

# Separación y clasificación de puntos rojos y azules\*

Delia Garijo<sup>†</sup>    Alberto Márquez<sup>‡</sup>    Joseph S. B. Mitchell<sup>§</sup>    Carlos Seara<sup>¶</sup>

## Resumen

En este trabajo estudiamos el problema de determinar si dos conjuntos disjuntos de  $n$  puntos en el plano son separables mediante una estructura de 2-level tree, compuesta por una recta y dos semirrectas, y diseñamos algoritmos óptimos de tiempo  $\Theta(n \log n)$  para resolver este problema.

## 1 Introducción

Sean  $R$  y  $B$  dos conjuntos disjuntos de puntos en el plano, clasificados como puntos rojos y puntos azules, respectivamente. Suponemos que todos los puntos están en posición general (no existen tres puntos alineados).

El criterio más simple de separabilidad aplicable a los conjuntos  $R$  y  $B$  es la separabilidad lineal. Dichos conjuntos se dice que son *linealmente separables* si existe una recta que los separa. La separabilidad lineal puede decidirse en tiempo  $O(n)$ , calculando además una recta separadora [5]. Incluso la mejor recta que maximiza la anchura de una banda que separa  $R$  y  $B$ , en caso de ser linealmente separables, puede calcularse en tiempo  $O(n)$  [6].

Si los conjuntos no son separables por una recta, existen diversos criterios para separar  $R$  y  $B$  utilizando dos rectas: separabilidad por banda (dos rectas paralelas), por cuña (dos semirrectas), o por doble cuña (dos rectas que intersecan). Todos estos criterios pueden decidirse y calcularse en tiempo óptimo  $\Theta(n \log n)$  [1, 4]. Otro criterio de separabilidad utilizado es el descrito por una poligonal convexa. Edelsbrunner y Preparata [2] resuelven el problema de calcular la mínima poligonal convexa (abierta o cerrada) que separa  $R$  y  $B$  en tiempo  $O(n \log n)$ , y en [1] se demuestra que este tiempo es óptimo. El problema, conocido como *Min-Link Red-Blue Separation Problem*, de determinar la mínima poligonal separadora de  $R$  y  $B$  (aquella con el menor número de aristas) es NP-completo [3].

En este trabajo estudiamos un criterio de separabilidad entre los conjuntos  $R$  y  $B$ , la *separabilidad mediante un 2-level tree*, que engloba como casos particulares algunos de los criterios ya mencionados: (1) separabilidad lineal, (2) separabilidad por banda, cuña y doble cuña, (3) separabilidad por una 3-poligonal convexa, (4) separabilidad por una 3-poligonal no convexa formada por dos semirrectas y un segmento, estructura conocida como zig-zag, y (5) separabilidad por tres rectas paralelas, que puede ser vista como un caso degenerado de la separación mediante un zig-zag.

En general un  $k$ -level tree es un árbol con raíz de profundidad  $k$ , con  $k + 1$  niveles. La raíz (nivel 0) es la única recta de separación de la estructura, cada nodo interno corresponde a una semirrecta de separación, y las hojas son subconjuntos monocromáticos de  $R$  y  $B$  (ver Figura 1). Dados los conjuntos  $R$  y  $B$ , un  $k$ -level tree divide al plano en regiones monocromáticas.

Aparte de su interés como estructura, la principal motivación para el estudio de la separabilidad mediante un  $k$ -level tree es obtener un sistema de clasificación rápido de puntos rojos y azules por

---

\*Parcialmente subvencionado por los proyectos O.R.I MTM2005-08441-C02-01, PAI FQM-0164, MTM2006-01267 y DURSI 2005SGR00692.

<sup>†</sup>Departamento de Matemática Aplicada I, Universidad de Sevilla, [dgarijo@us.es](mailto:dgarijo@us.es).

<sup>‡</sup>Departamento de Matemática Aplicada I, Universidad de Sevilla, [almar@us.es](mailto:almar@us.es).

<sup>§</sup>Applied Mathematics and Statistics, State University of New York, Stony Brook, NY, 11794-3600, USA, [jshb@ams.sunysb.edu](mailto:jshb@ams.sunysb.edu).

<sup>¶</sup>Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, [carlos.seara@upc.edu](mailto:carlos.seara@upc.edu).

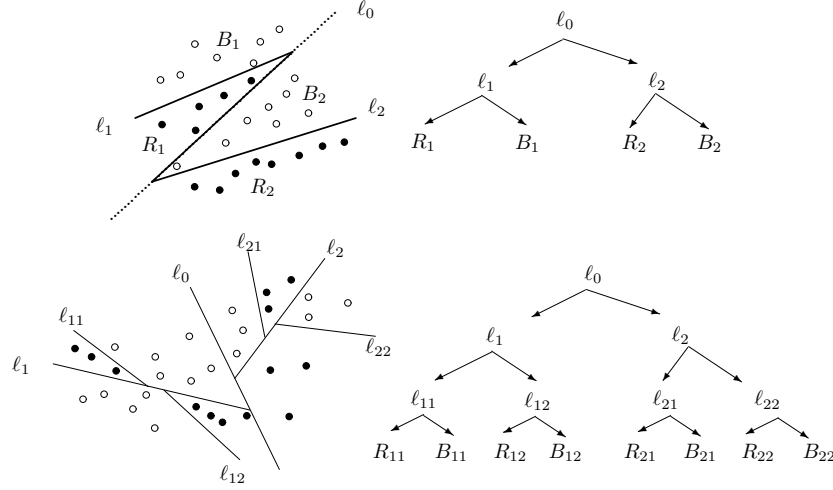


Figura 1: Ejemplos de 2-level tree y de 3-level tree.

medio de  $2^k - 1$  rectas o semirrectas. Así, cuando llega un nuevo punto podemos decidir en tiempo  $k$  donde clasificarlo. En el caso del 2-level tree, suponemos que los conjuntos  $R$  y  $B$  no son separables por una  $t$ -poligonal convexa,  $t \leq 3$ , ya que como hemos comentado, este criterio se puede decidir y calcular en tiempo  $O(n \log n)$ . Esto implica que podemos considerar que las envolventes convexas de  $R$  y  $B$ , denotadas por  $CH(R)$  y  $CH(B)$  respectivamente, no son monocromáticas.

El resultado principal de este trabajo es la resolución en tiempo óptimo  $\Theta(n \log n)$  del problema de decidir y calcular si dos conjuntos  $R$  y  $B$  de  $n$  puntos del plano pueden separarse y clasificarse mediante una estructura de 2-level tree. Para ello, diseñamos algoritmos óptimos para cada una de las situaciones posibles. Estas situaciones se obtienen mediante un análisis exhaustivo del problema, atendiendo a las posiciones de las semirrectas y de la recta que componen el 2-level tree, y también a las alternancias de colores de las cuatro regiones monocromáticas en las que el 2-level tree divide al plano. La primera distinción que realizamos, una vez excluido el criterio de separabilidad por  $t$ -poligonal convexa con  $t \leq 3$ , es la siguiente:

1. Los conjuntos  $R$  y  $B$  que son separables por un zig-zag y en consecuencia por un 2-level tree.
2. Los conjuntos  $R$  y  $B$  que no son separables por zig-zag pero sí lo son por un 2-level tree.

Estos dos casos dan lugar a las Secciones 2 y 3, respectivamente.

## 2 Separabilidad por zig-zag

En esta sección estudiamos la separabilidad de  $R$  y  $B$  mediante una 3-poligonal no convexa, formada por dos semirrectas orientadas  $\ell_1$ ,  $\ell_2$  y un segmento  $s$  que une los orígenes de las semirrectas, es decir, un zig-zag. Demostramos que se puede decidir la existencia de la separabilidad por zig-zag y calcular una solución en tiempo óptimo  $\Theta(n \log n)$ . Como solución óptima podemos considerar:

1. La 3-poligonal más *convexa* posible, es decir, la que se aproxima más a la separación lineal.
2. La 3-poligonal que tiene los dos ángulos más pequeños, es decir, la que se aproxima más a la separación por 3 rectas paralelas.

Sea  $l_s$  la recta que contiene al segmento  $s$ . Hay 3 tipos posibles de zig-zag según sean los ángulos  $\alpha$  y  $\beta$  formados por  $\ell_1$  y  $s$ , y por  $\ell_2$  y  $s$ , respectivamente. La Figura 2 muestra los tres tipos de zig-zag. En esta figura también se ilustra la partición del plano generada por  $l_s$ ,  $l_1$  y  $l_2$  en cuatro

cuñas monocromáticas,  $R_1$ ,  $R_2$ ,  $B_1$ , y  $B_2$ , que clasifican los puntos de  $R$  y  $B$  según la cuña a la que pertenezcan. Nótese que estos cuatro conjuntos son no vacíos y que los ángulos  $\alpha$  y  $\beta$  del zig-zag  $(\ell_1, s, \ell_2)$  pueden variar produciendo particiones  $R_1$ ,  $R_2$ ,  $B_1$ , y  $B_2$  diferentes.

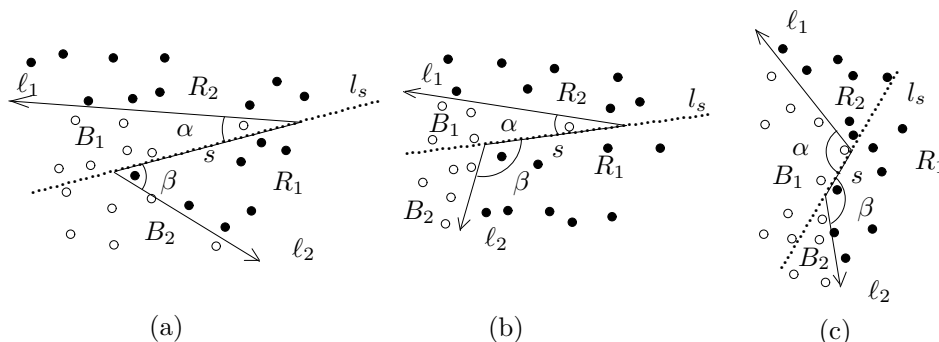


Figura 2: Tipos de zig-zag: (a)  $0 < \alpha < \pi/2$ ,  $0 < \beta < \pi/2$ , (b)  $0 < \alpha < \pi/2$ ,  $\pi/2 \leq \beta < \pi$ , (c)  $\pi/2 \leq \alpha, \beta < \pi$ .

**Lema 2.1.** Si existe un zig-zag  $(\ell_1, s, \ell_2)$  que separa  $R$  y  $B$ , entonces se pueden girar las dos semirrectas  $\ell_1$ ,  $\ell_2$  y el segmento  $s$  hasta que cada uno pase por dos puntos de diferente color.

Dado un zig-zag separador  $(\ell_1, s, \ell_2)$  de  $R$  y  $B$  denotamos por  $\ell'_i$ ,  $i = 1, 2$ , la recta orientada que contiene a la semirrecta orientada  $\ell_i$ .

**Lema 2.2.** Todo zig-zag separador  $(\ell_1, s, \ell_2)$  de  $R$  y  $B$  verifica al menos uno de los siguientes apartados:

1.  $CH(R_2)$  es linealmente separable de  $CH(B)$  por  $\ell'_1$  y además,  $\ell'_1$  es recta de soporte común entre  $CH(R_2)$  y  $CH(B)$ .
2.  $CH(B_2)$  es linealmente separable de  $CH(R)$  por  $\ell'_2$  y además,  $\ell'_2$  es recta de soporte común entre  $CH(B_2)$  y  $CH(R)$ .

**Lema 2.3.** Sea  $I_{B,R}$  el número de intersecciones entre pares de aristas de las envolventes convexas  $CH(B)$  y  $CH(R)$ . Si existe una separación por zig-zag de  $R$  y  $B$ , entonces  $I_{B,R} \in \{0, 2, 4, 6\}$ .

Las Figuras 3 y 4 muestran que todos estos casos de intersecciones son factibles.

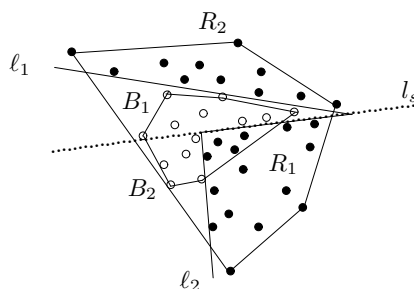


Figura 3: Separabilidad por zig-zag con  $I_{B,R} = 0$ .

Mediante el número de intersecciones  $I_{B,R}$  determinamos una nueva partición de los conjuntos de puntos rojos y azules. Denotamos por  $R_2$ ,  $R'_1$ , y  $R'_2$  ( $B_2$ ,  $B'_1$  y  $B'_2$ , resp.) a los, a lo más, tres conjuntos disjuntos de puntos rojos (azules, resp.) que son exteriores a  $CH(B)$  ( $CH(R)$ , resp.). Además, por  $R_I$  y  $B_I$  denotamos los conjuntos de puntos rojos y azules interiores a  $CH(B)$  y  $CH(R)$ , respectivamente. La Figura 5 muestra un ejemplo de estos conjuntos.

**Lema 2.4.** Sea  $(\ell_1, s, \ell_2)$  el zig-zag más convexo que separa los conjuntos  $R$  y  $B$ . Entonces, se verifica que:

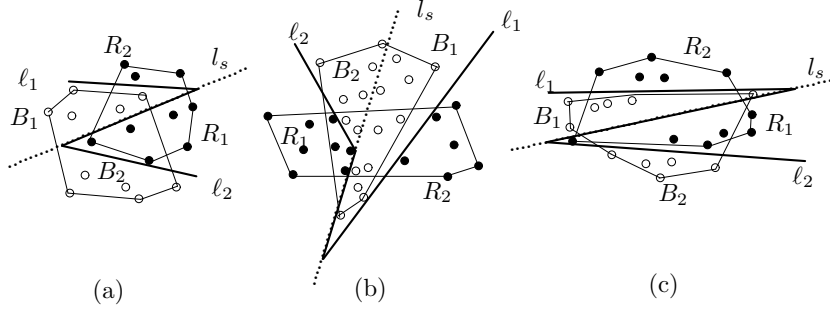


Figura 4: (a)  $I_{B,R} = 2$ , (b)  $I_{B,R} = 4$ , (c)  $I_{B,R} = 6$ .

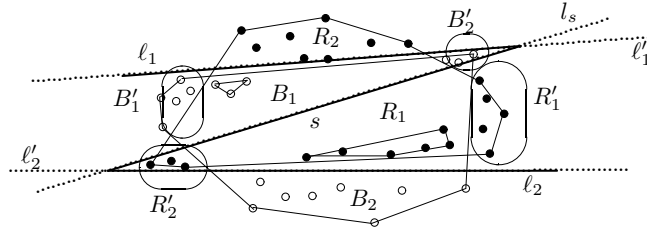


Figura 5: Partición determinada por un zig-zag separador e  $I_{B,R} = 6$ .

1.  $R_I$  es separable de  $B$  por la cuña  $(\ell_1, s)$  o  $B_I$  es separable de  $R$  por la cuña  $(\ell_2, s)$ .
2. La recta  $l_s$  es recta de soporte de alguno de los 6 polígonos convexos siguientes:  $CH(R_I)$ ,  $CH(B_I)$ ,  $CH(R'_1)$ ,  $CH(R'_2)$ ,  $CH(B'_1)$ , y  $CH(B'_2)$ .

Los resultados anteriores, y principalmente el Lema 2.4, permiten diseñar el siguiente algoritmo de tiempo  $O(n \log n)$  que decide la separabilidad por zig-zag de  $R$  y  $B$ , y calcula un zig-zag separador.

**Procedure:** SEPARABILIDAD-ZIG-ZAG

**Input:**  $R, B$ .

**Output:**  $(\ell_1, s, \ell_2)$

1. Calcular  $CH(R)$ ,  $CH(B)$ ,  $CH(R_I)$ ,  $CH(B_I)$  y el valor  $I_{B,R}$ . Si  $I_{B,R} \notin \{0, 2, 4, 6\}$  entonces no existe un zig-zag separador. En caso contrario, determinar el número de aristas de  $CH(R)$  y  $CH(B)$  que intersecan.
2. Si  $|R_I| = |\{r_1\}| = 1$  y  $|B_I| = |\{b_1\}| = 1$ , procedemos de la siguiente manera. Si  $r_1 \in CH(R)$  y  $b_1 \in CH(B)$ , entonces  $R$  y  $B$  son separables por zig-zag como muestra la Figura 6(a). Si  $r_1 \in CH(R)$  y  $b_1$  es interior de  $CH(B)$  o viceversa se procede de forma análoga (ver Figura 6(b)). A partir de ahora podemos suponer que  $|R_I| \geq 2$  ó  $|B_I| \geq 2$ .
3. Sea  $P$  un convexo del conjunto  $\{CH(R_I), CH(B_I), CH(B'_1), CH(B'_2), CH(R'_1), CH(R'_2)\}$  junto con sus puntos interiores. Para cada convexo  $P$ , ordenar los puntos de  $(R \cup B) - P$  de acuerdo a un barrido rotacional sobre  $P$  con una recta de soporte orientada  $l_s$ .
4. Cada vez que un punto rojo o azul pasa de un lado a otro de la recta  $l_s$ , mantener y actualizar las envolventes convexas  $CH(R_2)$ ,  $CH(B_1)$  ( $CH(R_1)$ ,  $CH(B_2)$ , resp.) de los puntos rojos y azules que están a la izquierda (derecha, resp.) de  $l_s$  al hacer el barrido rotacional.
5. En cada parada del barrido sobre  $P$ , comprobar si las envolventes convexas  $CH(R_2)$  y  $CH(B_1)$  ( $CH(R_1)$  y  $CH(B_2)$ , resp.) son linealmente separables mediante el cálculo de las rectas de soporte interiores entre ellas (ver Figura 7). En caso afirmativo, existe un zig-zag separador de  $R$  y  $B$ , y el proceso descrito nos permite determinar dicho zig-zag.

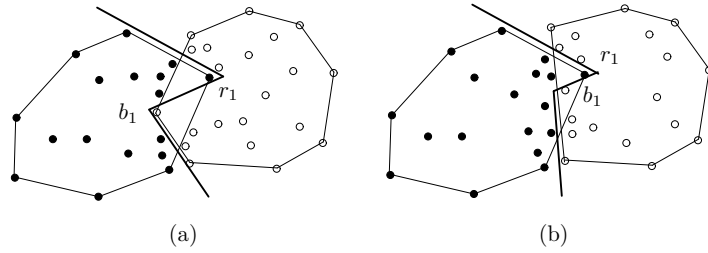


Figura 6: Separabilidad por zig-zag con  $|R_I| = |\{r_1\}| = 1$  y  $|B_I| = |\{b_1\}| = 1$ .

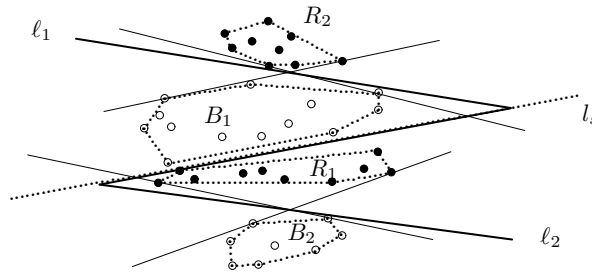


Figura 7: Rectas de soporte interiores entre envolventes convexas.

Nótese que en caso de existir más de un zig-zag separador, el algoritmo los encuentra todos ya que hace un análisis exhaustivo. Por tanto tenemos el siguiente resultado.

**Teorema 2.5.** Sean  $R$  y  $B$  dos conjuntos disjuntos de  $n$  puntos en el plano. Se puede decidir la separabilidad por zig-zag de  $R$  y  $B$ , y determinar un zig-zag separador en tiempo  $O(n \log n)$ .

A continuación mostramos una cota inferior  $\Omega(n \log n)$  para el problema de decisión de la separación por zig-zag. La Figura 8 ilustra la construcción que se usa para la demostración de dicha cota inferior. Como consecuencia obtenemos el siguiente resultado.

**Teorema 2.6.** Sean  $R$  y  $B$  dos conjuntos disjuntos de  $n$  puntos en el plano. Decidir si dichos conjuntos son separables por zig-zag y determinar un zig-zag separador puede hacerse óptimamente en tiempo  $\Theta(n \log n)$ .

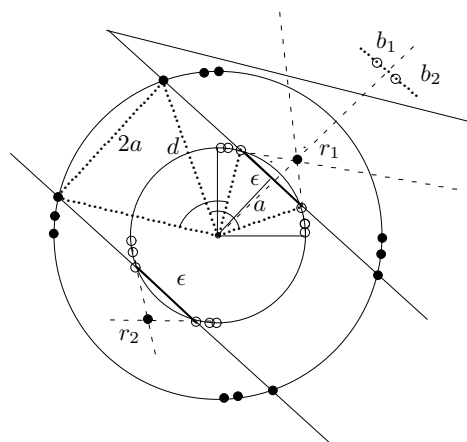


Figura 8: Construcción usada en la demostración de la cota inferior para la separabilidad por zig-zag.

### 3 Separabilidad por un 2-level tree

En esta sección enunciamos los resultados principales que nos permiten resolver en tiempo  $\Theta(n \log n)$  el problema de determinar la existencia de un 2-level tree separador de los conjuntos  $R$  y  $B$ , y calcular una solución.

Sean  $\ell_0$ ,  $\ell_1$ , y  $\ell_2$  la recta orientada, la semirrecta izquierda, y la semirrecta derecha respectivamente, de un 2-level tree. Denotamos por  $m(\ell_i)$  la pendiente de la recta o semirrecta  $\ell_i$ ,  $i = 0, 1, 2$ . Al igual que en la sección anterior,  $\ell'_1$  ( $\ell'_2$ , resp.) es la recta que contiene a la semirrecta  $\ell_1$  ( $\ell_2$ , resp.). Sean  $p$  y  $q$  los puntos de intersección entre la recta  $\ell_0$  y las semirrectas  $\ell_1$  y  $\ell_2$ , respectivamente. Suponemos que los conjuntos  $R$  y  $B$  no son separables por zig-zag, ya que esta situación ha sido estudiada en la sección anterior. Nótese que en tal caso, y teniendo en cuenta que  $R$  y  $B$  tampoco son separables por una  $t$ -poligonal convexa,  $t \leq 3$ , los puntos situados en las cuñas  $(l_1, l_0)$  y  $(l_0, l_2)$  tienen que ser del mismo color.

Para realizar una clasificación sistemática de todas las situaciones posibles en la separabilidad por 2-level tree, seguimos los siguientes criterios en el orden establecido.

1. La pendiente de la recta orientada  $\ell_0$ . Distinguimos los siguientes casos:
  - (a)  $\ell_0$  tiene pendiente positiva o pendiente negativa.
  - (b)  $\ell_0$  es vertical u horizontal.
2. La posición relativa de los puntos  $p$  y  $q$ . Estos puntos están situados en  $\ell_0$  y de acuerdo a su orientación. O bien la posición de  $p$  es anterior a la de  $q$ , o bien la de  $q$  es anterior a la de  $p$ . En ambos casos incluimos la posibilidad de que  $p = q$ .
3. Las posiciones de las semirrectas  $\ell_1$  y  $\ell_2$  atendiendo a los criterios siguientes:
  - (a) Pendientes de las rectas  $\ell'_1$  y  $\ell'_2$ .
  - (b)  $\ell_1$  y  $\ell_2$  son paralelas; o la prolongación de una de las semirrectas corta a la otra.

Una vez aplicados estos criterios diseñamos un algoritmo para cada uno de los casos obtenidos, que son los siguientes:

1.  $\ell_0$  tiene pendiente positiva y la posición de  $q$  es anterior a la de  $p$ , incluyendo la posibilidad de  $p = q$ . Además, hacemos una nueva distinción en función de las pendientes de  $\ell'_1$  y  $\ell'_2$ .
  - (a)  $m(\ell'_1), m(\ell'_2) \in (m(\ell_0), \pi]$  (Figura 9(a)).
  - (b) O bien  $m(\ell'_1) \in [0, m(\ell_0))$  o bien  $m(\ell'_2) \in [0, m(\ell_0))$ . Consideramos tres sub-casos:
    - (i) Las semirrectas  $\ell_1$  y  $\ell_2$  son paralelas (Figura 9(b)).
    - (ii) La prolongación de la semirrecta  $\ell_1$  corta a  $\ell_2$  (Figura 9(c)).
    - (iii) La prolongación de la semirrecta  $\ell_2$  corta a  $\ell_1$  (Figura 9(d)).

Los casos en que  $\ell_0$  tiene pendiente negativa o la posición de  $p$  es anterior a la de  $q$  son simétricos y se resuelven de forma similar.

2. La recta  $\ell_0$  es vertical u horizontal (Figura 9(e)).

Una idea general del procedimiento utilizado para decidir y calcular la existencia de un 2-level tree separador es la siguiente. En primer lugar, clasificamos uno de los dos conjuntos correctamente según el 2-level tree calculando una banda (recta) que separa dicho conjunto de forma adecuada. El siguiente resultado nos asegura la existencia de esta banda.

**Lema 3.1.** *Si  $R$  y  $B$  son separables por un 2-level tree  $(l_1, l_0, l_2)$ , entonces la recta  $\ell_0$  genera una banda (recta) que separa correctamente  $R$  ó  $B$  en los dos subconjuntos monocromáticos definidos por el 2-level tree. Además, o bien dicha banda (recta) tiene pendiente positiva, o es horizontal, o bien es vertical.*

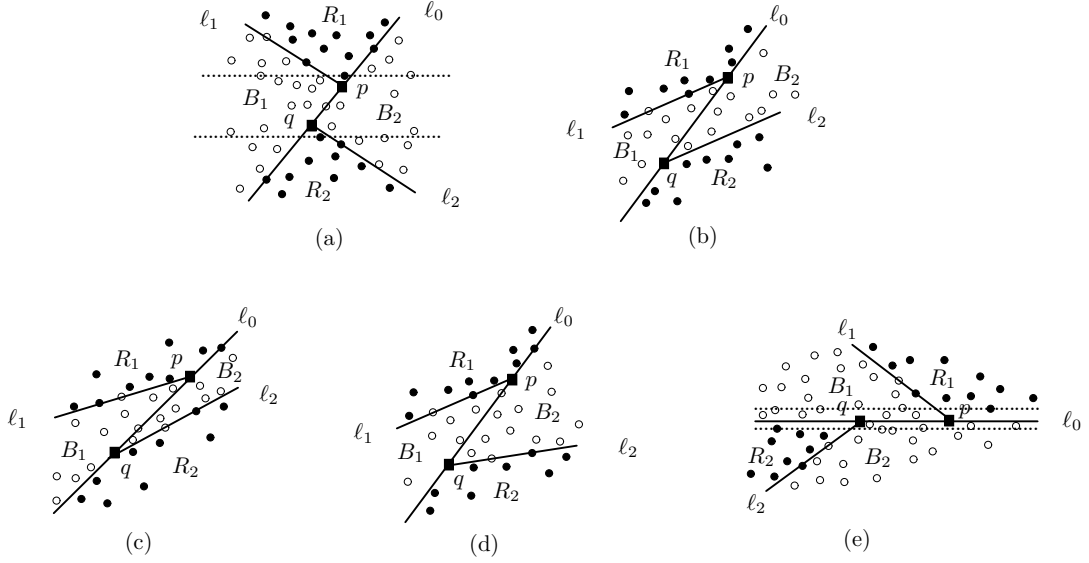


Figura 9: (a)  $\ell_0$  tiene pendiente positiva y  $m(\ell'_1), m(\ell'_2) \in (m(\ell_0), \pi]$  (b) la recta  $\ell_0$  tiene pendiente positiva,  $m(\ell'_1) \in [0, m(\ell_0))$ ,  $m(\ell'_2) \in [0, m(\ell_0))$ , y  $\ell_1, \ell_2$  son paralelas (c)  $\ell_0$  con pendiente positiva,  $m(\ell'_1) \in [0, m(\ell_0))$ ,  $m(\ell'_2) \in [0, m(\ell_0))$ , y la prolongación de  $\ell_1$  corta a  $\ell_2$  (d)  $\ell_0$  con pendiente positiva,  $m(\ell'_1) \in [0, m(\ell_0))$ ,  $m(\ell'_2) \in [0, m(\ell_0))$ , y la prolongación de  $\ell_2$  corta a  $\ell_1$  (e)  $\ell_0$  horizontal.

Sin pérdida de generalidad, podemos suponer que es el conjunto de puntos rojos  $R$  el que está clasificado por la banda en los subconjuntos  $R_1$  y  $R_2$ . A continuación, usando esta clasificación, el algoritmo clasifica el conjunto de puntos azules  $B$  en los subconjuntos  $B_1$  y  $B_2$ , y calcula una solución  $(\ell_1, \ell_0, \ell_2)$ . Para caracterizar las soluciones del problema utilizamos el siguiente resultado.

**Lema 3.2.** *Si existe una separación de los conjuntos  $R$  y  $B$  mediante un 2-level tree, entonces:*

- O bien la recta  $\ell_0$  pasa por un punto rojo de  $CH(R_1)$  o de  $CH(R_2)$  y por un punto azul de  $CH(B_1)$  o  $CH(B_2)$ ; o bien la recta  $\ell_0$  pasa por dos puntos rojos, uno de  $CH(R_1)$  y el otro de  $CH(R_2)$ .
- La semirrecta  $\ell_1$  pasa por un punto rojo de  $CH(R_1)$  y por un punto azul de  $CH(B_1)$ .
- La semirrecta  $\ell_2$  pasa por un punto rojo de  $CH(R_2)$  y por un punto azul de  $CH(B_2)$ .

Los casos posibles de un 2-level tree quedan reducidos a los mostrados en la Figura 9. De estos casos, el caso (e) es similar al caso (a) y se resuelve con el mismo algoritmo. El Caso (d) se determina girando la configuración de puntos 180 grados, cambiando las semirrectas  $\ell_2$  por  $\ell_1$  y viceversa con lo cual obtenemos el caso (c). El caso (b) realmente es un caso particular del caso (c) en el que las semirrectas  $\ell_1$  y  $\ell_2$  son paralelas. En resumen, hemos diseñado algoritmos que resuelven los casos (a), el caso (b) y el caso (c). En el diseño de los algoritmos usamos el siguiente resultado.

Sea  $(b_1 \leq b_2 \leq \dots \leq b_n)$  la secuencia de los puntos azules ordenados por coordenada  $x$ , así  $b_1$  y  $b_n$  son el primer y último punto azul en esta ordenación.

**Lemma 3.3.** *Si existe una separación por 2-level tree, los puntos azules  $b_1$  y  $b_n$  verifican uno de los siguientes casos: (1)  $b_1 \in B_1, b_n \in B_2$ ; (2)  $b_1 \in B_2, b_n \in B_1$ ; (3)  $b_1, b_n \in B_2$ ; (4)  $b_1, b_n \in B_1$ .*

Para resolver el caso (a) hacemos un barrido con una recta horizontal que clasifica los puntos rojos y azules dependiendo si están encima o debajo de la recta de barrido, hasta encontrar la solución.

En los casos (b) y (c) hacemos un barrido rotacional dependiendo de las situaciones de los puntos con un estudio de los casos según el Lemma 3.3. Debido a la falta de espacio no mostraremos los algoritmos diseñados para cada uno de los casos, pero todos se pueden resolver en tiempo  $O(n \log n)$ .

Una vez resueltos en tiempo  $O(n \log n)$  el número constante de casos posibles en la separabilidad mediante un 2-level tree, y como consecuencia de la cota inferior  $\Omega(n \log n)$  especificada en la sección anterior para el caso del zig-zag, obtenemos el siguiente resultado.

**Teorema 3.4.** Sean  $R$  y  $B$  dos conjuntos disjuntos de  $n$  puntos en el plano. Decidir y calcular si  $R$  y  $B$  pueden ser separados y clasificados mediante un 2-level tree puede hacerse óptimamente en tiempo  $\Theta(n \log n)$ .

Queremos hacer notar que existen configuraciones de puntos rojos y azules tales que, la profundidad  $k$  del mínimo  $k$ -level tree que clasifica los dos conjuntos de puntos es  $k = O(\log n)$ . Para construir tal configuración, suponemos que tenemos un conjunto  $S$  de  $n$  puntos de un sólo color en posición general. Sustituimos cada punto  $p \in S$  por un grupo de 4 puntos, dos rojos y dos azules de la forma indicada en la Figura 10. Obtenemos así, dos conjuntos disjuntos de  $2n$  puntos rojos y  $2n$  puntos azules, denotados por  $R$  y  $B$ , respectivamente. Todo  $k$ -level tree separador de  $R$  y  $B$  tiene que separar cada uno de los 2 pares de puntos rojos y azules definidos por cada punto  $p \in S$ . Por tanto, los nodos del penúltimo nivel del árbol son las  $n$  semirrectas que separan estos pares de puntos rojos y azules. Así, obtenemos un  $O(\log n)$ -level tree cuyas hojas son los  $4n$  puntos de  $R \cup B$ .

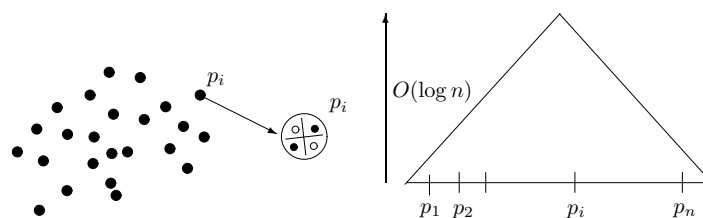


Figura 10:  $O(\log n)$ -level tree.

Para finalizar esta sección, queremos comentar que los algoritmos diseñados para el estudio de la separabilidad de dos conjuntos de puntos mediante un 2-level tree pueden ser aplicados al caso en que tengamos tres o cuatro conjuntos disjuntos de  $n$  puntos, por ejemplo  $R$ ,  $B$ ,  $G$ , y  $Y$ . En particular el caso de cuatro conjuntos puede resolverse en tiempo  $O(n)$  comprobando separaciones lineales entre los diferentes pares de los conjuntos dependiendo de su ordenación radial. En el caso de tres conjuntos disjuntos obtenemos el siguiente resultado.

**Teorema 3.5.** Sean  $R$ ,  $B$ ,  $G$  tres conjuntos disjuntos de  $n$  puntos en el plano. Decidir y calcular si dichos conjuntos pueden ser separados y clasificados por un 2-level tree puede hacerse óptimamente en tiempo  $\Theta(n \log n)$ .

## Referencias

- [1] E. M. Arkin, F. Hurtado, J. S. B. Mitchell, C. Seara, and S. S. Skiena. Some lower bounds on geometric separability problems. *International Journal of Computational Geometry and Applications*, Vol. 16, No. 1, 2006, pp. 1-26.
- [2] H. Edelsbrunner and F. P. Preparata. Minimum polygonal separation. *Information and Computation*, 77, 1988, pp. 218–232.
- [3] S. Fekete. On the complexity of min-link red-blue separation problem. *Manuscript*, 1992.
- [4] F. Hurtado, M. Noy, P. A. Ramos, C. Seara. Separating objects in the plane by wedges and strips. *Discrete Applied Mathematics*, Vol. 109, 2000, pp. 109–138.
- [5] N. Megiddo. Linear-time algorithms for linear programming in  $\mathbb{R}^3$  and related problems. *SIAM Journal of Computing*, Vol. 12, No. 4, 1983, pp. 759–776.
- [6] V. Sacristán. Lower bounds for some geometric problems. *Technical Report MA-IR-98-0034*, Universitat Politècnica de Catalunya, Departament de Matemàtica Aplicada II, 1998. Abstract in *Actas VIII Encuentros de Geometría Computacional*, 1999, pp. 371–372.