Information Processing Letters 131 (2018) 55–59

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A fast algorithm for the gas station problem

Kleitos Papadopoulos [a], Demetres Christofides [b,*]

[a] InSPIRE, Agamemnonos 20, Nicosia, 1041, Cyprus
[b] School of Sciences, UCLan, 7080 Pyla, Larnaka, Cyprus

ARTICLE INFO

Article history:
Received 25 August 2017
Received in revised form 19 November 2017
Accepted 25 November 2017
Available online xxxx
Communicated by B. Doerr

Keywords:
Gas station problem
Transportation
Algorithm complexity
Algorithms

ABSTRACT

In the gas station problem we want to find the cheapest path between two vertices of an n-vertex graph. Our car has a specific fuel capacity and at each vertex we can fill our car with gas, with the fuel cost depending on the vertex. Furthermore, we are allowed at most Δ stops for refuelling.
In this short paper we provide an algorithm solving the problem in $O(\Delta n^2 + n^2 \log n)$ steps improving an earlier result by Khuller, Malekian and Mestre.

1. Introduction

There are numerous problems in the literature in which the task is to optimise the travel from one location to another or to optimise a tour visiting a specific set of locations. The problems usually differ in the restrictions that we may put into the way we can travel as well as in the notion of what an 'optimal' route means. Of course one could theoretically check all possible ways to travel and pick out the optimal one. However we care about finding the optimal route in a much quicker way as usually checking all possibilities is impractical.

One of the most widely known abstractions of travel optimisation problems is that of the shortest paths which although very general in their definition, fail to take into consideration most of the aspects that arise in real world. Perhaps one of their more practical generalisations is the 'gas station problem', introduced by Khuller, Malekian and Mestre in [2]. Out of the infinitude of possible parameters it includes one more central aspect of the travelling agent, that of its limited fuel capacity and fuel consumption during the travelling. As it is the case for the shortest paths problem, it is also with the gas station problem that it can be used in a variety of problems that are not directly related with travelling optimisation

The setting of the gas station problem is as follows:

We are given a complete graph $G = (V, E)$, two specific vertices s, t of G and functions $d : E \to \mathbb{R}^+$ and $c : V \to \mathbb{R}^+$. Finally we are also given positive numbers U and Δ.

Each vertex v of G corresponds to a gas station and the number $c(v)$ corresponds to the cost of the fuel at this station. Given an edge $e = uv$ of G, the number $d(e) = d(u, v)$ corresponds to the distance between the vertices u and v, or what is essentially equivalent, to the amount of gas needed to travel between u and v. Finally, the number U corresponds to the maximum gas capacity of our car.

Our task is to find the cheapest way possible to move from vertex s to vertex t if we are allowed to make at most Δ refill stops.

We make two further assumptions:

Our first assumption concerns the function $d : E \to \mathbb{R}^+$. We will follow the natural assumption that it satisfies the triangle inequalities. I.e.

$$d(u, v) + d(v, w) \geqslant d(u, w) \quad \text{for every } u, v, w \in V.$$

* Corresponding author.
E-mail addresses: kleitospa@gmail.com (K. Papadopoulos), dchristofides@uclan.ac.uk (D. Christofides).

https://doi.org/10.1016/j.ipl.2017.11.009
0020-0190/

Our second assumption concerns the amount of fuel that we have initially in our car. We will make the assumption that we start with an empty fuel tank. We also consider the filling in our tank at this vertex as one of the refill stops. This does not make much difference. Indeed suppose that initially we have an amount g of gas. Instead of solving the gas station problem for the graph G, we modify this graph by adding a new vertex s'. We define the new distances by $d(s', v) = U - g + d(s, v)$ for every $v \in V(G)$. I.e. the vertex s' has distance $U - g$ from s and furthermore the shortest path from s' to any other vertex v of G is via s. We also define $c(s') = 0$. It is then obvious that if we start from s', we should completely fill our tank and then move to vertex s. The only difference is that we are now allowed one fewer refill stop than before. So solving the gas station problem for G starting from s with g units of gas is equivalent to solving the gas station problem for G' starting from s' with no gas.

An algorithm solving the gas station problem that runs in $O(\Delta n^2 \log n)$ was introduced by Khuller, Malekian and Mestre in [2]. The main result of our article is the following:

Theorem 1. *Given an n vertex graph G, there is an algorithm which solves the gas station problem with Δ stops in at most $O(\Delta n^2 + n^2 \log n)$ steps.*

We should point out that the algorithm in [2] makes similar assumptions to ours. It explicitly mentions the assumption that the car starts with an empty fuel tank. It does not mention explicitly the assumption that the distances in the graph need to satisfy the triangle inequalities. However it does use it implicitly in its Lemma 1. (See the proof of our Lemma 3 which makes explicit why we do indeed need the distances to satisfy the triangle inequalities.)

We will prove Theorem 1 in the next section. In a couple of instances our algorithm will call some familiar algorithms with known running time. The interested reader can find more details about those algorithms in many algorithms or combinatorial optimisation books, for example in [1].

2. Proof of Theorem 1

We start by ordering all edge distances. Since there are $O(n^2)$ edges, this can be done in $O(n^2 \log n)$ steps, using e.g. heapsort. In fact we will not use the full ordering of the edge distances. What we will need are the following local orderings:

For each $v \in V$ we create an ordering $v_1, \dots, v_{n-1}$ of the vertices of $V \setminus \{v\}$ such that $d(v, v_i) \leqslant d(v, v_j)$ for $i \leqslant j$. We will call this the **local edge ordering** at v. (Note that this definition might be a bit misleading as the local edge ordering at v is an ordering of the vertices of $V \setminus \{v\}$. Of course, this gives an ordering of the edges incident to v and this is where it gets its name from.)

Of course all of these local orderings can also be computed in $O(n^2 \log n)$ steps.

So it is enough to show how to solve the gas station problem in $O(\Delta n^2)$ time assuming that the edges are already ordered by distance.

The fact that the edge distances satisfy the triangle inequalities is needed to prove the following simple lemma:

Lemma 2. *There is an optimal route during which we fill our car with a positive amount of gas at every station (apart from the last one).*

Proof. Amongst all optimal routes, pick one passing through the smallest number of vertices. Suppose that it passes through vertices $v_1, v_2, \dots, v_k$ in that order. We definitely need to fill our car with gas at v_1 as we start with an empty tank. Suppose now for contradiction that in this optimal path we do not fill our car at station v_i for some $1 < i < k$. Then, instead of moving from v_{i-1} to v_{i+1} through v_i, we could have moved to it directly. This is indeed possible as

$$d(v_{i-1}, v_{i+1}) \leqslant d(v_{i-1}, v_i) + d(v_i, v_{i+1})$$

and it is a contradiction as we assumed that our optimal path is both optimal and minimal. □

Lemma 2 is needed to prove the following slightly modified lemma from [2]. Even though it looks completely obvious, we nevertheless provide a detailed proof in order to make explicit the need for using Lemma 2 and thus to require that the distances in the graph satisfy the triangle inequalities. Lin [3] also makes explicit this requirement. Essentially the same lemma also appears in [4].

Lemma 3. *There is an optimal route, say passing through vertices $v_1, v_2, \dots, v_k$ in that order, where $v_1 = s$ and $v_k = t$, for which an optimal way to refill the tank is as follows:*

(i) *For $1 \leqslant i \leqslant k - 2$, if $c(v_i) < c(v_{i+1})$, then at station v_i we completely fill our tank.*
(ii) *For $i = k - 1$, if $c(v_i) < c(v_{i+1})$, then at station v_i we fill the tank with just enough gas in order to reach vertex v_{i+1} with an empty tank.*
(iii) *For $1 \leqslant i \leqslant k - 1$, if $c(v_i) \geqslant c(v_{i+1})$ then at station v_i we fill the tank with just enough gas in order to reach vertex v_{i+1} with an empty tank.*

Proof. Pick an optimal route as given by Lemma 2.

Suppose that at some point during travelling through this optimal route we reach vertex v_i, with $1 \leqslant i \leqslant k - 2$ and suppose that $c(v_i) < c(v_{i+1})$. By Lemma 2 we have filled some gas at station v_{i+1}. If we did not fully filled our gas at station v_i then we could have reduced our cost by filling more gas at v_i and less at v_{i+1}, a contradiction. So at v_i we definitely must completely fill our tank.

If we reach v_{k-1} then of course we fill the car with just enough gas in order to reach v_k with our tank completely empty.

Finally suppose that at some point during travelling through this optimal route we reach vertex v_i, with $1 \leqslant i \leqslant k - 1$ and $c(v_i) \geqslant c(v_{i+1})$. Suppose we fill the car at v_i