Análisis de Algoritmos II

Un algoritmo de barrido de línea para agrupación espacial.

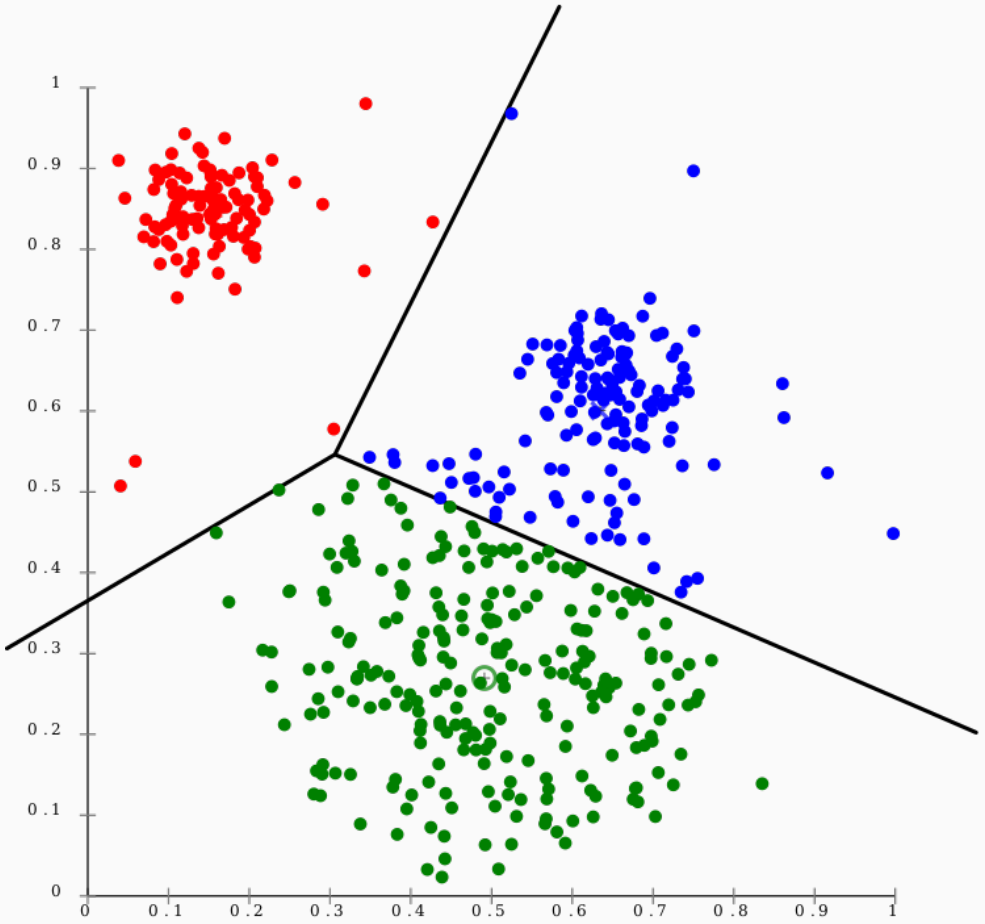
Profesores:
Jorge Urrutia Galicia
Adriana Ramírez Vigueras
Diego Jesús Favela Nava.

Aguilera Moreno Adrian.

Facultad de Ciencias, UNAM

Tabla de contenido.

Un conjunto de datos agrupados en *k* grupos “similares”.



Agrupamiento por K-means.

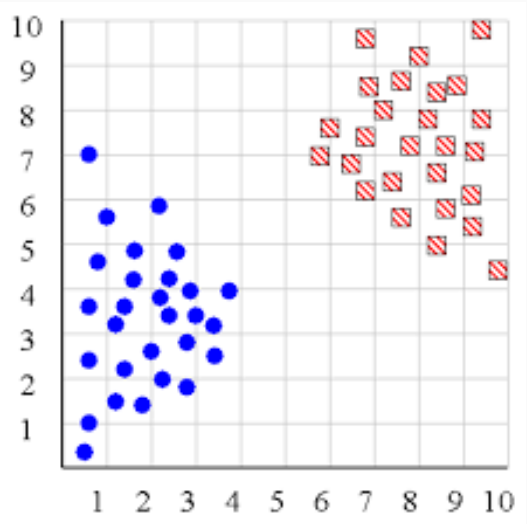
¿Cómo agrupar? ...

Categorías.

Existen dos categorías principales para realizar agrupamientos:

1. **Algoritmos de agrupamiento *jerárquico***.
   1.1 Aglomerantes.
   Inicialmente cada objeto es un grupo, conforme transcurren las iteraciones los objetos se deben ir fucionando.
   1.2 Divisivos.
   Inicialmente todos los objetos son un solo grupo, conforma transcurren las iteraciones se van subdividiendo estos grupos.
2. **Algoritmos de agrupamiento *particional***.
   Cada objeto se asocia con el centro de agrupamiento del más cercano.

Agrupamiento Particional.

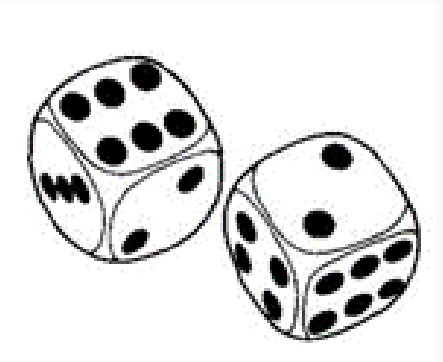


Agrupamiento Jerárquico.

Alternativas.

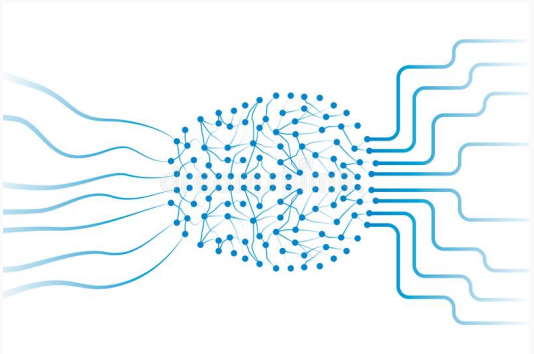
Algunas alternativas para agrupar son:

1. **Redes neuronales**.
2. **k-means** + **Algoritmos genéticos**.
3. **Muestreos aleatorios**.

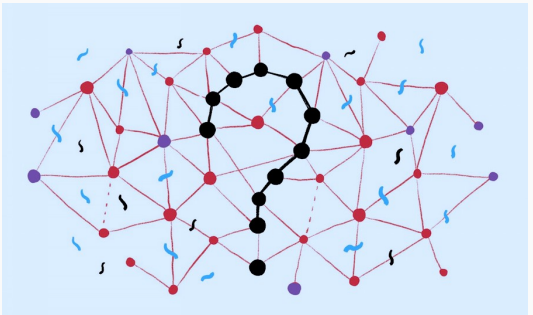


Aleatorios.

Algunas alternativas para agrupar basadas en el entrenamiento inteligente, búsquedas aleatorias (como las heurísticas), y uso de algoritmos genéticos (como las colonias de hormigas) son recurridas cuando no podemos garantizar un “buen” agrupamiento.



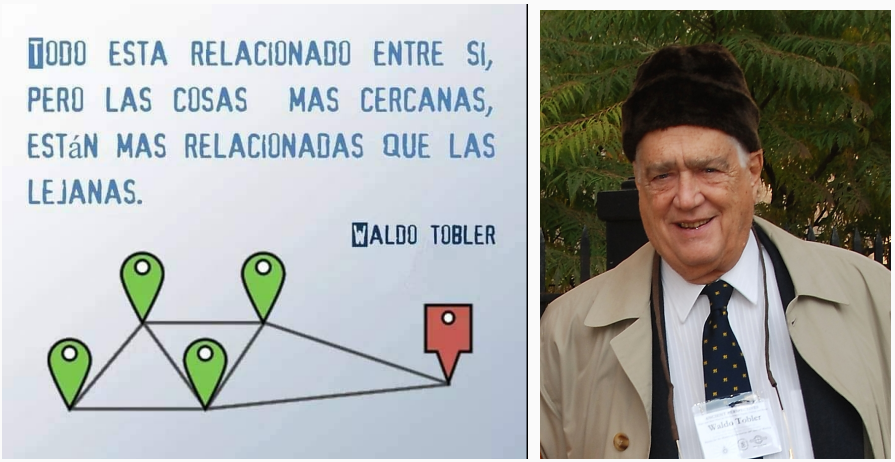
Redes Neuronales.



K-means + Genéticos.

Agrupación Espacial: Propuestas I.

La agrupación espacial es un subconjunto espacial de agrupación. Este tipo de agrupamiento es relacionado, con frecuencia, a métodos gráficos.



1era ley de la geografía.

Propuestas:

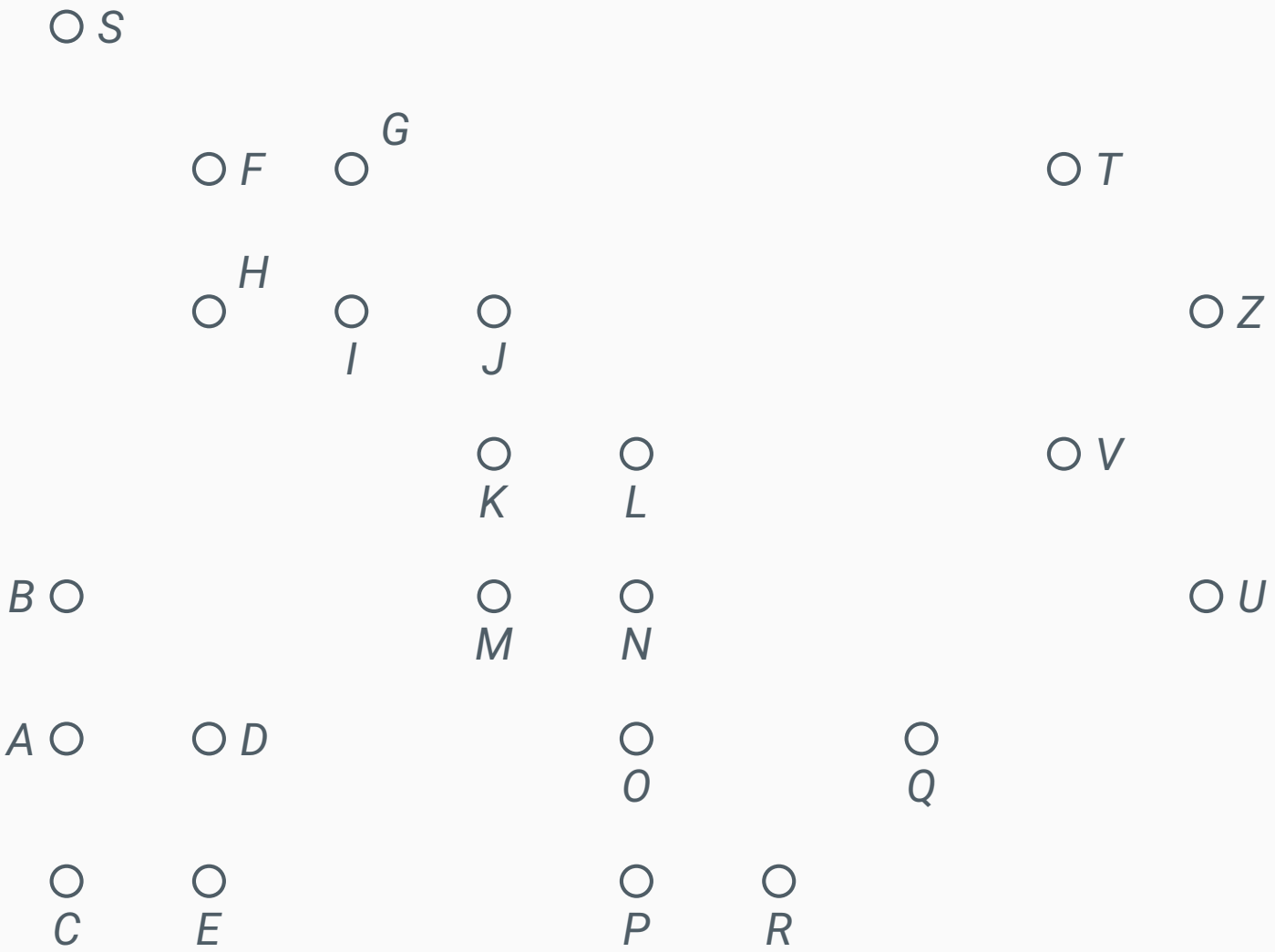
- Zahn sugiere trabajar con un gráfico completo (con vértices cada elemento en el espacio), construir el árbol de expansión mínima y eliminar los “bordes” más largos comparando las longitudes de los arcos con la longitud promedio, eliminando aquellos con longitud mayor al doble de la longitud promedio.

Propuestas:

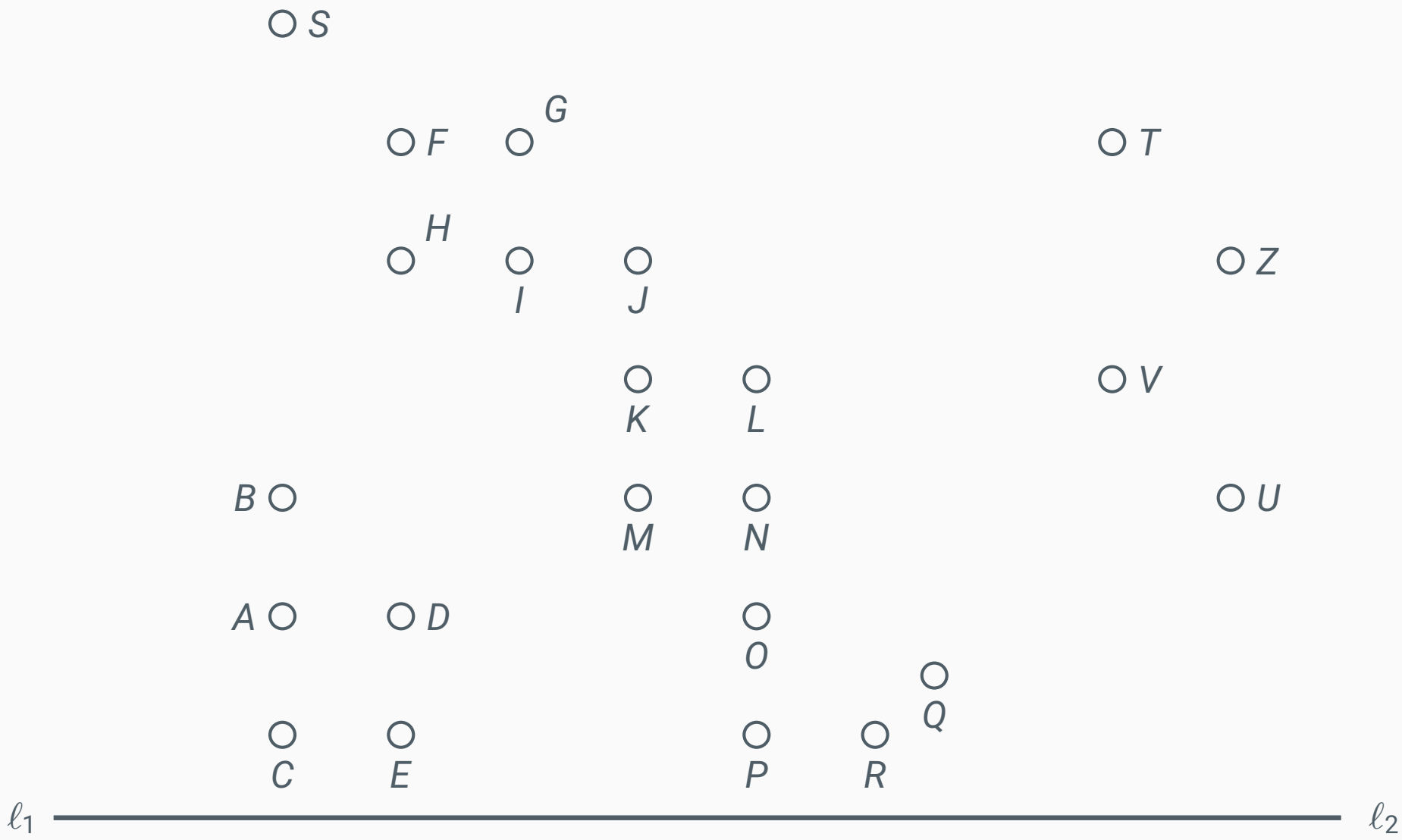
- Narendra sugiere el uso de diagramas de Voronoi para agrupar en tiempo $\mathcal{O}(n \log n)$. El problema de esta solución es que los algoritmos son dificiles de implementar.
- Kang usó triangulaciones de Delaunay y un diagrama dual de Voronoi. Después de construir la triangulación en $\mathcal{O}(n \log n)$, eliminamos las aristas con longitudes mayores a d.
- Yujian presentó un algoritmo de agrupamiento en subárboles máximos en distancia.

¿Problemas? ...

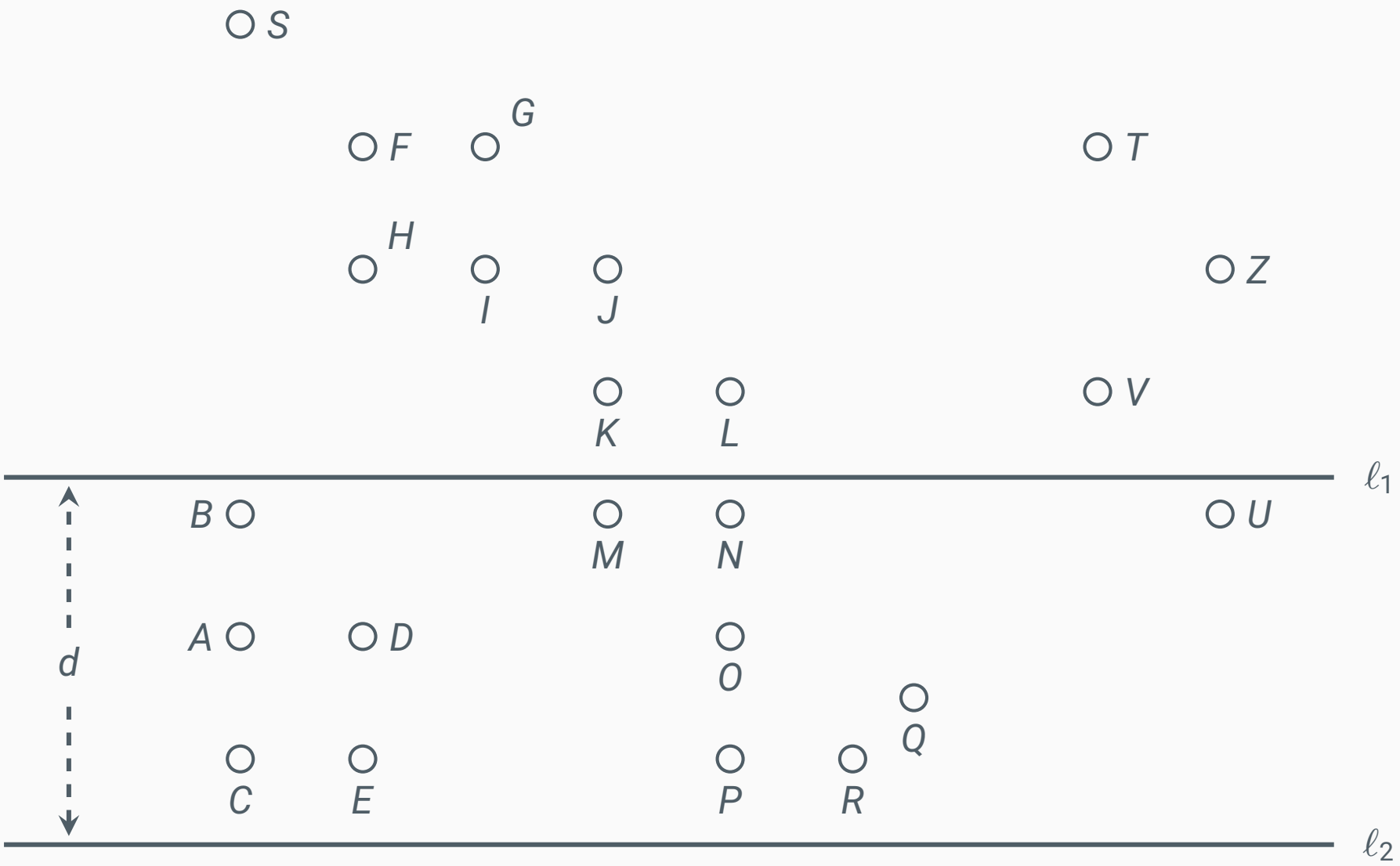
Algoritmo: Conjunto de puntos.



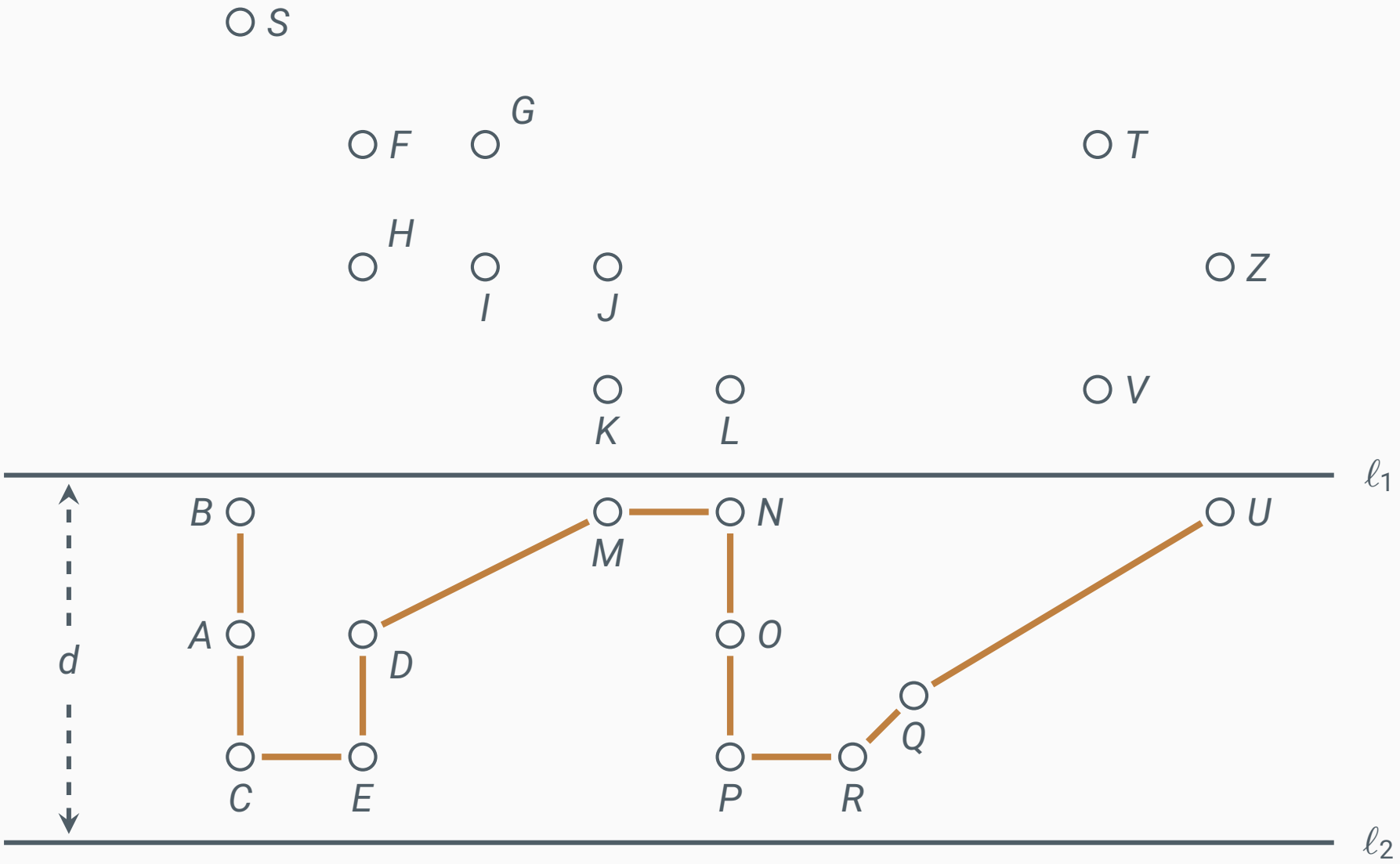
Algoritmo: Iteración 1.



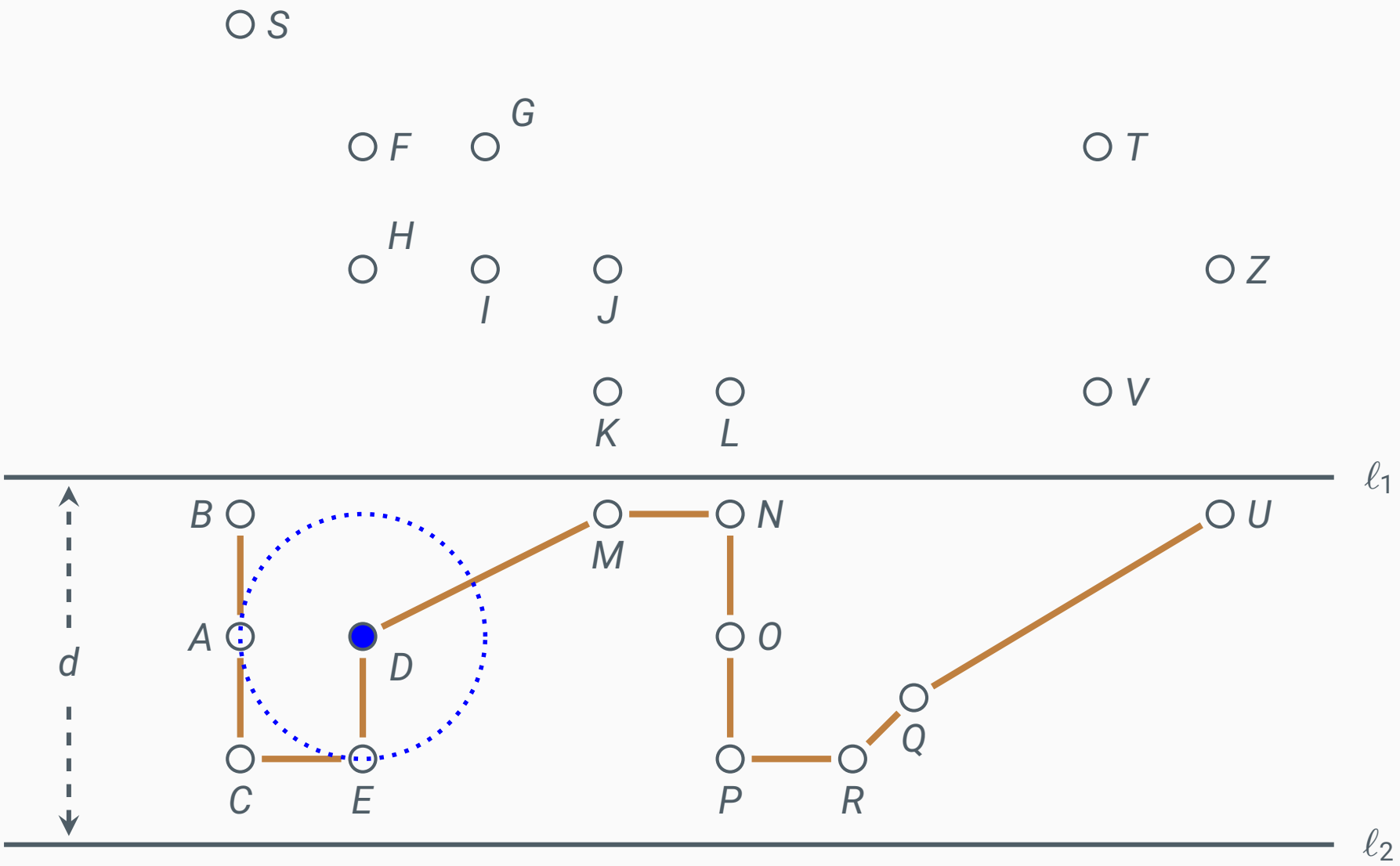
Algoritmo: Iteración 1.



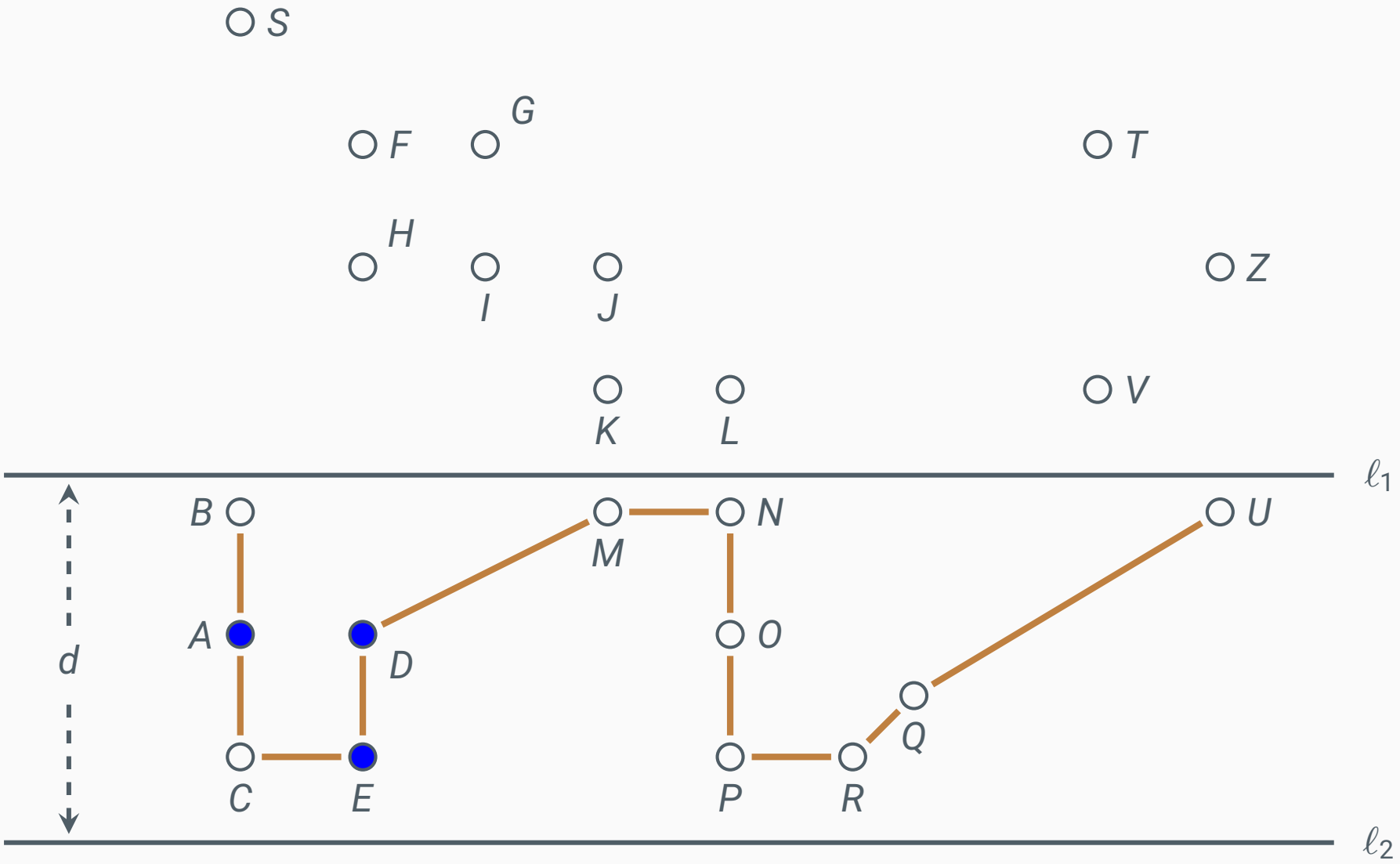
Algoritmo: Iteración 1.



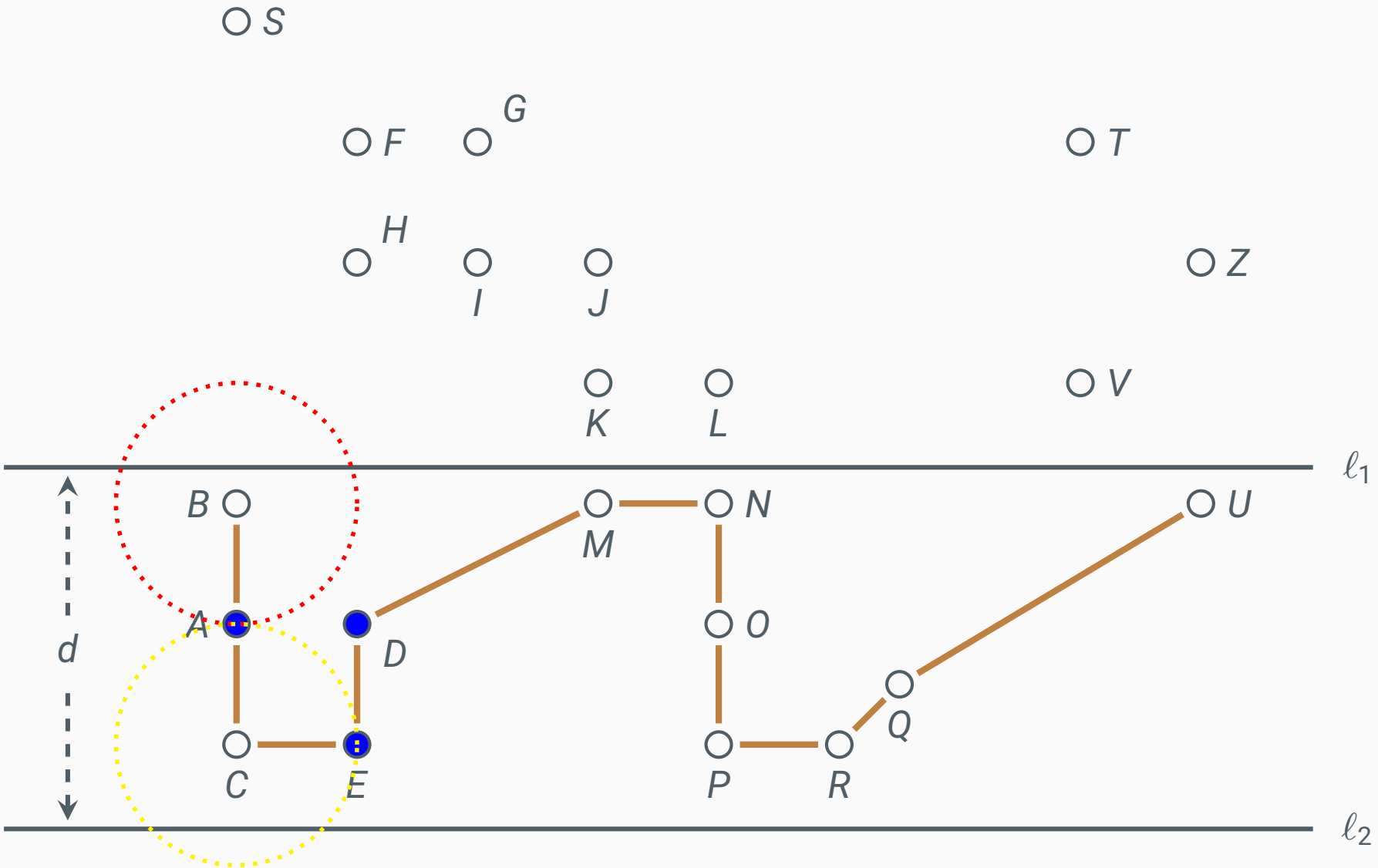
Algoritmo: Iteración 1.



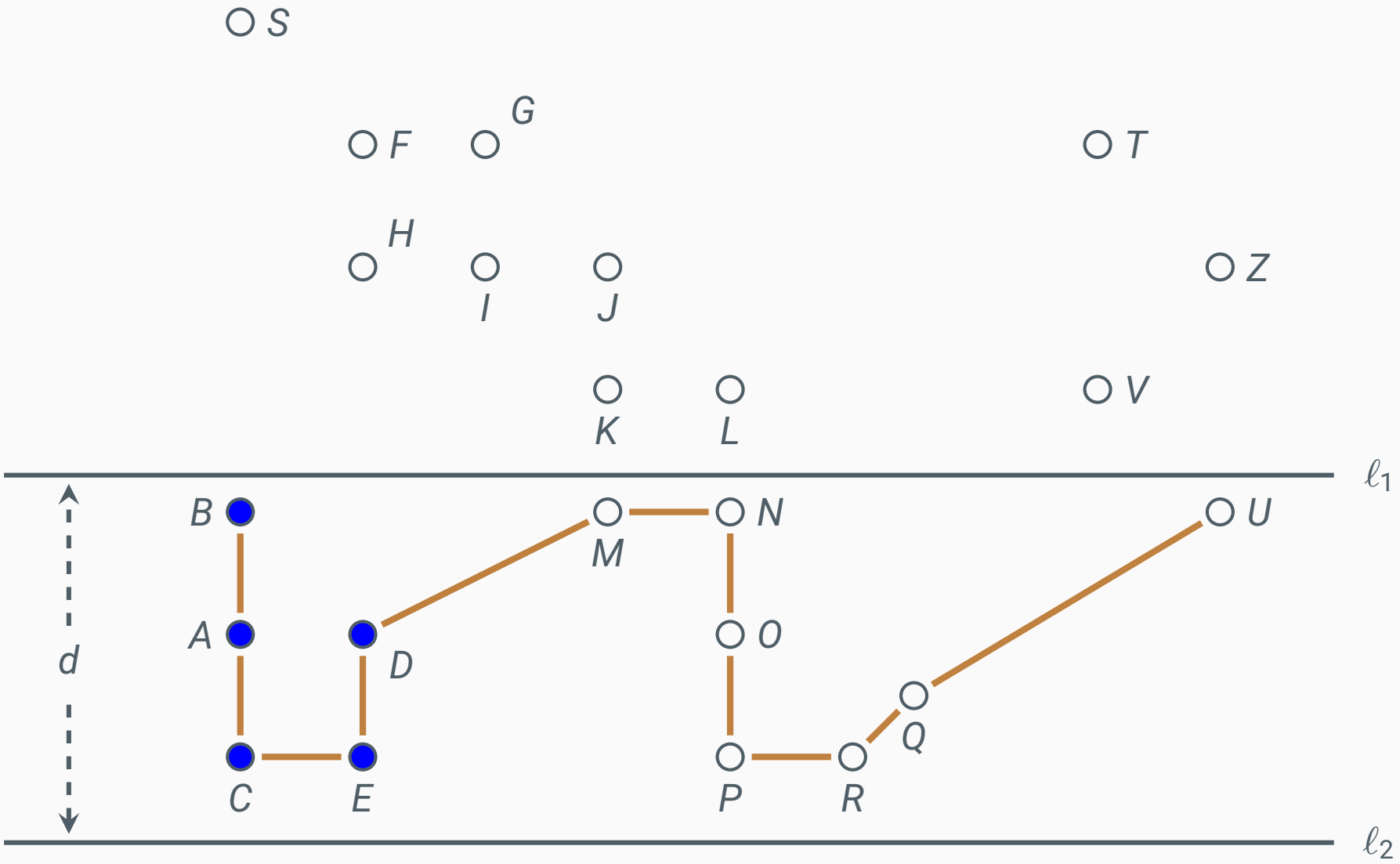
Algoritmo: Iteración 1.



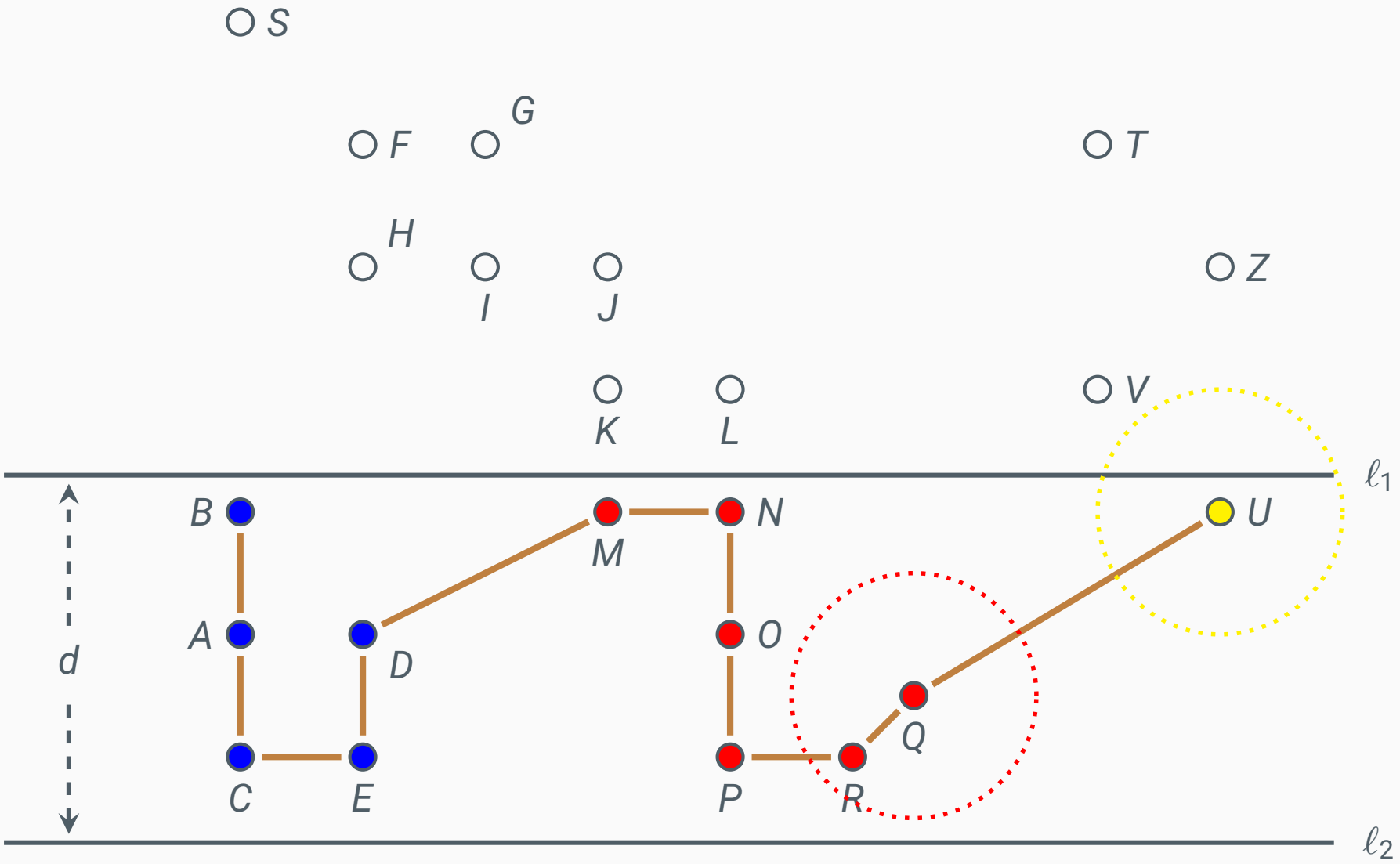
Algoritmo: Iteración 1.



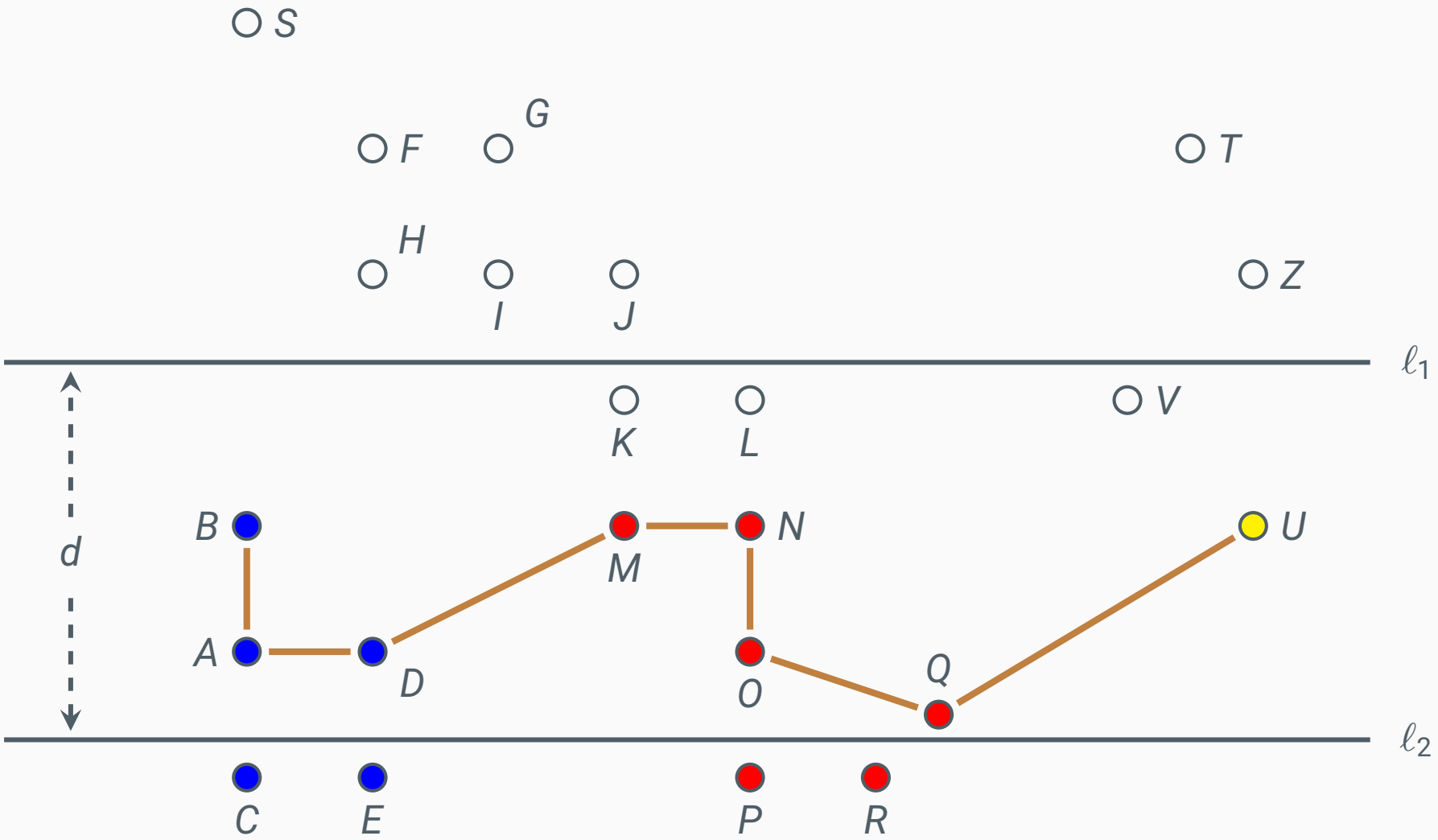
Algoritmo: Iteración 1.



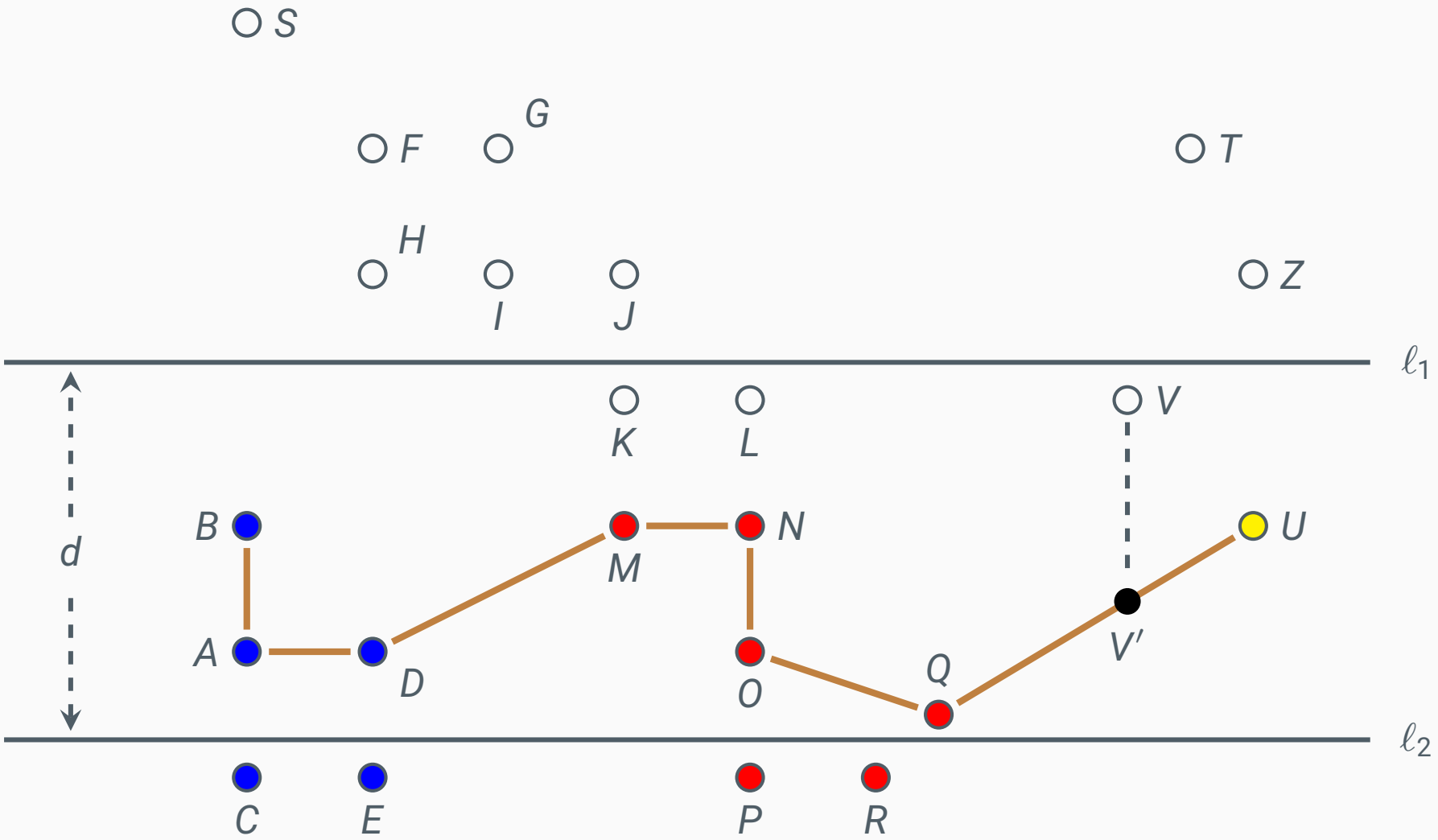
Algoritmo: Iteración 1.



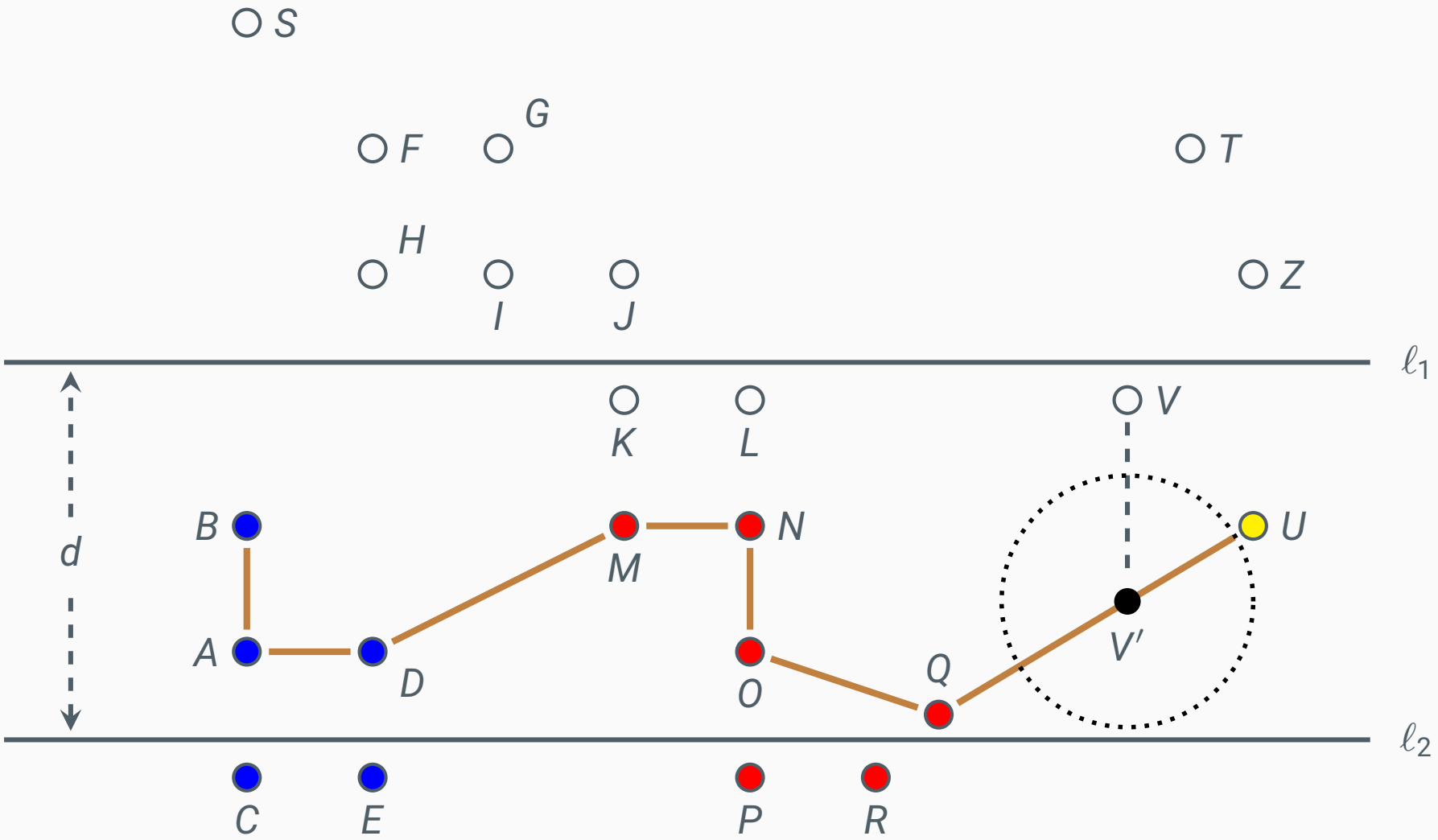
Algoritmo: Iteración 2.



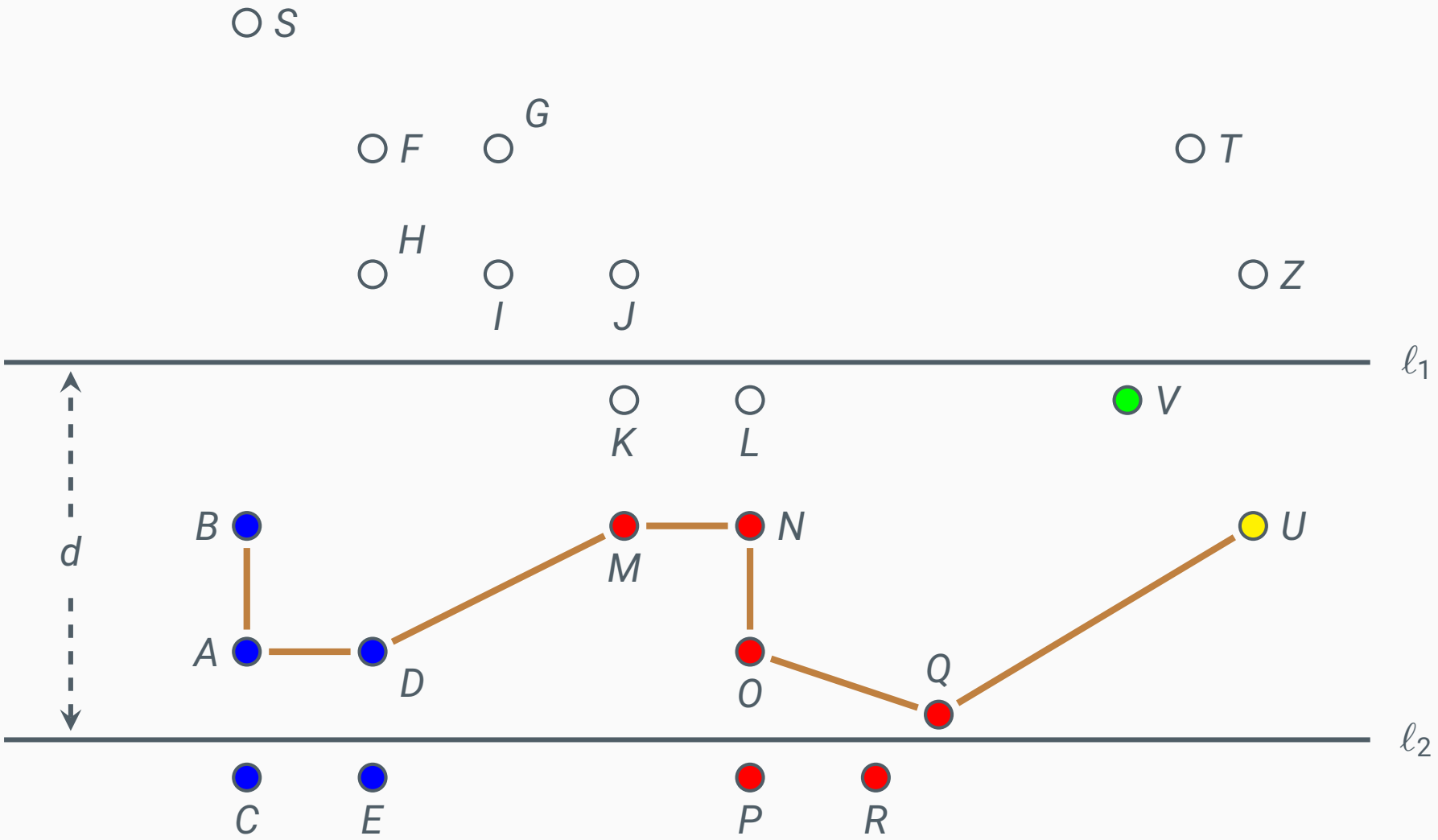
Algoritmo: Iteración 2.



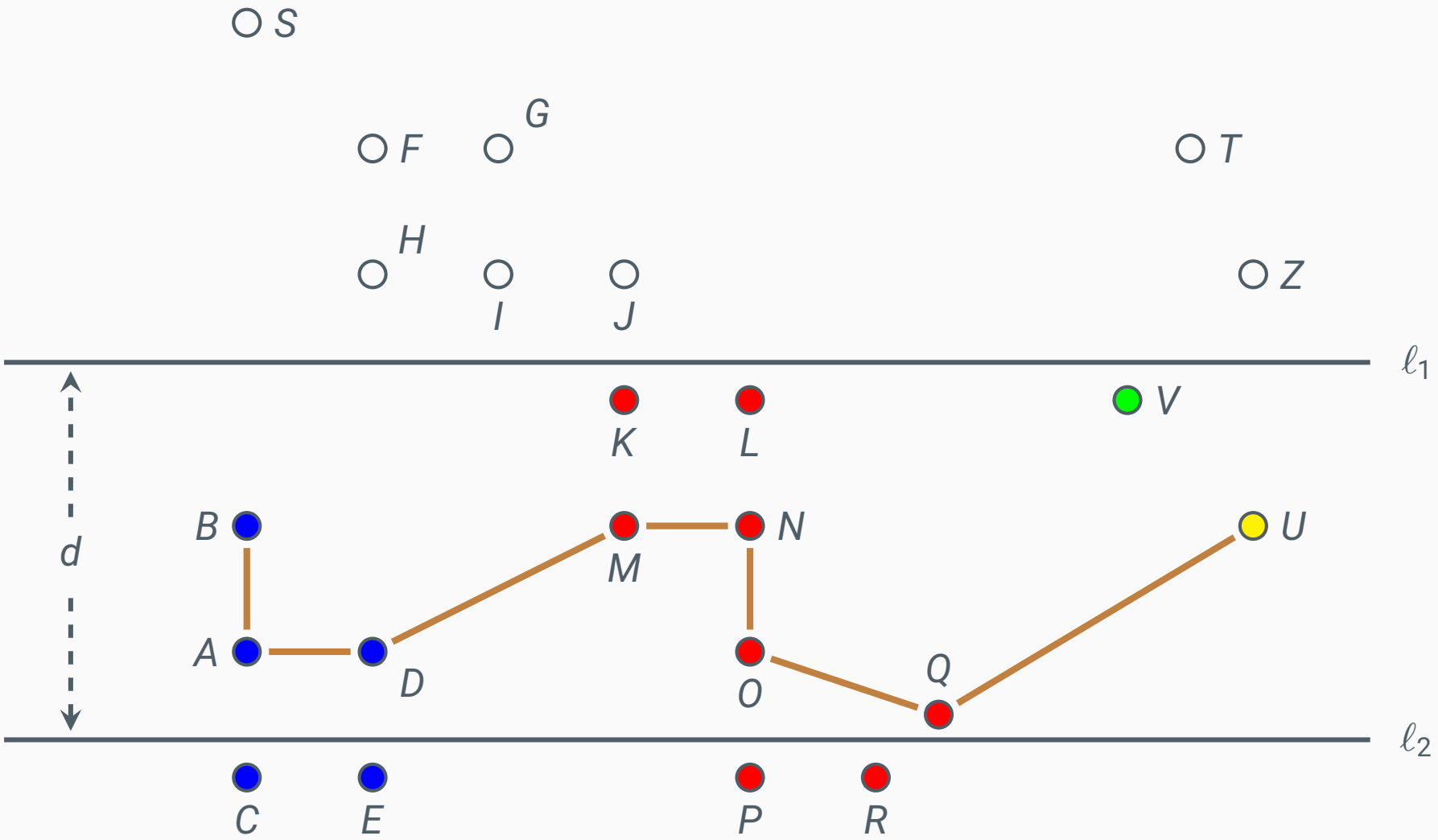
Algoritmo: Iteración 2.



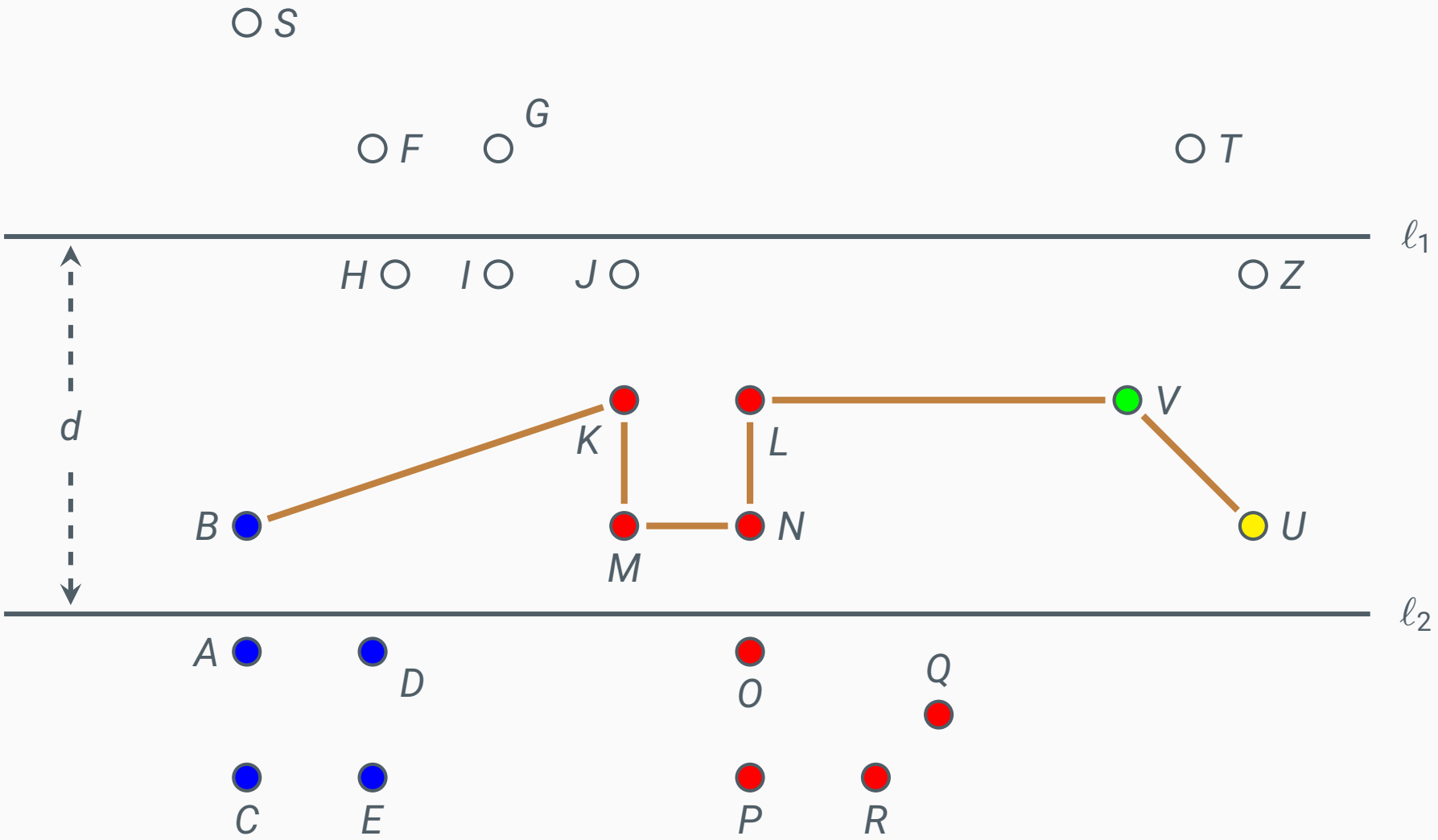
Algoritmo: Iteración 2.



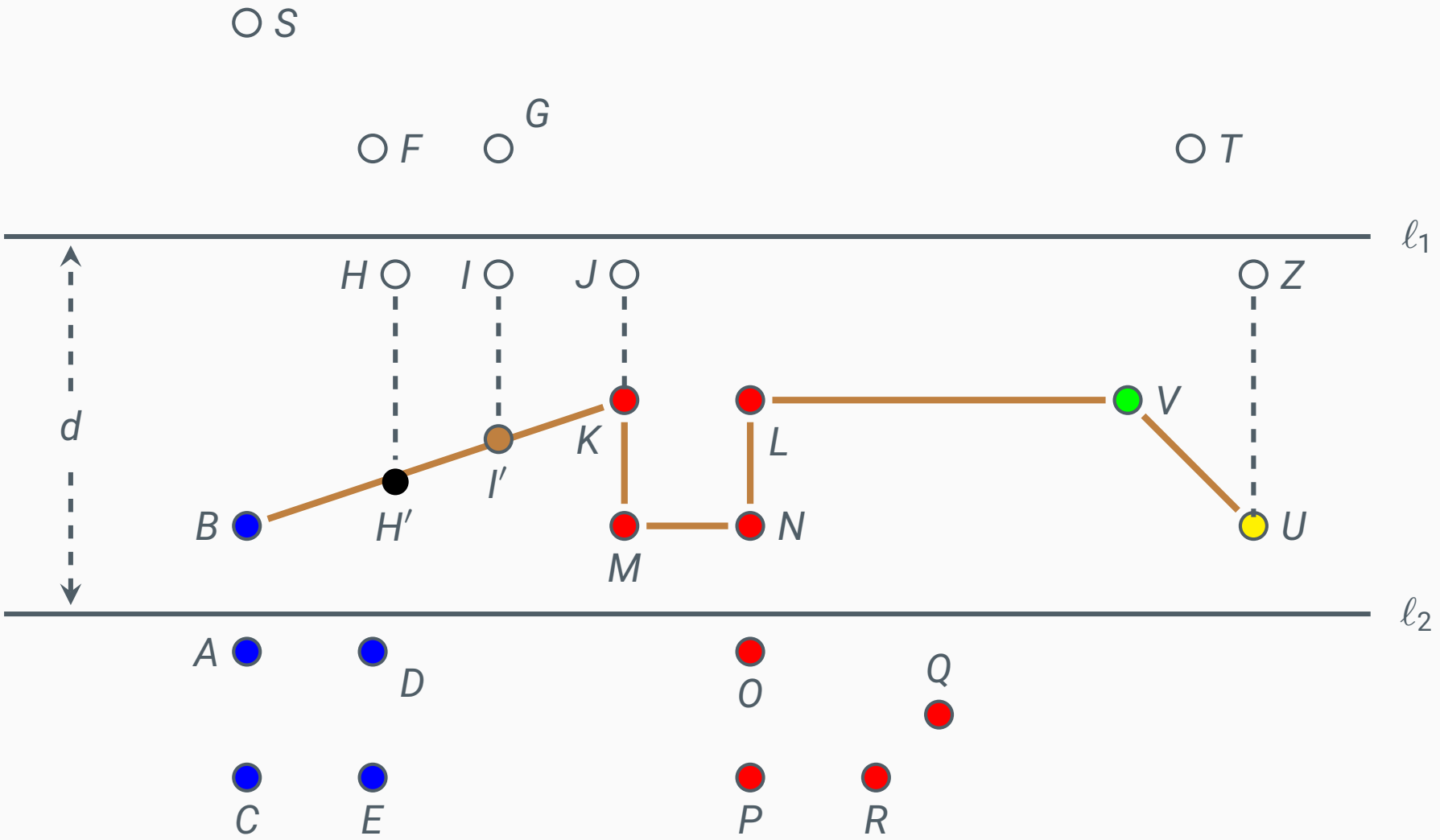
Algoritmo: Iteración 2.



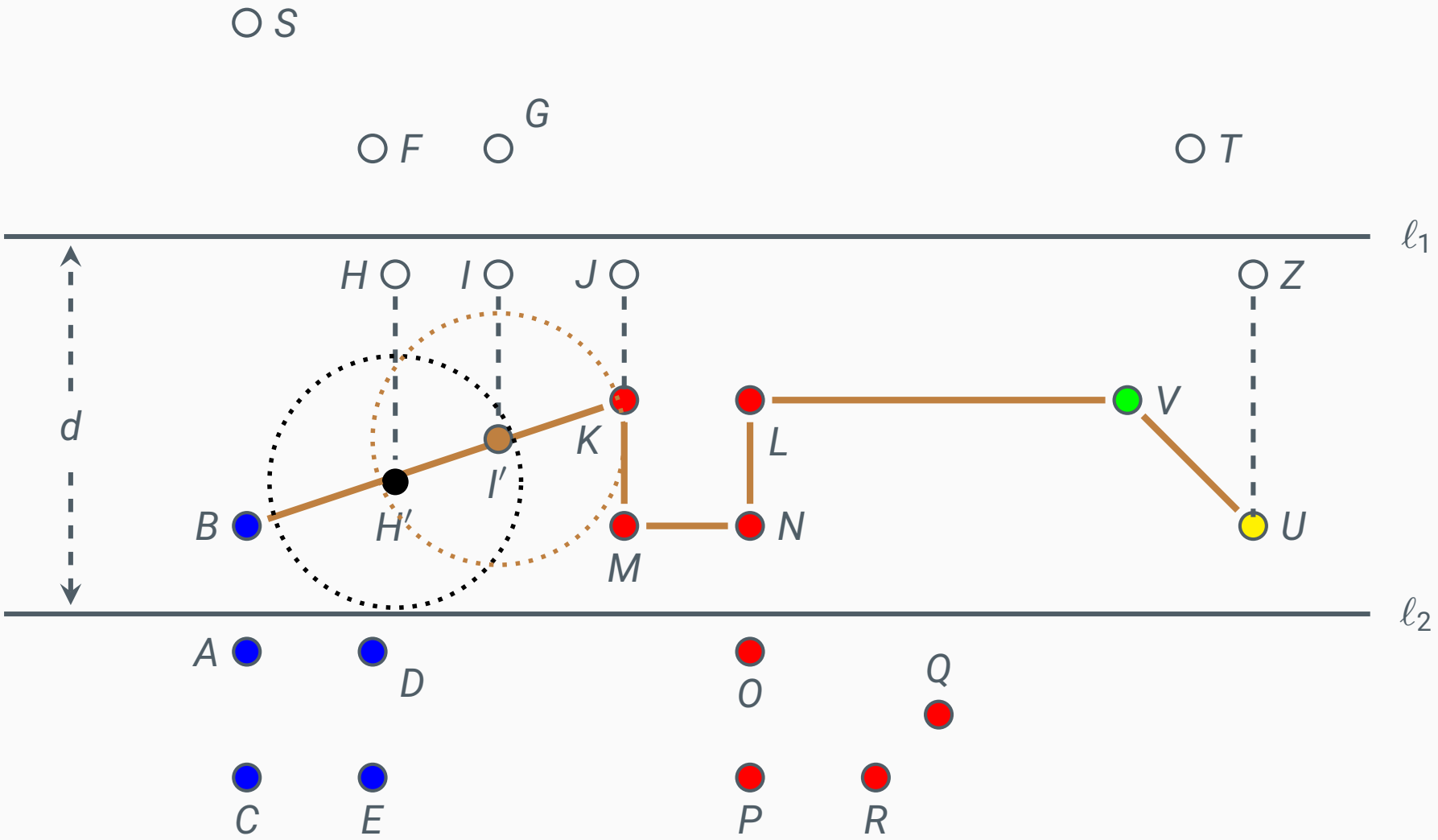
Algoritmo: Iteración 3.



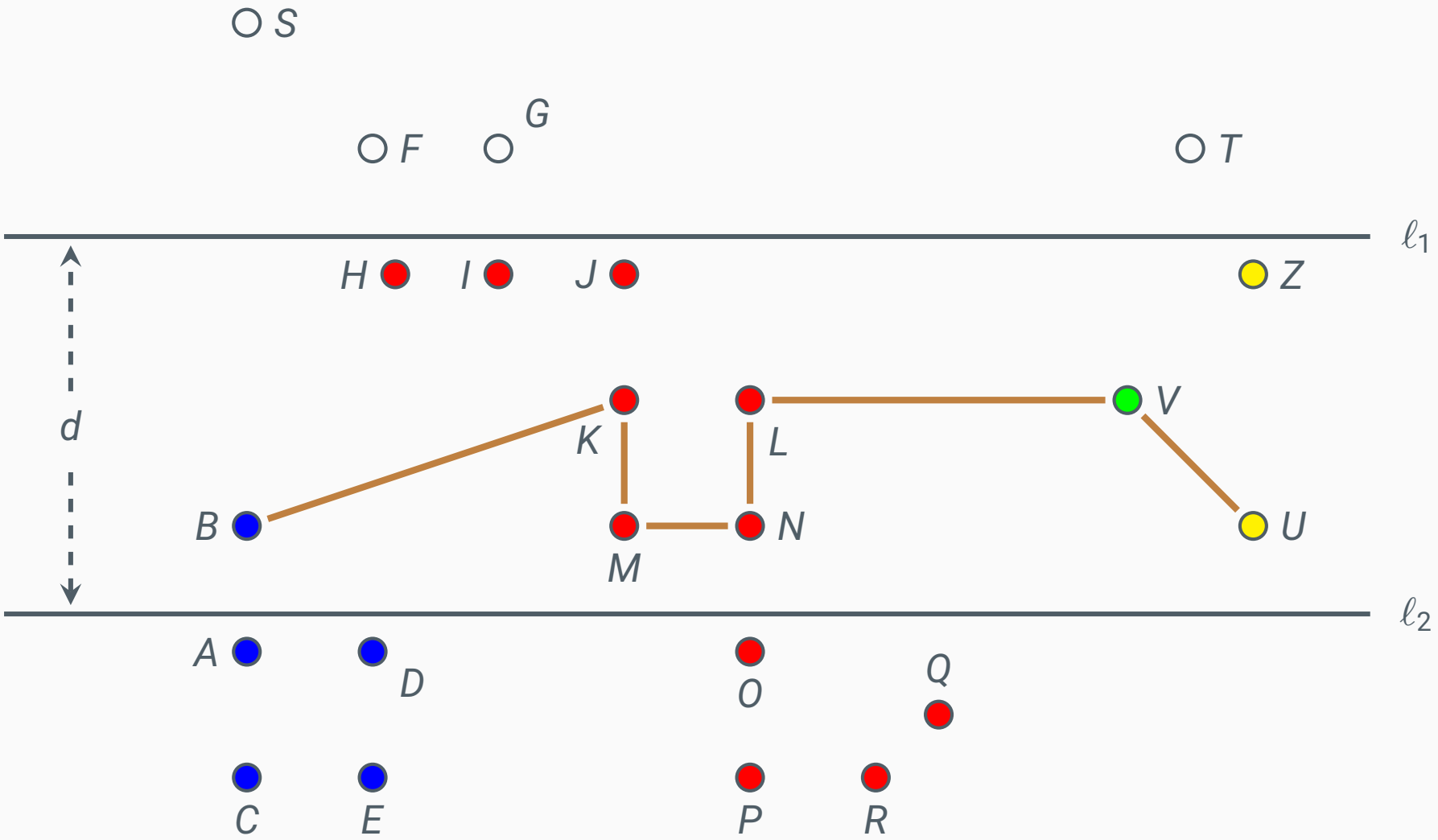
Algoritmo: Iteración 3.



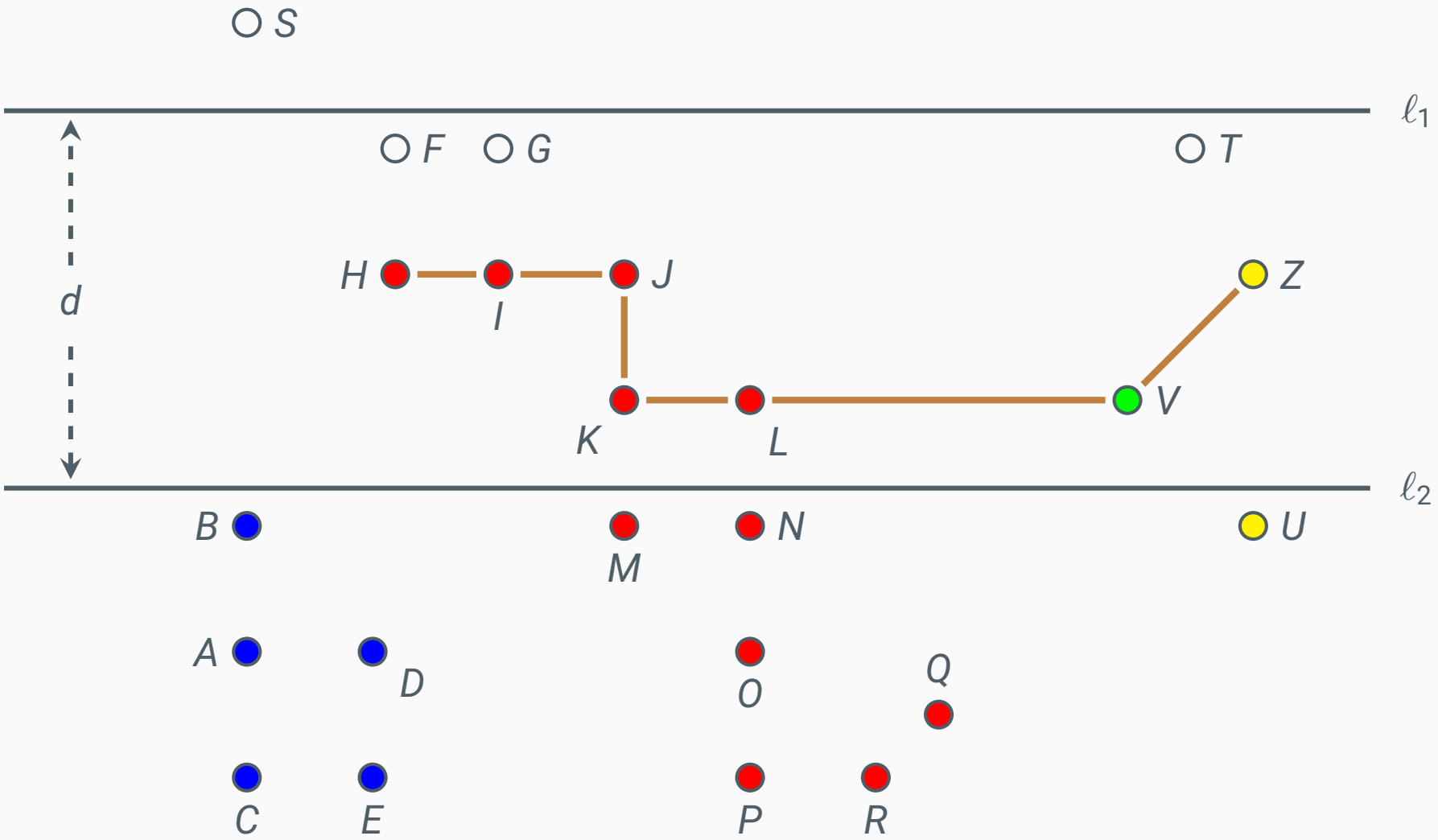
Algoritmo: Iteración 3.



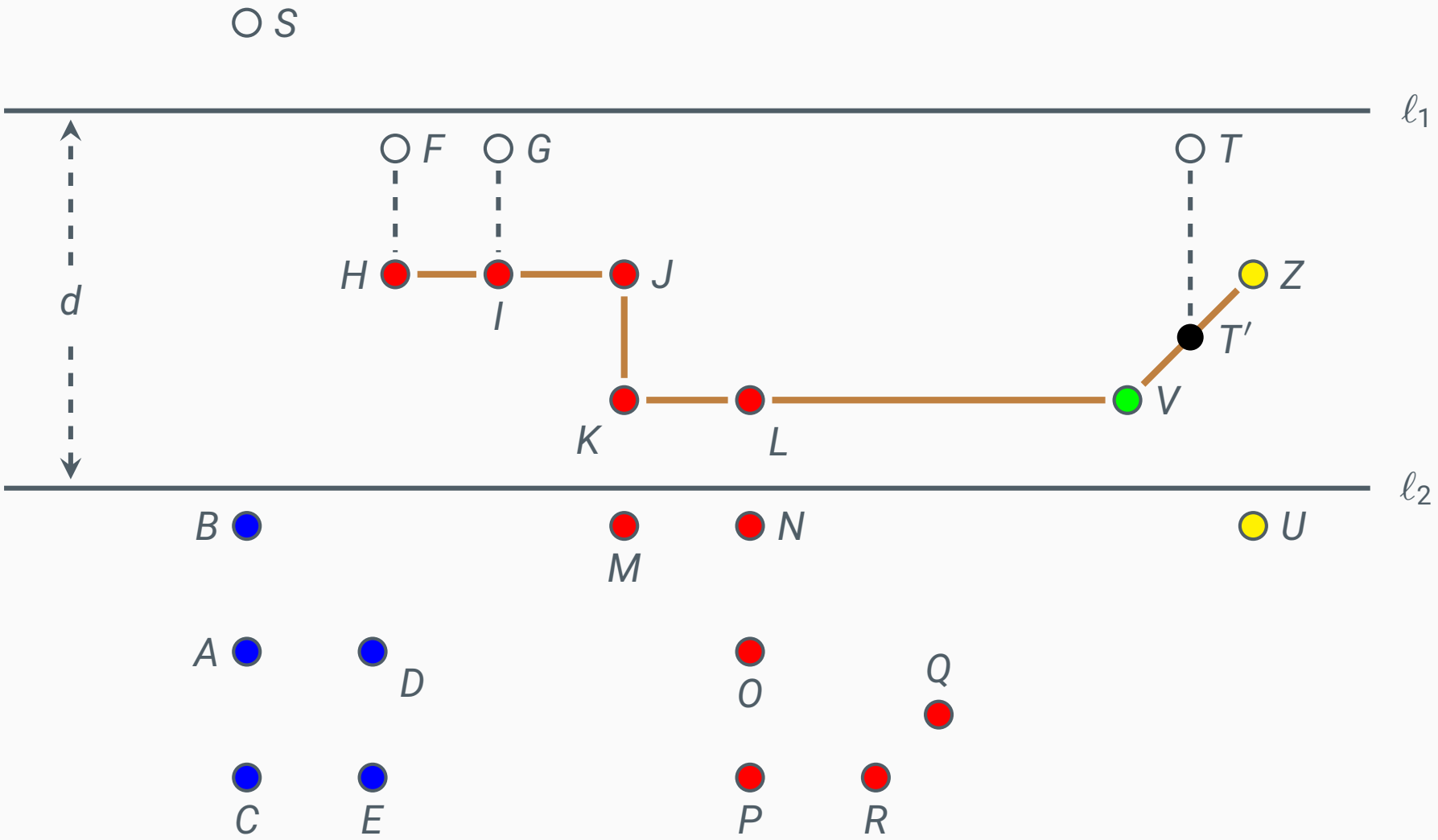
Algoritmo: Iteración 3.



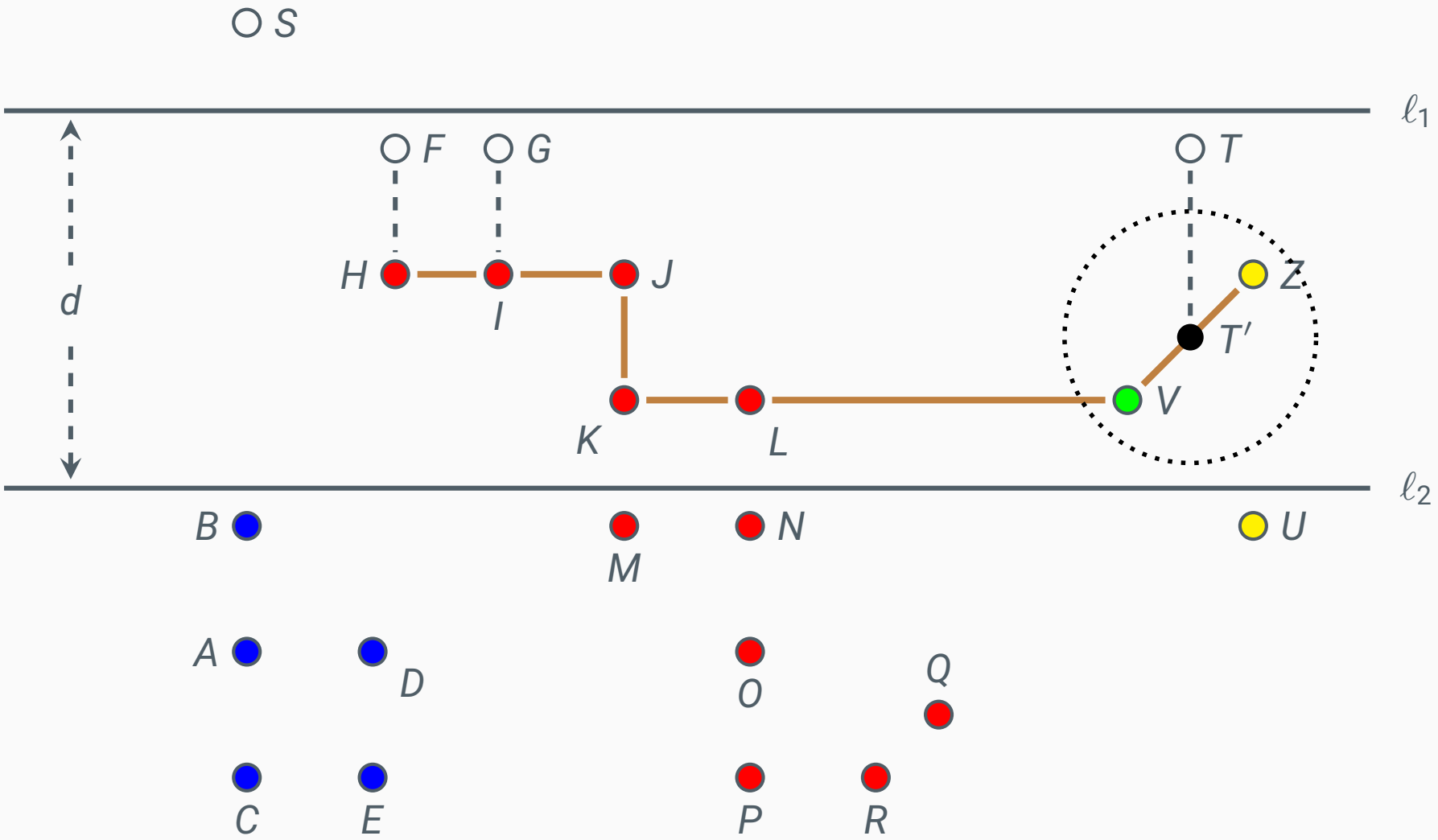
Algoritmo: Iteración 4.



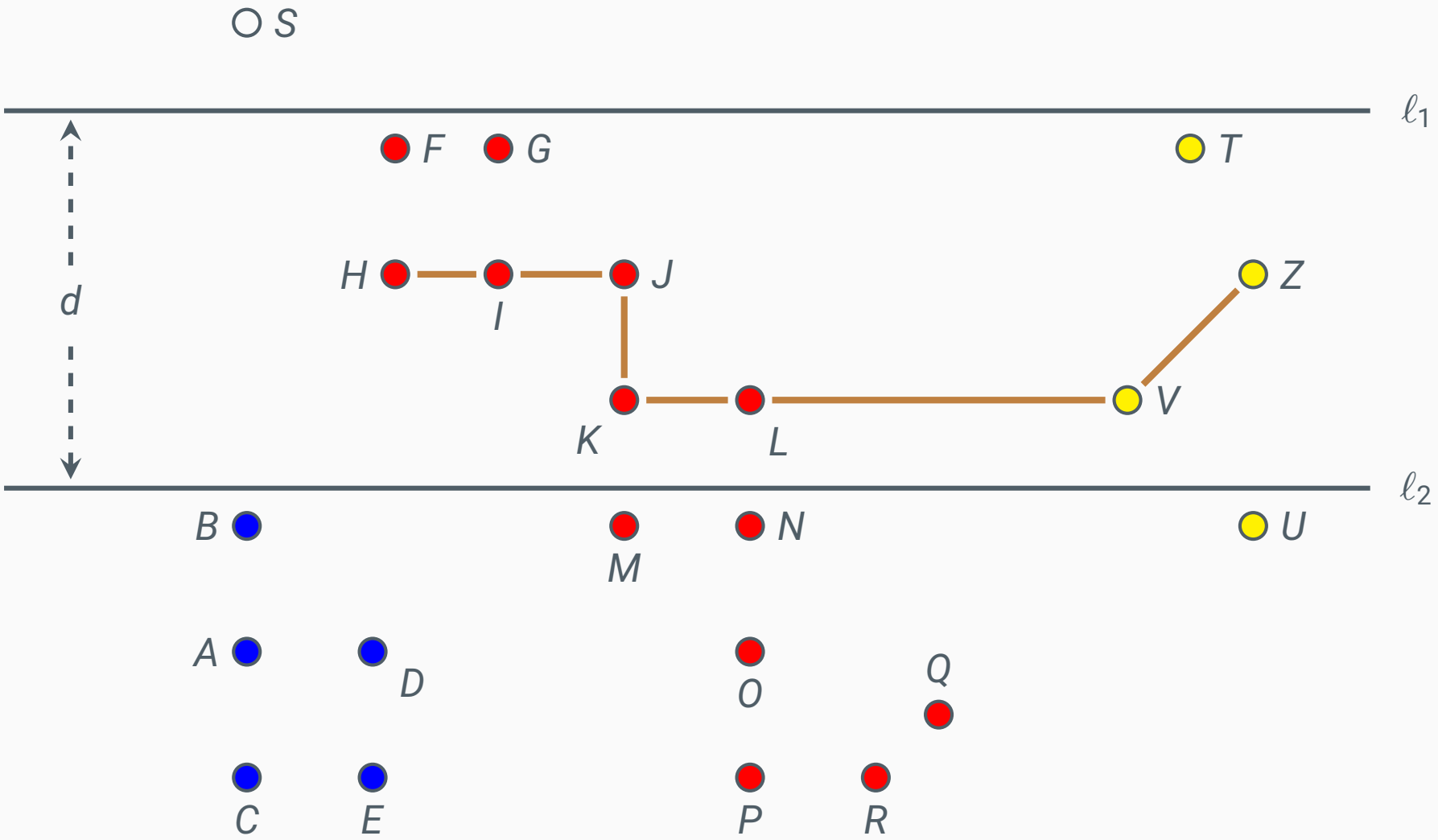
Algoritmo: Iteración 4.



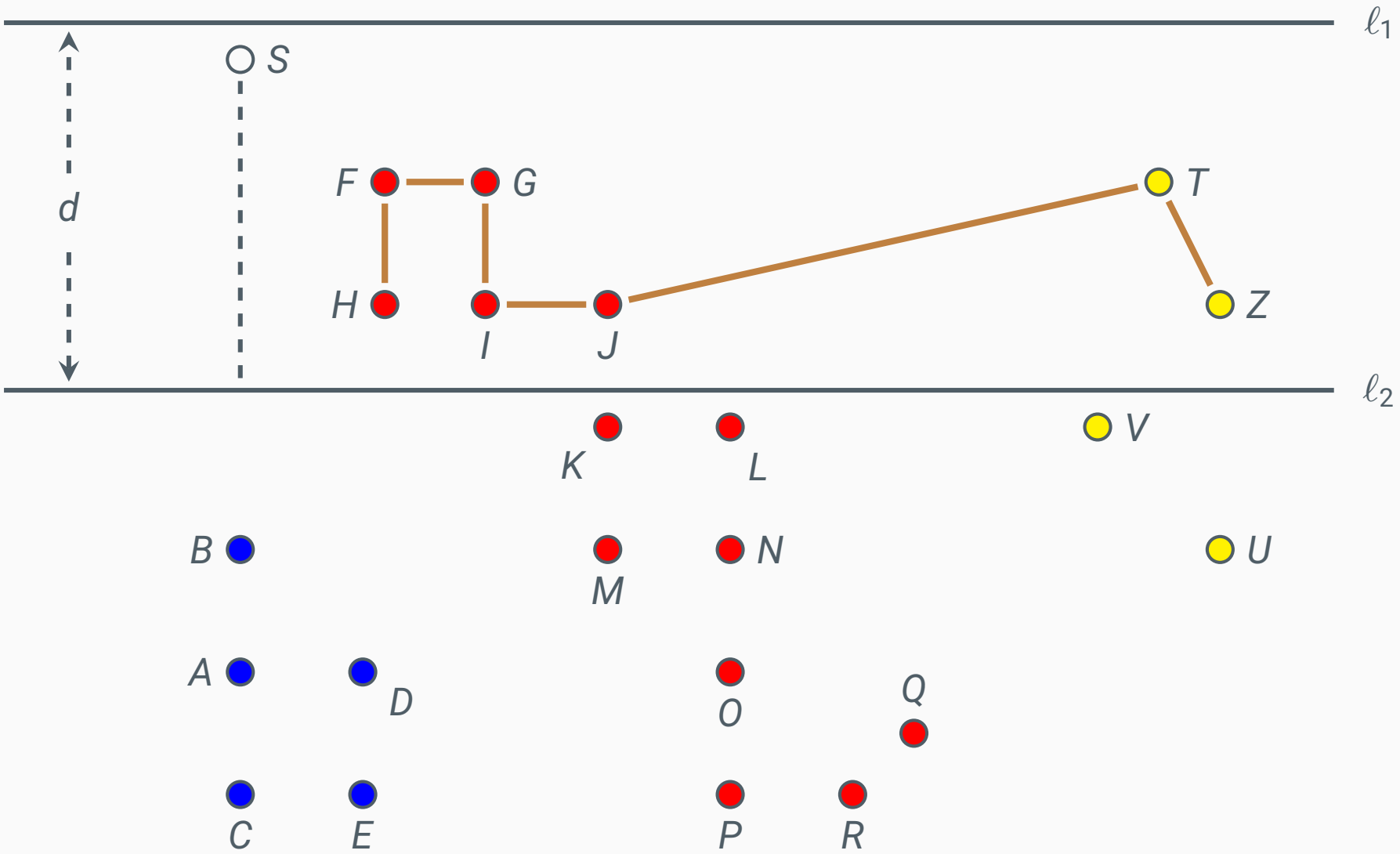
Algoritmo: Iteración 4.



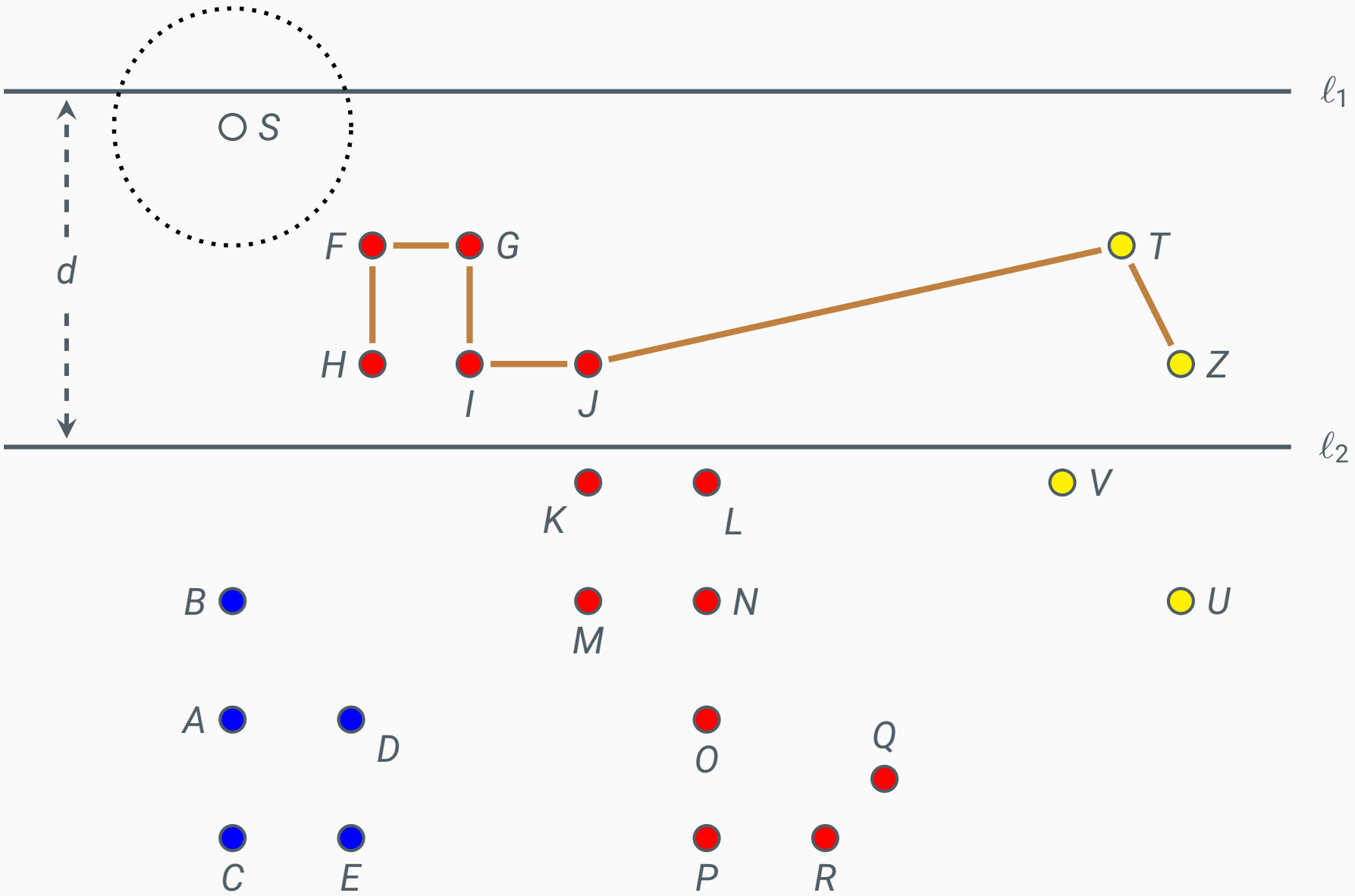
Algoritmo: Iteración 4.



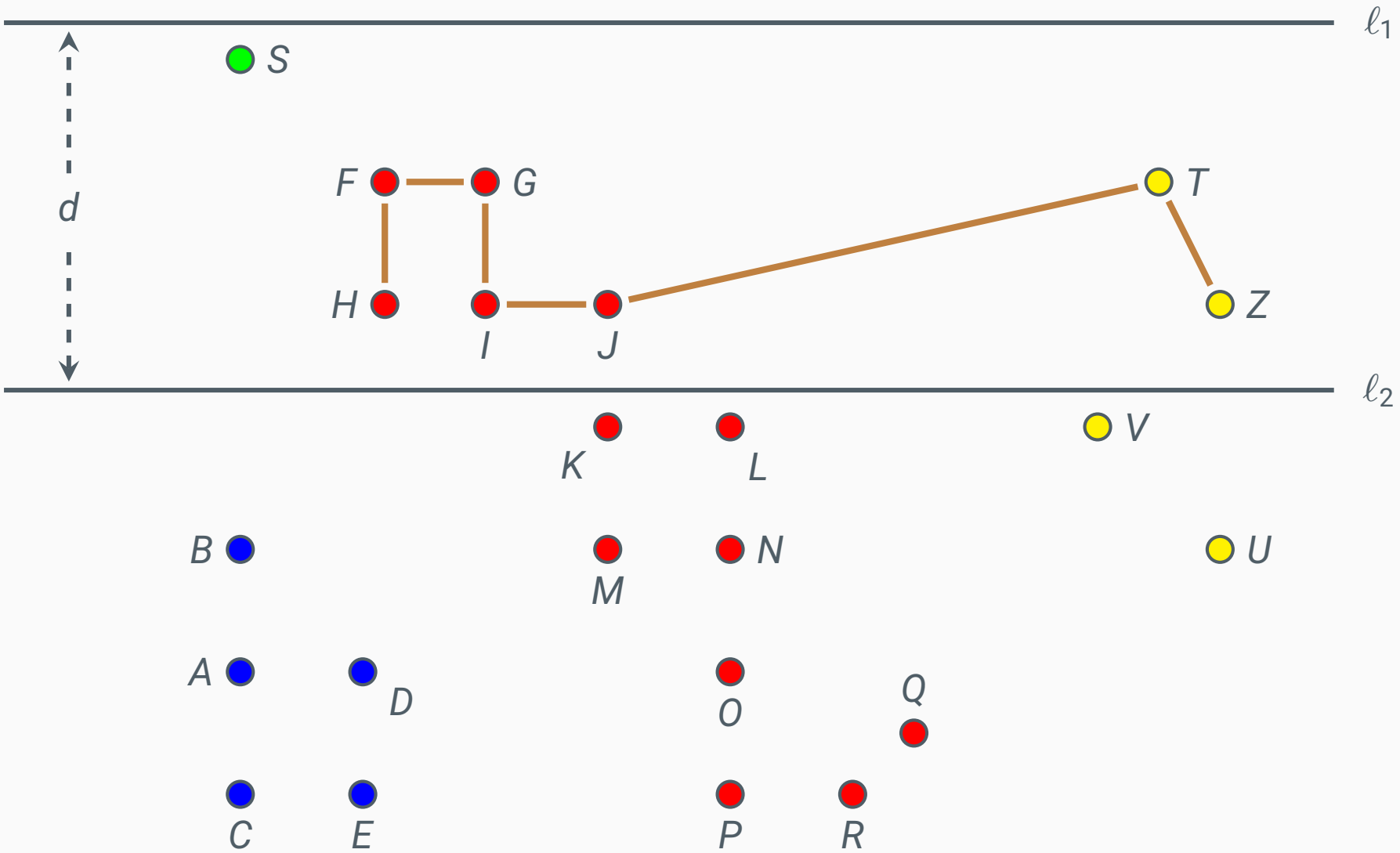
Algoritmo: Iteración 5.



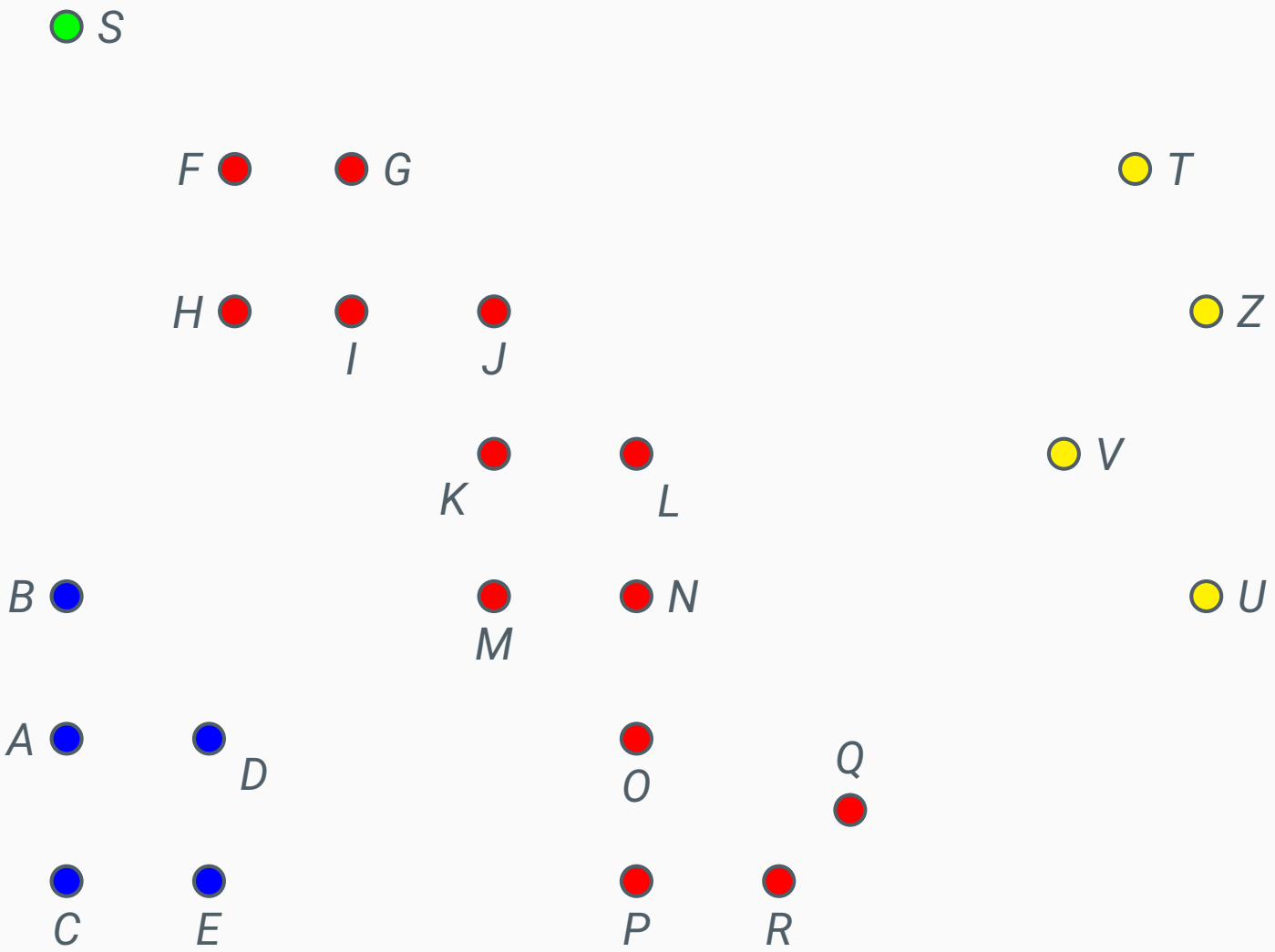
Algoritmo: Iteración 5.



Algoritmo: Iteración 5.



Algoritmo: Iteración 5.



Algoritmo: Especificación.

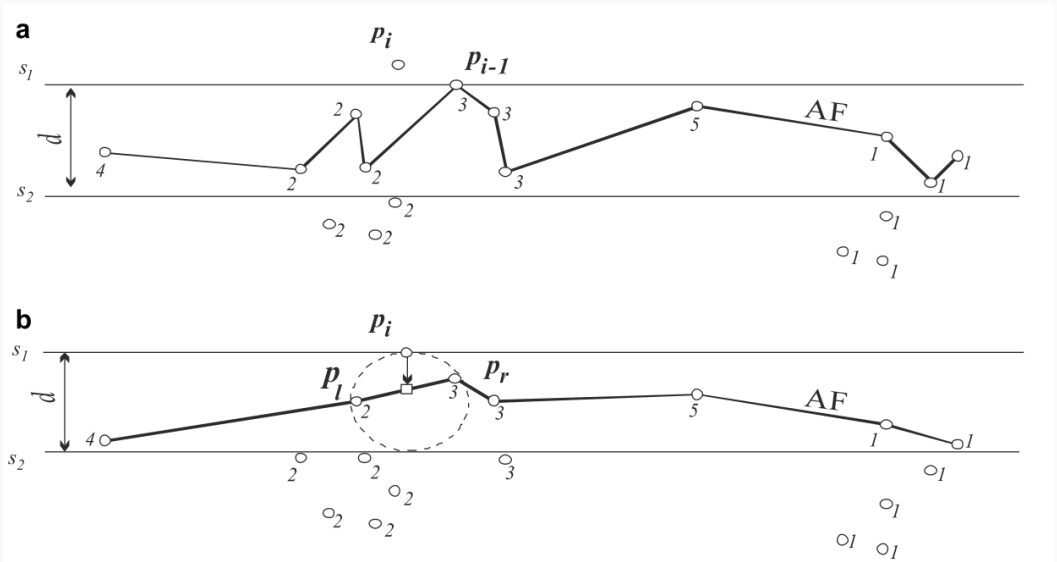
Nuestro algoritmo estara dividido en 3 fases:

1. **Inicialización**. Los puntos de entrada son ordenados con respecto a la dirección del movimiento de barrido.
2. **Barrido**. Los grupos se construyen durante esta face, por medio de dos líneas de barrido s_1 y s_2.
3. **Finalización**. Debemos unir los conglomerados.

Algoritmo: Barrido I.

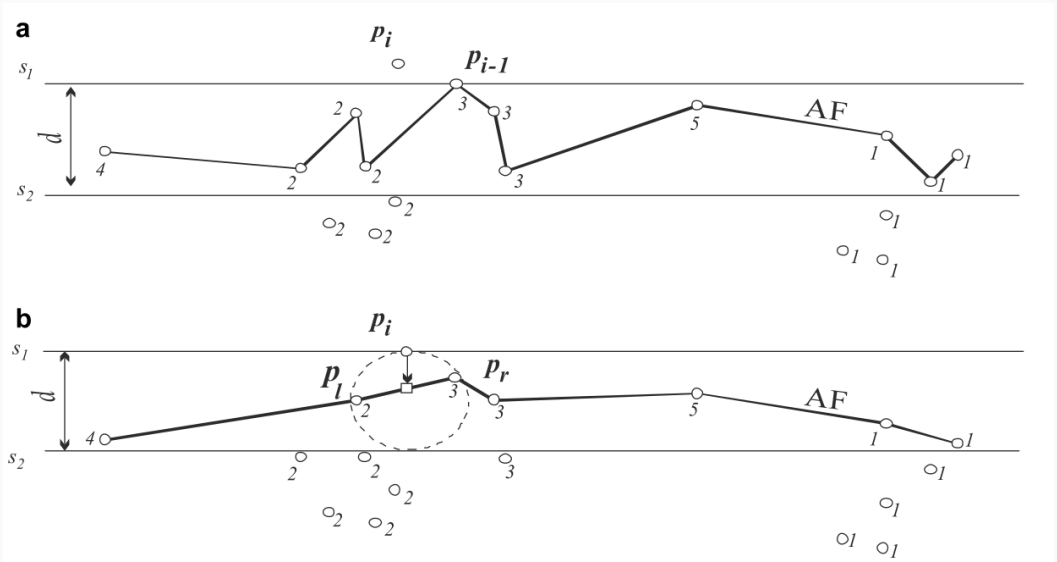
Barrido de línea. Con s_1 y s_2 líneas, supongamos que estamos en la *i*-ésima iteración, entonces:

1. Los puntos entre ambas líneas de barrido se enlazan según sus coordenadas *X*, para formar una polilínea denominada FRENTE DE AVANCE (AF).
2. Todos los puntos que han pasado por s_1 ya están contenidos en grupos C_i de acuerdo con el parámetro de proximidad *d*.
3. Los puntos que han sido barridos por s_2 se eliminan de *AF*.



i-ésima iteración.

Algoritmo: Barrido II.

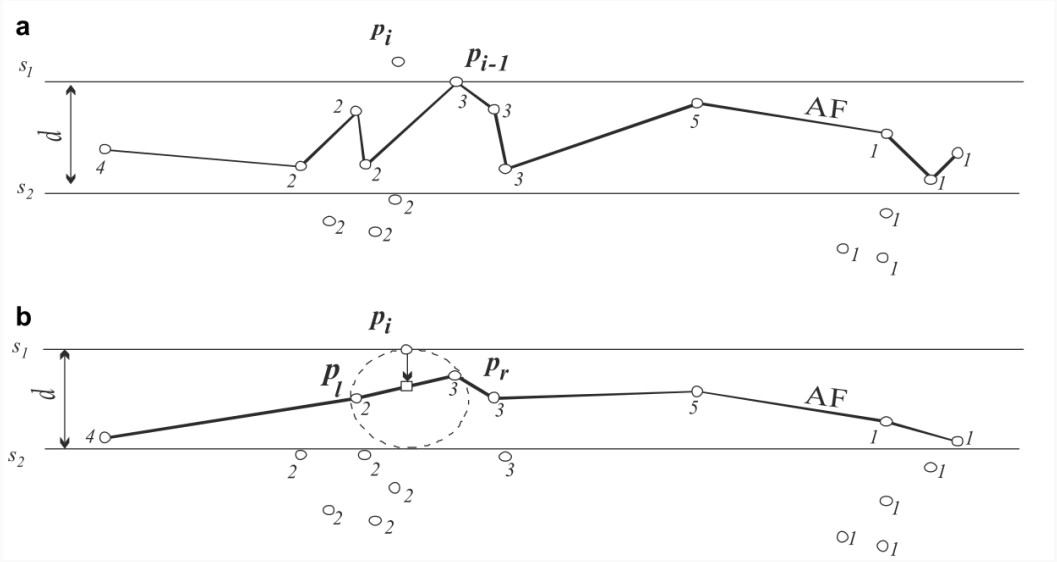


i-ésima iteración.

4. En la siguiente iteración s_1 se mueve al siguiente punto y s_2 la sigue a distancia d.
5. Cuando un punto p_i ingresa a la manga entre las líneas s_1 y s_2, se determina su proyección con AF. Entonces puede pasar que:
   - 5.1 La proyección alcanza el AF.
   - 5.2 La proyección no alcanza el AF.

   Obs. Diremos que $d_l = ||p_i - p_l||$ y $d_l = ||p_i - p_r||$.

Algoritmo: Barrido III.



i-ésima iteración.

5.1 La proyección alcanza el AF. Entonces, sucede que

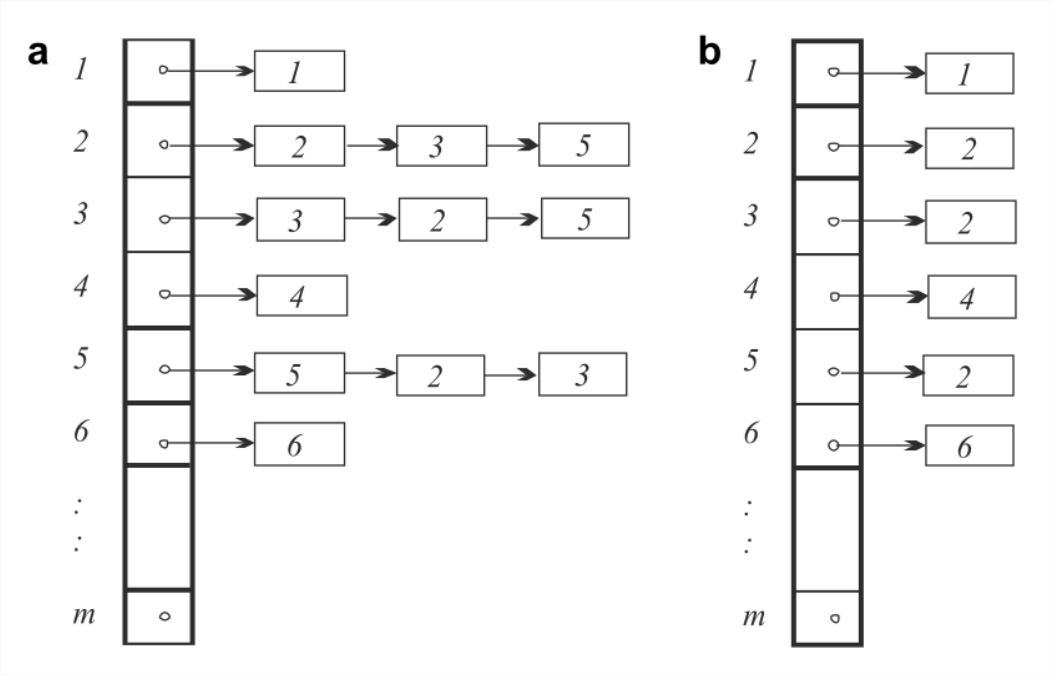
- 5.1.1 Si $d_l > d$ y $d_r > d$. Entonces, p_i forma un nuevo grupo.
- 5.1.2 Si $d_l \leq d$ y $dr > d$. Entonces, p_i forma parte del grupo de p_l.
- 5.1.3 Si $d_l > d$ y $dr \leq d$. Entonces, p_i forma parte del grupo de p_r.
- 5.1.4 Si $d_l \leq d$ y $dr \leq d$. Entonces, p_r, p_l, p_i forman un grupo.

5.2 La proyección no alcanza el AF. Se compara contra el último más cercano en AF, si no pertenece a ningún grupo existente forma uno nuevo.
Obs. Diremos que $d_l = ||p_i - p_l||$ y $d_l = ||p_i - p_r||$.

Algoritmo: Finalización.

Los índices conglomerador, que deben fusionarse se ajustan durante la fase de finalización.

La estructura de datos, que almacena los índices de los grupos es una matriz de listas.



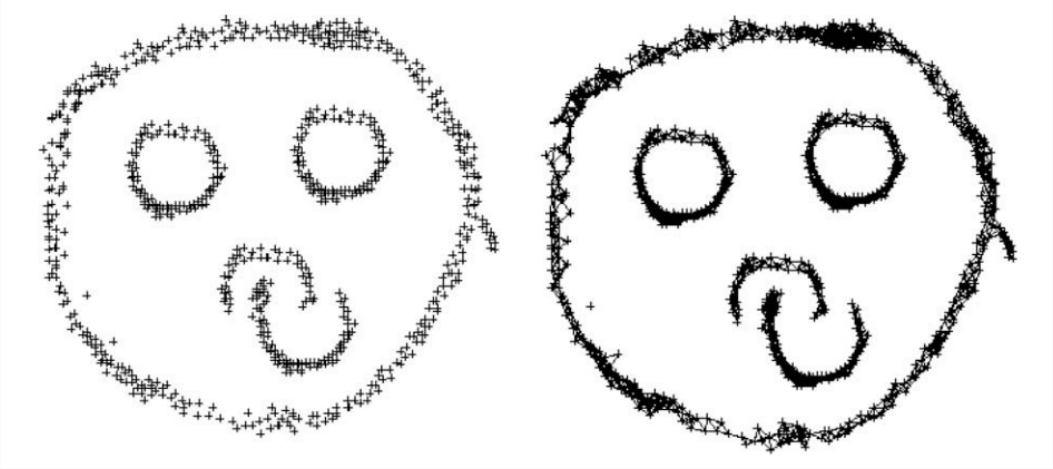
Estructura auxiliar.

En cada lista se conserva el registro que tiene el valor de índices más pequeño, mientras se eliminan.

1. *Inicialización.* Ordenar nuestro conjunto de puntos por comparaciones respecto al sentido de nuestro barrido nos toma $\mathcal{O}(n \log n)$.
2. *Barrido.* Al realizar una proyección para p_i en AF debemos realizar una búsqueda en nuestro conjunto formado por AF (que hereda el orden de nuestro conjunto de puntos). Esto nos toma $\mathcal{O}(\log m)$ donde m es el tamaño de AF. La complejidad total es $\mathcal{O}(n \log n)$.
3. *Finalización.* Fusionar los conglomerados se realiza en $c \cdot n \in \mathcal{O}(n)$.

$\therefore$ La complejidad total del algoritmo esta contenida en $\mathcal{O}(n \log n)$.

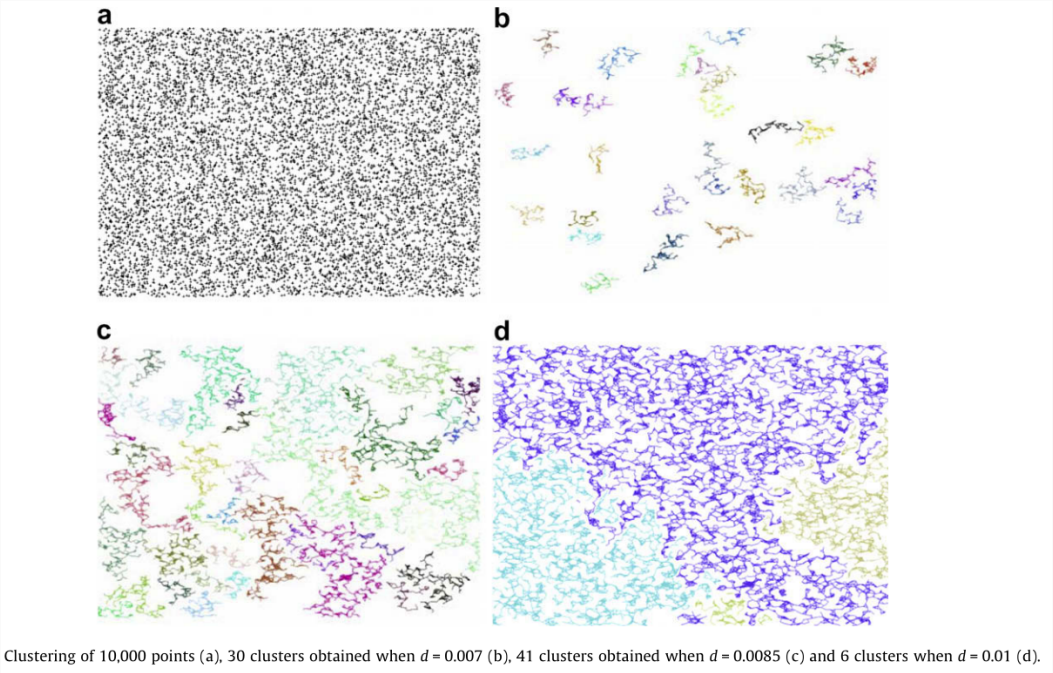
Para un conjunto de datos arbitrarios, nuestro algoritmo de agrupamiento reconoce grupos anidados en una forma arbitraria.



Agrupamiento anidado.

Conjunto con 753 puntos de datos obtenidos al trazar la imagen con MS Paint. *d* fue establecida con respecto al 1% del total de puntos. La cardinalidad se fijó en 1 y por consecuencia existen puntos aislados.

Experimento realizado con $10,000$ puntos de información.



Clustering of 10,000 points (a), 30 clusters obtained when $d = 0.007$ (b), 41 clusters obtained when $d = 0.0085$ (c) and 6 clusters when $d = 0.01$ (d).

Agrupamiento con variación en d.

La cardinalidad de los grupos se fijo en 50 y se vario el parámetro d.

Fin.