



Introducción a Redes y Problema de Ruta más Corta

Universidad Michoacana de San Nicolás de Hidalgo
Facultad de Contaduría y Ciencias Administrativas
Dr. Víctor G. Alfaro-García

De Programación Lineal a Modelos de Redes

PROGRAMACIÓN LINEAL
(Semanas 2-6):

- Formulación algebraica
- Variables continuas
- Solver resuelve con Simplex
- Abstracto, simbólico

MODELOS DE REDES
(Semanas 7-11):

- Representación gráfica
- Flujos en arcos
- Algoritmos especializados
- Visual, intuitivo

CONEXIÓN IMPORTANTE:

Los modelos de redes SON casos especiales de Programación Lineal, pero tienen estructura tan particular que existen algoritmos mucho más eficientes que el Simplex general.

Podríamos resolver todo con Solver, pero los algoritmos especializados son órdenes de magnitud más rápidos para problemas grandes.

Hillier & Lieberman: Capítulo 9, Secciones 9.1-9.3

¿Qué es una Red?

Una RED es un conjunto de puntos conectados por líneas:

NODOS (vértices):

- **Representan ubicaciones, ciudades, almacenes, eventos, estados**
- **Se dibujan como círculos**

ARCOS (aristas):

- **Representan conexiones, rutas, canales, relaciones, flujos**
- **Se dibujan como flechas**

EJEMPLOS COTIDIANOS DE REDES:

- Mapa de carreteras: ciudades (nodos) + carreteras (arcos)
- Red de distribución: almacenes (nodos) + rutas de envío (arcos)
- Internet: computadoras (nodos) + conexiones (arcos)
- Organigrama: personas (nodos) + relaciones de reporte (arcos)
- Planeación de proyecto: actividades (nodos) + precedencias (arcos)
- Red eléctrica: subestaciones (nodos) + líneas de transmisión (arcos)

Si pueden dibujarlo con círculos y flechas, es una red.

Terminología Fundamental de Redes

RED DIRIGIDA:
Arcos tienen dirección (flechas)
Ej: calles de un solo sentido

RED NO DIRIGIDA:
Arcos sin dirección (líneas)
Ej: carreteras bidireccionales

LONGITUD/COSTO DE ARCO:
Número asociado a cada arco
Ej: distancia (km), tiempo (min), costo ($), capacidad (ton)

CAMINO (path):
Secuencia de arcos conectados desde nodo origen a destino

LONGITUD DE CAMINO:
Suma de longitudes de arcos en el camino

NODO ORIGEN:
Punto de inicio

NODO DESTINO:
Punto final

NODOS INTERMEDIOS:
Nodos entre origen y destino

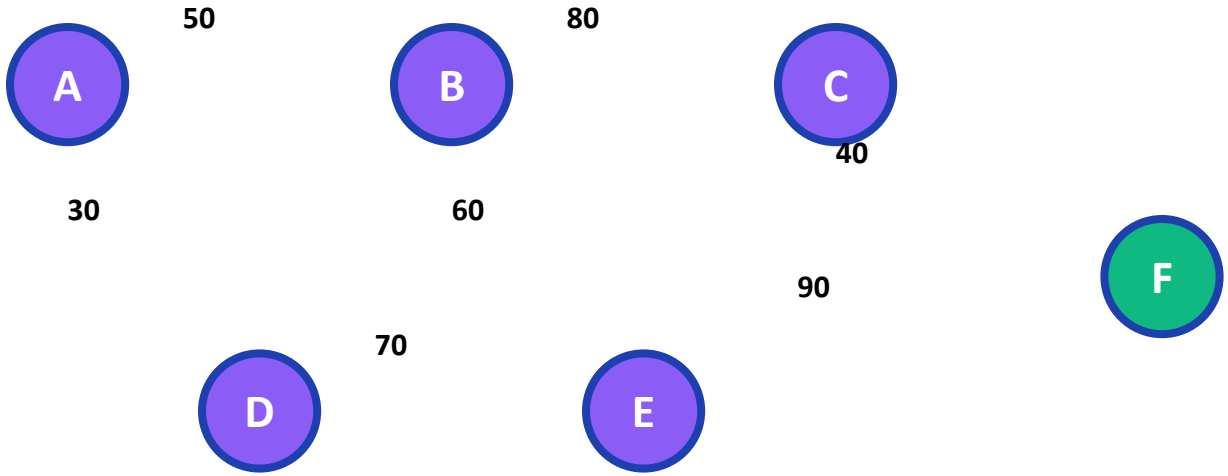
RUTA MÁS CORTA:
Camino con longitud mínima entre origen y destino

RED CONEXA:
Existe al menos un camino entre cualquier par de nodos

Esta terminología es universal en teoría de redes

Ejemplo: Red de Transporte entre Ciudades

Red con 6 ciudades, distancias en kilómetros:



PREGUNTA: ¿Cuál es la ruta más corta de A a F?

Caminos posibles: A→B→C→F, A→D→E→F, A→B→E→F, etc.
¿Cuál tiene la menor distancia total?

Problema de Ruta Más Corta

DEFINICIÓN FORMAL:

Dada una red con longitudes no negativas en los arcos, encontrar el camino desde un nodo origen hasta un nodo destino que tenga la mínima longitud total.

APLICACIONES EMPRESARIALES:

LOGÍSTICA:

- Ruta de entrega más corta
- Minimizar tiempo de viaje
- Reducir consumo de combustible
- Planificación de rutas de flotilla

OTRAS ÁREAS:

- Telecomunicaciones: ruteo de paquetes en internet
- Manufactura: secuenciación óptima de tareas
- Finanzas: conversión de divisas con menor costo
- Proyectos: ruta crítica (veremos después)

Es uno de los problemas más fundamentales y útiles en redes

Algoritmo de Dijkstra: La Solución Sistemática

Desarrollado por Edsger Dijkstra en 1956. Es el método estándar para ruta más corta.

IDEA CENTRAL DEL ALGORITMO:

Construir progresivamente rutas más cortas desde el origen,
expandiéndose como ondas desde el centro,
marcando permanentemente cada nodo cuando se encuentra su ruta óptima.

Es un algoritmo GREEDY (codicioso): en cada paso, elige la mejor opción local.

GARANTÍA MATEMÁTICA:

Si todas las longitudes son NO NEGATIVAS, Dijkstra siempre encuentra la ruta más corta. Es óptimo y eficiente (complejidad $O(n^2)$ o mejor).

HOY APRENDERÁN:

✓ La LÓGICA del algoritmo (conceptual, no memorizar pasos mecánicos)
✓ EJECUTARLO a mano en problema pequeño (para entender cómo funciona)
✓ INTERPRETAR resultados (no solo números, sino insights de red)
✓ Cuándo y cómo usarlo en Excel Solver (para problemas más grandes)

Algoritmo de Dijkstra: Pasos (Conceptual)

PASO 0 (Inicialización):
- Marcar distancia del origen = 0
- Marcar distancia de todos los demás nodos = ∞ (infinito)
- Todos los nodos están SIN RESOLVER

PASO 1 (Selección):
- Entre los nodos SIN RESOLVER, elegir el que tenga menor distancia
- Marcarlo como RESUELTO (permanente)

PASO 2 (Actualización):
- Para cada vecino SIN RESOLVER del nodo recién marcado:
  - Calcular distancia alternativa: distancia_actual + longitud_arco
  - Si esta distancia es MENOR que la registrada, actualizarla

PASO 3 (Repetición):
- Regresar al PASO 1 hasta que el nodo destino esté RESUELTO

PASO 4 (Reconstrucción):
- Reconstruir el camino siguiendo los pasos hacia atrás desde destino

Parece complicado, pero es muy intuitivo al ejecutarlo

Dijkstra Paso a Paso: Inicialización e Iteración 1

Problema: Encontrar ruta más corta de A a F

INICIALIZACIÓN (Paso 0):

Distancias:
- **A: 0 (origen)**
- **B, C, D, E, F: ∞**

Todos sin resolver

ITERACIÓN 1:

Elegir nodo con menor distancia:
A (distancia 0) ✓

Marcar A como RESUELTO

Actualizar vecinos de A:
- **B: 0+50 = 50**
- **D: 0+30 = 30**

```
ESTADO DESPUÉS DE ITERACIÓN 1:

Nodos RESUELTOS: A

Distancias actuales:
  A: 0 ✓  |  B: 50  |  C: ∞  |  D: 30  |  E: ∞  |  F: ∞

Ruta conocida a B: A→B (50)
Ruta conocida a D: A→D (30)
```

Siguiente: elegir D (menor distancia entre sin resolver: 30 < 50)

Dijkstra Paso a Paso: Iteraciones 2-3

ITERACIÓN 2:

Elegir D (distancia 30) ✓
Marcar D como RESUELTO
Actualizar vecinos de D:
B: min(50, 30+40) = 50 (no mejora)
E: ∞ → 30+60 = 90

ITERACIÓN 3:

Elegir B (distancia 50) ✓
Marcar B como RESUELTO
Actualizar vecinos de B:
C: ∞ → 50+80 = 130
E: min(90, 50+70) = 90 (no mejora)

```
ESTADO DESPUÉS DE ITERACIÓN 3:

Nodos RESUELTOS: A, D, B

Distancias actuales:
  A: 0 ✓  |  B: 50 ✓  |  C: 130  |  D: 30 ✓  |  E: 90  |  F: ∞

Mejor ruta conocida a E: A→D→E (90)
Mejor ruta conocida a C: A→B→C (130)
```

OBSERVACIÓN CLAVE:

Cada vez que marcamos un nodo como resuelto, esa distancia es ÓPTIMA y nunca cambiará. El algoritmo garantiza esto matemáticamente.

Dijkstra: Iteraciones Finales y Solución

ITERACIÓN 4:

Elegir E (distancia 90) ✓
Marcar E como RESUELTO
Actualizar vecinos de E:
C: min(130, 90+25) = 115 ✓ mejora
F: ∞ → 90+40 = 130

ITERACIÓN 5:

Elegir C (distancia 115) ✓
Marcar C como RESUELTO
Actualizar vecinos de C:
F: min(130, 115+90) = 130 (no mejora)

SOLUCIÓN FINAL:
Ruta más corta de A a F:
A → D → E → F
Distancia total: 130 km
(A→D: 30 + D→E: 60 + E→F: 40 = 130)
Esta es la ruta óptima garantizada por Dijkstra.

¡Hemos encontrado la ruta más corta sistemáticamente!

Por Qué Dijkstra es Tan Poderoso

✓ **VENTAJAS:**

- Siempre encuentra óptimo (si longitudes ≥ 0)
- Eficiente: $O(n^2)$ básico, $O(n \log n)$ optimizado
- Encuentra rutas a TODOS los nodos, no solo destino
- Conceptualmente simple
- Implementación directa

❌ **LIMITACIONES:**

- Requiere longitudes NO NEGATIVAS
- Si hay arcos negativos, usar Bellman-Ford
- Calcula desde un origen fijo
- Para todos-a-todos, usar Floyd-Warshall

BONUS: Dijkstra encuentra rutas óptimas a TODOS los nodos
En nuestro ejemplo, una sola ejecución de Dijkstra desde A nos dio:

A → A: 0 km | A → D: 30 km | A → E: 90 km
A → B: 50 km | A → C: 115 km | A → F: 130 km

Si mañana necesitamos la ruta de A a C, ya la tenemos sin recalcular.
Esto es muy valioso en aplicaciones prácticas.

Usar Dijkstra en Excel Solver (Problemas Grandes)

Para redes pequeñas (5-10 nodos): ejecutar Dijkstra a mano es rápido y didáctico.
Para redes grandes (50+ nodos): usar Excel Solver es más práctico.

FORMULACIÓN COMO PROBLEMA DE PL:

Variables: x_{ij} = flujo en arco (i,j) [0 si no se usa, 1 si se usa en la ruta]

Objetivo: Minimizar Σ(distancia_ij × x_{ij})

Restricciones:
- Flujo que sale del origen = 1
- Flujo que entra al destino = 1
- Para nodos intermedios: flujo_entrada = flujo_salida (conservación)

EN SOLVER:

1. Definir variables binarias para cada arco (0 o 1)
2. Función objetivo = suma de (distancia × variable)
3. Restricciones de conservación de flujo en cada nodo
4. Resolver con Simplex LP (o Solver automáticamente reconoce que es red)

Solver encuentra la misma solución que Dijkstra a mano, pero en milisegundos incluso para redes con miles de arcos.

Aplicación Práctica: Optimización de Rutas de Entrega

CASO REAL: Una empresa de paquetería necesita encontrar la ruta más rápida desde su centro de distribución en Morelia hasta un cliente en Guadalajara, considerando tiempos de viaje por diferentes carreteras y tráfico.

MODELADO DEL PROBLEMA:

- Nodos = ciudades intermedias (Morelia, Zamora, La Barca, León, etc.)
- Arcos = carreteras disponibles
- Longitudes = tiempo promedio de viaje (minutos)
- Origen = Morelia
- Destino = Guadalajara

VALOR EMPRESARIAL:
✓ Minimizar tiempo de entrega → mayor satisfacción del cliente
✓ Reducir kilómetros recorridos → menor costo de combustible
✓ Optimizar uso de flotilla → mayor eficiencia operativa
✓ Actualizar rutas con tráfico en tiempo real → adaptabilidad
Empresas como UPS y FedEx resuelven millones de estos problemas diariamente.

Otras Aplicaciones del Problema de Ruta Más Corta

TELECOMUNICACIONES:

- Ruteo de paquetes en internet
- Minimizar latencia de transmisión
- Balanceo de carga en servidores
- Diseño de redes de fibra óptica

FINANZAS:

- Arbitraje de divisas
- Encontrar secuencia de conversiones con menor costo
- Optimización de portafolio de inversiones
- Planificación de flujos de efectivo

MANUFACTURA:

- Secuenciación de operaciones
- Minimizar tiempo de ciclo
- Ruteo de materiales en planta
- Planificación de mantenimiento

PROYECTOS (CPM/PERT):

- Encontrar ruta crítica
- Identificar actividades que no pueden retrasarse
- Calcular holguras
- (Lo veremos en detalle después)

El problema de ruta más corta aparece disfrazado en muchas áreas

Recursos y Próxima Semana

RECURSOS DISPONIBLES:

📚 Lectura obligatoria: Hillier & Lieberman, Cap. 9, Secc. 9.1-9.3
📼 Video tutorial: Dijkstra paso a paso (en plataforma)

SEMANA 8 - ADELANTO:

Problema de Flujo de Costo Mínimo
- **Modelo de transporte**
- **Conservación de flujo**
- **Aplicaciones en logística y distribución**
- **Problemas de asignación y transporte multiproducto**

Practiquen Dijkstra esta semana. Es fundamental.

Recapitulación y Cierre

HOY APRENDIMOS:

- ✓ Qué es una red (nodos y arcos) y su representación gráfica
- ✓ Terminología fundamental de redes
- ✓ El problema de ruta más corta y sus aplicaciones
- ✓ Algoritmo de Dijkstra conceptualmente
- ✓ Ejecución manual paso a paso
- ✓ Cómo usar Solver para redes grandes
- ✓ Aplicaciones reales en logística, telecom, finanzas, manufactura

Los problemas de redes son visualmente intuitivos pero matemáticamente rigurosos.

Dijkstra es elegante, eficiente, y profundamente útil.

¿Dudas o preguntas?

Dr. Víctor G. Alfaro-García • victor.alfaro@umich.mx