Fairness Driven Efficient Algorithms for Sequenced Group Trip Planning Query Problem

Napendra Solanki
Indian Institute of Technology Ropar
Rupnagar, India
napendra.21csz0001@iitrpr.ac.in

Shweta Jain
Indian Institute of Technology Ropar
Rupnagar, India
shwetajain@iitrpr.ac.in

Suman Banerjee
Indian Institute of Technology Jammu
Jammu, India
suman.banerjee@iitjammu.ac.in

Yayathi Pavan Kumar S
Indian Institute of Technology Ropar
Rupnagar, India
yayathi.21csz0014@iitrpr.ac.in

ABSTRACT

The Group Trip Planning Query Problem (GTP) is a well-researched spatial database problem. Given a city road network with Point-of-Interests (PoIs) representing vertices divided into different categories, GTP aims to suggest one PoI from each category to minimize the group's total distance traveled. This paper focuses on sequenced GTP with pre-determined category visit order, studied under the constraints of fairness, and referred to as sequenced *Fair Group Trip Planning Query Problem (Fair-GTP)*. While GTP aims to minimize the group's total travel time, Fair-GTP seeks to minimize the maximum time difference between any two agents in the group. Although solving group trip planning queries is NP-hard, we present polynomial time algorithms for finding optimal paths for both sequenced GTP and Fair-GTP. Our second significant result provides a bound on the price of fairness (PoF) representing the ratio of optimal path cost in sequenced Fair-GTP to optimal path cost in sequenced GTP. We show that while the PoF can go arbitrarily bad for general sequenced Fair-GTP solutions, restricting to Pareto-optimal solutions bounds the PoF by $(2b - 1)$, where b denotes the number of agents traveling in the group. We further show that this bound is tight. Finally, we present the performance analysis of our algorithms on real-world datasets, demonstrating that our solution approach recommends PoIs within reasonable computational time, and in practice, PoF is below 2.

KEYWORDS

Spatial Database; Group Trip Planning; Road Network; Fairness; Pareto-optimal paths

ACM Reference Format:
Napendra Solanki, Shweta Jain, Suman Banerjee, and Yayathi Pavan Kumar S. 2023. Fairness Driven Efficient Algorithms for Sequenced Group Trip Planning Query Problem. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Due to the advancement of wireless internet and handheld devices tracking the location of moving objects has become easy. Hence, several trajectory datasets are publicly available on different repositories [15, 21, 39]. These datasets are used to solve many real-life problems, including driving behavior prediction [28], trip planning problem [11], route recommendations [14], and many more. Hence managing (storing, querying, etc.) and mining these datasets are of utmost importance and have led to a new domain called Spatial Database [33]. Several internet giants (such as Google) specialize in analyzing spatial databases.

The Group Trip Planning Query Problem is a spatial database challenge centered around a weighted, undirected graph representing a city's road network. The graph comprises vertices denoting point of interest (PoI), and edges indicating distance-weighted road segments. PoIs are categorized according to various groups such as parks, cafeterias, restaurants, bars, movie theaters, etc. To comprehend the GTP problem, one can consider a scenario where a group of friends intends to plan a trip to the city, visiting distinct PoIs from each category, beginning from their respective starting points and concluding at their final destination. In sequenced GTP, the order in which the friends visit each PoI category is predetermined. For instance, they have decided to visit coffee, then a movie, followed by lunch, bowling, and finally dinner in a particular order. GTP or sequenced GTP aims to identify a path that includes at least one node from each category. Until now, researchers have focused on minimizing the total distance traveled by all the friends in sequenced GTP [2, 6, 20, 37].

This paper studies the problem of sequenced GTP from the lens of fairness. We quantify the fairness of a path by the maximum difference in the distances traveled by any two friends. Thus, the goal of the Fair-GTP problem is to output the path, minimizing the maximum difference. In this paper, we show that an optimal path (minimizing the aggregated distance) can be extremely unfair to an agent, requiring an agent to travel a much larger distance than the other agents. Such a path is not desirable due to two reasons. Firstly, it makes one agent envy another agent, which may strain their friendship or cause them not to take the trip altogether. Secondly, this path requires one agent to travel a longer distance alone instead of the other potentially longer path, but traveling together may be more fun. Apart from the group trip planning problem, the Fair-GTP problem finds its applications in other applications such as supply chain[42] and federated learning[40]. To understand the connection between the Fair-GTP and supply chain, consider the source nodes representing the location of various producers from

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

where the raw material is coming. Different categories represent different manufacturing processes, with each node in one category representing the same task. Finally, the combined products need to reach a warehouse. In this case, the destinations will not be different points but will be common points. Then a Fair-GTP problem will lead to choosing that manufacturing process that will minimize the maximum difference traveled by the raw material from any producer, thus reducing envy among the producers. In the federated learning application, consider that different clients possess the data and are trying to learn a global model which could be trained on several servers available. The problem is then to choose the server for training the global model to minimize the envy between the two clients; that is, a fair solution would be to choose a server minimizing the difference between the communication times of any two clients. Since the problem of Fair-GTP is useful in many applications, we will refer to friends, producers, and clients as agents for the remainder of the paper.

Motivated by real-world applications described above, we study the problem of sequenced Fair-GTP. To the best of our knowledge, the only work considering the fairness notion in sequenced GTP [34] provides an exhaustive search method to find the solution to Fair-GTP leading to a non-practical solution. Further, a naive fair solution could be arbitrarily bad. To illustrate, imagine a path where each agent travels an equal but extremely long distance versus a slightly longer path for one agent but significantly shorter for all others (see figure 2). To avoid this, we focus on Pareto-optimal paths that offer a better outcome for at least one agent without harming others. We also limit the price of fairness, the cost of the fair path relative to the optimal path. Our findings show that while the price of fairness can be poor for any fair path, it can be bounded for the Pareto-optimal fair path.

It is also worth noting that all the current approaches to solve sequenced GTP problem involve generating all possible paths for each of the k categories, resulting in a time complexity of $O(n^k)$ where n is the number of nodes in each category. To address this issue, we create a complete $(k+2)-$partite graph, providing the shortest distance between nodes of categories i and $i+1$. We then observe that all agents travel via the same path once the PoIs from the first and last categories are fixed, leading to a further reduction of the graph to a complete $4-$partite graph with nodes from the source, the first category, last category, and the destination. These constructions reduce the time complexity from $O(n^k)$ to $O(n^4)$. To this end, our specific contributions in sequenced Fair-GTP include:

- We demonstrate that naive implementation of Fair-GTP can lead to poor solutions (Lemma 3).
- We introduce the problem of finding fair solutions in the domain of Pareto-optimal solutions, with tight bounds on the price of fairness (Lemma 6 and 7).
- We present polynomial-time algorithms for sequenced GTP and sequenced Fair-GTP that produce Pareto-optimal and fair solutions (Algorithm 2).
- We provide empirical evidence of the superiority of our proposed algorithms over existing approaches in terms of optimal cost and running time.

The paper is structured as follows. Section 2 reviews related literature. Section 3 introduces preliminary concepts related to group trip planning queries, fairness, and other definitions. Section 4 shows theoretical guarantees. Section 5 presents the proposed solution approach with detailed analysis and illustrations. Section 6 reports on the experimental evaluation of the proposed approach. Section 7 concludes the study and outlines future research directions.

2 RELATED WORK

2.1 Group Trip Planning

When considering road networks, trip, and route planning queries are primarily viewed from the lens of the traveling salesman problem [19]. In a typical trip planning problem, a single user needs to travel PoIs from each category and wants to minimize the total travel time. Since this problem is a special version of the traveling salesman problem, it is NP-hard to compute the optimal trip. Several works propose the approximation algorithm [23]. While general trip planning is hard to solve, it has been shown that sequenced trip planning exhibit polynomial time algorithms[32]. All these techniques primarily assume a single user and only work for group trip planning. The group trip planning problem was first introduced by Hashem et al. [20] and has been extensively studied after that through various techniques, including nearest neighbor (NN), group nearest neighbor (GNN), trip planning (TP)[24] queries in both euclidean space and road networks. Many of these techniques use spatial data structures such as R-tree or R*-tree. Subsequent techniques were proposed to obtain the solutions to group trip planning much more efficiently. Ahmadi and Nascimento [1] studied this problem and introduced the Progressive Group Neighbour Exploration framework that leads to the optimal solution. Ahmadi and Nascimento [2] proposed a dynamic programming-based solution, Iterative Backward Search, which uses the suffix optimality principle. Many techniques provide efficient algorithms using the pruning technique based on geometric properties of ellipses [12, 19, 23, 25, 32] of efficient paper. None of the existing works propose a polynomial-time efficient algorithm to produce the optimal solution to the group trip planning problem. In this paper, we fill this research gap and propose a graph-based algorithm that provides the optimal solution in polynomial time.

2.2 Variants of Group Trip Planning

Many variants to group trip planning problems have been looked at in the literature. For example, Barua et al. [6] studied the weighted version where the PoIs are associated with the weights that signify the utility if the PoI is visited. The goal is, thus, not to minimize the total traveled distance but to maximize the overall utility. Tabassum et al. [37] introduced the concept of dynamic group in the Group Trip Planning Query Problem. In contrast with the traditional GTP Query Problem, in this case, the group can change dynamically throughout the trip, i.e., members can leave or join the group. The proposed solution methodology works for Euclidean Plane only. Mahin and Hashem [30] introduced the notion of flexible PoI in the GTP Query Problem. In particular, they studied the GTP Query Problem considering the following three noble features: (i) ensures a complete trip for visiting more than two locations, (ii) allows visiting both fixed and flexible locations, and (iii) provides true ridesharing services instead of taxi like ride-sourcing services by matching a group of riders' flexible trips with a driver's fixed trip. Their proposed solution methodology process the queries in real-time, and its efficiency depends upon the number of riders and the PoIs to be explored. Lima and Hashem [26] studied a variant of the GTP Query Problem where the group can expand and shrink during the trip, and the underlying space is obstructed. Even though many variants were considered, the primary goal remained the same, i.e., to minimize the total travel time to the agents.

We study the following variant of the group trip planning problem. Instead of minimizing the total travel distance to the agents, we aim to find the Pareto-optimal path, i.e., the path that is not worst for all the agents and minimizes the maximum envy between any two agents. An elementary study of Minimizing the maximum envy in group trip planning is considered by Singhal and Banerjee [34], but this paper does not consider the Pareto-optimal solution. We show that a naive path with minimum envy can lead to a path with arbitrary high cost, so it is necessary to consider a Pareto-optimal solution. Apart from considering Pareto-optimal solutions, we significantly contribute over Singhal and Banerjee [34] by providing the polynomial time optimal algorithm to sequenced Fair-GTP problem and by theoretically bounding the price of fairness.

2.3 Fairness in Other Domains

The concept of envy-free is not new and has been extensively studied in the context of resource allocation to multiple agents. Course matching, rent sharing among roommates, spitting tax fees, and division on the contested territory are a few instances of equitable division in the actual world. This includes divisible goods allocation[5, 17, 36] and indivisible goods allocation [3, 7, 8, 27] . In the resource allocation problem, envy-freeness implies the value of the good that one agent is receiving should not be more than the value of the good received by another agent. While there are some algorithms to find envy-free solutions in divisible good setting[4, 5], it is shown that envy-free solutions may not exist in indivisible good setting[9]. To circumvent this, appropriate relaxations (envy-free up to one good) are proposed in the literature [9]. An envy-free solution could be really bad in terms of the primary objective of maximizing social welfare. Recent works [10] have also studied the impact of fairness on social welfare in the form of the price of fairness. Our problem is fundamentally different from the fair division problem, as in GTP, the edges (resources) are shared amongst the agents as opposed to the distribution of the resources.

When considering a graph or road network, the fairness issue has been considered while solving routing and load balancing problems in the literature. These studies fit into the broad spectrum ranging from fair resource allocation in optical networks [31] to fair network design for hazmat road transportation [41]. In particular, several kinds of problems have been studied under different fairness notions such as Fair Bandwidth Allocation [13], Fair Routing for Underwater Networks [16], Fair Cost Allocation for Ride-sharing Services [29], Fairness-Aware Load Balancing [22] and many more. The setting is fundamentally different from our setting since, in all the above cases, fairness concerning edges is considered, whereas we are considering fairness among the agents traveling through this road network.

This paper defines the appropriate notion of envy-freeness in GTP and shows that an envy-free solution may not exist. We also provide appropriate relaxation to envy-free solution $\epsilon-$envy free solution. Finally, we connect fairness with the primary goal of minimizing the total travel time and study the price of fairness in this setting.

3 PRELIMINARIES

Let $B = \{1, 2, \ldots, b\}$ be the set of agents planning a trip to visit one PoI from each of the k categories. These k categories may include coffee shops, theatres, shopping malls, game zones, etc. The set of PoIs, therefore, partitions all the nodes in k sets denoted by $\{V^1, V^2, \ldots, V^k\}$. Let the origin and destination of B agents be denoted by sets $O = \{o_1, o_2, \ldots, o_b\}$ and $T = \{t_1, t_2, \ldots, t_b\}$ respectively. Note that the number of sources and the number of destinations could be different, as in the case of distributed learning with b sources and 1 destination. Such cases can always be tackled by keeping $t_1 = t_2 = \ldots = t_b$. Let G be a weighted graph representing a city's road network and is provided in the group trip planning (GTP) problem. The vertices of interest are the sequenced vertices $V = \{O \cup T \cup V^1 \cup \ldots \cup V^k\}$. The edges $E(G) = \{e_1, e_2, e_3, \ldots.., e_m\}$ of graph G are the highways connecting the PoIs, sources, and destinations of these agents. Each edge weight represents the distance of one vertex to another. For notation convenience, we will further denote v_i^j to be the i^{th} PoI from j^{th} category. The agents plan to travel in a predetermined order, such as first going to a coffee shop, then a movie, lunch, gaming area, etc. Without loss of generality, we assume that the sets of PoIs are arranged in this predetermined sequence, i.e., V^1 represents the set of vertices where the agents plan to visit first, and V^k represents the set of vertices from the last category. We can now define a valid path for sequenced group trip planning as follows:

Definition 1 (Valid Path). *We say a path $P = \{v^1_{\alpha_1}, p_1, v^2_{\alpha_2}, p_2, \ldots, v^k_{\alpha_k}\}$ is valid for sequenced set of vertices $\{V^1, V^2, \ldots, V^k\}$ iff each $v^j_{\alpha_j}$ belong to j^{th} category, i.e., $v^j_{\alpha_j} \in V^j$. Here, p_i represent an arbitrary path connection $v^i_{\alpha_i}$ and $v^{i+1}_{\alpha_{i+1}}$.*

A valid path is thus a path that visits at least one PoI from each category in a predetermined sequence. Note that fixing the set of PoIs visited may make different paths possible. For example, one valid path could be $P = \{v^1_{\alpha_1}, p_1, v^2_{\alpha_2}, p_2, \ldots, v^k_{\alpha_k}\}$ and another path could be $P' = \{v^1_{\alpha_1}, p'_1, v^2_{\alpha_2}, p'_2, \ldots, v^k_{\alpha_k}\}$. Out of all these possible paths, the paths of interest are the ones having the shortest distance, and such paths are denoted by $P = \{v^1_{\alpha_1}, v^2_{\alpha_2}, \ldots, v^k_{\alpha_k}\}$. Path connecting vertices from different categories does not impact fairness. Fairness comes only from source and destination nodes. Let the shortest route from PoI $v^{j_1}_\alpha$ to PoI $v^{j_2}_\beta$ be denoted by the $D(v^{j_1}_\alpha, v^{j_2}_\beta)$. Then the cost of $p = \{v^1_{\alpha_1}, v^2_{\alpha_2}, \ldots v^k_{\alpha_k}\}$ is given by:

$$C(p) = \sum_{i \in B} D(o_i, v^1_{\alpha_1}) + b \sum_{j \in [k-1]} D(v^j_{\alpha_j}, v^{j+1}_{\alpha_{j+1}}) + \sum_{i \in B} D(v^k_{\alpha_k}, t_i) \quad (1)$$

Let $\mathcal{P}$ denote the set of all valid paths. The fundamental problem of sequenced group trip planning is to find a *valid path* to minimize the total distance of the trip, which is defined as follows:

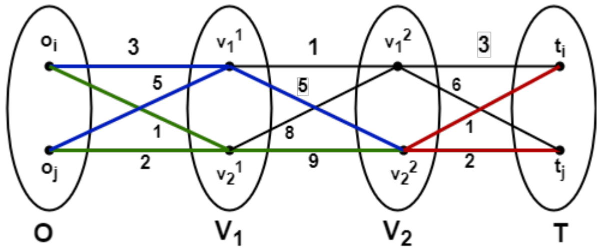


Figure 1: Motivation for Studying Fairness in Group Trip Planning

Definition 2 (Optimal Sequenced Group Trip Planning Problem). *Given the graph, G, Sets $O, V^1, V^2, \ldots, V^k, T$, the optimal sequenced group trip planning problem is to find an optimal valid path p^* that minimizes the aggregated sum of the distances traveled by agents, i.e., $p^* = \text{argmin}_{p \in \mathcal{P}} C(p)$.*

A naive algorithm focused on finding the path and minimizing the total distance could be heavily biased and may result in agents envying each other. Consider a simple example in Figure 1. This figure represents the two agents i and j planning to visit two PoIs from the respective sources to the respective destinations. The (v_1^1, v_2^2) path (total cost from O to T 21) shows the optimal path that any group trip planning algorithm will produce. However, in this path, agent j will travel significantly higher than agent i. Whereas an alternative (v_2^1, v_2^2) path (total cost from O to T 24) is a better option in terms of fairness due to two reasons, first, the distance covered by both the agents is comparable. Second, they are traveling more distance together (in the company of each other) rather than traveling alone. In order to tackle such issues, we introduce the problem of sequenced fair GTP problem. Our work aligns with the work by Singhal and Banerjee [34] where the authors formulated the envy-free trip planning problem. However, there are several things that could be improved in that work. First, it enumerates all possible paths, checks for feasibility, and subsequently searches for the optimal path. As a whole, it is based upon the exhaustive search approach, which leads to the running time of $O(n^k)$ where n is the number of PoIs and k is the number of categories. Secondly, apart from providing the path with minimum envy, we also provide the price of fairness, which bounds the ratio of the cost of the path with minimum envy to the optimal cost. In particular, our results show that the price of fairness can go arbitrarily bad in general. Imagine an example where there is a path minimizing envy but arbitrarily bad for all the agents. Any of the agents will never prefer such a path. Hence we restrict to the paths that are *Pareto-optimal*. We then show that, when restricted to the domain of Pareto-optimal solution, one can guarantee a constant price of fairness. Finally, we provide a polynomial time algorithm to find a Pareto-optimal path with minimum envy. We first define a Pareto-optimal path, followed by the fairness notions we use in our paper.

Definition 3 (Pareto-optimal Path). *In a sequenced Group trip planning (GTP), a path p' is called a Pareto-optimal path if there does not exist any other path p'' such that:*

$$\forall i \in [b],\ D_i(p'') \leq D_i(p') \text{ and } \exists j \in [b] \text{ s.t. } D_j(p'') < D_j(p')$$

Here, $D_i(p)$ denotes the distance traveled by i^{th} agent in path p.

3.1 Fairness notions

We now define the fairness notion, which we call Envy-freeness motivated by resource allocation problem [18, 38].

Definition 4 (Envy-Free Path (EFP)). *We say that a given path p is an envy-free path (EFP) iff $D_i(p) \leq D_j(p)\ \forall i, j \in [b]$.*

It is easy to see that each agent will travel an equal distance, irrespective of their source and destination, in an envy-free path. Further, such a path may not even exist (Figure 1). The reason is that there may be an agent far from all the vertices of each of the PoIs, and hence they will have to travel a longer distance. There are two possibilities to circumvent the issue: minimizing the envy or defining an approximate envy-free path. We now define envy of a valid path and approximate envy-free path below. For a pair of agents i and j, and a given path p, define the envy by agent i from agent j as follows: $\mathcal{E}_{i \to j}(p) = \max\{D_i(p) - D_j(p), 0\}$.

Definition 5 (Envy of a Path). *Envy of a path p is defined as $\max_{i,j} \mathcal{E}_{i \to j}(p)$. It is the maximum difference of distances traveled by a pair of agents in the path p.*

In some cases, a minimum envy path may result in a very high total path cost (See Figure 2 for example). Therefore, it may be required to define a trade-off parameter between the envy and the total path length. This leads us to have the following definition:

Definition 6 (ε–Envy-Free Path (ε–EFP)). *We say that a given path p is ε–envy-free path iff $D_i(p) \leq D_j(p) + \varepsilon\ \forall i, j \in B$.*

It is interesting to ask the following question. For what value of ε does the ε–EFP exist? It is easy to see the following result:

Lemma 1. *ε–EFP always exists for $\varepsilon \geq min_{p \in \mathcal{P}} \max_{i,j} \mathcal{E}_{i \to j}(p)$.*

Proof. Let $p^* = \text{argmin}_{p \in \mathcal{P}} \max_{i,j} \mathcal{E}_{i \to j}(p)$ be the path minimizing the maximum envy and let i^*, j^* be the pair of agents having maximum envy in p^* i.e. $(i^*, j^*) = \text{argmax}_{i,j} \mathcal{E}_{i \to j}(p^*)$. Then for any pair of agents i and j, we have $D_i(p^*) \leq D_j(p^*) + \mathcal{E}_{i \to j}(p^*) \leq D_j(p^*) + \mathcal{E}_{i^* \to j^*}(p^*) \leq D_j(p^*) + \varepsilon$. □

Finally, we define a minimum envy path as follows:

Definition 7. *(Minimum Envy Path) We say that a path p is a minimum envy path (MEP) iff it is ε–envy free path with $\varepsilon = min_{p \in \mathcal{P}} \max_{i,j} \mathcal{E}_{i \to j}(p)$.*

One could also ask for the cost of friendship, which captures how much extra distance an individual has to travel to enjoy the company of friends.

Definition 8 (Cost of Friendship). *Let p_i^* denote the optimal shortest valid path for agent i, which he would have preferred if he traveled alone. Then, for a given path p, we define the cost of friendship of an agent i as $cof_i(p) = D_i(p) - D_i(p_i^*)$. The cost of friendship for the path p is then given as $cof(p) = \sum_{i \in [b]} cof_i(p)$.*

The cost of friendship for an agent i will always be positive because, if the agent were traveling alone, she would have always chosen the quickest route from her source, even though it is likely that she would have had to cover a greater distance with her group.

It is easy to see that the path that minimizes the cost of friendship will also minimize the sum of the distances. And this could lead to high envy between the pair of agents.

Lemma 2. *The path that minimizes the cost of friendship is also a path that minimizes the total distance.*

Proof. For any path p, the Cost of Friendship is given as:

$$cof(p) = \sum_{i \in B} cof_i(p) = \sum_{i \in B} \left[D_i(p) - D_i(p_i^*) \right]$$
$$= \sum_{i \in B} D_i(p) - \sum_{i \in B} D_i(p_i^*)$$

Here, the second term denotes the sum of optimal distances by taking optimal individual paths for a given graph which is a constant irrespective of the chosen path. Hence minimizing the cost of friendship is equivalent to minimizing the total distance. □

Therefore, we will focus on minimizing envy in the rest of the paper. Based on the fairness and efficiency notions defined above, we describe the following problems.

3.2 Problem Definitions

Definition 9 (Sequenced Fair Group Trip Planning Query Problem). *Given a sequenced GTP Query Problem instance $\mathcal{I}$, a sequenced fair group trip planning query problem asks to recommend one PoI from every category such that the envy of the path is minimum. If multiple such paths exist, then output a path that minimizes the cost of friendship.*

Definition 10 (Sequenced ϵ-Fair Group Trip Planning Query Problem). *Given a sequenced GTP Query Problem instance $\mathcal{I}$ along with a value ϵ, the sequenced ϵ-Fair Group Trip Planning Query Problem asks to recommend one PoI from every category such that the total distance traveled by the group is minimized and for any pair of agents the difference between their distance travelled is less than or equal to ϵ.*

The envy-freeness property comes with a price. We quantify this via the price of fairness, which is defined as follows:

Definition 11 (Price of Fairness). *In a group trip planning problem, the price of fairness (PoF) is the ratio of the total distance of the fair path to the total distance of the optimal path traveled by the group. The PoF is a numerical representation of the welfare loss (here, defined as distance) that the group must incur to ensure fairness.*

Let the optimal path be denoted by p^* and the fair path with the least cost be denoted by p_f^*. Then, PoF is given by:

$$PoF = \frac{\sum_{i \in B} D_i(p_f^*)}{\sum_{i \in B} D_i(p^*)} \qquad (2)$$

It is clear that PoF is always greater than or equal to 1. In the next section, we provide the upper and lower bounds on the price of fairness in sequenced group trip planning problems.

4 THEORETICAL RESULTS

Our first result is a negative result which suggests that, in general, the price of fairness can be arbitrarily high.

Lemma 3. *In group trip planning, the price of fairness is unbounded.*

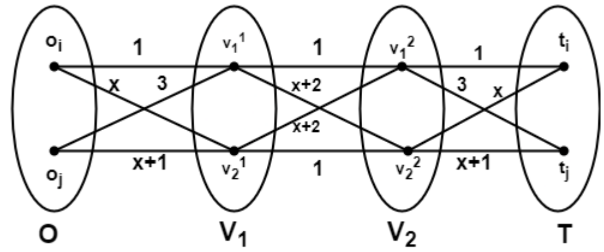


Figure 2: Figure showing PoF is unbounded (Lemma 3)

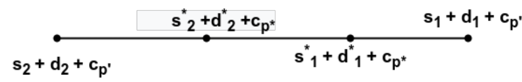


Figure 3: First Case for Lemma 4

Proof. Consider the example shown in figure 2. For any value of $x \in \mathbb{R}^+$, with $x > 2$, the path p^* via the edge $e(v_1^1, v_1^2)$ incurs a total cost of 10 with total envy of 4. While minimum envy path p_f^* with the least cost is the path via the edge $e(v_2^1, v_2^2)$ incurs the total cost of $2x + 4$ with envy of 2. Thus, the price of fairness is given as $\frac{2x+4}{9}$, which can be arbitrarily high as x increases. □

The above example is not very good as the minimum envy path leads to agents traveling much more distance which is not favorable to any of the agents. Therefore, it makes more sense to ask the following question. Does there exist a bound on the price of fairness among all Pareto-optimal solutions? We answer this question affirmatively in the below lemma:

Lemma 4. *The price of fairness among all Pareto-optimal solutions is bounded by 3 when the number of agents is 2.*

Proof. For any path $p = \{v_{\alpha_1}^1, v_{\alpha_2}^2, \ldots, v_{\alpha_k}^k\}$, let us denote the cost incurred by an agent in traveling from $v_{\alpha_1}^1$ to $v_{\alpha_k}^k$ as c_p. The total cost of the path can then be written as $C(p) = O(p) + c_p + T(p)$. Here $O(p)$ denotes the sum of the distance of all the agents from the respective source nodes to the first PoI $v_{\alpha_1}^1$, and $T(p)$ denotes the sum of the distances of all the agents from the last PoI $v_{\alpha_k}^k$ to the respective destinations. Let us compare an optimal path p^* with any Pareto-optimal path p' with minimum envy. Since we are considering a simple case of two agents, we further split $O(p^*) = s_1^* + s_2^*$ and $T(p^*) = d_1^* + d_2^*$. Here, s_1^*, s_2^* (d_1^*, d_2^*) denote the distance of the two agents to the first PoI in p^* from their respective sources (destinations). Similarly, for path p', we have $O(p') = s_1 + s_2$ and $T(p') = d_1 + d_2$. Further, without loss of generality, assume that $s_1^* + d_1^* \geq s_2^* + d_2^*$. The proof for the other case will be similar. If we plot these distances on a real line (Figure 3 and 4), it is easy to see that there are only two possibilities for a Pareto optimal path p' with $C(p') > C(p^*)$.

Case 1 (Figure 3): When $s_2 + d_2 + c_{p'} < s_2^* + d_2^* + c_{p^*}$ and $s_1 + d_1 + c_{p'} > s_1^* + d_1^* + c_{p^*}$. For this case, we have $\mathcal{E}(p') > \mathcal{E}(p^*)$, which leads to contradiction to p' being the minimum envy path.

Case 2 (Figure 4): When $s_2 + d_2 + c_{p'} > s_2^* + d_2^* + c_{p^*}$ and $s_1 + d_1 + c_{p'} < s_1^* + d_1^* + c_{p^*}$. For this case, the cost can be bounded

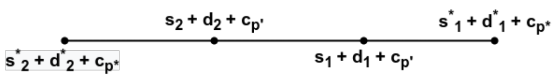


Figure 4: Second Case for Lemma 4

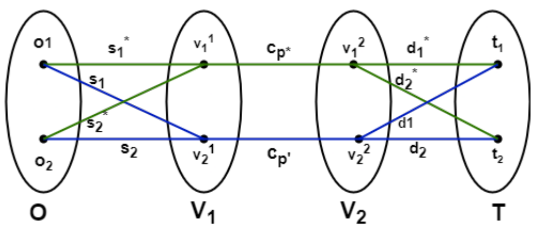


Figure 5: Understanding Triangular Inequality for Lemma 4

by triangular inequality, which can be best explained from Figure 5. Let path $p^* = \{v_1^1, v_1^2\}$ and path $p' = \{v_2^1, v_2^2\}$. We have $s_2 \leq s_2^* + s_1 + s_1^*$ and $d_2 \leq d_2^* + d_1 + d_1^*$ Then:

$$C(p') = s_1 + d_1 + c_{p'} + s_2 + d_2 + c_{p'} \leq s_1^* + d_1^* + c_{p^*} + s_2 + d_2 + c_{p'}$$

$$\leq s_1^* + d_1^* + c_{p^*} + s_1^* + d_1^* + s_2^* + d_2^* + s_1 + d_1 + c_{p'}$$

$$\leq 2(s_1^* + d_1^* + c_{p^*}) + s_1^* + d_1^* + s_2^* + d_2^* \leq 3C(p^*) \qquad \square$$

We now prove that the bound on the price of fairness is tight.

Lemma 5. *The price of fairness among all Pareto-optimal solutions with* 2 *agents is approximately equal to* 3.

Proof. The idea is motivated from the example provided in Figure 4. The price of fairness will be worst when the distance between $s_2^* + d_2^* + c_{p^*}$ and $s_2 * + d_2^* + c_{p'}$ is maximum and the distance between $s_1^* + d_1^* + c_{p^*}$ and $s_1 + d_1 + c_{p'}$ is minimum while satisfying the triangular inequality. Let $s_2^* = d_2^* = \frac{\epsilon}{2}$, $s_1^* = d_1^* = x$, $c_{p^*} = y$. Further, let $s_1 = d_1 = x - \epsilon$, $s_2 = d_2 = 2x - \frac{7}{2}\epsilon$, and $c_{p'} = y + \epsilon$. It is easy to see that these values satisfy triangular inequality. Since the values mimic Figure 4, we have p^* as the optimal path and p' as the Pareto-optimal path minimizing the envy. The price of fairness is given as: $PoF = \frac{6x+2y-7\epsilon}{2x+2y+\epsilon} = \frac{3x+y-7/2\epsilon}{x+y+\epsilon/2}$ As ϵ tends to 0 and the value of x is sufficiently large, the PoF will tend to 3. $\square$

We now extend the result for general b agents in the next lemma:

Lemma 6. *The price of fairness among all Pareto-optimal solutions is bounded by* $(2b - 1)$ *when the number of agents is* b.

Proof. We can bound the price of fairness for b agents using the idea presented for 2 agents case. From Pareto-optimality, we know at least one agent i such that $s_i + d_i + d < s_i^* + d_i^* + d^*$. We can now bound the cost of any agent $j \neq i$ as follows:

$$s_j + d_j + d \leq s_j^* + s_i^* + s_i + d_j^* + d_i^* + d_i + d$$

$$\leq s_j^* + d_j^* + s_i^* + d_i^* + s_i^* + d_i^* + d^*$$

Summing over all the agents, we will get the following: $\sum_{j \in B} D_j(p_f^*) \leq 2(b-1)(d_i^* + s_i^*) + \sum_{j \in B} D_i(p^*) \leq (2b-1) \sum_{j \in B} D_j(p^*)$. $\square$

Lemma 7. *The price of fairness among all Pareto-optimal solutions with* b *agents is approximately equal to* $(2b - 1)$.

Proof. The example is similar to the one considered in Lemma 5, but an extension to b agents. Consider agent i as considered in Lemma 6. We can now assign similar values as was assigned in Lemma 4 by considering agent 1 as i and agent 2 as other agents. This gives us the following values $s_j^* = d_j^* = \frac{\epsilon}{2}\ \forall j \neq i$, $s_i^* = d_i^* = x$, $c_{p^*} = y$. Further, let $s_i = d_i = x - \epsilon$, $s_j = d_j = 2x - \frac{7}{2}\epsilon\ \forall j \neq i$, and $c_{p'} = y + \epsilon$. The price of fairness is then given as:

$$PoF = \frac{2(b-1)(2x - 7/2\epsilon) + 2(x - \epsilon) + b(y - \epsilon)}{2x + 2(b-1)\epsilon/2 + by}$$

$$= \frac{2x(2b-1) - \epsilon(8b-5) + by}{2x + by + (b-1)\epsilon}$$

Substituting $y = 0$, and ϵ close to 0, we get the bound of $2b - 1$. $\square$

We now present the algorithm for finding the Pareto-optimal solution to minimize envy.

5 PO-MINENVYGTP: PROPOSED ALGORITHM

We begin by providing a polynomial time algorithm to find the optimal cost path. We will later see how this algorithm can be extended to find a Pareto-optimal path with minimum envy. Even without fairness constraints, the best-known algorithm in this literature is large polynomial $O(n^k)$ [34], which checks for each possible valid path (number of such paths is $O(n^k)$) and outputs the path that has the minimum distance with n being the number of nodes in each category. In general, there can be different numbers of nodes in each category. However, it can be shown that the worst time complexity occurs when all the categories contain an equal number of nodes[34]. Hence, we show our complexity results when the number of nodes in each category is equally distributed. However, our algorithms run for any number of nodes in each category. Our algorithm first creates a complete $(k+2)-$partite representing the shortest path between the vertices in the sets O, V_1, V_2, …, V_k, and T, respectively. Such a complete $(k+2)-$partite graph can be computed by running a Floyd-Warshall algorithm once on the complete graph with a time complexity of $O(n^3)$. If the number of vertices of interests is much smaller than the complete city network graph, then a complete (k+2)-partite graph can also be obtained by running a single source shortest path algorithm on the vertices of interests. Upon completion of the Floyd-Warshall, we will have the shortest distance between all possible pairs of nodes in G, but we are only interested in the shortest distance between each pair of nodes $(i, j) \in \{(O, V_1), (V_1, V_2), \ldots, (V_k, T)\}$ so we will keep only these edges and remove the other edges.

Once such a $(k+2)-$partite graph is formed, our algorithm 4partiteGTP again runs the Floyd Warshall algorithm on the obtained $(k+2)-$partite graph from the vertex set V_1 to V_k to form a 4−partite graph containing the shortest distance edge for each pair of nodes $(i, j) \in \{(O, V_1), (V_1, V_k), (V_k, T)\}$. If $k = 1$, add a dummy category V_k between (V_1, T) such that $|V_1| = |V_k|$ and connect $(i, j) \in |V_1| \times |V_k|$ with weight 0 to obtain the required 4−partite graph.

Lemma 8. *Any edge* (α, β) *connecting a vertex in* V_1 *and a vertex in* V_k *in* 4*−partite graph will represent a valid path.*

Proof. Any edge in 4−partite graph connecting a vertex α in V_1 to a vertex β in V_k is obtained by complete $(k+2)$−partite graph where the only possible paths from V_1 to V_k are the ones which cover exactly one vertex from intermediary categories. Hence, the resulting path will be a valid path. □

Lemma 9. *The set of all Pareto-optimal paths will be preserved in the* 4−*partite graph.*

Proof. Note that the 4−partite graph contains all shortest paths from each vertex in V_1 to each vertex in V_k. Since all the agents are taking the same path from category 1 vertices to category k, we are not losing any Pareto-optimal path. □

From the above Lemma, it is enough to restrict to the final complete 4−partite graph. We next compute the path matrix presented in line numbers 6-11 of algorithm 1, which essentially saves the cost incurred by each agent for each path present in obtained 4−partite graph. We now have the following main theorem:

Theorem 10. *There exists an $O(n^3)$ algorithm (4partiteGTP) which produces the efficient path in polynomial time.*

Proof. The Floyd-Warshall method on the graph takes $O(n^3)$. Construction of $(k+2)$−partite graph and 4−partite graph construction take $O((k+2)n^2)$ and $O(n^3)$ time respectively. Finally, the minimum cost path can be computed via the path matrix in $O(bn^2)$ time. □

Algorithm 1 4partiteGTP

Input: The road network $G(V,E,W)$, Source and Destination vertices sets, $O = \{o_1, o_2, ..., o_b\}$ and $T = \{t_1, t_2, ..., t_b\}$, PoI categories $\{V^1, V^2, \ldots, V^k\}$

Output: A $4 - partite$ graph and a path-matrix P

1: Run Floyd-Warshall algorithm on the road network graph G and let the matrix representing the shortest path be denoted by M

2: Generate a complete $(k+2)$-partite graph $(K_{O,V^1,\ldots,V^k,T})$ from M retaining the edges connecting O to V^1, V^1 to $V^2, \ldots, V^{k-1}$ to V^k, and V^k to T

3: Run Floyd-Warshall algorithm again on $(k+2)-partite$ graph $(K_{O,V^1,\ldots,V^k,T})$ to generate the matrix M_1

4: Generate a complete 4-partite graph $(K_{O,V^1,V^k,T})$ by retaining the edges from M_1 connecting O to V^1, V^1 to V^k, and V^k to T.

5: Let M_2 be the matrix representing edge weights in $(K_{O,V^1,V^k,T})$

6: **for** $(i,j) \in |V^1 \times V^k|$ **do** ▷ Path-matrix computation

7: $\quad p = 0$

8: $\quad$ **for** each agent $b \in B$ **do**

9: $\quad\quad P[b][p] = M_2[o_b][i] + M_2[i][j] + M_2[j][t_b]$

10: $\quad$ **end for**

11: $\quad p = p + 1$

12: **end for**

5.1 Computing Desired Paths from Path Matrix

From path matrix P, one can compute the following:

Optimal Path: The column with a minimum sum in the path matrix P will give the optimal path.

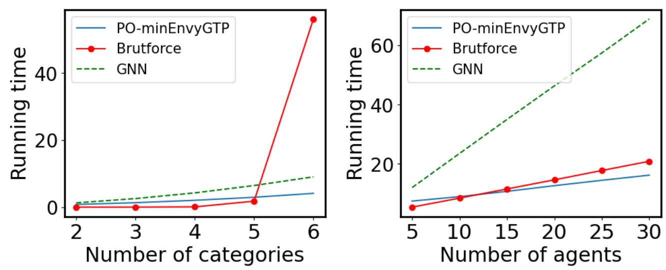
Pareto-optimal path with minimum envy: The algorithm for the same is provided in Algorithm 2. It can again be computed from the path matrix P as follows: For any path p, check if there exists a path p' such that the distance of all the agents in path p is greater than or equal to the distance of all the path in p' (checked with flag parameter). If such a path exists, then p cannot be a Pareto-optimal path. It should also be checked that for at least one agent, the distance in path p' should be strictly greater than the distance in path p. This is taken care of by the flag1 parameter. Once we have identified that a path is a Pareto-optimal path, we also save the envy of that path which is essentially the maximum difference in the distances traveled by any two agents. Finally, the algorithm returns that Pareto-optimal path and minimizes envy. We require to check all possible combinations of $O(n^2)$ paths in 4-partite graphs, then the algorithm 2 will take $O(n^4 b)$ time to compute Pareto-optimal paths. We now present the experimental results on real-world city graph datasets.

Algorithm 2 PO-MinEnvyGTP

Input: The path-matrix P produced by Algorithm 1

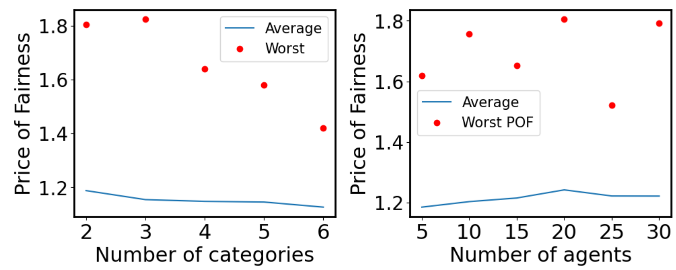
Output: Set of Pareto-optimal paths PO

flag = 0
for $p \in P$ **do**
$\quad$ **for** $p' \in P \setminus p$ **do**
$\quad\quad$ flag1 = 1
$\quad\quad$ **for** each agent $b \in B$ **do**
$\quad\quad\quad$ **if** $P[b][p] \geq P[b][p']$ **then**
$\quad\quad\quad\quad$ flag = 0
$\quad\quad\quad$ **else**
$\quad\quad\quad\quad$ flag = 1
$\quad\quad\quad\quad$ Break
$\quad\quad\quad$ **end if**
$\quad\quad\quad$ if $P[b][p] > P[b][p']$ then flag1 = 0
$\quad\quad$ **end for**
$\quad\quad$ **if** flag = 0 And flag1 = 0 **then**
$\quad\quad\quad$ Break
$\quad\quad$ **end if**
$\quad$ **end for**
$\quad$ **if** flag = 0 and flag1 = 0 **then**
$\quad\quad$ Break
$\quad$ **else**
$\quad\quad$ $pareto = \{pareto \cup p\}$
$\quad\quad$ **for** each $(i,j) \in B$ **do**
$\quad\quad\quad$ $M[i][j] = |P[i][p] - P[j][p]|$
$\quad\quad$ **end for**
$\quad\quad$ $maximum[p] = max(M)$
$\quad$ **end if**
end for
Return the Pareto-path p with minimum value of $max[p]$



(a) Fixed number of Agents **(b) Fixed Number of Categories**

Figure 6: Runtime Comparison with existing approaches



(a) Fixed number of Agents **(b) Fixed Number of Categories**

Figure 7: Impact on PoF with an increase in the number of agents and number of categories

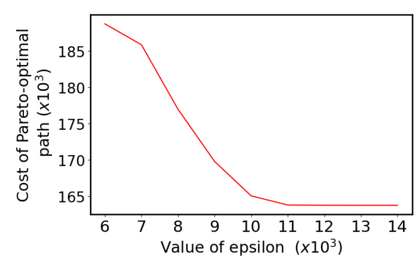


Figure 8: Epsilon-Envy Free path cost

6 EXPERIMENTAL EVALUATION

We now discuss the performance of the proposed algorithm PO-MinEnvyGTP on real-world datasets. The PO-MinEnvyGTP algorithm is compared with the existing naive algorithms such as brute force and GNN [34]. We first show the run-time comparison on the road network dataset of Oldenburg city [24]. We then show the effect of the change in the number of agents and the number of categories on the price of fairness. The codes of the experiments are publicly available on Github[35].

Figure 7 shows the run time comparison of PO-MinEnvyGTP against brute force and GNN over 100 runs. In Figure 6a, the number of agents is fixed to 15, and the number of categories is increased from 2 to 6. In each experiment, the number of options is increased by the interval of five in each category, i.e., category-1 contains five options, category-2 contains ten options, and following similar lines, category-6 contains thirty options. These values are in-line with the existing literature [34]. We chose fewer options for comparison against the brute-force technique, although our algorithms work well for any number of options and categories. It can be seen that the run-time of PO-MinEnvyGTP has no significant impact on the increased number of categories, whereas brute-force [34] grows drastically with the number of categories. It can also be seen that the run-time of GNN also grows (not drastically compared to brute force) as the number of categories increases. Similarly, in Figure 6b, when the number of categories is fixed to 4, PO-MinEnvyGTP has a little impact with the increase in the number of agents, whereas the run-time of GNN increases steeply with the number of agents. This shows the efficiency of our proposed algorithm. Not only on efficiency, but PO-MinEnvyGTP also ensures fairness, and we show that the price of fairness is not much. Our algorithm computes the Pareto-optimal path with minimum envy and gives the price of fairness for different sizes of categories and a different number of agents. Here in Figure 7a, the algorithm computes the price of fairness for different sizes of categories. In this figure, Y-axis represents the average price of fairness, and X-axis represents the number of categories for 100 runs. Similarly, Figure 7b shows the price of fairness for the different number of agents. In both cases, it is seen that though the worst price of fairness is bounded by $(2b-1)$, the price of fairness never goes beyond 2 on a real-world dataset. Similar trends can be observed in San Joaquin County dataset. Figure 8 computes the cost of the optimal path by varying the value of ϵ for 15 agents and 4 categories. $\epsilon-$envy-free path can be computed from the path matrix returned by Algorithm 1 where a path with minimal cost is returned which has envy less than ϵ. Each time we increase the ϵ, the number of paths satisfying the $\epsilon-$increases, and thus the solution converges to the optimal unrestricted solution. One interesting observation is that the average distance traveled by one agent increased from 11000($\approx 165000/15$) to 13000($\approx 185000/15$) only; thereby, leading to a significant reduction of envy from 11000 to 6000 which denotes the additional distance traveled by a single agent.

7 CONCLUSION AND FUTURE WORK

This paper proposed a polynomial time algorithm 4partiteGTP to sequenced group trip planning problems in spatial databases. Through the experiments on real-world datasets, we showed that the proposed algorithm is much faster than existing algorithms. We further proposed a polynomial time algorithm PO-MinEnvyGTP that produces the path with minimum envy. We then showed that the price of fairness depicting the ratio of cost of the Pareto-optimal and minimum envy path to that of the optimal path is bounded by $(2b-1)$, with b being the number of agents.

This is the first step towards providing an efficient, fair solution to the group-trip planning problem. We believe such solutions also apply to numerous applications ranging from the supply chain, distributed learning, route planning, etc. We considered the notion of envy; one could also look for maximizing the minimum distance by any agent. It would be interesting to see the relationship between these two notions. One natural extension is to unordered group trip planning where there is no pre-determined order of the categories. We leave such extensions to the future.

ACKNOWLEDGEMENT

The authors thank Google Research for providing travel grant to the primary author to allow him to attend and present the work to the conference. We would also like to acknowledge by Department of Science & Technology, India with grant number SRG/2020/001138 for providing computational resources required for the project.

REFERENCES

[1] Elham Ahmadi and Mario A Nascimento. 2015. A mixed breadth-depth first search strategy for sequenced group trip planning queries. In *2015 16th IEEE International Conference on Mobile Data Management*, Vol. 1. IEEE, 24–33.

[2] Elham Ahmadi and Mario A Nascimento. 2017. Ibs: An efficient stateful algorithm for optimal sequenced group trip planning queries. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 212–221.

[3] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2022. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2208.08782* (2022).

[4] Haris Aziz. 2020. Developments in multi-agent fair allocation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13563–13568.

[5] Haris Aziz and Simon Mackenzie. 2016. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 416–427.

[6] Sukarna Barua, Roksana Jahan, and Toufique Ahmed. 2017. Weighted optimal sequenced group trip planning queries. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 222–227.

[7] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. 2021. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence* 293 (2021), 103436.

[8] Umang Bhaskar, AR Sricharan, and Rohit Vaish. 2020. On approximate envy-freeness for indivisible chores and mixed resources. *arXiv preprint arXiv:2012.06788* (2020).

[9] Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

[10] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7, 3 (2019), 1–32.

[11] Chao Chen, Daqing Zhang, Yasha Wang, and Hongyu Huang. 2021. TripPlanner: Personalized Trip Planning Leveraging Heterogeneous Trajectory Data. In *Enabling Smart Urban Services with GPS Trajectory Data*. Springer, 225–247.

[12] Haiquan Chen, Wei-Shinn Ku, Min-Te Sun, and Roger Zimmermann. 2008. The multi-rule partial sequenced route query. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. 1–10.

[13] Stefano Coniglio, Luca Giovanni Gianoli, Edoardo Amaldi, and Antonio Capone. 2020. Elastic traffic engineering subject to a fair bandwidth allocation via bilevel programming. *IEEE/ACM Transactions on Networking* 28, 6 (2020), 2407–2420.

[14] Jian Dai, Bin Yang, Chenjuan Guo, and Zhiming Ding. 2015. Personalized route recommendation using big trajectory data. In *2015 IEEE 31st international conference on data engineering*. IEEE, 543–554.

[15] Trajectory Datasets. 2023. Trajectory Datasets:. (2023). http://vis.cs.kent.edu/DL/.

[16] Roee Diamant, Paolo Casari, Filippo Campagnaro, Oleksiy Kebkal, Veronika Kebkal, and Michele Zorzi. 2018. Fair and throughput-optimal routing in multimodal underwater networks. *IEEE Transactions on Wireless Communications* 17, 3 (2018), 1738–1754.

[17] Lester E Dubins and Edwin H Spanier. 1961. How to cut a cake fairly. *The American Mathematical Monthly* 68, 1P1 (1961), 1–17.

[18] George Gamov and Marvin Stern. 1958. Puzzle math. *Viking, New York* (1958).

[19] Tanzima Hashem, Sukarna Barua, Mohammed Eunus Ali, Lars Kulik, and Egemen Tanin. 2015. Efficient computation of trips with friends and families. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 931–940.

[20] Tanzima Hashem, Tahrima Hashem, Mohammed Eunus Ali, and Lars Kulik. 2013. Group trip planning queries in spatial databases. In *International Symposium on Spatial and Temporal Databases*. Springer, 259–276.

[21] Kaggle. 2015. Taxi Trajectory Data:. (2015). https://www.kaggle.com/datasets/crailtap/taxi-trajectory.

[22] Avadh Kishor, Rajdeep Niyogi, and Bharadwaj Veeravalli. 2020. Fairness-aware mechanism for load balancing in distributed systems. *IEEE Transactions on Services Computing* (2020).

[23] Feifei Li, Dihan Cheng, Marios Hadjieleftheriou, George Kollios, and Shang-Hua Teng. 2005. On trip planning queries in spatial databases. In *International symposium on spatial and temporal databases*. Springer, 273–290.

[24] Feifei Li, Dihan Cheng, Marios Hadjieleftheriou, George Kollios, and Shang-Hua Teng. 2005. On Trip Planning Queries in Spatial Databases. In *Advances in Spatial and Temporal Databases*, Claudia Bauzer Medeiros, Max J. Egenhofer, and Elisa Bertino (Eds.). 273–290.

[25] Hongga Li, Hua Lu, Bo Huang, and Zhiyong Huang. 2005. Two ellipse-based pruning methods for group nearest neighbor queries. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*. 192–199.

[26] Sweety Lima and Tahsina Hashem. 2019. Dynamic Obstructed Group Trip Planning Queries in Spatial Databases. In *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 1–6.

[27] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. 125–131.

[28] Peng Liu, Arda Kurt, and Ümit Özgüner. 2014. Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification. In *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 942–947.

[29] Wei Lu and Luca Quadrifoglio. 2019. Fair cost allocation for ridesharing services–modeling, mathematical programming and an algorithm to find the nucleolus. *Transportation Research Part B: Methodological* 121 (2019), 41–55.

[30] Mehnaz Tabassum Mahin and Tanzima Hashem. 2019. Activity-aware ridesharing group trip planning queries for flexible POIs. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 5, 3 (2019), 1–41.

[31] Tania Panayiotou and Georgios Ellinas. 2020. Fair resource allocation in optical networks under tidal traffic. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–6.

[32] Mehdi Sharifzadeh, Mohammad Kolahdouzan, and Cyrus Shahabi. 2008. The optimal sequenced route query. *The VLDB journal* 17, 4 (2008), 765–787.

[33] Shashi Shekhar. 2007. *Spatial databases*. Pearson Education India.

[34] Mayank Singhal and Suman Banerjee. 2022. Envy-Free Trip Planning in Group Trip Planning Query Problem. In *International Conference on Network-Based Information Systems*. Springer, 212–223.

[35] Napendra Solanki. 2023. Group-Trip-Planning code:. (2023). https://github.com/Napendra001/Group-Trip-Planning.

[36] Hugo Steinhaus. 1949. Sur la division pragmatique. *Econometrica: Journal of the Econometric Society* (1949), 315–319.

[37] Anika Tabassum, Sukarna Barua, Tanzima Hashem, and Tasmin Chowdhury. 2017. Dynamic group trip planning queries in spatial databases. In *Proceedings of the 29th international conference on scientific and statistical database management*. 1–6.

[38] Hal R Varian. 1973. Equity, envy, and efficiency. (1973).

[39] Sheng Wang, Zhifeng Bao, J Shane Culpepper, and Gao Cong. 2021. A survey on trajectory data management, analytics, and learning. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–36.

[40] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. 2019. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 13, 3 (2019), 1–207.

[41] Haodong Yin, Huijun Sun, Shanshan Peng, Jianjun Wu, Ying-en Ge, and Yizhou Chen. 2020. Designing a safe and fair network for hazmat road transportation. *Journal of Transportation Safety & Security* 12, 4 (2020), 482–500.

[42] Mortaza Zolfpour-Arokhlo, Ali Selamat, and Siti Zaiton Mohd Hashim. 2013. Route planning model of multi-agent system for a supply chain management. *Expert Systems with Applications* 40, 5 (2013), 1505–1518.