National Technical University of Athens
School of Electrical and Computer Engineering

Game Theory
Spring 2025 - Exercise Series 3

Submitted by
Evangelos Chaniadakis
echaniadakis@gmail.com
03400279

Submission Date
Wednesday 25th June, 2025

Problems

Problem 1 Price of Anarchy in Congestion Games

We shall analyze the congestion game defined over the network given below.

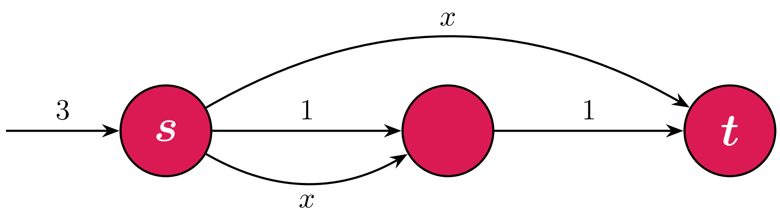

Figure 1: Network under analysis

As we can easily observe, it is comprised of three parallel paths with their corresponding latency functions.

- Top path: $\ell_{\text{top}}(x) = x$
- Middle path: $\ell_{\text{mid}}(x) = 2$
- Bottom path: $\ell_{\text{bot}}(x) = x + 1$

We shall study the game under two models:

1. **Non-atomic routing**, where infinitesimal agents control negligible flow.
2. **Atomic routing**, where three players each control one unit of indivisible flow.

In both cases, we compute the Wardrop/Nash equilibrium, the social optimum and the resulting Price of Anarchy.

1.1 Non-Atomic Routing

We consider a non-atomic routing game over the network shown in Figure 1 with three parallel paths connecting a source node s to a destination node t. A total of 3 units of infinitesimal flow must be routed from s to t.

1.1.1 Wardrop Equilibrium

In a Wardrop equilibrium, all used paths have equal and minimal latency. Let x_1, x_2 and x_3 denote the flows on the top, middle and bottom paths respectively. The total flow constraint is $x_1 + x_2 + x_3 = 3$. We consider an equilibrium where only the top and bottom paths are used. Thus, $x_2 = 0$ and the latencies must then be equal:

$$\begin{aligned} x_1 + x_3 &= 3 \\ \ell_{\text{top}}(x_1) &= \ell_{\text{bot}}(x_3) \\ x_1 &= x_3 + 1 \end{aligned}$$

By substituting, we get

$$x_1 + (x_1 - 1) = 3 \Rightarrow 2x_1 = 4 \Rightarrow x_1 = 2, \quad x_3 = 1$$

Since the latency of the middle path is also 2, not using it does not violate equilibrium conditions. Thus, the flow at equilibrium is

$$(x_1, x_2, x_3) = (2, 0, 1)$$

with a total cost of

$$C_{\text{eq}} = x_1 \cdot \ell_{\text{top}}(x_1) + x_3 \cdot \ell_{\text{bot}}(x_3) = 2 \cdot 2 + 1 \cdot 2 = 6$$

1.1.2 Socially Optimal Flow

To compute the optimal flow minimizing total latency, we use the standard technique of replacing each latency function $\ell_e(x)$ with its marginal-cost counterpart:

$$\hat{\ell}_e(x) = \ell_e(x) + x \cdot \ell_e'(x)$$

This yields:

$$\begin{aligned} \hat{\ell}_{\text{top}}(x) &= 2x \\ \hat{\ell}_{\text{mid}}(x) &= 2 \\ \hat{\ell}_{\text{bot}}(x) &= 2x + 1 \end{aligned}$$

Assuming all three paths are used, their marginal costs must equal a common value L.

$$\begin{aligned} \hat{\ell}_{\text{top}}(x_1) = 2x_1 = L \quad &\Rightarrow \quad x_1 = \frac{L}{2} \\ \hat{\ell}_{\text{bot}}(x_3) = 2x_3 + 1 = L \quad &\Rightarrow \quad x_3 = \frac{L-1}{2} \end{aligned}$$

Using the total flow constraint $x_1 + x_2 + x_3 = 3$, we solve for x_2:

$$x_2 = 3 - x_1 - x_3 = 3 - \frac{L}{2} - \frac{L-1}{2} = \frac{7-2L}{2}$$

This is valid as long as $x_1, x_2, x_3 \geq 0$, implies $1 \leq L \leq 3.5$. We express the total latency $C_{\text{opt}}(L)$ as a function of L, using the original latency functions:

$$C_{\text{opt}}(L) = x_1^2 + 2x_2 + x_3(x_3 + 1) = \frac{1}{2}L^2 - 2L + \frac{27}{4}$$

We differentiate in order to minimize the cost

$$\frac{dC_{\text{opt}}}{dL} = L - 2 = 0 \Rightarrow L = 2$$

This gives $x_1 = \frac{2}{2} = 1,\ x_2 = \frac{7-4}{2} = 1.5$ & $x_3 = \frac{2-1}{2} = 0.5$. Thus, the socially optimal flow is

$$\boldsymbol{(x_1^*,\ x_2^*,\ x_3^*) = (1,\ 1.5,\ 0.5)}$$

We do finally compute the total cost, though now, using the original latency functions:

$$\begin{aligned}C_{\text{opt}} &= x_1 \cdot \ell_{\text{top}}(x_1) + x_2 \cdot \ell_{\text{mid}} + x_3 \cdot \ell_{\text{bot}}(x_3) \\ &= 1 \cdot 1 + 1.5 \cdot 2 + 0.5 \cdot (0.5 + 1) = 1 + 3 + 0.75 \\ &= 4.75\end{aligned}$$

1.1.3 Price of Anarchy

The Price of Anarchy compares the cost of selfish routing (equilibrium) to the optimal centralized routing.

$$\text{PoA} = \frac{C_{\text{eq}}}{C_{\text{opt}}} = \frac{6}{4.75} = \frac{24}{19} = 1.26315$$

This result demonstrates that in this network, selfish routing leads to a total latency that is approximately 26% higher than the socially optimal outcome.

1.2 Atomic Routing

We consider an atomic selfish routing game with three players, each controlling one unit of indivisible flow, over the same network shown in Figure 1. Here, x denotes the number of players choosing a given path. Since each player is atomic, $x \in \{0, 1, 2, 3\}$. In this atomic setting, a pure Nash equilibrium is a profile of path choices (a_1, a_2, a_3) such that for each player i, the latency on their chosen path is less than or equal to the latency they would experience by deviating unilaterally to another path.

Enumeration of Profiles

There are $3^3 = 27$ pure strategy profiles when treating players as distinct. However, since we are only interested in the number of players assigned to each path (not their identities), it suffices to consider the distinct distributions of players across paths. This reduces the analysis to counting the non-negative integer solutions of the equation $x_1 + x_2 + x_3 = 3$, where x_i is the number of players assigned to each path. Using the classical *stars and bars* argument, the number of such solutions is:

$$\binom{n + r - 1}{r - 1} = \binom{3 + 3 - 1}{3 - 1} = \binom{5}{2} = 10$$

where $n = 3$ is the total number of players (*stars*) and $r = 3$ is the number of paths (*bins*). Each solution corresponds to an unordered profile describing how many players use each path. We analyze some of these profiles to identify pure Nash equilibria.

1.2.1 Nash Equilibrium

Case: One Player per Path (1,1,1)

- $\ell_{\text{top}}(1) = 1$
- $\ell_{\text{mid}} = 2$
- $\ell_{\text{bot}}(1) = 1 + 1 = 2$

The player on the top path experiences lower latency. A player on another path has incentive to switch to the top path. Thus, it is not a Nash equilibrium.

Case: All Players on Top Path (3,0,0)

- $\ell_{\text{top}}(3) = 3$
- $\ell_{\text{mid}} = 2$
- $\ell_{\text{bot}}(0) = 1$

At least two players can strictly reduce their latency by switching. Not a Nash equilibrium either.

Case: A Mixed Strategy(2,0,1)

Two players choose the top path and one player chooses the bottom path.

- $\ell_{\text{top}}(2) = 2$
- $\ell_{\text{bot}}(1) = 1 + 1 = 2$
- $\ell_{\text{mid}} = 2$

All players experience latency 2. Any unilateral deviation leads to the same or worse latency. Hence, this is a pure Nash equilibrium.

All remaining configurations (e.g., (2,1,0), (0,2,1), (1,0,2)) result in at least one player facing an incentive to deviate. Hence, they are not equilibria. Our unique pure Nash equilibrium (up to player relabeling) is

$$(x_{\text{top}}, x_{\text{mid}}, x_{\text{bot}}) = (2, 0, 1)$$

Each player experiences latency 2 and no player benefits from deviating. Thus, our flow in equilibrium incurs a cost of

$$C_{\text{eq}} = 2 \cdot \ell_{\text{top}}(2) + 1 \cdot \ell_{\text{bot}}(1) = 2 \cdot 2 + 1 \cdot 2 = 6$$

1.2.2 Socially Optimal Flow

To determine the socially optimal flow assignment, we compute the total latency cost for all 10 possible configurations of player distributions. For each configuration $(x_{\text{top}}, x_{\text{mid}}, x_{\text{bot}})$, we evaluate the total cost as:

$$C = x_{\text{top}} \cdot \ell_{\text{top}}(x_{\text{top}}) + x_{\text{mid}} \cdot \ell_{\text{mid}} + x_{\text{bot}} \cdot \ell_{\text{bot}}(x_{\text{bot}})$$

We analyze each distribution:

- (3,0,0): $3 \cdot 3 = 9$
- (2,1,0): $2 \cdot 2 + 1 \cdot 2 = 6$
- (2,0,1): $2 \cdot 2 + 1 \cdot (1 + 1) = 6$
- (1,2,0): $1 \cdot 1 + 2 \cdot 2 = \mathbf{5}$
- (1,1,1): $1 \cdot 1 + 1 \cdot 2 + 1 \cdot (1 + 1) = \mathbf{5}$
- (1,0,2): $1 \cdot 1 + 2 \cdot (2 + 1) = 1 + 6 = 7$
- (0,3,0): $3 \cdot 2 = 6$
- (0,2,1): $2 \cdot 2 + 1 \cdot (1 + 1) = 4 + 2 = 6$
- (0,1,2): $1 \cdot 2 + 2 \cdot (2 + 1) = 2 + 6 = 8$
- (0,0,3): $3 \cdot (3 + 1) = 3 \cdot 4 = 12$

Among all possible profiles, the configurations (1,2,0) and (1,1,1) both yield the minimal total latency. Hence, the socially optimal total cost is $C_{\text{opt}} = 5$.

1.2.3 Price of Anarchy

Thus, the price of anarchy is

$$\text{PoA} = \frac{C_{\text{eq}}}{C_{\text{opt}}} = \frac{6}{5}$$

This quantifies the inefficiency of equilibria in the system, indicating that the worst-case Nash equilibrium incurs a 20% higher total cost than the social optimum.

Problem 2 How Bad Can Quadratics Get?

2.1 Worst-Case PoA for $ax^2 + c$ Delays

The worst-case Price of Anarchy over all non-atomic congestion games with delay functions in class $\mathcal{L}$ is achieved on a two-link Pigou network. We thus consider a network of this type, which comprises of two parallel links from s to t and total demand $r = 1$.

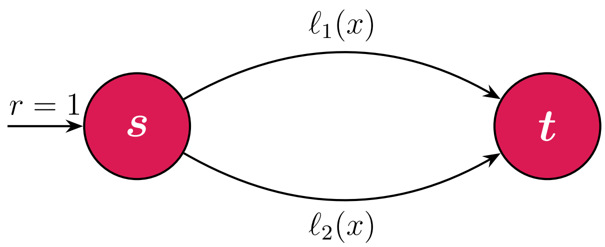

Figure 2: Two-Link Pigou Network

The top path is given by the function $\ell_1(x) = ax^2 + c$, where $a, c \geq 0$, whereas the bottom path is the constant function $\ell_2(x) = 1$.

2.1.1 Wardrop equilibrium

At Wardrop equilibrium, all flow is routed along paths with minimum latency. The equilibrium behavior depends on the value of $\ell_1(1)$ relative to 1:

- if $\ell_1(1) < 1$: the first path has strictly lower latency than the second, so the entire unit of flow is routed through the first path. Then:

$$C_{\text{eq}} = \ell_1(1) = a + c.$$

- if $\ell_1(1) > 1$: the second path has strictly lower latency, so all flow is routed through the second path:

$$C_{\text{eq}} = 1.$$

- if $\ell_1(1) = 1$: both paths have equal latency when fully loaded. In this case, any flow split between the two paths (e.g., any $x \in [0, 1]$ on path 1) is an equilibrium and the equilibrium cost is:

$$C_{\text{eq}} = 1.$$

2.1.2 Socially Optimal Flow

Let x be the fraction of flow routed through the first path. Then the total cost is:

$$C(x) = x \cdot \ell_1(x) + (1 - x) \cdot 1 = x(ax^2 + c) + 1 - x = ax^3 + (c - 1)x + 1$$

Setting the derivative of C equal to zero gives us:

$$C'(x) = 3ax^2 + (c - 1)$$

$$\Downarrow$$

$$3ax^2 + (c - 1) = 0 \quad \Rightarrow \quad x^2 = \frac{1 - c}{3a}$$

We will now consider the case $a = 1$, $c = 0$, so $\ell_1(x) = x^2$. Based on that, we minimize the total cost:

$$C(x) = x^3 + 1 - x \quad \Rightarrow \quad C'(x) = 3x^2 - 1 = 0 \Rightarrow x = \frac{1}{\sqrt{3}}$$

We plug $x = \frac{1}{\sqrt{3}}$ into the original cost function $C(x) = x \cdot \ell_1(x) + (1 - x) \cdot 1$ and we get

$$C_{\text{opt}} = \left(\frac{1}{\sqrt{3}} \cdot \left(\frac{1}{\sqrt{3}}\right)^2\right) + \left(1 - \frac{1}{\sqrt{3}}\right) = \frac{1}{3\sqrt{3}} + 1 - \frac{1}{\sqrt{3}} = \frac{3\sqrt{3} - 2}{3\sqrt{3}}$$

2.1.3 Price of Anarchy

$$PoA = \frac{C_{\text{eq}}}{C_{\text{opt}}} = \frac{1}{\frac{3\sqrt{3}-2}{3\sqrt{3}}} = \frac{3\sqrt{3}}{3\sqrt{3}-2}$$

This value represents the worst-case Price of Anarchy for delay functions of the form $\ell(x) = ax^2 + c$ and it is exactly reached when $\ell(x) = x^2$.

2.2 Worst-Case PoA for $ax^2 + bx + c$ Delays

We now extend the previous analysis to the broader class of polynomial delay functions of the form $\ell(x) = ax^2 + bx + c$, where $a, b, c \geq 0$. We aim to show that the same tight upper bound on the Price of Anarchy holds for this class. As before, by the Pigou Bound Theorem, the worst-case PoA is achieved in a two-link Pigou network:

- Top path: $\ell_1(x) = ax^2 + bx + c$
- Bottom path: $\ell_2(x) = 1$ (constant)

2.2.1 Wardrop equilibrium

We consider the behavior of the Wardrop equilibrium for a total demand $r = 1$:

- If $\ell_1(1) < 1$, then all flow is routed through the top path and the equilibrium cost is

$$C_{\text{eq}} = \ell_1(1) = a + b + c.$$

- But if $\ell_1(1) \geq 1$, the entire flow uses the bottom path and the equilibrium cost is

$$C_{\text{eq}} = 1.$$

2.2.2 Socially Optimal Flow

Let $x \in [0, 1]$ be the fraction of flow routed through the top path. The total cost is:

$$C(x) = x(ax^2 + bx + c) + (1 - x) \cdot 1 = ax^3 + bx^2 + (c - 1)x + 1$$

To minimize $C(x)$, we compute the derivative:

$$C'(x) = 3ax^2 + 2bx + (c - 1)$$

Solving $C'(x) = 0$ gives the socially optimal flow split. While the exact solution depends on the values of a, b and c, we focus on an *example* that leads to the worst-case Price of Anarchy. Consider the delay functions:

$$\ell_1(x) = x^2 + x \quad \& \quad \ell_2(x) = 1$$

Since $\ell_1(1) > \ell_2(1)$, all flow is routed through the second edge, so the equilibrium cost is $C_{\text{eq}} = 1$. To find the optimal split, define:

$$C(x) = x(x^2 + x) + (1 - x) = x^3 + x^2 + 1 - x \Rightarrow C'(x) = 3x^2 + 2x - 1$$

Solving $C'(x) = 0$:

$$3x^2 + 2x - 1 = 0 \Rightarrow x = \frac{-2 + \sqrt{16}}{6} = \frac{-2 + 4}{6} = \frac{1}{3}$$

Then, cost of socially oprimal flow is

$$C_{\text{opt}} = \frac{1}{3} \cdot \ell_1\left(\frac{1}{3}\right) + \left(1 - \frac{1}{3}\right) = \frac{1}{3} \cdot \left(\left(\frac{1}{3}\right)^2 + \frac{1}{3}\right) + \frac{2}{3} = \frac{1}{3} \cdot \left(\frac{1}{9} + \frac{1}{3}\right) + \frac{2}{3} = \frac{1}{3} \cdot \frac{4}{9} + \frac{2}{3} = \frac{4}{27} + \frac{18}{27} = \frac{22}{27}$$

2.2.3 Price of Anarchy

$$PoA = \frac{C_{\text{eq}}}{C_{\text{opt}}} = \frac{1}{22/27} = \frac{27}{22} \approx 1.227$$

The colormap below shows how the inefficiency of selfish routing evolves as linear and constant terms are added. The highest value on the colorbar corresponds to the theoretical and proven worst-case PoA $\frac{3\sqrt{3}}{3\sqrt{3}-2}$, which occurs at $b = 0$ *&* $c = 0$.

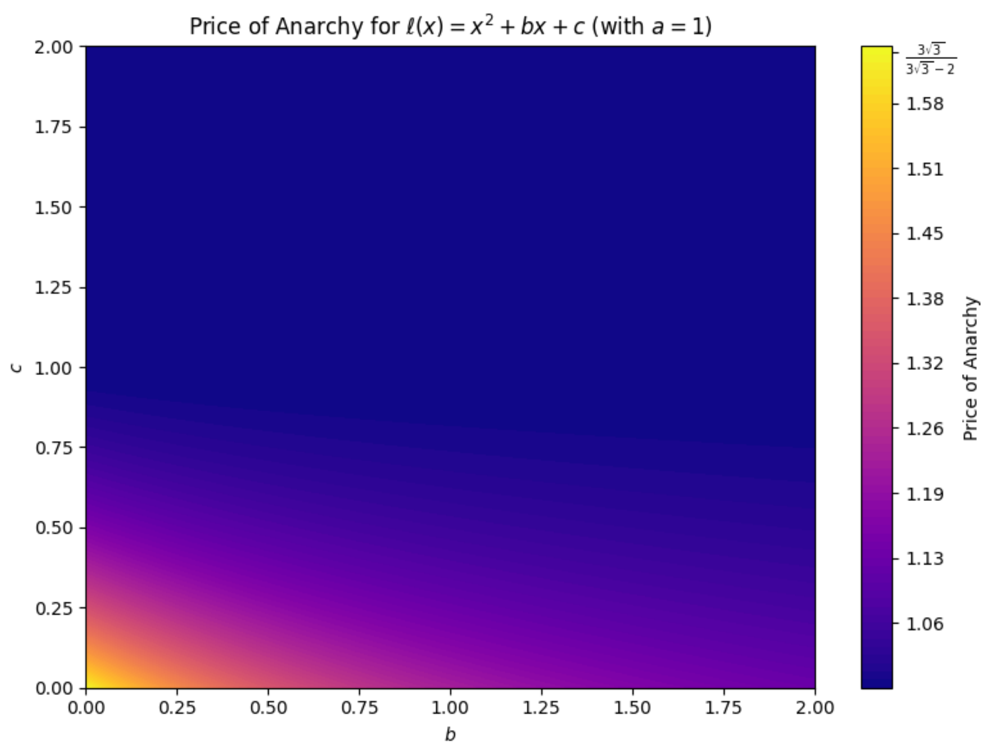

Figure 3: Price of Anarchy for latency functions of the form $\ell(x) = x^2 + bx + c$, with fixed $a = 1$ and parameters $b, c \in [0, 2]$.

2.3 Bigger Delay, Less Anarchy?

At first glance, one might expect that adding a linear term bx or a constant term c to a delay function would worsen congestion and increase inefficiency. However, the opposite is true: the worst-case Price of Anarchy for delay functions of the form $\ell(x) = ax^2 + bx + c$ (with $a, b, c \geq 0$) is achieved when $b = c = 0$, that is, when the function is purely quadratic: $\ell(x) = ax^2$.

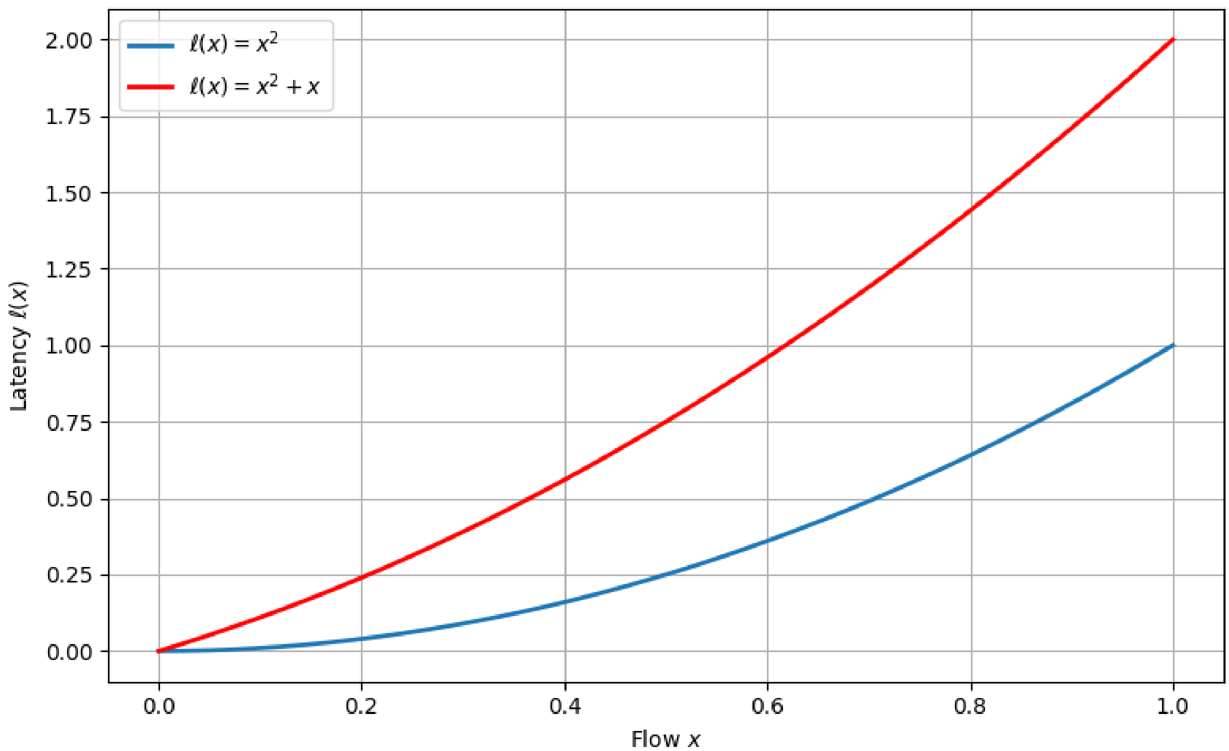

Figure 4: Comparison of latency functions $\boldsymbol{\ell(x) = x^2}$ and $\boldsymbol{\ell(x) = x^2 + x}$.

As the figure shows, the function $\ell(x) = x^2 + x$ ($a = b = 1, c = 0$) grows faster at low flow levels than the purely quadratic function $\ell(x) = x^2$ ($a = 1, b = c = 0$). This means selfish users are penalized earlier when choosing the lower-cost path. As a result, they self-distribute in a way that is closer to the socially optimal flow.

In contrast, for $\ell(x) = x^2$, the latency remains relatively low until flow gets large, which encourages overloading the faster path under selfish behavior. This causes a larger deviation from the optimal solution and leads to a higher total cost at equilibrium.

Thus, the presence of linear ($b > 0$) or constant ($c > 0$) terms moderates the inefficiency of selfish routing. The worst-case inefficiency within the class $\ell(x) = ax^2 + bx + c$ arises when $b = c = 0$ and the Price of Anarchy attains its maximum value of $\dfrac{3\sqrt{3}}{3\sqrt{3} - 2} \approx 1.625$.

Problem 3 Does the EF1 Algorithm Always Finish?

We revisit the EF1 allocation algorithm for general valuations, presented in the last lecture, which assigns goods iteratively by selecting an agent with no incoming envy and giving them an item. When no such agent exists, the algorithm resolves envy cycles by rotating bundles along those cycles. While correctness has been shown, we have not yet determined if this process does always terminate.

Algorithm 1 EF1 Cycle Elimination Algorithm

Initialize $A_i \leftarrow \emptyset$ for all agents i
Initialize envy graph $G \leftarrow (V, E)$ with $E = \emptyset$
for each good g in fixed order **do**
    while no agent has in-degree 0 in G **do**
        Find a directed cycle C in G
        for each $(i \to j)$ in C **do**
            $A_i \leftarrow A_j$
        end for
        Update envy graph G
    end while
    Let i be any agent with in-degree 0
    $A_i \leftarrow A_i \cup \{g\}$
    Update envy graph G
end for

We must prove that this process cannot continue forever.

3.1 A Potential Function That Increases at Every Step

Our goal is to define a potential function that increases strictly during both types of operations performed by the EF1 algorithm:

1. When a good is assigned to an agent with in-degree 0 in the envy graph.
2. When a cycle in the envy graph is eliminated by rotating bundles among the agents in the cycle.

To track progress, we define the following lexicographic potential function:

$$\Phi = (k,\ -|E|),$$

where:

- k denotes the number of goods that have been assigned so far,
- $|E|$ is the number of directed edges in the envy graph G, representing strict envy relations between agents.

We interpret Φ as a lexicographically ordered pair. The algorithm modifies the state through two mechanisms:

- **Item assignment:** When a new good is assigned to an agent with no incoming envy, k increases by 1. The number of envy edges may increase or remain the same, but the first component of Φ increases, hence Φ increases.
- **Cycle elimination:** No new item is assigned, so k is unchanged. However, the envy cycle is eliminated through bundle rotation and this strictly reduces the number of envy edges: $|E|$ decreases. Therefore, the second component of Φ increases (becomes less negative), so Φ increases lexicographically.

This potential function captures the notion that each step makes measurable progress, either by allocating a new resource or simplifying the envy structure.

3.2 Why the Algorithm Always Terminates

To establish termination, we show that the EF1 algorithm performs a finite number of steps by tracking progress through the potential function $\Phi = (k, -|E|)$, where k is the number of assigned goods and $|E|$ is the number of envy edges. The key insight is that Φ increases strictly at every step and ranges over a finite domain. We begin by observing the following bounds:

- The total number of goods m is finite. Therefore, the first component k of Φ can take at most $m + 1$ distinct values.
- For n agents, the envy graph has at most $n(n-1)$ directed edges. Thus, the second component $-|E|$ belongs to a finite set $\{-n(n-1), \dots, 0\}$.

In each iteration, the algorithm may first eliminate one or more cycles in the envy graph, each of which strictly decreases $|E|$. It then assigns the next good to an agent with in-degree zero, increasing k. As a result, the potential function $\Phi = (k, -|E|)$ increases strictly in lexicographic order.

Since Φ is strictly increasing at each step and takes values from a finite set of lexicographically ordered pairs, the algorithm cannot continue indefinitely. There are only finitely many possible values for Φ and each step moves to a strictly greater one. Eventually, no further increase is possible, so the algorithm must terminate.

Therefore, the Cycle Elimination Algorithm is *guaranteed to halt* after a finite number of steps, ensuring not only EF1 correctness but also termination.

Problem 4 Nonatomic Selfish Routing with Tolls

We are given a nonatomic routing game with 12 units of flow entering a network with three parallel edges from source s to target t.

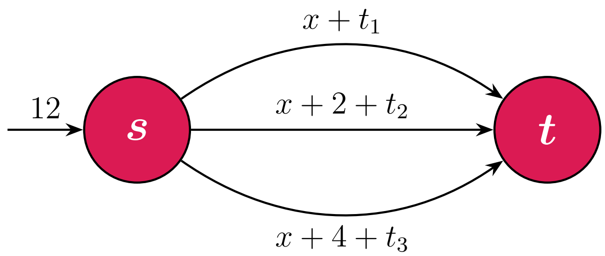

Figure 5: Three-path network with tolls

The delay on each edge is given by:

- Top edge: $\ell_1(x) = x + t_1$
- Middle edge: $\ell_2(x) = x + 2 + t_2$
- Bottom edge: $\ell_3(x) = x + 4 + t_3$

Initially, all tolls are zero: $t_1 = t_2 = t_3 = 0$.

4.1 Optimal Flow and Marginal-Cost Tolls

Let x_1, x_2 and x_3 denote the flow on the top, middle and bottom edges respectively. The total flow is:

$$x_1 + x_2 + x_3 = 12$$

The original latency functions, without tolls, are:

$$\ell_1(x) = x, \quad \ell_2(x) = x + 2, \quad \ell_3(x) = x + 4$$

To compute the optimal flow, we use marginal-cost latency functions:

$$\hat{\ell}(x) = \ell(x) + x \cdot \ell'(x)$$

$$\hat{\ell}_1(x) = x + x = 2x, \quad \hat{\ell}_2(x) = x + 2 + x = 2x + 2, \quad \hat{\ell}_3(x) = x + 4 + x = 2x + 4$$

We find the flow where all used paths have equal marginal cost:

$$2x_1 = 2x_2 + 2 = 2x_3 + 4$$

$$\Downarrow$$

$$x_1 = x_2 + 1 \quad \& \quad x_1 = x_3 + 2$$

Substituting the above into the total flow constraint, gives us:

$$x_1 + x_2 + x_3 = x_1 + (x_1 - 1) + (x_1 - 2) = 3x_1 - 3 = 12 \Rightarrow x_1 = 5 \quad \& \quad x_2 = 4 \quad \& \quad x_3 = 3$$

To induce the optimal flow as a Nash equilibrium, we assign tolls so that all paths used in the flow have equal total cost, meaning the sum of latency and toll is the same across those paths. Given the latencies $\ell_1(x_1) = 5$, $\ell_2(x_2) = 6$ and $\ell_3(x_3) = 7$, we require $5 + t_1 = 6 + t_2 = 7 + t_3$, which yields the tolls $t_1 = 5$, $t_2 = 4$ *&* $t_3 = 3$. Thus, the optimal flow is $x_1 = 5,\ x_2 = 4,\ x_3 = 3$ *and* the corresponding tolls are $t_1 = 5,\ t_2 = 4,\ t_3 = 3$.

4.2 Incentives of Toll-Setting Edge Managers

Suppose each edge is managed by a toll-setter who chooses t_i to maximize their revenue:

$$\text{Revenue}_i = t_i \cdot x_i,$$

where x_i is the equilibrium flow on edge i. The current toll profile is $(t_1, t_2, t_3) = (5, 4, 3)$, which induces the equilibrium flow $(x_1, x_2, x_3) = (5, 4, 3)$. Since each used edge has equal total cost,

$$\ell_1(5) + 5 = \ell_2(4) + 4 = \ell_3(3) + 3 = 10,$$

we are at a Wardrop equilibrium. We aim to show that no manager can increase their revenue by unilaterally reducing their toll, assuming others remain fixed.

- **Edge 1.** The latency function is $\ell_1(x) = x$, so the total cost on edge 1 is $C_1(x) = x + t_1$. Suppose the toll-setter lowers the toll to $t_1' = 5 - \varepsilon$ for some small $\varepsilon > 0$. To maintain equilibrium, users will shift to edge 1 until the total cost matches that of the other paths:
$$x_1' + t_1' = 10 \quad \Rightarrow \quad x_1' = 10 - t_1' = 5 + \varepsilon.$$
The new revenue becomes:
$$R_1' = t_1' \cdot x_1' = (5 - \varepsilon)(5 + \varepsilon) = 25 - \varepsilon^2.$$
This is strictly less than the original revenue 25. Hence, lowering the toll decreases revenue.

- **Edge 2.** Similarly, for $\ell_2(x) = x + 2$, we have $\ell_2'(x) = 1$ and the current flow is $x_2 = 4$, yielding optimal toll $t_2 = 4 \cdot 1 = 4$. Reducing the toll attracts more flow, but increased latency offsets gains and revenue decreases.

- **Edge 3.** With $\ell_3(x) = x + 4$, we again get $\ell_3'(x) = 1$, so the revenue-maximizing toll is $t_3 = 3 \cdot 1 = 3$, which matches the current toll. Any decrease leads to $R_3' = (3 - \varepsilon)(3 + \varepsilon) = 9 - \varepsilon^2 < 9$.

Thus, given the equilibrium flow induced by tolls $t_1 = 5$, $t_2 = 4$ and $t_3 = 3$, no manager benefits from unilaterally lowering their toll. Therefore, the toll profile is a Nash equilibrium among the managers.

4.3 Fair Allocation of Edge Revenues Among Two Managers

The tolls introduced above are chosen to support the socially optimal flow and correspond to *Pigouvian tolls*, that is, tolls equal to the marginal externality imposed by an additional infinitesimal user on each edge. Formally, for latency function $\ell_i(x)$, the Pigouvian toll is $t_i = x_i \cdot \ell_i'(x_i)$, ensuring that selfish routing aligns with social optimality. Given these tolls, the resulting edge revenues are:

$$r_1 = 5 \cdot 5 = 25, \quad r_2 = 4 \cdot 4 = 16, \quad r_3 = 3 \cdot 3 = 9.$$

We aim to allocate the three edges to our two managers in such a way that each of the following fairness properties is demonstrated by one allocation:

- **Envy-Free (EF):** An allocation is envy-free if no agent prefers another agent's bundle over their own. Formally, for every pair of agents i and j, we require $v_i(A_i) \geq v_i(A_j)$, where A_k denotes the bundle allocated to agent k and v_i is agent i's valuation function. EF guarantees full fairness in the sense that every manager believes they received at least as much value as any other.
- **EFX (Envy-Free up to any good):** An allocation is EFX if no agent envies another after the removal of any single good from the other agent's bundle. That is, for all i, j and for all $g \in A_j$, it holds that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$.
- **EF1 (Envy-Free up to one good):** EF1 relaxes EF even further by requiring that envy disappears after removing *some* single good from the envied bundle. Formally, for all i, j, there exists a good $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$.
- **Not even EF1:** This class includes allocations where an agent still envies another even after the removal of any single item from the envied bundle. That is, there exists i, j such that for all $g \in A_j$, $v_i(A_i) < v_i(A_j \setminus \{g\})$.

We denote the revenue bundle assigned to a manager as the sum of revenues of the edges they control.

4.3.1 Envy-Free Allocation

We do assign edge e_1 (with revenue 25) to Manager A and edges e_2 and e_3 (with revenues 16 and 9) to Manager B. Then we have:

$$\text{A's utility} = 25, \quad \text{B's utility} = 16 + 9 = 25.$$

Since both managers receive equal utility, no one strictly prefers the other's bundle. Hence, the **allocation is envy-free**. By definition, it also satisfies EF1 and EFX.

4.3.2 EFX but not EF Allocation

An EFX but not EF allocation does not exist in this setting due to the structure of the item values and the number of agents. We consider the allocation of three indivisible items with values 25, 16 and 9 to the two agents whose utility functions are additive. The total value of the items is 50 and all items must be allocated.

Suppose an allocation exists that is EFX but not EF. Then, by definition, some agent envies the other, but this envy disappears upon the removal of any single item from the envied bundle. In this instance, at least one agent must receive two items and the other receives one. There are exactly three possible partitions of the item set into a singleton and a pair.

$$\{\{25\},\{16,9\}\} \quad \& \quad \{\{16\},\{25,9\}\} \quad \& \quad \{\{9\},\{25,16\}\}.$$

In the first case, if one agent receives the item of value 25 and the other receives the items worth 16 and 9, the resulting utilities are equal. This allocation is EF and hence cannot be used to demonstrate an allocation that is EFX but not EF. In the second and third case, the agent receiving the singleton item has strictly less utility than the other, leading to envy. To satisfy EFX, removing either item from the envied bundle must eliminate this envy. However, due to the magnitude of the highest-valued item, removing the lower-valued item from the envied bundle does not sufficiently reduce the envied utility to eliminate envy. Specifically, the value gap remains strictly positive even after the removal of any single item, violating the EFX condition. Therefore, in all non-EF allocations, the agent experiencing envy continues to do so after the removal of at least one item from the envied bundle. As a result, no allocation satisfies the EFX condition without also satisfying EF. This proves that, in our specific game, **no allocation exists that is EFX but not EF**.

4.3.3 EF1 but not EFX Allocation

Consider the following allocation of edges:

- Manager A receives edge e_2, with revenue $r_2 = 16$.
- Manager B receives edges e_1 and e_3, with total revenue $r_1 + r_3 = 25 + 9 = 34$.

The resulting utilities are $u_A = 16$ & $u_B = 34$. Manager A envies Manager B since $16 < 34$, so the allocation is not EF. We examine whether envy disappears when a single item is removed from B's bundle. If item e_1 is removed, B is left with only e_3, which has a revenue of 9. Since $16 > 9$, agent A no longer envies B. Therefore, the *allocation satisfies the EF1 criterion.*

- If item e_1 is removed from B's bundle, B retains only e_3, which yields utility 9. Since A's utility is 16 and $16 > 9$, A does not envy B in this case.
- If item e_3 is removed from B's bundle, B retains only e_1, which yields utility 25. Since $16 < 25$, A still envies B.

Since there exists at least one item whose removal does not eliminate A's envy, the *allocation fails the EFX criterion.* It **satisfies EF1 but not EFX**.

4.3.4 Allocation That Is Not Even EF1

Consider the following allocation:

- Manager A receives edge e_3, with revenue $r_3 = 9$.
- Manager B receives edges e_1 and e_2, with total revenue $r_1 + r_2 = 25 + 16 = 41$.

The resulting utilities are $u_A = 9$ *&* $u_B = 41$. Manager A envies B since $9 < 41$, so the allocation is not EF. We examine whether removing any single good from B's bundle eliminates A's envy.

- If e_1 is removed, B retains e_2, which yields utility 16. Since A's utility is 9 and $9 < 16$, A continues to envy B.
- If e_2 is removed, B retains e_1, which yields utility 25. Since $9 < 25$, A still envies B.

As A continues to envy B regardless of which item is removed from B's bundle, the *allocation does not satisfy the EF1 condition.* We thus conclude that this allocation is **not EF**, **not EFX** and **not even EF1**.

4.3.5 Got No Incentive to Undercut Pigouvian Tolls

Our claim is that if each edge is managed by a profit-maximizing agent and the tolls are set to the Pigouvian values $(5, 4, 3)$, then no agent has an incentive to unilaterally lower their toll below this value, assuming the other tolls remain fixed.
Let manager e consider deviating by lowering t_e to $t'_e < t_e$ while other tolls remain fixed and the resulting equilibrium flow be x'_e. Then, the manager's revenue is:

$$R_e(t'_e) = t'_e \cdot x'_e.$$

We analyze whether this revenue is higher than the original revenue $R_e(t_e) = t_e \cdot x_e$. Assume differentiability of $x_e(t_e)$ and define:

$$R_e(t_e) = t_e \cdot x_e(t_e).$$

Then the derivative of revenue w.r.t. the toll is:

$$\frac{dR_e}{dt_e} = x_e(t_e) + t_e \cdot \frac{dx_e}{dt_e}.$$

Since demand is routed according to Wardrop equilibrium, increasing t_e reduces x_e ($\frac{dx_e}{dt_e} < 0$). Setting the derivative to zero to find the local maximum:

$$0 = x_e(t_e) + t_e \cdot \frac{dx_e}{dt_e} \quad \Rightarrow \quad t_e = -\frac{x_e(t_e)}{\frac{dx_e}{dt_e}}.$$

But this is exactly the toll that internalizes the marginal congestion externality, that is, $t_e = x_e \cdot \ell'_e(x_e)$. Hence, the revenue function is maximized at the Pigouvian toll. Therefore, any deviation $t'_e < t_e$ leads to:

$$R_e(t'_e) < R_e(t_e),$$

and the manager earns less revenue.

Since each manager maximizes their revenue by setting the toll to the marginal cost and any decrease in toll leads to strictly lower revenue, no manager has an incentive to lower their toll below the threshold and the toll profile $(5, 4, 3)$ is a Nash equilibrium among toll-setting managers, even when one manager owns two edges.