

# DIBUJO A TRAZADO CONTINUO BASADO EN EL PROBLEMA DEL AGENTE VIAJERO

CORONADO, CARLOS      PEREZ, MARITZA      ROJAS, ADRIAN  
LARA, HUGO\*

## ABSTRACT:

The traveling salesman problem (TSP) is a classical combinatoric optimization problem. It consists on finding a closed path of minimum length, when connecting a collection of points on the plane. This paper describes how to use the TSP to create continuous drawline images from a picture in digital format. To do it we built TSP instances by manipulating the digital file, defining points on the plane (cities) and simplifying the pixel density in the target image in grayscale format. Then, we solve the instances by using an application which calculates the optimal tours, and so drawing the images.

## RESUMEN:

El problema de agente viajero (TSP, por sus siglas en inglés) es un problema clásico de optimización combinatoria. Consiste en encontrar una trayectoria cerrada de longitud mínima al conectar una colección de puntos en el plano. El presente trabajo describe cómo usar el problema de agente viajero para crear atractivos dibujos a trazos continuos desde una imagen en formato digital. Para ello construimos instancias de problemas de agente viajero, manipulando el archivo digital, definiendo los puntos del plano (ciudades) y simplificando la densidad de pixels en escala de grises de la imagen dada. Luego, resolvemos las instancias usando una aplicación que encuentra los recorridos óptimos en un TSP, y así realizamos el dibujo a trazado continuo.

## I. INTRODUCCIÓN

La mayoría de las personas han realizado dibujos a trazado continuo, bien sea en revistas de pasatiempos o en una clase de dibujo. Generalmente se presentan dibujos sencillos que se obtienen observando la imagen que desea dibujar, o por medio de puntos enumerados para descubrir el dibujo. El propósito es producir el mejor trazado de la imagen o figura sin despegar la punta del lápiz del papel. Cuando se está realizando un dibujo a trazado continuo se está resolviendo un problema de optimización. Muchos problemas de este tipo no pueden ser resueltos con el uso de métodos exactos; es decir, no es posible encontrar su solución óptima con esfuerzos computacionales aceptables, aunque se pueda contar con computadores de alta velocidad operando en paralelo. Esto debido al fenómeno llamado explosión combinatoria, lo cual significa, que cuando crece el número de variables de decisión del problema, el número de decisiones factibles y el costo computacional se incrementan en forma exponencial. Esta clase de problemas combinatorios es llamada NP-completo. El problema del agente viajero (TSP) es de la clase de problemas de decisión NP-Completo. Este problema consiste en que, dada una colección de ciudades en un mapa, el vendedor debe encontrar una ruta para visitar ese conjunto de ciudades, visitando cada una sólo una vez de tal forma que se minimice la distancia total viajada. También se puede ver como, encontrar una trayectoria cerrada de longitud mínima al conectar una colección de puntos en el plano. Desde la década de los años 50 muchos algoritmos han sido desarrollados para encontrar la solución a este problema, encontrando buenas soluciones pero no necesariamente las soluciones óptimas. Desde los años 80 las técnicas de solución se han centrado en la aplicación de metaheurísticas de propósito general, entre las que se pueden mencionar recocido simulado, algoritmos genéticos, colonia de hormigas, búsqueda tabú, etc (ver [4]). También se han aplicado distintas técnicas computacionales como heurísticas evolutivas, redes de Hopfield, entre otras. Éste trabajo de investigación tiene como propósito realizar dibujos a trazado continuo basados en el problema del agente viajero, al usar aplicaciones de manejo de diferentes formatos de imágenes en formato digital a color o en escala de grises, y manipularlas para obtener una versión de la imagen numéricamente manejable que genere instancias de TSP desde dicha imagen para resolverlas. Del mismo modo aplicar las heurísticas adecuadas para obtener atractivas imágenes de trazado continuo. En [2] se presenta un modelo de optimización combinatoria para realizar dibujos de trazado continuo basándose en el problema del agente viajero. Nuestro trabajo se inspira en el artículo de Bosh. Nosotros encontramos imágenes con una calidad similar, pero haciendo uso de un número menor de ciudades, lo cual incide favorablemente en el desempeño de los algoritmos que resuelven el problema del agente viajero. Este trabajo forma parte de un proyecto de investigación, cuyo objetivo principal es la aplicación de heurísticas en la resolución de problemas de optimización combinatoria. La siguiente sección describe el procedimiento para construir las instancias del agente viajero. La sección 3 se dedica a mostrar algunas imágenes generadas por nuestro algoritmo, y la última propone algunas conclusiones, recomendaciones y enfoques futuros.

---

\*Universidad Centrocidental "Lisandro Alvarado", Decanato de Ciencias y Tecnología, Maestría en Optimización

## II. EL PROCEDIMIENTO

Para crear un dibujo de trazado continuo de una imagen digital, se siguen las siguientes instrucciones, de manera similar a Bosh [2]:

Paso 1: Formato de la imagen.

Se utilizaron imágenes digitales en formato jpg que fueron transformadas a escala de grises (una imagen de escala de grises es una imagen que contiene hasta 255 tonos de gris). La imagen de  $km \times kn$  pixels es almacenada en una matriz. Las entradas de dicha matriz son números entre 0 (completamente claro) y 255 (completamente oscuro).

Paso 2: División la matriz de la imagen en cuadrados.

Se divide la imagen en  $m \times n$  cuadrados de  $k \times k$  pixels cada uno. Para cada fila  $i$  y columna  $j$ , se genera la media  $\mu_{i,j}$  de la escala de grises de los píxeles en los cuadrados  $(i, j)$ .

Sea  $\lambda$  el número máximo de puntos (ciudades) designado para cada cuadrado. Se define por  $g_{i,j} = \lambda - \lfloor \lambda \mu_{i,j} / 256 \rfloor$  al número de puntos (ciudades) asignado a cada cuadrado  $(i, j)$ .  $g_{i,j}$  representa la cantidad de oscuridad o claridad de cada cuadrado  $(i, j)$  (i.e., si  $g_{i,j}$  toma el valor de 0 el cuadrado es completamente claro, y si toma  $\lambda$  el cuadrado es oscuro).

Paso 3: Construcción de la instancia TSP.

Para cada par  $(i, j)$  se generan  $g_{i,j}$  puntos colocados de manera aleatoria en los cuadrados  $(i, j)$ , los cuales representan las ciudades para resolver la instancia TSP.

Paso 4: Resolución de la instancia TSP.

Se calculan las distancias, usando la distancia euclídea entre los puntos del plano, para completar la data que es requerida por un resolvidor de TSP. Luego se resuelve aproximadamente el problema TSP para usar el recorrido óptimo en el dibujo de trazado continuo.

## III. EL TRAZADO

Las figuras que se presentan a continuación ilustran los pasos anteriormente descritos. La figura 1. ilustra la construcción de una instancia

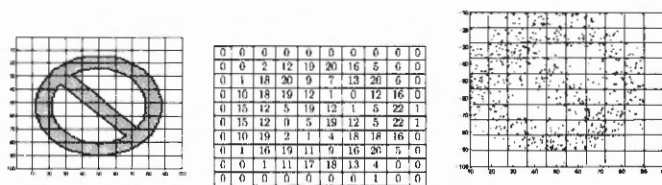


Figura 1: A la izquierda signo de prohibido de  $100 \times 100$  píxeles. En el centro la tabla con la distribución de las 689 ciudades. A la derecha las ciudades calculadas según la tabla con  $k = 10$  y  $\lambda = 50$ .

TSP en un fragmento de una imagen fuente. Luego de haber producido la distribución de las ciudades convenientes para el TSP, estas las llevamos a un programa que hace el recorrido del viaje de TSP, entre todos puntos calculados en los diferentes cuadros. El resultado es una clase de trayectoria cerrada que se asemeje a la imagen original (o imagen fuente). A través del Solver de Concorde (software que resuelve TSP [1]) se produce el viaje final de estas ciudades obtenidas anteriormente. No se intenta encontrar una solución óptima verdadera, pero Concorde produce, con una heurística adecuada, una solución que es bastante aceptable estéticamente. Cabe destacar que la trayectoria que produce nunca se cruza. Una vez obtenido el recorrido, podemos dibujarlo fácilmente como un polígono cerrado personalizando el estilo de la línea. En los anexos mostramos varios ejemplos.

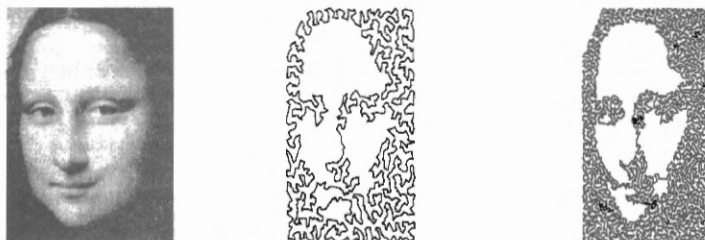


Figura 2: La Mona Lisa de Da Vinci. A la izquierda la imagen original; en el centro la imagen calculada con 1167 ciudades y a la derecha con solo una vez de tal forma que 618 ciudades.



Figura 3: Marilyn Monroe. A la izquierda la imagen original, y a la derecha con 4195 ciudades.



Figura 4: Retrato de Simon Bolivar. A la izquierda la imagen original, en el centro la imagen calculada con 4098 ciudades y a la derecha con 16360 ciudades.

#### IV. CONCLUSIONES

El procedimiento de dibujo a trazado continuo de Bosch y Herman [2] consta de dos etapas: La primera es la construcción de una instancia de problema TSP que proviene de la simplificación de la imagen fuente en escala de grises. Se divide la imagen en cuadrados de tamaño  $k \times k$  y se dibuja en cada cuadrado, en una localización aleatoria dentro del mismo, un número de ciudades que depende de la escala promedio de grises dentro del cuadrado en la imagen fuente. Dos parámetros juegan un papel importante en esta etapa: el tamaño  $k$  de la división, y la densidad de píxeles máxima  $\lambda$  utilizada para calibrar el número de ciudades en cada cuadrado. Con el objetivo de generar imágenes con calidad estética, pero con un tamaño de problema TSP manejable, se debe jugar con estos parámetros. Si  $k$  es muy grande, aunque implique reducción de la complejidad, se pierden detalles de la imagen fuente, ya que las ciudades serán colocadas en una posición aleatoria dentro del cuadrado, distorsionando la información local original. Se desea entonces mantener los cuadrados pequeños ( $k = 2$  parece ser buen número). Ahora, con  $\lambda$  pequeño resultan imágenes con pocas ciudades, pero se corre el riesgo de que muchas áreas queden sin ciudades que representen mejor las imágenes. Al aumentar  $\lambda$  crece también la posibilidad de colocar ciudades en cada cuadrado, pero si  $\lambda$  es excesivamente grande, entonces los cuadrados quedan muy llenos y no se distinguen los detalles. Es conveniente entonces colocar  $\lambda$  de manera que no exceda el número de píxeles en el cuadrado ( $\lambda \leq k$ ). Por la experimentación,  $\lambda = k$  es un nivel que no hace crecer mucho el problema, a su vez produce imágenes estéticamente aceptables. La colocación aleatoria de las ciudades dentro de cada cuadrado no es la única posibilidad que se tiene. Se puede generar un procedimiento que organice las ciudades a ser colocadas en cada cuadrado, y así mejorar la calidad estética de la imagen. Como trabajo futuro, planeamos generar un procedimiento que coloque las ciudades con regularidad geométrica, y no en lugar generado al azar.

La segunda etapa consiste en resolver la instancia de TSP que produce el trazado continuo. Es conocido que el problema del agente viajero es NP-duro (ver [4] por ejemplo). Esto significa que es difícil de resolver, y su complejidad crece con el número de ciudades. En algunos de los problemas utilizados, el número de ciudades llega hasta cerca de 16000. Los problemas de TSP con este número de ciudades son considerados grandes, y tratar de resolverlos de manera exacta es utópico. En el presente trabajo usamos heurísticas disponibles en la aplicación Concorde [1]. En algunas de ellas, como la estrategia voraz, ni con los problemas medianos (5000 ciudades) daba respuesta. La estrategia que se mostró más robusta para nuestras instancias fué Lin Kernigham [1] la cual prioriza búsquedas locales, produciendo trazados que tienden a unir las ciudades cercanas, y así trazados estéticamente agradables. En proyecto tenemos la utilización de otras estrategias de solución para problemas TSP, como algoritmos evolutivos, ya que el tamaño de nuestros problemas son retadores para cualquier estrategia.