VLDB 2025
WELCOME TO LONDON

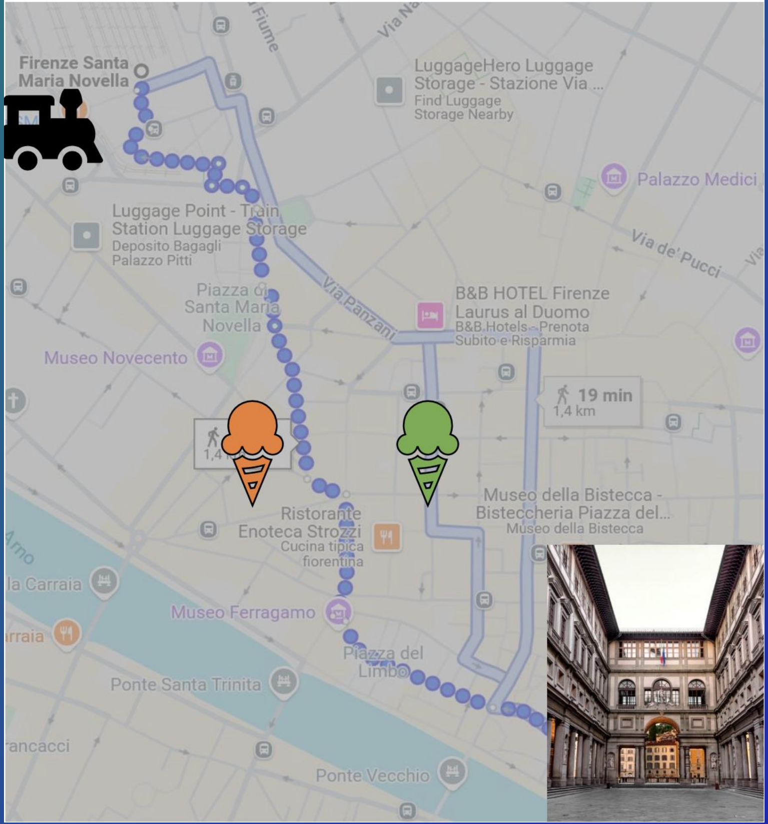
Beyond Shortest Paths: Node Fairness in Route Recommendation

Antonio Ferrara, David Garcia-Soriano, and Francesco Bonchi

CENTAI

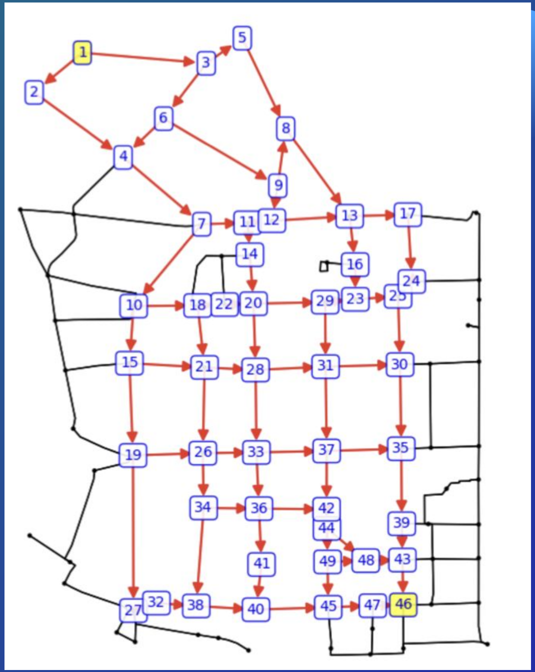
Motivating Scenario

The tourism office of the municipality of Florence produces an **app for helping tourists walking around the city center**, moving in between the key attractions and historic landmarks. Shortly after the introduction of the app, an **ice-cream parlor**, located in a strategic position for tourists flowing in between the Central Train Station and the Uffizi Gallery, sees a sudden **drop of its sales.** Investigating the issue, they realize that the new app was routing all the tourists moving in between the two landmarks, **always through the same shortest path**, greatly reducing the visibility of the ice-cream parlor which was located along a slightly longer path. At the same time, a newly opened ice-cream parlor located on the recommended shortest path, sees its sales skyrocketing.

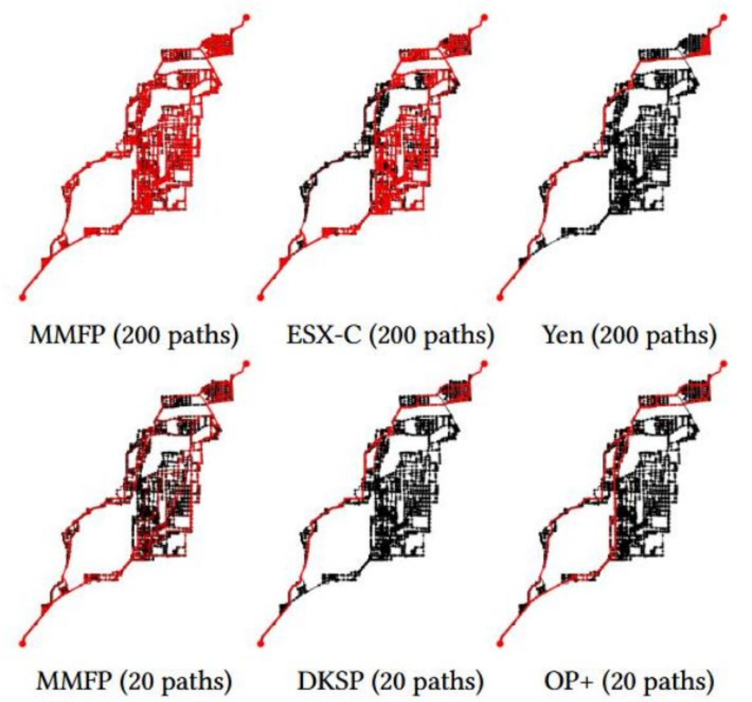


Goal:

"Guarantee a fair distribution of visits across network nodes while maintain near-optimal routes."



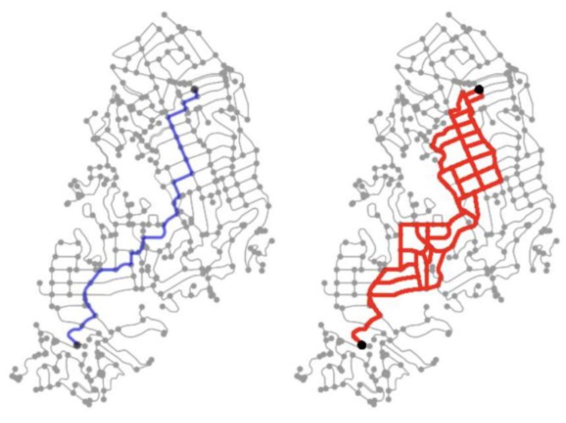
Related work: Diversity in point-to-point path queries



- Lots of work, several slightly different problem statements:
  - E.g., find a set of k paths that are sufficiently dissimilar and as short as possible
- Diversity and Fairness are related notions, however:
  - Fairness guarantees some level of diversity
  - while diversity does not provide any fairness guarantee
- Diversity is necessary yet not sufficient condition for fairness

What kind of paths should be allowed?

- In a weighted graph the s-t shortest path is often unique
- We need to relax the requirement of being the shortest to have a basis for fairness
- Instead of using a tolerance parameter, we define forward paths

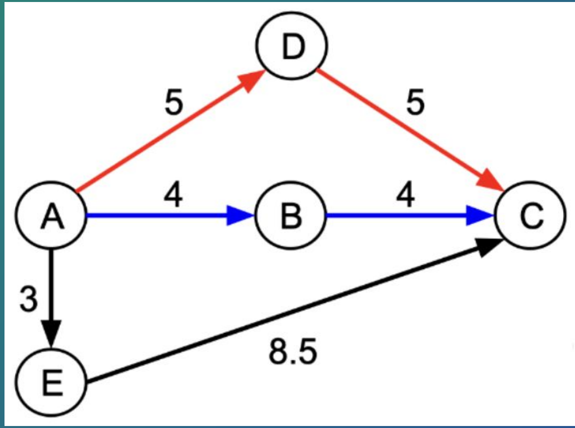


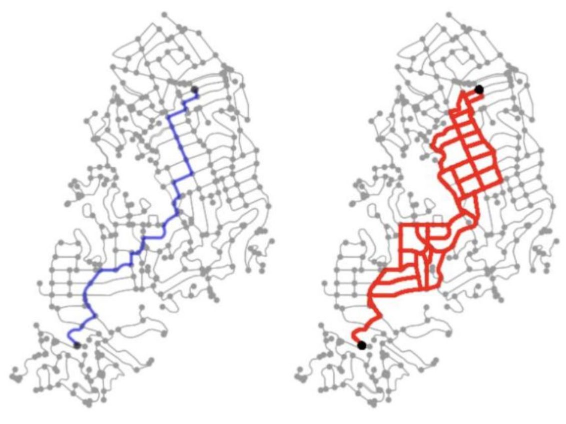
First contribution: Forward Paths

Definition*:*

A s-t forward path is a path that, at each step, decreases the distance to the target

- Aligns well with user preferences in real-world applications
  - Always progressing toward their destination, no backtracking or moving in circles





First contribution: Forward Paths

Definition 1 (s-t-Forward path). *An s-t-path $P = (v_1 = s, \ldots, v_k = t)$ in a graph G is called forward if for $i = 1, \ldots, k-1$,*

$$d(v_{i+1}, t) < d(v_i, t) \ ,$$

where d represents the shortest path distance in G.

Proposition 1: Let G be a directed (weighted or unweighted) graph. Any shortest path in G is a forward path.

Proposition 2: Let G be a directed unweighted graph. An s-t-path is a shortest path if and only if it is a forward path.

First contribution: Forward Paths

- We empirically show that in many real-world weighted graphs, **forward paths are very close in length to the shortest path** and **they're many more**
  - with a minimal increase in the worst-case forward path length
  - the number of locations visited by the union of all forward paths increases significantly

Dataset	# nodes	# edges	SP length	LFP length	LFP/SP length	SP nodes	FP nodes	FP/SP nodes
Piedmont, California	352	937	1 860.07	2 061.58	1.11 ± 0.11	19.91	39.48	1.81 ± 0.66
Essaouira, Morocco	1 277	3 429	3 650.36	3 967.11	1.13 ± 0.14	37.20	107.03	2.52 ± 1.33
Florence, Italy	6 096	11 737	6 909.52	7 287.29	1.05 ± 0.05	75.10	121.27	1.58 ± 0.65
Buenos Aires, Argentina	17 890	37 474	9 065.83	9 642.57	1.06 ± 0.04	89.10	387.50	4.00 ± 2.12
Kyoto, Japan	44 828	118 087	8 501.42	9 243.75	1.09 ± 0.06	117.90	536.77	4.16 ± 2.21
Florida, USA	1 070 376	2 687 902	$4.12 \cdot 10^6$	$4.22 \cdot 10^6$	1.02 ± 0.02	1 194.27	2 720.63	2.33 ± 0.95
Eastern USA	3 598 623	8 708 058	$4.39 \cdot 10^6$	$4.57 \cdot 10^6$	1.04 ± 0.02	1 924.43	7 046.97	3.28 ± 1.45

Max-Min Fairness

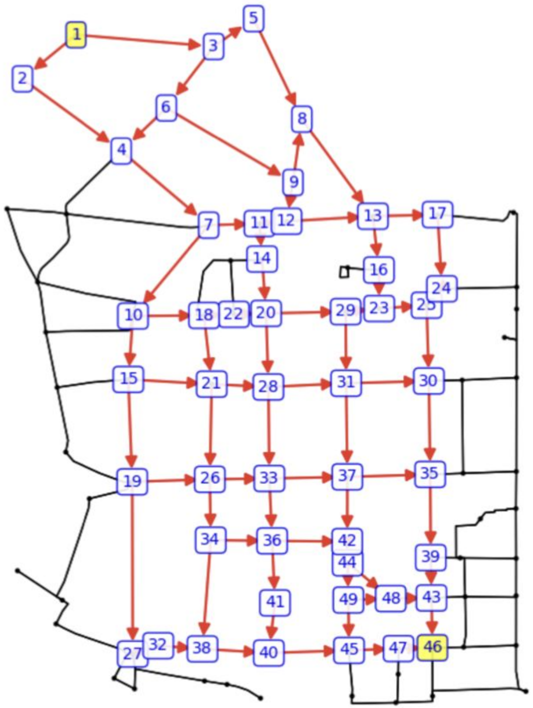
Definition: A distribution of paths is **Maxmin-Fair** if it is impossible to improve the probability of visiting a node without decreasing the probability of visiting another node which is already visited less

- Grounded in **game theory**
- Based on Rawls' theory of **distributive justice**
- Introduced by Garcia-Soriano and Bonchi [1,2]
    - Many authors built up on this definition of fairness
- Definition of **individual fairness**
    - Ex-ante fairness

1) "Fair-by-design matching", D. García-Soriano and F. Bonchi, DAMI, 2020
2) "Maxmin-fair ranking: individual fairness under group-fairness constraints", D. García-Soriano and F. Bonchi, SIGKDD, 2021

Our Problem

Given a weighted graph (road network), a source node s, a destination node t, produce a **probabilistic distribution over forward paths** from s to t such that it is **maxmin-fair for all the eligible nodes** (those that belong to at least a forward path).



Path	Probability (%)
[1, 2, 4, 7, 10, 15, 19, 27, 32, 38, 40, 45, 47, 46]	10.4
[1, 3, 6, 9, 12, 13, 16, 23, 25, 30, 35, 39, 43, 46]	8.3
[1, 3, 6, 9, 12, 13, 17, 24, 25, 30, 35, 39, 43, 46]	8.3
[1, 3, 5, 8, 13, 16, 23, 25, 30, 35, 39, 43, 46]	8.3
[1, 3, 5, 8, 13, 17, 24, 25, 30, 35, 39, 43, 46]	8.3
[1, 3, 6, 4, 7, 10, 15, 19, 27, 32, 38, 40, 45, 47, 46]	6.2
[1, 2, 4, 7, 10, 18, 22, 20, 29, 31, 37, 42, 44, 49, 48, 43, 46]	5.2
[1, 2, 4, 7, 11, 14, 20, 29, 31, 37, 42, 44, 49, 48, 43, 46]	5.2
[1, 2, 4, 7, 10, 18, 22, 20, 28, 33, 36, 41, 40, 45, 47, 46]	3.5
[1, 2, 4, 7, 11, 14, 20, 28, 33, 36, 41, 40, 45, 47, 46]	3.5
[1, 2, 4, 7, 10, 15, 21, 26, 34, 36, 41, 40, 45, 47, 46]	3.5
[1, 2, 4, 7, 10, 18, 21, 26, 34, 36, 41, 40, 45, 47, 46]	3.5
[1, 3, 6, 4, 7, 10, 18, 22, 20, 29, 31, 37, 42, 44, 49, 48, 43, 46]	3.1
[1, 3, 6, 4, 7, 11, 14, 20, 29, 31, 37, 42, 44, 49, 48, 43, 46]	3.1
[1, 3, 6, 4, 7, 10, 18, 22, 20, 28, 33, 36, 41, 40, 45, 47, 46]	2.1
[1, 3, 6, 4, 7, 11, 14, 20, 28, 33, 36, 41, 40, 45, 47, 46]	2.1
[1, 3, 6, 4, 7, 10, 15, 21, 26, 34, 36, 41, 40, 45, 47, 46]	2.1
[1, 3, 6, 4, 7, 10, 18, 21, 26, 34, 36, 41, 40, 45, 47, 46]	2.1
[1, 2, 4, 7, 10, 18, 22, 20, 28, 33, 37, 42, 44, 49, 48, 43, 46]	1.7
[1, 2, 4, 7, 11, 14, 20, 28, 33, 37, 42, 44, 49, 48, 43, 46]	1.7
[1, 2, 4, 7, 10, 15, 21, 26, 34, 38, 40, 45, 47, 46]	1.7
[1, 2, 4, 7, 10, 18, 21, 26, 34, 38, 40, 45, 47, 46]	1.7
[1, 3, 6, 4, 7, 10, 18, 22, 20, 28, 33, 37, 42, 44, 49, 48, 43, 46]	1.0
[1, 3, 6, 4, 7, 11, 14, 20, 28, 33, 37, 42, 44, 49, 48, 43, 46]	1.0
[1, 3, 6, 4, 7, 10, 15, 21, 26, 34, 38, 40, 45, 47, 46]	1.0
[1, 3, 6, 4, 7, 10, 18, 21, 26, 34, 38, 40, 45, 47, 46]	1.0

Technical challenges and contributions

1. How to enumerate all forward paths (which can be exponential in number)?

→ We can compute a Directed Acyclic Graph (**DAG**) representing all the forward paths

→ Avoid the need to explicitly enumerate the set of all possible forward paths

2. How to compute the maxmin-fair distribution over forward paths?

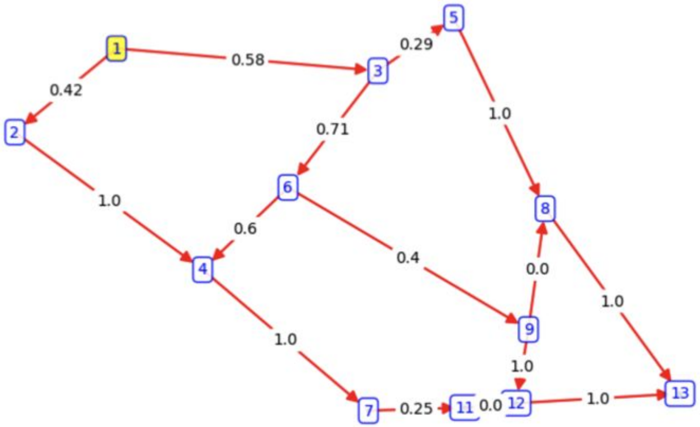
→ We design a **Linear Programming Flow Problem** which allows us to derive the maxmin-fair distribution over forward paths

3. How to store and sample from the maxmin-fair distribution?

→ Our method does not materialize the distribution explicitly

→ Just assigns the correct transition probability to each arc of the DAG

→ We show that a **random-walk** on this probabilistic graph from s to t is equivalent to sampling from the underlying maxmin-fair distribution



Algorithm 1: DAG of Forward Paths

Algorithm 1 DAG-FP: Creates a DAG containing the Forward Paths from s to t

Input: Graph $G = (V, E, \ell)$, source node s, target node t

Output: DAG with Forward Paths

$dist\_from\_s \leftarrow \text{distance}(G, s, V)$

$dist\_to\_t \leftarrow \text{distance}(G, V, t)$

$reachable\_from\_s \leftarrow \{i \mid dist\_from\_s[i] < \infty\}$

$reachable\_to\_t \leftarrow \{i \mid dist\_to\_t[i] < \infty\}$

$reachable \leftarrow reachable\_from\_s \cap reachable\_to\_t$

$G' \leftarrow G.\text{induced\_subgraph}(reachable)$

for all (u, v) in $G'.edges$ **do**

  if $dist\_to\_t[u] \leq dist\_to\_t[v]$ **then**

    remove (u, v) from G'

  end if

end for

return G'

Same asymptotic computational runtime as solving a single shortest-path query

Algorithm 2: Maxmin-Fair Forward Paths

$$
\begin{aligned}
\text{maximize} \quad & \lambda \in \mathbb{R} \quad \text{subject to} \\
& \sum_{u \mid (u,t) \in E} f_{u,t} = 1 \\
& \sum_{u \mid (s,u) \in E} -f_{s,u} = -1 \\
& \sum_{u \mid (u,v) \in E} f_{u,v} - \sum_{w \mid (v,w) \in E} f_{v,w} = 0 \quad \forall v \neq s,t \\
& \lambda - \sum_{u \mid (u,v) \in E} f_{u,v} \leq 0 \quad \forall v \notin K \\
& - \sum_{u \mid (u,v) \in E} f_{u,v} \leq -\alpha_v \quad \forall v \in K \\
& f_{u,v} \geq 0 \quad \forall (u,v) \in E
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
\text{minimize} \quad & d_t - d_s - \sum_{v \in K} \alpha_v w_v \quad \text{subject to} \\
& d_v - d_u - w_v \geq 0 \quad \forall (u,v) \in E \\
& \sum_{v \notin K} w_v = 1 \\
& w_v \geq 0 \quad \forall v \in V \\
& d_v \in \mathbb{R} \quad \forall v \in V
\end{aligned} \tag{2}
$$

Algorithm 2 MMFP - MaxMin Fair Forward Paths

Input: DAG G, source node s, target node t
Output: Encoding of a maxmin-fair distribution for s, t-forward paths

```
K ← ∅
while K ≠ V do
    Solve LP (1) and its dual LP (2);
    let λ*, {f*_{u,v}}, {w*_v} be the optimum values
    K' ← {v ∉ K | w*_v > 0}
    K ← K ∪ K'
    for all v ∈ K' do
        α_v ← λ*
    end for
end while
Return the flow values f*_{u,v}
```

- It can be solved in **polynomial time** by solving a sequence of small linear programs
- MMFP is **Optimal**

Practical implications and deployment

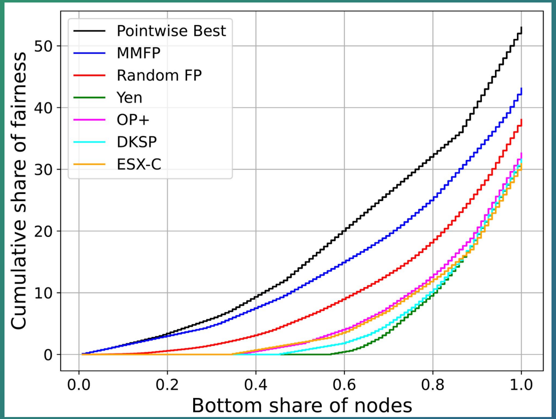
- Same query can be served many times for different users
  → Well suited when computing high volumes of different alternative paths
  → Methods from the literature on diverse path recommendations face significant challenges to recommend several different alternative paths

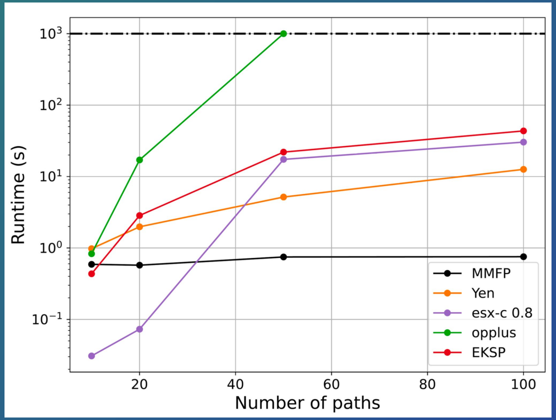
- Each *s*-*t* pair defines its own problem instance
  → Highly parallelizable

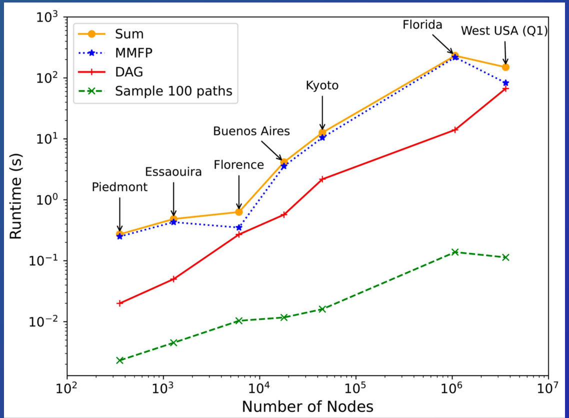
- The maxmin-fair distribution for an *s*-*t* pair needs only be computed once and stored
  → At query time only sampling from the distribution is needed

Main results and takeaways

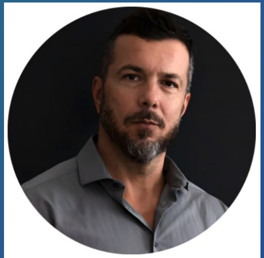
- MMFP reduces inequalities
- Well suited to serve a high amount of paths
- Empirical results confirm theoretical runtimes







THANKS!



Antonio Ferrara
David Garcia-Soriano
Francesco Bonchi

antonio.ferrara@centai.eu

Scan me to download the paper

CENTAI

TU Graz

UPC
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

eurecat
Centre Tecnològic de Catalunya