Maximización de la utilidad de la red (Network Utility Maximization,NUM)

Network Utility Maximization

$$\begin{aligned} \max \quad & \sum_{r \in R} U_r(x_r) \\ \text{sujeto a} \quad & Rx \leq C \\ & x \geq 0 \end{aligned}$$

x_r : tasa de transferencia del usuario *r*

$U_r(x_r)$: función de utilidad del usuario en función de xr.

Se asume que la utilidad es aditiva, la utilidad del sistema es la suma de las utilidades de todos los usuarios.

Asumiremos que la utilidad es estrictamente cóncava.

C: vector de capacidad con *Cj*: capacidad del recurso (enlace) j

R: Matriz de ruteo

Rjr = 1 si el enlace j está en la ruta r

Rjr = 0 en otro caso

Nociones de Justicia

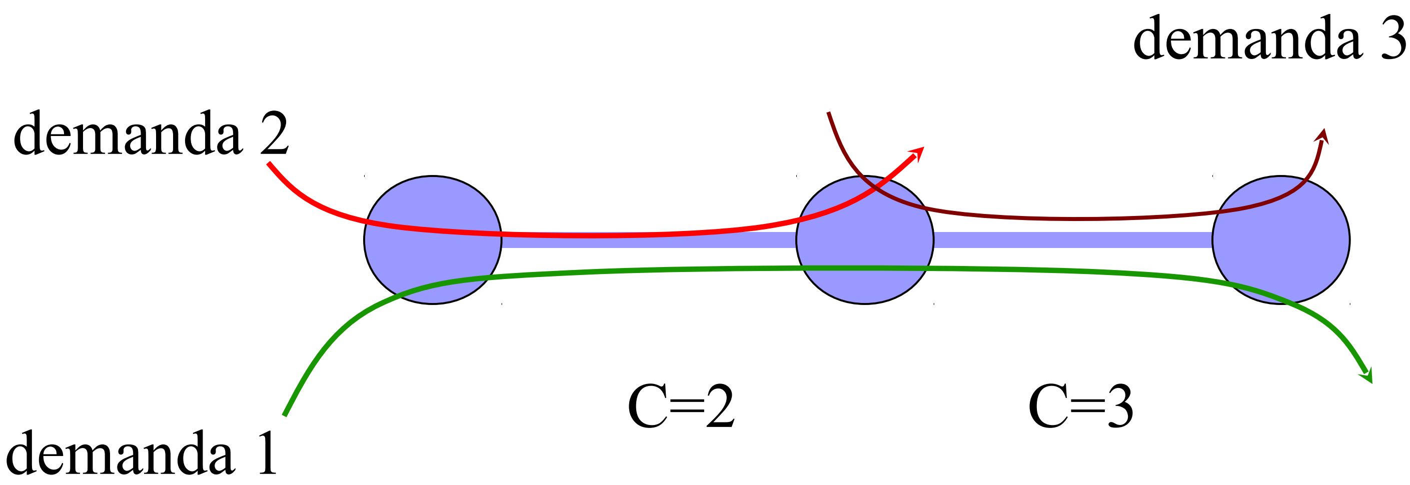
- **Proportional fairness** [Kelly, Maulloo & Tan, '98]
- Un vector factible *x* is proporcionalmente justo si para todo otro vector factible *y* se cumple que

$$\sum \frac{(y_i - x_i)}{x_i} \leq 0$$

¿Cuál es la relación entre este tipo de justicia y NUM?

- Se puede *interpretar* la justicia proporcional como una asignación que maximiza la utilidad global, siendo la utilidad de cada fuente el logaritmo de su tasa (ejercicio práctico)

Ejemplo



- ¿cuál es el reparto proporcionalmente justo?

  max (log(x1)+log(x2)+log(x3)) s.t. x1+x2≤ 2, x1+x3 ≤ 3

  Solución: x1=0.8, x2=1.2, x3=2.2

- ¿cuál es el reparto que maximiza el throughput?

  max (x1+x2+x3) s.t. x1+x2≤ 2, x1+x3 ≤ 3

  Solución: x1=0, x2=2, x3=3

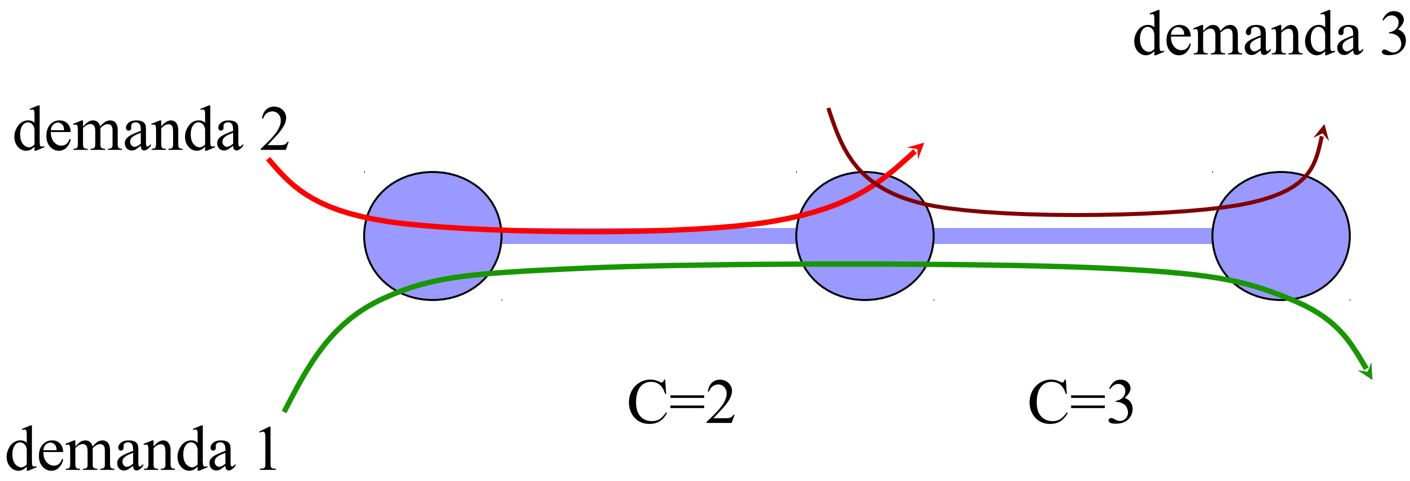
Asignación de recursos Max-Min justa

- Definición : Una tasa factible R^* es max-min justa si ninguna tasa R_i puede ser incrementada sin decrementar R_k sujeto a que $R_k \leq R_i$

¿Cómo encontrar una asignación Max-Min justa?

- Idea: compartir los enlaces por igual hasta que sea posible
- Procedimiento
  1. Comenzar con tasa 0 para todas las demandas
  2. Incrementar la tasa a la misma velocidad para todas las demandas,hasta que algún enlace sature.
  3. Remover los enlaces saturados y las demandas que utilizan esos enlaces.
  4. Volver al paso 2 con las capacidades remanentes hasta que no quede ninguna demanda.

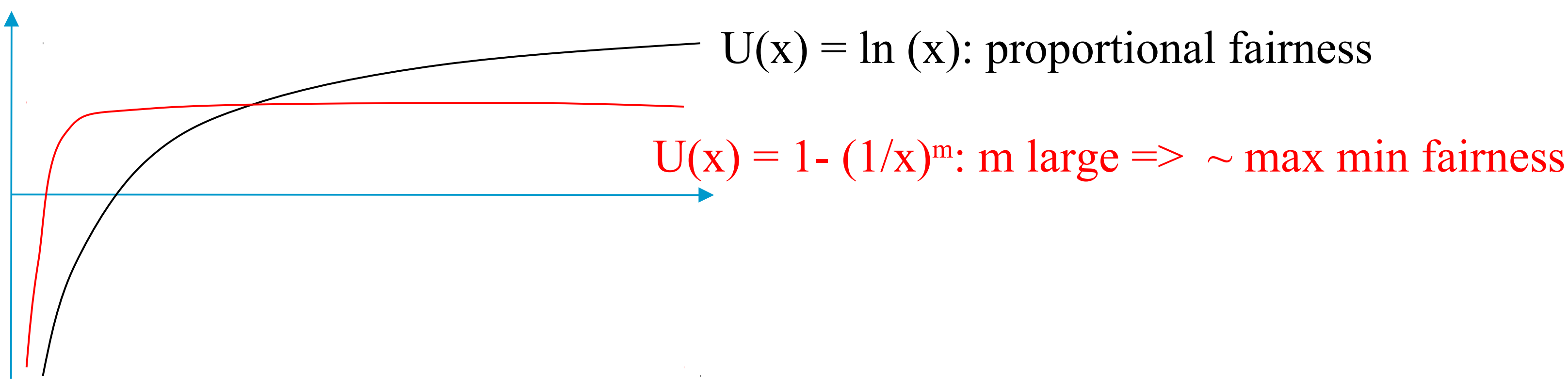
Ejemplo



- ¿cuál es el reparto max-min?
- Óptimo max-min: x1=1, x2=1, x3=2

Justicia basada en Utilidades

- Si se toman otras funciones utilidad se tienen otros criterios de justicia
- La justicia Max-Min se demuestra que se puede ver como el límite de un problema NUM donde la función utilidad converge a la función escalón.



Algoritmos distribuidos para NUM

$$\max_{x_r} \sum_{r \in \mathcal{S}} U_r(x_r) \qquad \sum_{r: l \in r} x_r \leq c_l, \ \forall l \in \mathcal{L},$$

$$x_r \geq 0, \ \forall r \in \mathcal{S}.$$

- Algoritmos
  - Primal
  - Dual
  - Primal-dual
- Estabilidad

Algoritmo primal

- Consideramos maximizar el siguiente sistema modificado con una función de barrera asociada a las restricciones de capacidad

$$V(x) = \sum_{r \in \mathcal{S}} U_r(x_r) - \sum_{l \in \mathcal{L}} B_l\left(\sum_{s:l \in s} x_s\right)$$

- Donde B es una función de barrera convexa y creciente de la forma

$$B_l\left(\sum_{s:l \in s} x_s\right) = \int_0^{\sum_{s:l \in s} x_s} f_l(y)dy$$

- Siendo f una función que asumimos creciente y diferenciable.

Algoritmo primal

- La condición que se debe cumplir para maximizar V(x) es entonces

$$U_r'(x_r) - \sum_{l:l\in r} f_l\left(\sum_{s:l\in s} x_s\right) = 0, \qquad r \in \mathcal{S}.$$

- Utilizando el algoritmo del gradiente, cada fuente calcula una nueva tasa y lo envía a los enlaces que le envían luego su precio a la fuente:

$$\dot{x}_r = k_r(x_r)\left(U_r'(x_r) - \sum_{l:l\in r} f_l\left(\sum_{s:l\in s} x_s\right)\right)$$

$$\begin{aligned} p_l(t) &= f_l\left(\sum_{s:l\in s} x_s\right) \\ &= f_l(y_l(t)). \end{aligned} \qquad q_r = \sum_{l:l\in r} p_l(t).$$

Estabilidad

$$\dot{x} = g(x), \qquad x(0) = x_0.$$

En un sistema dinámico donde se asume que *g(x)= 0* tiene solución única

El punto de equilibrio se dice:

1. Estable, si para cada $\varepsilon > 0$, existe $\delta = \delta(\varepsilon) > 0$ tal que

$$\|x(t)\| \leq \varepsilon, \quad \forall t \geq 0, \qquad \text{si} \quad \|x_0\| \leq \delta.$$

2. Asintóticamente estable si existe $\delta > 0$ tal que

$$\lim_{t\to\infty} \|x(t)\| = 0$$

para todo $\|x_0\| \leq \delta$

3.Globalmente asintóticamente estable si $\lim_{t\to\infty} \|x(t)\| = 0$ para cualquier condición inicial.

El algoritmo Dual

La función dual de Lagrange es $D(p) = \max_{\{x_r > 0\}} \sum_r U_r(x_r) - \sum_l p_l \left(\sum_{s:l \in s} x_s - c_l \right)$

Y el problema dual $\min_{p \geq 0} D(p)$

$$U_r'(x_r) = q_r,$$

Por un lado se debe cumplir que:

$$x_r = U_r'^{-1}(q_r),$$

Por otra parte:

$$\begin{aligned} \frac{\partial D}{\partial p_l} &= \sum_{r:l \in r} \frac{U_r'(x_r)}{U_r''(x_r)} - (y_l - c_l) - \sum_l p_l \sum_{r:l \in r} \frac{1}{U_r''(x_r)} \\ &= c_l - y_l, \end{aligned}$$

El algoritmo dual entonces comprende las siguientes ecuaciones:

$$\begin{aligned} x_r &= U_r'^{-1}(q_r) \\ \dot{p}_l &= h_l(y_l - c_l)^+_{p_l} \end{aligned}$$

El algoritmo Primal y Dual

- Implementación distribuida
- Estabilidad

El algoritmo Primal-Dual

$$\dot{x} = K\left[\frac{\partial D}{\partial x}\right]_x^+ = K\left[U'(x) - R^T p\right]_x^+$$

$$\dot{p} = \Gamma\left[\frac{\partial D}{\partial p}\right]_p^+ = \Gamma\left[Rx - c\right]_p^+$$

- Implementación distribuida
- Se puede probar también su estabilidad.