Pricing of Geometric Transportation Networks

Jean Cardinal[†] Martine Labbé[†] Stefan Langerman[†] Belén Palop[‡]

Abstract

We propose algorithms for pricing a transportation network in such a way that the profit generated by the customers is maximized. We model the transportation network as a subset of the plane and take into account the fact that the customers minimize their own transportation cost. The underlying theory is a two-player game model called *Stackelberg games*. We propose algorithms for the cases where the fare does or does not depend on the distance traveled. In particular, we propose an $O(n \log n)$ algorithm for optimal pricing of a highway, and an $O(nk \log n \log^3 k)$ algorithm for orthogonally convex networks of complexity $O(k)$ using the Manhattan distance.

1 Introduction

Suppose a person wishes to go from one point to another using public transportation. Taking into account the transportation fare, this person may consider different routes, each with different costs in terms of time and money, and choose the one considered best. In this paper, we play the role of the transportation company whose goal is to maximize the profit generated by the customers. We propose algorithms for pricing the transportation network, taking into account the fact that each customer uses a minimum-cost route. This kind of *pricing problems* is nicely modeled by games known as *Stackelberg games*. These are games with two players: a *leader* and a *follower*. The first decision is made by the leader: in our case the transportation company fixes the fare. Then the follower responds by optimizing its own objectives, taking into account the decision of the leader. In our case, customers decide of their optimal route, taking into account the fare that has been fixed by the transportation company. The problem is to find an optimal strategy for the leader. Stackelberg games have found applications in both transportation planning and telecommunications: see Altman et al. for a survey [4]. They are also known as bilevel programs or hierarchical optimization problems. Labbé et al. [5] have studied a family of such pricing problems on graphs.

We define geometric versions of these problems. We let $\{(s_i, t_i)\}_{i=1}^n$ be a collection of pairs of points in the plane, and define the transportation network $\mathcal{N}$ as a compact subset of the plane. Each single customer i wishes to go from s_i to t_i using a minimum cost path. The cost is the sum of the distance traveled outside the network and the fare.

We propose two distinct models for defining this fare. In both cases the model is parameterized by a number R and the profit $B(R)$ is the sum of the amounts paid by all customers. The algorithms we propose in the following sections compute a value of R that maximizes the profit.

In the first model, that we call *fixed pricing*, the fare R that a customer has to pay for using the network does not depend on the distance traveled within the network. The total cost for one customer is the sum of the distance traveled outside the network and R.

We call the other model *proportional pricing*. In that case, when customers use the network, the fare is equal to the length of their trajectory inside the network multiplied by the real number $R \in [0, 1]$, that we call the *pricing factor*. The total cost for one customer is the sum of the distance traveled outside the network and the distance within the network multiplied by R. Since each customer minimizes its cost, the length of the trajectory inside the network is a nonincreasing function of R. Note that the algorithms presented in this paper for the proportional pricing model can be easily adapted to finer models in which, for instance, customers minimize a weighted sum of the time and money spent on the trip. The distance measure is also part of the model. We use both Euclidean (L_2) and Manhattan (L_1) distances.

The cost of traveling between two points in the proportional pricing model induces what is known as a *time distance*: if we define $v = 1/R$, then the cost is equal to the total trip time when v is the speed inside the network. Time distances have been studied recently by Aichholzer et al. [3], and Abellanas et al. [2, 1]. Another related problem is finding shortest paths in planar subdivisions, in which a speed is assigned to each subdivision [6]. Snell's law of refraction is known to apply to such shortest paths when the Euclidean distance is used. The use of the Euclidean distance however severely restricts the range of problems (e.g. network shapes) that can be addressed: even the simplest examples may lead to computing high-degree polynomials or inverse trigonometric functions [1]. This motivates the use of the simpler, yet meaningful, Manhattan distance.

In section 2, we examine the fixed pricing case and give simple algorithms that work for many different

[†]Université Libre de Bruxelles, Belgium, {jcardin, mlabbe, slanger}@ulb.ac.be

[‡]Universidad de Valladolid, Spain, b.palop@infor.uva.es

kinds of network shapes. Section 3 proposes a $O(n \log n)$ algorithm for proportional pricing of a straight line (e.g. a highway). In section 4 and 5, we consider proportional pricing together with the Manhattan distance and propose $O(nk \log n \text{ polylog } k)$ for orthoconvex networks of complexity k.

2 Railpass

We first consider the fixed pricing case, where customers have to pay a fixed fare R for using the network, independently of the portion of the network they use. The cost is then the sum of R and the distance traveled outside the network. We propose algorithms for finding the value of R maximizing the profit $B(R)$ for any metric.

We first assume that the network $\mathcal{N}$ is a connected subset of the plane. Then if a customer uses the network, the trajectory will enter the network at the nearest neighbor of s_i and leave it at the nearest neighbor of t_i. Denote by $n(x)$ the point of $\mathcal{N}$ that is the nearest to x, and by $d(x, y)$ the minimum length of a path between x and y outside the network. The ith customer uses the network whenever

$$\begin{aligned} & d(s_i, n(s_i)) + R + d(n(t_i), t_i) \leq d(s_i, t_i) \\ \Leftrightarrow\quad & R \leq d(s_i, t_i) - (d(s_i, n(s_i)) + d(n(t_i), t_i)). \end{aligned}$$

We call *breakpoints* values of R for which equality holds, i.e. values of R above which one customer decides not to use the network. For any value of R, the overall profit $B(R)$ is the number of customers using the network multiplied by R. By computing the breakpoints given above for all i and sorting them, we obtain an algorithm for finding a value of R maximizing the profit. This algorithm is valid for any network shape and any distance measure.

Theorem 1 *When the network $\mathcal{N}$ is a connected subset of the plane, we can find a fixed fare R maximizing $B(R)$ in $O(n(\log n + C))$ time, where C is the cost of finding the nearest neighbor in $\mathcal{N}$ of a point.*

In cases where the Voronoi diagram of a network of complexity k can be computed in $O(k \log k)$ time and point location in this partition can be performed in $O(\log k)$ time, we can proceed by first computing this diagram and then using it to answer the nearest neighbor queries. Hence for many networks, we can find an optimal value of R in $O(n \log n + (n+k) \log k)$ time. This is true for instance when the network is a connected set of line segments.

Finally, if the network is not connected, then it is not true anymore that customers using the network will enter and leave the network at the nearest neighbor points. First, a shortest path using the network has to be computed for each customer. The overall complexity of the

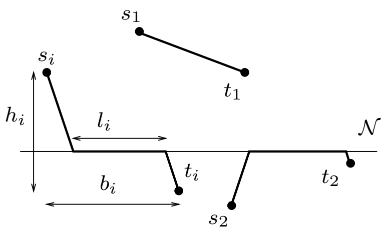


Figure 1: minimum cost paths from s_i to t_i

algorithm is $O(n(\log n + S))$, where S is the cost of computing a shortest path using the network.

3 Australian Highways

We now consider the model in which the network $\mathcal{N}$ is a straight line, the Euclidean distance is used, and proportional pricing is applied.

Denote by b_i the distance between the orthogonal projections of s_i and t_i on $\mathcal{N}$, and by h_i the sum of the distances between s_i and its projection and t_i and its projection. This is illustrated in Fig. 1. Finally, let d_i be the euclidean distance between s_i and t_i. If s_i and t_i are separated by $\mathcal{N}$, then $d_i = \sqrt{h_i^2 + b_i^2}$. We denote by $l_i(R)$ the length of the highway segment in a minimum cost path from s_i to t_i. We wish to find a pricing factor R maximizing the overall profit $B(R) = R \sum_{i=1}^{n} l_i(R)$.

Let us first describe the behavior of the customers when the pricing factor R is fixed.

Snell's law of refraction implies that in a minimum cost path from s_i to t_i using the highway, the angles of the trajectory from s_i to the highway and from the highway to t_i are the same. It is observed in [1] that the minimum cost of a path from s_i to t_i is actually the minimum between their Euclidean distance and the distance between s_i and the two halflines through t_i (or its symmetrical point with respect to $\mathcal{N}$) with slopes respectively R and $-R$. In the following, we call α the angle such that $\sin \alpha = R$.

We omit the proof of the following lemma.

Lemma 2 *The ith customer will use the highway if and only if $R \leq r_i$, with $r_i = (b_i d_i - h_i \sqrt{b_i^2 - d_i^2 + h_i^2})/(b_i^2 + h_i^2)$. Moreover, If the trajectory of the ith customer uses the highway, then it is composed of a highway segment of length $l_i(R) = b_i - h_i \tan \alpha$, and its total cost is $J_i(R) = h_i \cos \alpha + b_i \sin \alpha$.*

Note that the expression for r_i has a special form when s_i and t_i are separated by $\mathcal{N}$. In that case, we have $d_i = \sqrt{h_i^2 + b_i^2}$ and r_i is equal to $b_i/d_i = \cos \beta$, where β is the angle between $\mathcal{N}$ and the segment $[s_i, t_i]$. Hence the decision of the customer is only based on whether $\sin \alpha \leq \cos \beta$, and is independent of the distance between s_i and t_i. Also note that in this case, the breakpoint is also the solution of the equation $l_i = 0$, which means that as R increases, the trajectory of the customer tends to the straight line segment $[s_i, t_i]$ in a continuous fashion.

Theorem 3 *When the network $\mathcal{N}$ is a straight line and the L_2 distance is used, we can find a pricing factor R maximizing $B(R)$ in $O(n \log n)$ time.*

Proof. The algorithm first sorts the pair (s_i, t_i) according to the values r_i in $O(n \log n)$ time. The breakpoints partition the interval $[0, 1] \ni R$ in subintervals of the form $[r_i, r_{i+1}]$. Between two successive subintervals, the number of minimum cost paths using the highway decreases by one. Hence as we walk from one interval to the next, the sum $\sum_{i:l_i(R)>0} l_i(R)$ can be maintained in constant time using the decomposition:

$$\sum_{i:l_i(R)>0} l_i(R) = \left[\sum_{i:l_i(R)>0} b_i\right] - \left[\sum_{i:l_i(R)>0} h_i\right] \tan\alpha.$$

The values in brackets are constant in each interval, hence we have a closed form of the function $R\sum_{i:l_i(R)>0} l_i(R)$ to maximize in each interval. We can show that it can be solved analytically in constant time by letting the unknown be $Z = \tan\alpha$. The minimization step therefore takes $O(n)$ time, and the whole algorithm is $O(n \log n)$. □

4 Manhattan Buses

In this section, we consider proportional pricing of more complex networks but simpler shortest paths, since we allow customers to move only following Manhattan paths and switch thus to the L_1 distance. Consider the network $\mathcal{N}$ having the shape of an isothetic staircase defined by an origin (u, v), and a sequence of length k of pairs (x_i, y_i) of positive real numbers.

We define the profit function $B_i(R)$ for a customer i as the fraction of the profit that is generated by this customer. We use $J_i(R)$ to denote the minimum cost of a path from s_i to t_i for a pricing factor R. We describe an algorithm that finds the optimal pricing factor by computing each function $J_i(R)$ recursively. The cost and profit functions are both piecewise linear. To represent such functions, we store the breakpoints in increasing order together with the slope and offset of the supporting line between each pair of successive breakpoints. This representation makes it easy to compute the sum $f_1(R) + f_2(R)$ or the lower envelope $\min\{f_1(R), f_2(R)\}$ of two functions $f_1(R)$ and $f_2(R)$ in linear time.

A first observation is that each customer may enter the network only at a discrete number of entrance points. These points are a subset of the staircase corners that are oriented towards the point s_i, together with the vertical and horizontal projections of s_i on $\mathcal{N}$. Similarly, a customer always leaves the network at one of the corners that are oriented towards t_i or one of the projection of t_i on $\mathcal{N}$.

We start the description of the algorithm with a special case.

Lemma 4 *If $t_i \in \mathcal{N}$, then a description of $J_i(R)$ can be computed in $O(k)$ time.*

Proof. We assume without loss of generality that the staircase defining the network $\mathcal{N}$ is going down and from left to right, and that s_i is above $\mathcal{N}$. We consider the rectangle having s_i and t_i as two opposite corners.

When s_i is on the right of t_i this rectangle does not intersect $\mathcal{N}$ and the trajectory of the customer always contains one of the two other corners s' and s'' of the rectangle. This is illustrated on Fig. 2(a). The cost function J_i can be constructed in linear time from the function obtained by replacing s_i first by s' and then by s''. The two cases are symmetric and equivalent to the case where s_i has the same y-coordinate as t_i, as shown on Fig. 2(c).

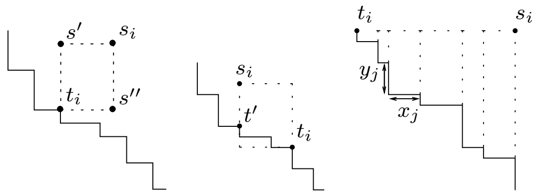
When s_i is on the left of t_i, then the intersection of the rectangle and $\mathcal{N}$ contains the vertical projection t' of s_i on $\mathcal{N}$. This case is illustrated on Fig. 2(b). We can safely assume that any minimum cost trajectory of the ith customer contains t'. The cost function J_i can be constructed from the cost function obtained by replacing t_i by t'. Since t' has by definition the same x-coordinate as s_i, this case is again symmetric to that shown on Fig. 2(c). Overall, considering only the case on Fig. 2(c) is sufficient.

We assume without loss of generality that t_i is the origin of the staircase and that the last point of the staircase is the vertical projection of s_i on $\mathcal{N}$. We call $L_c(R)$ the function giving the minimum cost of the path from s_i to t_i given that the customer enters the network at the cth corner of the staircase:

$$L_c(R) = R\sum_{j=1}^{c}(y_j + x_j) + \sum_{j=1}^{c} y_j + \sum_{j=c+1}^{k} x_j.$$

The profit generated in that case equals the first term $R\sum_{j=1}^{c}(y_j + x_j)$. The functions L_c are linear in R.

We also define $L_0(R)$ as the horizontal line whose y-coordinate is equal to the horizontal distance between s_i and t_i. This value corresponds to the case where the customer does not use the network. The cost function $J_i(R)$ is the lower envelope of the set of lines with equations $\{L_c(R) : c = 0, 1, \ldots, k\}$. Since $\sum_{j=1}^{c}(y_j + x_j)$ is strictly increasing with respect to c, the slopes of these lines are found in increasing order. Computing the lower envelope is equivalent to finding the convex hull of the corresponding set of points after the dual transformation that maps the line $y = ax + b$ to the point (a, b). Since the slopes of the lines L_c are increasing with respect to c, this is equivalent to finding the lower convex



(a) (b) (c)

Figure 2: Special case when t_i is on the network.

hull of a set of points given in increasing order of their x-coordinates. This can be done in linear time using Graham scan. □

We now lift the assumption that t_i is on the network.

Lemma 5 *A description of $J_i(R)$ can be computed in $O(k \log k)$ time.*

Proof. We fix a point p on $\mathcal{N}$ that splits $\mathcal{N}$ in two equal-sized parts A and B. We call $f_1(R)$ the function giving the minimum cost of a path from s_i to p, and $f_2(R)$ the function giving the minimum cost of a path from p to t_i. From the previous lemma, these two functions can be computed in $O(k)$ time. We can also obtain their sum $f_1(R) + f_2(R)$ in linear time. We call $f_A(R)$ and $f_B(R)$ the functions giving the minimum cost of a path from s_i to t_i using only the portions A or B of the network. These can be computed recursively. Now having computed the three functions $f_1(R) + f_2(R)$, $f_A(R)$ and $f_B(R)$, we can find the actual minimum cost $J_i(R)$ of a path from s_i to t_i by computing their lower envelope in linear time. The total number of operations per recursion level is linear and the depth of the recursion is $O(\log k)$, so the complexity of computing $J_i(R)$ is $O(k \log k)$. □

Theorem 6 *When the network $\mathcal{N}$ is a staircase and the L_1 distance is used, we can find a pricing factor R maximizing $B(R)$ in $O(nk \log n \log k)$ time.*

Proof. We first compute the functions $J_i(R)$ for all i, which takes $O(nk \log k)$ time, then sum them in $O(nk \log n \log k)$ time by sorting their breakpoints. We obtain the piecewise linear cost function $J(R)$. The function $B(R)$ is obtained from $J(R)$ by letting the offsets of the supporting lines between two breakpoints be zero. Then we can find a maximum of $B(R)$ in linear time. □

5 Manhattan Polygons

We now consider the case where the network $\mathcal{N}$ is a polygon having all its edges either horizontal of vertical, and such that the intersection of any horizontal or vertical line with $\mathcal{N}$ is a line segment (orthoconvex polygon). We stick to the proportional pricing model with the L_1 distance. We proceed as previously by computing a description of the cost function $J_i(R)$ for each customer i.

The boundary of $\mathcal{N}$ can be divided in four staircases, each having a different orientation. This division in turn divides the points outside $\mathcal{N}$ in four quadrants according to the location of their nearest neighbor in $\mathcal{N}$ with respect to the L_1 distance. In the case where both s_i and t_i do not belong to $\mathcal{N}$ and are in the same quadrant, we can assume without loss of generality that the customer uses only the boundary of $\mathcal{N}$, and we find ourselves in the same situation as in the previous section. The function J_i in this case can therefore be computed in $O(k \log k)$ time.

We now state three lemmas corresponding to different cases. All three consist of divide-and-conquer algorithms similar to the algorithm of Lemma 5.

Lemma 7 *If $s_i \notin \mathcal{N}$ and $t_i \in \mathcal{N}$, then a description of $J_i(R)$ can be computed in $O(k \log k)$ time.*

Lemma 8 *If s_i and t_i do not belong to $\mathcal{N}$ and are in adjacent quadrants, then a description of $J_i(R)$ can be computed in $O(k \log^2 k)$ time.*

Lemma 9 *If s_i and t_i do not belong to $\mathcal{N}$ and are in opposite quadrants, then a description of $J_i(R)$ can be computed in $O(k \log^3 k)$ time.*

Now having computed all cost functions $J_i(R)$, we can sum them, deduce the profit function $B(R)$ and find its maximum.

Theorem 10 *When the network $\mathcal{N}$ is an orthoconvex polygon and the L_1 distance is used, we can find a pricing factor R maximizing $B(R)$ in $O(nk \log n \log^3 k)$ time.*

References

[1] M. Abellanas, F. Hurtado, and B. Palop. Transportation networks and voronoi diagrams. In *Proc. International Symposium on Voronoi Diagrams in Science and Engineering*, 2004.

[2] M. Abellanas, F. Hurtado, V. Sacristan, C. Icking, L. Ma, R. Klein, E. Langetepe, and B. Palop. Voronoi diagrams for services neighboring a highway. *Inform. Process. Lett.*, 86:283–288, 2003.

[3] O. Aichholzer, F. Aurenhammer, and B. Palop. Quickest paths, straight skeletons, and the city voronoi diagram. *Discrete & Computational Geometry*, 31(1):17–35, 2004.

[4] E. Altman and L. Wynter. Equilibrium, games, and pricing in transportation and telecommunications networks. *Networks and Spacial Economics*, 4(1), 2004.

[5] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard. A bilevel model for for toll optimization on a multicommodity transportation network. *Transportation Science*, 35:345–358, 2001.

[6] J. S. B. Mitchell. Geometric Shortest Paths and Network Optimization Handbook of Computational Geometry, Elsevier Science, (J.-R. Sack and J. Urrutia, eds.), 2000. Chapter 15, pages 633–701.