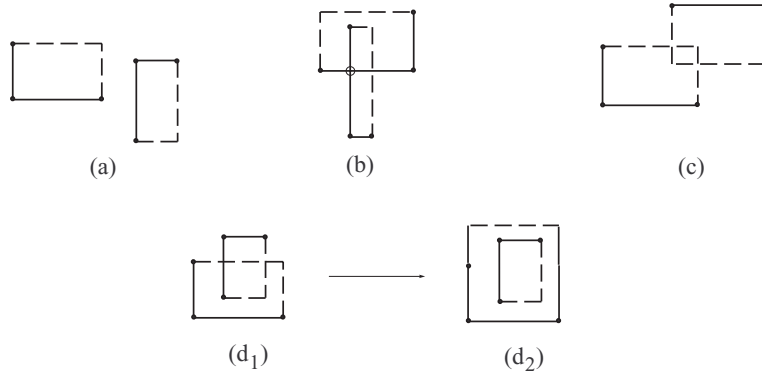


Figura 5: Intersecciones posibles de triángulos ortogonales

Los casos 2 y 3 implican que el problema $OT(2)$ no tiene solución. Los casos 1 y 4 serán estudiados dinámicamente.

Construimos ahora en tiempo $O(n \log n)$ las siguientes estructuras de datos a partir de las anteriores: cuatro listas, una por cada dirección (N, E, S, W), que nos muestran el orden en que se encuentran las aristas correspondientes de cada rectángulo. Usaremos estas listas para estudiar cuál es el rectángulo que puede intersectar al que estamos incluyendo en cada momento. Para ello, definiremos para cada arista de cada rectángulo punteros a las posiciones que ocupan, en las dos listas correspondientes a su orientación (N, S ó E, W). El coste total de esta operación es de nuevo $O(n \log n)$.

Una vez construidas estas estructuras de datos, en un tiempo total del orden de $O(n \log n)$, estamos en condiciones de usar un algoritmo incremental para resolver (si es posible) el problema. El algoritmo que encuentra los trazados ortogonales, si existen, es el siguiente:

Suponemos que hemos conseguido encontrar los trazados para m triángulos ortogonales y vamos a incluir el triángulo ortogonal $m + 1$. Los m rectángulos ya trazados son disjuntos o incluidos unos en otros. La relación de inclusión ha sido reflejada en una estructura de datos de tipo *bosque*, dónde cada árbol representa un conjunto de rectángulos incluidos unos en otros, siendo la raíz de cada árbol el rectángulo exterior. Al añadir el rectángulo exterior $m + 1$, éste es cortado a lo máximo por dos aristas con bandera 1 de rectángulos exteriores a conjuntos de rectángulos incluidos (árboles), puesto que en caso contrario, el problema no tiene solución (Figura 6(a)). El estudio de las intersecciones se hace en tiempo constante, puesto que se estudia a partir de las listas de direcciones N, S y E, W .

Si el nuevo rectángulo añadido no intersecta ningún árbol, creamos un nuevo árbol cuyo única hoja es precisamente ese rectángulo.

Si interseca a un árbol únicamente, hay tres posibilidades:

1. La intersección implica irresolubilidad del problema, bien por corte de aristas con bandera 0, bien por ser intersección *de esquinas*. En este caso, el problema no tiene solución.
2. Una arista con bandera 1 del nuevo rectángulo interseca dos aristas del rectángulo exterior del árbol. En este caso prolongamos el rectángulo nuevo en la dirección de la arista señalada hasta *engullir* al árbol, actualizando si es preciso las banderas (marcando a 0 la otra posible bandera que esté a 1), actualizando las listas de orden de las direcciones N, S ó E, W y actualizamos también la estructura dinámica del bosque, situando al nuevo rectángulo como raíz del árbol.
3. Una arista con bandera 1 del rectángulo exterior del árbol interseca dos aristas del nuevo rectángulo. En este caso, el rectángulo exterior *engulle* al nuevo por prolongación en la dirección de su arista señalada. Se actualizan las estructuras y se repite el proceso con los rectángulos del interior del árbol.

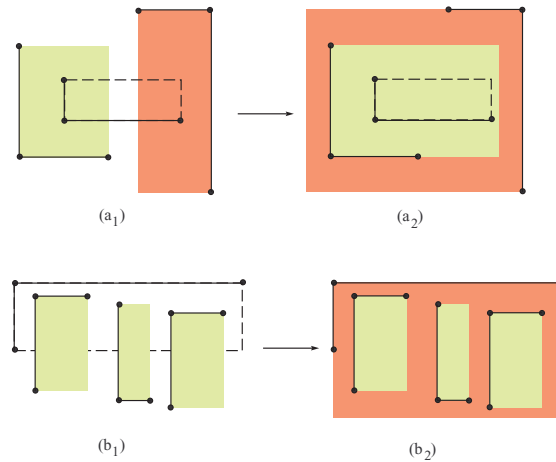


Figura 6: Inserción de un nuevo rectángulo

En el caso de que el rectángulo nuevo sea intersectado por aristas de bandera 1 de dos árboles, sólo hay una forma de que ésto ocurra (Figura 6(a₁)). En este caso, o uno de los rectángulos exteriores de los árboles engulle a ambos rectángulos (nuevo y el otro exterior) uniendo ambos árboles en uno sólo (Figura 6(a₂)) o bien el problema no tiene solución. Si hay prolongación, se repite el estudio dentro del nuevo árbol.

Finalmente, si corta a más de dos rectángulos exteriores, es el triángulo nuevo el que debe ser ampliado (Figura 6(b)).

El número de operaciones a realizar es de orden cuadrático, puesto que cada pareja de rectángulos se compara como máximo dos veces. Efectivamente, una vez que un rectángulo crece hasta abarcar a otro, no vuelven a compararse. Y todas las actualizaciones de banderas y listas se hacen en tiempo constante para cada pareja de rectángulos.

Por lo tanto, el algoritmo resuelve el problema $\text{OT}(2)$ en tiempo cuadrático.

□

4 Un mayor número de codos

En esta sección trabajamos con trazados ortogonales de triángulos en los que se permiten tres o más codos. Más concretamente, en lugar de considerar el problema general $\text{OT}(k)$, nos centramos en una variante en la que se fijan las inmersiones permitidas para cada triángulo. El nuevo problema queda así definido:

Triángulos Ortogonales con k codos y l inmersiones $l\text{-OT}(k)$

ENTRADA: Un conjunto de triángulos $\{T_1, T_2, \dots, T_n\}$, donde cada T_i es un conjunto de tres puntos en el plano, un número entero no negativo k y para cada triángulo T_i , a lo sumo, l inmersiones de K_3 en la cuadrícula del plano uniendo los puntos de T_i con k codos como máximo.

PREGUNTA: ¿Pueden unirse simultáneamente todos los puntos de cada conjunto T_i , usando las inmersiones ortogonales dadas para cada uno de ellos y sin intersecciones entre las aristas?

Podemos establecer el siguiente resultado:

Teorema 5 $3\text{-OT}(3)$ es un problema NP-completo.

Además, teniendo en cuenta que cualquier entrada del problema $3\text{-OT}(3)$ será entrada del problema $l\text{-OT}(k)$ ($l, k \geq 3$), obtenemos:

Corolario 6 $l\text{-OT}(k)$ es un problema NP-completo si, y sólo si, $l, k \geq 3$.

Puesto que la naturaleza NP-completa del problema hace que su demostración no se encuadre dentro de las áreas tratadas en el encuentro, no incluimos la prueba de dicho resultado, la cual se basa en una transformación del conocido problema de satisfacibilidad PLANAR 3-SAT.

5 Conclusiones y problemas abiertos

Podríamos presentar a título de conclusión, el siguiente cuadro de resultados y métodos empleados:

$\text{OT}(0)$	No tiene solución	
$\text{OT}(1)$	$O(n \log n)$	Intersección rectángulos
$\text{OT}(2)$	$O(n^2)$	Etiquetado
$l\text{-OT}(k)$	NP-completo $\Leftrightarrow l, k \geq 3$	PLANAR 3-SAT

Algunas líneas de investigación que pueden surgir del presente trabajo son las siguientes:

1) El problema $OT(2)$ se ha probado que podemos resolverlo en tiempo $O(n^2)$, pero no hemos conseguido obtener la optimalidad del algoritmo presentado. En cualquier caso, puesto que problemas muy similares de etiquetado han sido obtenidos con la misma complejidad, pensamos que el resultado es óptimo.

2) El problema $l-OT(3)$ es de naturaleza NP-completa, pero pensamos que podrían conseguirse aproximaciones a la solución del problema, en tiempo polinomial, introduciendo algunas restricciones sobre la entrada del problema.

3) En la actualidad estudiamos generalizar el problema en el sentido de considerar otras estructuras iniciales de los puntos a conectar; es decir ¿qué ocurrirá cuando en lugar de triángulos en el plano, tratemos de unir otro tipo de grafos? Podríamos en este sentido enunciar el siguiente problema general:

ENTRADA: Un conjunto de grafos conexos $\{G_1, G_2, \dots, G_n\}$, un conjunto de números enteros no negativos $\{k_1, k_2, \dots, k_n\}$ y para cada grafo G_i , un conjunto W_i de puntos del plano, tal que $|G_i| = |W_i|$.

PREGUNTA: ¿Pueden unirse simultáneamente todos los puntos de cada conjunto W_i , usando una inmersión ortogonal de G_i en la malla del plano con, a lo sumo, k_i codos y sin intersecciones?

Referencias

- [1] C. Batini, E. Nardelli y R. Tamassia. A Layout Algorithm for Data-Flow Diagrams. *IEEE Trans. on Software Engineering*, **SE-12.4** (1986), 538-546.
- [2] C. Batini, M. Talamo y R. Tamassia. Computer Aided Layout of Entity-Relationship Diagrams. *The Journal of Systems and Software*, **4** (1984), 163-173.
- [3] Th. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. Teubner/Wiley & Sons, (1990).
- [4] T. C. Biedl. Optimal orthogonal drawings of triconnected plane graphs. *Proc. of SWAT'96, LNCS 1097 Springer-Verlag* (1996), 333-344.
- [5] M. A. Garrido y A. Márquez. Embedding a graph in the grid of a surface with the minimum number of bends is NP-hard. In G. DiBattista, editor, *Graph Drawing (Proc. of GD'97)*, LNCS **1353** Springer-Verlag (1998), 124-133.
- [6] M. A. Garrido, A. Márquez, A. Morgana y J.R. Portillo. Single bend wiring on surfaces. *submitted* (1998), 12 pages.
- [7] Y. Liu, A. Morgana y B. Simeone. General theoretical results on rectilinear embeddability of graphs. *Acta Math. Appl. Sinica*, **7** (1991), 187-192.
- [8] Y. Liu, A. Morgana y B. Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Mathematics*, **8** (1998), 69-91.

- [9] R. Raghavan, J. Cohoon y S. Sahni. Single Bend Wiring. *Journal of Algorithms*, **7** (1986), 232–257.
- [10] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* **16 (3)** (1987) 421–444.
- [11] M. R. Garey y D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. Freeman, (1979).
- [12] H. Edelsbrunner. Dynamic data structures for orthogonal intersection queries. Rep. **F59**, Tech. Univ. Graz. Institute für Informationsverarbeitung, (1980).
- [13] E. M. MacCreight. Priority search trees. Tech. Rep. **CSL-81-5**, Xerox PARC, (1981).
- [14] E. Imhof. Positioning names on maps. *The American Cartographer*. (1975) 2,128-144.
- [15] M. Formann y F. Wagner. A packing problem with applications in lettering of maps. *Proceedings of the 7th ACM Symposium on Computational Geometry*. (1991) 281-288.
- [16] C. Iturriaga y A. Lubiw. Elastic Labels: The two-axis case. In G. DiBattista, editor, *Graph Drawing (Proc. of GD'97)*, LNCS **1353** Springer-Verlag (1998), 181–192.