On Trip Planning Queries in Spatial Databases*

Feifei Li, Dihan Cheng, Marios Hadjieleftheriou,
George Kollios, and Shang-Hua Teng

Computer Science Department, Boston University
{lifeifei, dcheng, marioh, gkollios, steng}@cs.bu.edu

Abstract. In this paper we discuss a new type of query in Spatial Databases, called the Trip Planning Query (TPQ). Given a set of points of interest P in space, where each point belongs to a specific category, a starting point S and a destination E, TPQ retrieves the *best* trip that starts at S, passes through at least one point from each category, and ends at E. For example, a driver traveling from Boston to Providence might want to stop to a gas station, a bank and a post office on his way, and the goal is to provide him with the best possible route (in terms of distance, traffic, road conditions, etc.). The difficulty of this query lies in the existence of multiple choices per category. In this paper, we study fast approximation algorithms for TPQ in a metric space. We provide a number of approximation algorithms with approximation ratios that depend on either the number of categories, the maximum number of points per category or both. Therefore, for different instances of the problem, we can choose the algorithm with the best approximation ratio, since they all run in polynomial time. Furthermore, we use some of the proposed algorithms to derive efficient heuristics for large datasets stored in external memory. Finally, we give an experimental evaluation of the proposed algorithms using both synthetic and real datasets.

1 Introduction

Spatial databases has been an active area of research in the last two decades and many important results in data modeling, spatial indexing, and query processing techniques have been reported [29,17,40,37,42,26,36,4,18,27]. Despite these efforts, the queries that have been considered so far concentrate on simple range and nearest neighbor queries and their variants. However, with the increasing interest in intelligent transportation and modern spatial database systems, more complex and advanced query types need to be supported.

In this paper we discuss a novel query in spatial databases, the Trip Planning Query (TPQ). Assume that a database stores the locations of spatial objects that belong to one or more categories from a fixed set of categories $\mathcal{C}$. The user specifies two points in space, a starting point S and a destination point E, and a subset of categories $\mathcal{R}$, ($\mathcal{R} \subseteq \mathcal{C}$), and the goal is to find the *best* trip (route) that starts at S, passes through at least one point from each category in $\mathcal{R}$ and ends at E. An example of a TPQ is the following: A user plans to travel from Boston to Providence and wants to stop

* This work was partially supported by NSF grants IIS-0133825, IIS-0308213, CCR-0311430, and ITR CCR-0325630.

C. Bauzer Medeiros et al. (Eds.): SSTD 2005, LNCS 3633, pp. 273–290, 2005.

at a supermarket, a bank, and a post office. Given this query, a database that stores the locations of objects from the categories above (as well as other categories) should compute efficiently a feasible trip that minimizes the total traveling distance. Another possibility is to provide a trip that minimizes the total traveling time.

Efficient TPQ evaluation could become an important new feature of advanced navigation systems and can prove useful for other geographic applications as has been advocated in previous work [12]. For instance, state of the art mapping services like MapQuest, Google Maps, and Microsoft Streets & Trips, currently support queries that specify a starting point and only one destination, or a number of user specified destinations. The functionality and usefulness of such systems can be greatly improved by supporting more advanced query types, like TPQ. An example from Streets & Trips is shown in Figure 1, where the user has explicitly chosen a route that includes an ATM, a gas station and a Greek restaurant. Clearly, the system could not only optimize this route by re-arranging the order in which these stops should be made, but it could also suggest alternatives, based on other options available (e.g., from a large number of ATMs that are shown on the map), that the user might not be aware of.

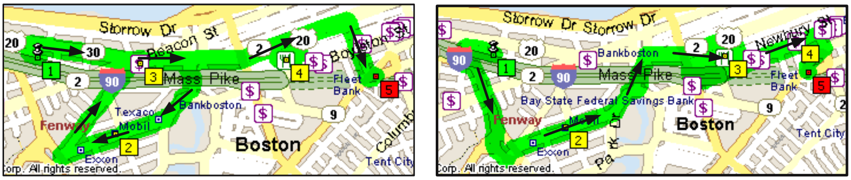

Fig. 1. A route from Boston University (1) to Boston downtown (5) that passes by a gas station (2), an ATM (3), and a Greek restaurant (4) that have been explicitly specified by the user in that order. Existing applications do not support route optimization, nor do they give suggestions of more suitable routes, like the one presented to the right

TPQ can be considered as a generalization of the Traveling Salesman problem (TSP) [2,1,10] which is NP-hard. The reduction of TSP to TPQ is straightforward. By assuming that every point belongs to its own distinct category, any instance of TSP can be reduced to an instance of TPQ. TPQ is also closely related to the group minimum spanning/steiner tree problems [24,20,16], as we discuss later. From the current spatial database queries, TPQ is mostly related to *time parameterized* and *continuous* NN queries [5,41,36,37], where we assume that the query point is moving with a constant velocity and the goal is to incrementally report the nearest neighbors over time as the query moves from an initial to a final location. However, none of the methods developed to answer the above queries can be used to find a good solution for TPQ.

Contributions. This paper proposes a novel type of query in spatial databases and studies methods for answering this query efficiently. Approximation algorithms that achieve various approximation ratios are presented, based on two important parameters: The total number of categories m and the maximum category cardinality ρ. In particular:

- We introduce four algorithms for answering TPQ queries, with various approximation ratios in terms of m and ρ. We give two practical, easy to implement solutions