

# **Herramientas para el diseño y análisis de redes de transporte urbano de pasajeros**

Tema 6: Redes de transporte privado (cont.)

## Óptimo del sistema (SO)

- Formulación similar a UE, pero se minimiza el tiempo total de viaje en la red. Es otro problema.
- La solución no necesariamente es consistente con las hipótesis de UE, el valor objetivo puede ser diferente.
- Representa una situación que podría considerarse ideal (más ventajosa para toda la sociedad), pero difícil de ser alcanzada dado el comportamiento individual de cada conductor.
- SO asume que algunos usuarios deciden no mejorar su tiempo de viaje, para contribuir a la disminución del tiempo total de viaje en la red.

## Formulación del SO

$$\min \sum_{a \in A} x_a t_a(x_a)$$

$$\text{s.a.} \quad \sum_{p \in P_k} f_p^k = R_k \quad \forall k \in K,$$

$$f_p^k \geq 0 \quad \forall p \in P_k, k \in K,$$

$$x_a = \sum_k \sum_p f_p^k \delta_{ap}^k \quad \forall a \in A.$$

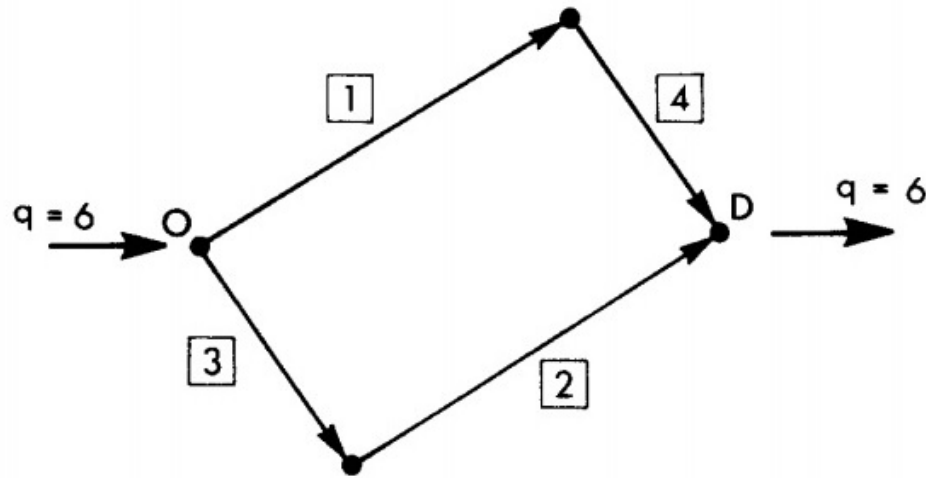
# Óptimo del sistema

- SO también tiene solución única respecto a los flujos en arcos.
- SO puede utilizarse para obtener una cota del UE.
- Cuando el tiempo de viaje en los arcos es independiente de los flujos, UE y SO coinciden:
  - Notar que si  $t_a(x_a) = t'_a$ , la función objetivo de SO es  $\sum_{a \in A} x_a t'_a$  y la de UE es  $\sum_{a \in A} \int_0^{x_a} t'_a d\omega = \sum_{a \in A} x_a t'_a$ .
  - El problema se reduce al del cálculo del camino de costo mínimo.
  - Para cada par OD  $k \in K$ , todo el flujo  $R_k$  seguirá el mismo camino.

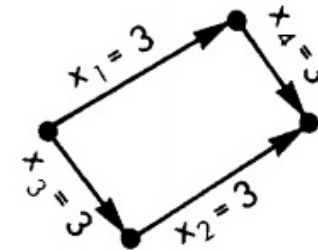
## Paradoja de Braess

- Aumento de infraestructura puede empeorar el desempeño global del sistema.
- Si el sistema se optimiza bajo las hipótesis del SO, luego en realidad se va a comportar según las de UE.
- Ejemplo sencillo con un par OD conectado por dos caminos y cuatro arcos. Se considera agregar un quinto arco a la red.

# Paradoja de Braess, ejemplo



User Equilibrium Solution



Performance Data

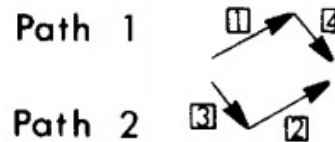
$$t_1(x_1) = 50 + x_1$$

$$t_2(x_2) = 50 + x_2$$

$$t_3(x_3) = 10x_3$$

$$t_4(x_4) = 10x_4$$

Path Definition

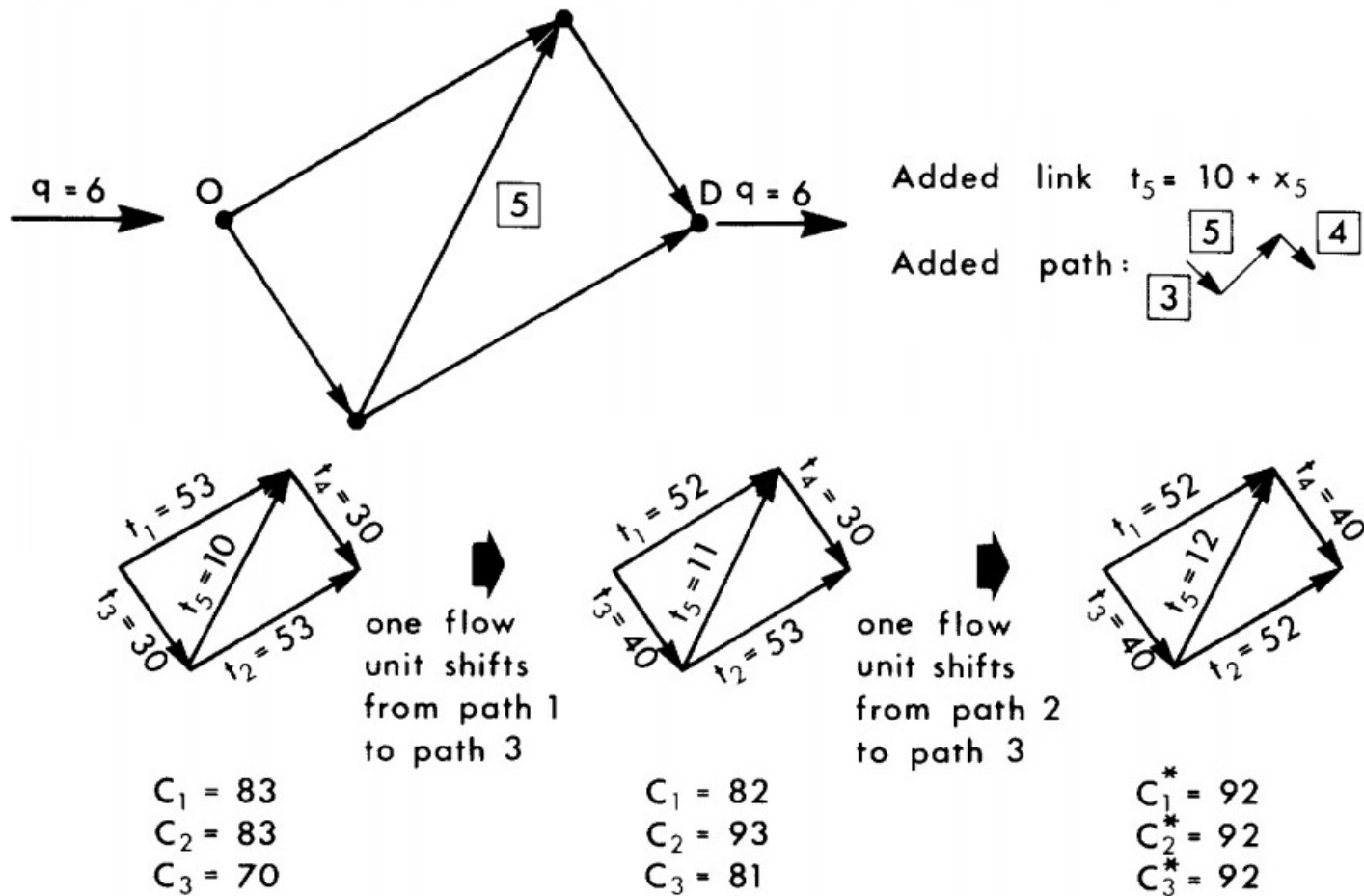


## Paradoja de Braess, ejemplo

Antes de agregar el quinto arco:

- El flujo total (6) se divide (3+3) entre dos caminos con tiempo de viaje igual a 83 unidades de tiempo.
- El tiempo total en la red (flujo por tiempo, función objetivo del SO) es de 498 unidades.
- El valor objetivo del UE es 399.

# Paradoja de Braess, ejemplo





## Paradoja de Braess, ejemplo

Después de agregar el quinto arco:

- El flujo total (6) se divide ( $2+2+2$ ) entre tres caminos con tiempo de viaje igual a 92 unidades de tiempo ( $> 83$ ).
- El tiempo total en la red (flujo por tiempo, función objetivo del SO) es de 552 unidades ( $> 498$ ).
- El valor objetivo del UE es 386 ( $< 399!$ ).
- El desempeño del sistema empeora debido a una redistribución de la demanda existente (no se considera demanda inducida).

## Paradoja de Braess, implicancias prácticas

- No todo incremento de capacidad necesariamente redundando en mejoras en el desempeño del sistema, de hecho, la situación podría empeorar.
- El estado de la práctica reconoce que ciertas restricciones en las decisiones de viaje y reducciones de capacidad pueden llevar a una mejor distribución de flujos.
- Ejemplo: semáforos con tiempos prolongados para entrar a una vía rápida (pero congestionada), causa que algunos conductores busquen caminos alternativos, lo que mejora la situación global de la red.

## Resolución computacional del UE

- Métodos heurísticos (restricción de capacidad, asignación incremental) y en base a la formulación matemática (combinación convexa).
- Carga de la red: asignar flujos a los arcos para ciertos valores (constantes) de tiempo de viaje, bajo la hipótesis de que los usuarios siguen los caminos más cortos.
- Algunos métodos heurísticos pueden producir soluciones que oscilan.

## Algoritmo de asignación incremental

**Paso 0:** Inicialización. Dividir cada demanda OD  $R_k$  en  $N$  porciones iguales (setear  $R_k^n = R_k/N$ ). Setear contador de iteraciones  $n = 1$  y flujos iniciales  $x_a^0 = 0, \forall a$ .

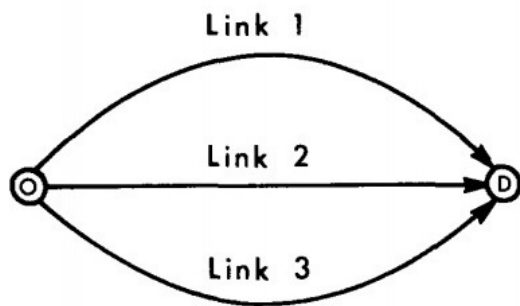
**Paso 1:** Actualización. Setear  $t_a^n = t_a(x_a^{n-1}), \forall a$ .

**Paso 2:** Carga incremental. Asignar la porción  $R_k^n$  para todo  $k$  al camino más corto según tiempo constante en los arcos  $t_a^n$ , obteniendo flujos  $w_a^n$ .

**Paso 3:** Incremento de flujo. Setear  $x_a^n = x_a^{n-1} + w_a^n, \forall a$ .

**Paso 4:** Condición de parada. Si  $n = N$  parar, si no setear  $n = n + 1$  y ejecutar paso 1.

## Algoritmo de asignación incremental (cont.)



$$t_1 = 10 \left[ 1 + 0.15 \left( \frac{x_1}{2} \right)^4 \right] \text{ time units}$$

$$t_2 = 20 \left[ 1 + 0.15 \left( \frac{x_2}{4} \right)^4 \right] \text{ time units}$$

$$t_3 = 25 \left[ 1 + 0.15 \left( \frac{x_3}{3} \right)^4 \right] \text{ time units}$$

$$x_1 + x_2 + x_3 = 10 \text{ flow units}$$

Iteration (Increment)	Algorithmic Step	Link		
		1	2	3
1	Update	$t_1^1 = 10$	$t_2^1 = 20$	$t_3^1 = 25$
	Incremental loading	$w_1^1 = 2.5$	$w_2^1 = 0$	$w_3^1 = 0$
	Summation	$x_1^1 = 2.5$	$x_2^1 = 0$	$x_3^1 = 0$
2	Update	$t_1^2 = 14$	$t_2^2 = 20$	$t_3^2 = 25$
	Incremental loading	$w_1^2 = 2.5$	$w_2^2 = 0$	$w_3^2 = 0$
	Summation	$x_1^2 = 5.0$	$x_2^2 = 0$	$x_3^2 = 0$
3	Update	$t_1^3 = 69$	$t_2^3 = 20$	$t_3^3 = 25$
	Incremental loading	$w_1^3 = 0$	$w_2^3 = 2.5$	$w_3^3 = 0$
	Summation	$x_1^3 = 5.0$	$x_2^3 = 2.5$	$x_3^3 = 0$
4	Update	$t_1^4 = 69$	$t_2^4 = 20.5$	$t_3^4 = 25$
	Incremental loading	$w_1^4 = 0$	$w_2^4 = 2.5$	$w_3^4 = 0$
	Summation	$x_1^4 = 5.0$	$x_2^4 = 5.0$	$x_3^4 = 0$
Travel time at convergence		$t_1^* = 69$	$t_2^* = 27.3$	$t_3^* = 25$

## Algoritmo de combinación convexa

- El algoritmo de asignación incremental puede mejorar su desempeño aumentándose el valor de  $N$ , aunque sigue siendo intrínsecamente heurístico, y para redes muy congestionadas es más difícil que logre encontrar el equilibrio.
- El algoritmo de combinación convexa pone énfasis en la determinación del largo del paso y utiliza dos conceptos fundamentales:
  - Método de combinación convexa para problemas multivariados de objetivo no lineal y restricciones lineales: determina la dirección de descenso factible en términos de la profundidad del descenso y la distancia que se puede avanzar en la misma.
  - Método de reducción de intervalo para problemas en una dimensión: bisección o Bolzano.

## Algoritmo de combinación convexa (cont.)

**Paso 0:** Inicialización. Carga inicial de la red mediante la distribución del flujo según tiempos de viaje  $t_a = t_a(0), \forall a$ . Con esto se obtienen los flujos iniciales  $x_a^1$ . Setear el contador de iteraciones  $n = 1$ .

**Paso 1:** Actualizar  $t_a^n = t_a(x_a^n), \forall a$ .

**Paso 2:** Búsqueda de dirección. Asignar el flujo en base a  $t_a^n$ , con lo que se obtienen flujos auxiliares  $y_a^n$ .

**Paso 3:** Determinación del paso. Encontrar el  $\alpha_n$  que resuelve

$$\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(\omega) d\omega$$

**Paso 4:** Avance. Setear  $x_a^{n+1} = x_a^n + \alpha(y_a^n - x_a^n), \forall a$ .

**Paso 5:** Test de convergencia. Si se cumple la condición de parada, parar (flujos en equilibrio  $x_a^{n+1}$ ); si no setear  $n = n + 1$  y ejecutar paso 1.

## Algoritmo de combinación convexa (cont.)

- La condición de parada puede establecerse en términos de cambios en flujos  $x_a^n$  o tiempos en caminos de origen a destino  $u_k^n$ .
- Cuanto más congestionada la red, más iteraciones requiere el algoritmo en converger.
- Para el ejemplo anterior, el algoritmo logra convergencia en pocas iteraciones.



## Algoritmo de combinación convexa (cont.)

Iteration Number	Algorithmic Step	Link			Objective Function	Step Size
		1	2	3		
0	Initialization	$t_1^0 = 10.0$ $x_1^1 = 10.00$	$t_2^0 = 20.0$ $x_2^1 = 0.00$	$t_3^0 = 25.0$ $x_3^1 = 0.00$		
1	Update Direction Move	$t_1^1 = 947.0$ $y_1^1 = 0$ $x_1^2 = 4.04$	$t_2^1 = 20.0$ $y_2^1 = 10$ $x_2^2 = 5.96$	$t_3^1 = 25.0$ $y_3^1 = 0$ $x_3^2 = 0.00$	$z(\mathbf{x}) = 1975.00$	$\alpha_1 = 0.596$
2	Update Direction Move	$t_1^2 = 35.0$ $y_1^2 = 10$ $x_1^3 = 3.39$	$t_2^2 = 35.0$ $y_2^2 = 0$ $x_2^3 = 5.00$	$t_3^2 = 25.0$ $y_3^2 = 0$ $x_3^3 = 1.61$	$z(\mathbf{x}) = 197.00$	$\alpha_1 = 0.161$
3	Update Direction Move	$t_1^3 = 22.3$ $y_1^3 = 10$ $x_1^4 = 3.62$	$t_2^3 = 27.3$ $y_2^3 = 0$ $x_2^4 = 4.83$	$t_3^3 = 35.3$ $y_3^3 = 0$ $x_3^4 = 1.55$	$z(\mathbf{x}) = 189.98$	$\alpha = 0.035$
4	Update Direction Move	$t_1^4 = 26.1$ $y_1^4 = 0$ $x_1^5 = 3.54$	$t_2^4 = 26.3$ $y_2^4 = 0$ $x_2^5 = 4.73$	$t_3^4 = 25.3$ $y_3^4 = 10$ $x_3^5 = 1.72$	$z(\mathbf{x}) = 189.44$	$\alpha = 0.020$
5	Update Direction Move	$t_1^5 = 24.8$ $y_1^5 = 10$ $x_1^6 = 3.59$	$t_2^5 = 25.8$ $y_2^5 = 0$ $x_2^6 = 4.70$	$t_3^5 = 25.4$ $y_3^5 = 0$ $x_3^6 = 1.71$	$z(\mathbf{x}) = 189.33$	$\alpha = 0.007$
	Update	$t_1^6 = 25.6$	$t_2^6 = 25.7$	$t_3^6 = 25.4$	$z(\mathbf{x}) = 189.33$	

# Modelos normativos para diseño de redes de transporte privado

- Decisiones respecto a la estructura y atributos de la red de calles.
- Uso menos difundido respecto a los modelos descriptivos.
- Variantes respecto a decisiones modeladas:
  - Continua: expansión de capacidad de calles existentes, instalación y configuración de semáforos y peajes urbanos.
  - Discreta: construir nuevas calles, agregar nuevas sendas, determinar el sentido de calles y restricciones de giros.
  - Mixta: decisiones continuas y discretas.

## Modelos normativos, características

- Unidad de demanda: vehículos.
- Hipótesis de comportamiento: equilibrio de usuario (UE) o su variante estocástica (SUE).
- Entrada: red existente, matriz OD, restricciones de cotas mínimas y máximas en parámetros, proyectos candidatos (nuevas calles, nuevos semáforos, etc), presupuesto, entre otras.

# Decisiones de naturaleza continua<sup>1</sup>

- Variables de decisión: capacidad en cada arco.
- Objetivo: minimización de tiempo total de viaje y presupuesto.
- Restricciones: límite superior de capacidad y flujos de acuerdo a UE.

$$\begin{aligned} \mathbf{Min}_y \quad & F(x, y) = \sum_{a \in E} (c_a(x_a(y), y_a) x_a(y) + \theta G_a(y_a)) \\ \text{s.t.} \quad & 0 \leq y_a \leq u_a \quad \forall a \in E \end{aligned}$$

where  $x(y)$  is the equilibrium flow defined by the following equivalent minimization problem:

$$\begin{aligned} \mathbf{Min}_x \quad & z = \sum_{a \in E} \int_0^{x_a} c_a(w, y_a) dw \\ \text{s.t.} \quad & \sum_{p \in R_w} h_p^w = T_w \quad \forall w \in W \\ & x_a = \sum_{w \in W} \sum_{p \in R_w} h_p^w \delta_{ap}^w \quad \forall a \in E \\ & h_p^w \geq 0 \quad \forall p \in R_w, w \in W \end{aligned}$$

---

<sup>1</sup>Chiou, S-W (2005) Bilevel programming for the continuous transport network design problem. Transportation Research Part B: Methodological 39(4):361-383

## Decisiones de naturaleza continua<sup>2</sup>

- Variables de decisión: capacidad en cada arco y duraciones de luces en semáforos.
- Objetivo: minimización de demoras en intersecciones y presupuesto.
- Restricciones: límite superior de capacidad, opciones de señalización y flujos de acuerdo a UE.

$$\begin{aligned} & \text{Min}_{\mathbf{y}, \Psi, \mathbf{f}} \sum_a D_a W_D M_D + S_a W_S M_S + \sigma V_a(y_a) \\ & \text{subject to } \Psi \in \Pi, \quad \mathbf{0} \leq \mathbf{y} \leq \mathbf{y}_{\max}, \quad \mathbf{f} \in \mathbf{K}_0(\mathbf{y}, \Psi), \end{aligned}$$

---

<sup>2</sup>Chiou, S-W (2008) A hybrid approach for optimal design of signalized road network. Applied Mathematical Modelling 32(2):195-207.

## Decisiones de naturaleza discreta<sup>3</sup>

- Variables de decisión: tramos a incluir en la red de calles.
- Objetivo: minimización de tiempo total de viaje .
- Restricciones: límite superior de presupuesto y flujos de acuerdo a UE.

$$\begin{aligned}
 \min Z &= \sum_{a \in A_1} x_a t_a(x_a) + \sum_{a \in A_2} x_a t_a(x_a) & \min F &= \sum_{a \in A} \int_0^{x_a} t_a(x) dx \\
 \text{s.t. } & \sum_{a \in A_2} c_a u_a \leq B & \text{s.t. } & \sum_k f_k^{rs} = q_{rs}, \quad \forall r \in R, s \in S \\
 & u_a = 0 \text{ or } 1 \quad \forall a \in A_2 & & f_k^{rs} \geq 0, \quad \forall r \in R, s \in S, k \in K_{rs} \\
 & & & x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A \\
 & & & x_a \leq M u_a, \quad \forall a \in A_2
 \end{aligned}$$

---

<sup>3</sup>Gao, Z; Wu, J; Sun, H (2005) Solution algorithm for the bi-level discrete network design problem. Transportation Research Part B: Methodological 39(6):479-495.

## Decisiones de naturaleza discreta<sup>4</sup>

- Variables de decisión: restricciones de giro en nodos.
- Objetivo: minimización de tiempo total de viaje .
- Restricciones: nodos donde imponer restricciones de giros y flujos de acuerdo a SUE.

$$\begin{aligned} \min \quad & G(\mathbf{x}, \mathbf{y}) = \sum_{ab} x_{ab}(\mathbf{y}) t_{ab}(\mathbf{x}(\mathbf{y}), \mathbf{y}) \\ \text{subject to} \quad & y_{\mu} = \{0, 1\} \quad \forall \mu \in \Gamma \end{aligned}$$

---

<sup>4</sup>Long, J; Gao, Z; Zhang, H; Szeto, WY (2010) A turning restriction design problem in urban road networks. European Journal of Operational Research 206(3):569-578.

## Bibliografía

- Sheffi, Y (1985) Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice Hall. Disponible en [http://web.mit.edu/sheffi/www/selectedMedia/sheffi\\_urban\\_trans\\_networks.pdf](http://web.mit.edu/sheffi/www/selectedMedia/sheffi_urban_trans_networks.pdf)
- Zanjirani, R; Miandoabchi, E; Szeto, WY; Rashidi, H (2013) A review of urban transportation network design problems. European Journal of Operational Research 229(2):281-302.
- Notas del docente del curso.