## Collective Travel Planning in Spatial Networks

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Conference Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Shang, Shuo; Chen, Lisi; Wei, Zhewei; Jensen, Christian S.; Wen, Ji-Rong; Kalnis, Panos</td>
</tr>
<tr>
<td>Eprint version</td>
<td>Post-print</td>
</tr>
<tr>
<td>DOI</td>
<td>10.1109/ICDE.2017.36</td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td>Journal</td>
<td>2017 IEEE 33rd International Conference on Data Engineering (ICDE)</td>
</tr>
<tr>
<td>Rights</td>
<td>(c) 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.</td>
</tr>
<tr>
<td>Download date</td>
<td>2024-01-28 07:50:20</td>
</tr>
<tr>
<td>Link to Item</td>
<td><a href="http://hdl.handle.net/10754/624983">http://hdl.handle.net/10754/624983</a></td>
</tr>
</tbody>
</table>
Collective Travel Planning in Spatial Networks

(Extended Abstract)

Shuo Shang\(^1\) Lisi Chen\(^2\) Zhewei Wei\(^3\) Christian S. Jensen\(^4\) Ji-Rong Wen\(^5\) Panos Kalnis\(^6\)

\(^1\)CEMSE, King Abdullah University of Science and Technology, Saudi Arabia
\(^2\)Department of Computer Science, Hong Kong Baptist University, Hong Kong
\(^3\)School of Information, Renmin University, China
\(^4\)Department of Computer Science, Aalborg University, Denmark
\(^5\)School of Information, Renmin University, China
\(^6\)School of Information, Renmin University, China

I. INTRODUCTION

We propose and investigate a novel query, the Collective Travel Planning (CTP) query, that finds the lowest-cost route connecting multiple query sources and a destination via at most \(k\) meeting points. This type of query is useful in organizing large events, and it can bring significant benefits to society and the environment: it can help optimize the allocation of transportation resources, reduce resource consumption, and enable smarter and greener transportation; and it can help reduce greenhouse-gas emissions and traffic congestion.

Given the current locations \(Q\) of a set of travelers, a set of meeting points \(S\), a destination \(d\), and an integer threshold \(k\) \((1 \leq k \leq \min\{|S|, |Q|\})\), we aim to identify a subset \(A\) of \(S\) with at most \(k\) elements that when used as meeting points results in the minimum global travel cost. The local travel cost includes two parts: a local travel cost and a connection travel cost. The local travel cost is the sum of the costs of travel from each traveler’s current location to their closest meeting point, and the connection travel cost is the sum of the costs of travel from each meeting point to the destination. The meeting point count \(k\) is expected to be set according to the resources that can be used.

An example of the CTP query is shown in Figure 1, where \(p_1, p_3, p_4,\) and \(p_5\) are selected meeting points and \(d\) is the destination. Let \(k = 5\) and subset \(A = \{p_1, p_3, p_4, p_5\}\). First, travelers go to their closest meeting point individually. Then the travelers at the same meeting point go together to the destination by collective transport. For example, for the traveler at \(q_1, p_1\) is the closest meeting point, so the traveler will follow the shortest path from \(q_1\) to \(p_1\) (local travel cost of \(q_1\)). A total of five travelers, \(q_1, q_2, q_3, q_4,\) and \(q_5\), meet at \(p_1\). They then follow the shortest path from \(p_1\) to the destination \(d\) by collective transport (connection travel cost of \(p_1\)).

To the best of our knowledge, this is the first study of the collective travel planning query in spatial networks. The CTP query is different from existing multi-source trip planning queries (e.g., the group nearest neighbor query [5] and the group trip planning query [2]) because they assume that travelers go to the destination individually and do not take into account collective travel. The CTP query is also different from most existing ridesharing (carpooling) services [4][7][8]. Generally, such services aim to plan a travel route with pick-up and drop-off locations for a small number of users with similar destination, while the CTP query aims to plan a collective travel route for public transport.

II. SOLUTION OVERVIEW

Exact Algorithm. Exact search is a straightforward method to compute the CTP query that evaluates each potential subset \(A (A \subseteq S \land |A| \leq k)\), of which there is \(\sum_{i=1}^{k} \binom{|S|}{i} = \sum_{i=1}^{k} \frac{|S|!}{(|S| - i)! i!}\). We define a pair of an upper and a lower bound to prune the search space during query processing. For a small threshold \(k\) (e.g., \(k = 2\)), the exact algorithm is capable of finding the optimal result of the CTP query in interactive time. However, \(\sum_{i=1}^{k} \frac{|S|!}{(|S| - i)! i!}\) is exponential in \(|S|\), and the CTP query cannot be computed in polynomial time.

The exact algorithm considers \(\sum_{i=1}^{k} \binom{|S|}{i} = \sum_{i=1}^{k} \frac{|S|!}{(|S| - i)! i!}\) combinations of subset \(A\). For each subset \(A\), it matches query points to their closest meeting point \(p \in A\). The time complexity is \(O(|Q| \cdot |A|) = O(|Q|)\) because \(|A|\) is a constant no larger than \(k\). Thus, the time complexity of the exact algorithm is

\[O \left( |Q| \sum_{i=1}^{k} \frac{|S|!}{(|S| - i)! i!} \right) = O \left( |Q| \cdot |S|^k \right) .\]

The last equation uses Stirling’s approximation.

Approximation Algorithm. To achieve better performance, an approximation algorithm is developed with a 5 approximation ratio. Initially, we arbitrarily select a subset \(A (|A| \leq k)\)

Fig. 1. An example of the CTP query trip for many users (e.g., tens or hundreds of users or more) located all over a city and targeting the same destination. In fact, the CTP query can be viewed as a variant of the metric \(k\) uncapacitated facility location problem (k-UFL) [3], as it asks for an optimal meeting point set \(A (A \subseteq S \land |A| \leq k)\). We prove that the CTP query is Max SNP-hard. To the best of our knowledge, no existing method can compute the CTP query efficiently.
from $S$. Then we define three operations based on local search [1][9]: add (add a new item $p \in (S \setminus A)$ to $A$, if $|A| < k$), drop (drop an item from $A$, if $|A| > 1$), and swap (swap an item in $A$ with another items in $(S \setminus A)$). We repeatedly apply a randomly selected operation to improve the global travel cost by a factor of $1 + \varepsilon$, where $\varepsilon$ is an arbitrary small constant. The search process terminates when no new operation can produce an improved result. The cost of the obtained result is guaranteed to be at most 5 times worse than that of the globally optimum result. The experimental results show that the approximate results are generally very close to the global optimum.

The main contribution in relation to the approximation algorithm is to “bridge theory and practice.” Several theoretical methods exist for the $k$-UFL problem (e.g., modify one item [1] or modify multiple items at one time [9]), and their target is to achieve a lower approximation ratio. However, the CTP query has to balance accuracy and efficiency. Although some theoretical methods can achieve a lower approximation ratio, their query efficiency is very low. Thus, our goal is to select a suitable theoretical method for the CTP query and then to make it practical. Through theoretical analysis, we only allow one item to be modified in an operation. We propose two effective pruning techniques that accelerate the approximation algorithm while retaining its approximation ratio. Experimental results show that the query efficiency is improved by at least an order of magnitude. It is worth noting that the theoretical method cannot be used by itself due to its low efficiency.

To find a valid operation (that can improve the global travel cost by a factor of $1 + \varepsilon$), we check all possibilities of add, drop, and swap, which has time complexity $O(|Q|(|S| - |A|) + O(|Q| |A|) + O(|Q|(|S| - |A||A|) = O(|Q| S|)]$ because $|A|$ is a constant no larger than $k$. The total number of $\log(1+\varepsilon) Q|S|/|A|$ operations is a constant.

Thus, the time complexity of the approximation algorithm is $O(\log(1+\varepsilon) Q|S|/|A|) = O(|Q| S|)]$.

**Extension.** We further extend the approximation algorithm to two practical scenarios where (1) the connection travel cost is dependent on the number of travelers, and (2) where a traveler close to the destination can go to the destination directly. We develop new metrics and bounds for these scenarios. The theoretical approximation ratio does not work here, and we demonstrate empirically that our extension is usable in the targeted scenarios (the approximate costs are less than 1.2 times larger than that of the global optimum).

### III. EXPERIMENTS

We evaluate the performance of the exact algorithm (Exact-Alg) and the approximation algorithm (Approx-Alg) on the North America Road Network (NRN$^1$). Query points are randomly selected vertices, and the meeting points are generated according to random distributions.

Figure 2 shows the effect of the query point count $|Q|$ on the performance of the two algorithms. Approx-Alg outperforms Exact-Alg by almost a factor of $10^5$ (for both CPU time and visited vertices). Figure 3 shows the effect of varying parameter $\varepsilon$ on the efficiency and accuracy of Approx-Alg. When $\varepsilon = 0.03$, Approx-Alg achieves a very good approximation ratio (less than 1.15) and low CPU time (less than 260ms).

---

**REFERENCES**


---

$^1$http://www.cs.utah.edu/lifeifei/SpatialDataset.htm