

Routing in Road Networks: the toll booth problem

Fernando Stefanello, Luciana S. Buriol, Marcus Ritt

Instituto de Informática – Universidade Federal do Rio Grande do Sul
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
[fstefanello,buriol,mrpritt]@inf.ufrgs.br

Michael J. Hirsch

Raytheon Company, Intelligence and Information Systems
300 Sentinel Drive Annapolis Junction, MD 20701, USA.
mjh8787@ufl.edu

Panos M. Pardalos

Dept. of Industrial and Systems Engineering, University of Florida
303 Weil Hall, Gainesville, FL 32611, USA.
pardalos@ufl.edu

Tania Querido

Linear Options Consulting
7450 SW 86th Way, Gainesville, FL 32608, USA.
tania@lineartoptions.com

Mauricio G. C. Resende

Internet and Network Systems Research Center, AT&T Labs Research
180 Park Avenue, Room C241, Florham Park, NJ 07932, USA.
mgcr@research.att.com

ABSTRACT

Due to population growth and the massive production of automotive vehicles, traffic congestion problems have become larger and more common. This is a reality that governments are facing everywhere, even in medium sized cities that were not used to this scenario. However, despite the growth of the number of vehicles, traffic congestion can be lessened using different strategies. One possibility, that is explored in this research, is assigning tolls to roads, inducing users to take alternative paths, and thus better distributing the traffic across the road network. This problem is called the toll booth problem and is NP-hard. We propose mathematical formulations for variations of the toll booth problem, using two piecewise linear functions to approximate the congestion cost. We test these models on a set of real-world instances, and apply a previously proposed genetic algorithm to all instances. The experimental results show that the proposed piecewise linear functions approximates the original convex function quite well, and the genetic algorithm produces high quality solutions.

KEYWORDS. Transportation networks, Genetic algorithms, Toll both problem, Combinatorial Optimization.

1. Introduction

The quality of transportation people have available plays an important role in modern life. Keeping the time spent driving as short as possible for doing daily activities directly impacts the quality of life for individuals in a society. If we consider the case where high quality roads have been already built, there are not many alternatives for reducing driving time. However, if a given road has high traffic, reducing or distributing the resulting congestion across the road network is certainly a possibility that should be explored.

Due to the population growth and the massive production of vehicles, traffic congestion problems become larger and more common every day, mainly in metropolitan areas. To a commuter or traveler, congestion means loss of time, potential missed business opportunities, stress, and frustration. To an employer, congestion means lost worker productivity, trade opportunities, delivery delays, and increased costs (W. and Wen, 2008). For example, one significant aspect is the value of wasted fuel and additional time. In 2010 traffic congestion cost about \$115 billion in the 439 urban areas of EUA (Schrank et al., 2011).

Plan improvements in these systems require a careful analysis of several factors. These alternatives are evaluated using different models that attempt to capture the nature of transportation systems and thus allow the estimation of the effect of future changes in system performance. Performance measures include efficiency in time and money, security, and social and environmental impacts, among others.

There are some strategies aimed at reducing traffic congestion. Among them, installing tolls on certain roads can induce drivers to choose alternative routes, which reduces congestion as a result of a better distribution of traffic flow. Naturally, tolls can increase the cost of a trip, but it can be compensated with less time, fuel and stress spent in traffic. In the 1950's Beckmann et al. (1956) proposed the use of tolls, and recently this idea has been used in practice. In 1975 Singapore implemented the "Electronic Road Pricing - ERP" program. Norway, London and some cities in the United States have implemented projects to use toll roads (Bai et al., 2010). Toll roads are also commonly applied in old European towns to reduce vehicular congestion in downtown areas.

Given a set of input data, defining the location of the toll booths and their tariffs can be formulated as a combinatorial optimization problem. This problem has aroused interest in the scientific community, both by its intrinsic difficulty as well as the social importance and impact of its solution.

Optimizing transportation network performance has been widely discussed in the literature. The minimum toll booth problem (MINTB), first introduced by Hearn and Ramana (1998), aims at minimizing the number of toll locations to achieve a system optimum. Yang and Zhang (2003) approach the second-best link-based pricing formulated as a bi-level program and solved it by a genetic algorithm. In Bai et al. (2010) it was shown that the problem is NP-hard and a local search metaheuristic was proposed. Another similar problem is to minimize total revenue without (MINSYS) or with (MINREV) the possibility for negative tolls or subsidies in some links (Hearn and Ramana, 1998; Dial, 1999a,b; Hearn and Yildirim, 2002; Bai et al., 2004). For a complete review of the design and evaluation of road network pricing schemes we refer to the survey by Tsekeris and Voß (2009).

Road and telecommunication routing problems have some similarities: they are modeled by a directed weighted graph, where each link has capacity and delay (or link travel time), and a demand matrix defines the amount of flow required between each pair of nodes. Different from a road network, whose flow depends on the routes used by the users, in telecommunication networks the flow is sent according to a protocol. One of the most commonly used protocols within autonomous systems is the Open Shortest First Protocol (OSPF) that sends flow between origin and destination by the shortest path, splitting traffic evenly among alternative paths. A classical NP-Hard optimization problem in this area is the weight setting problem (WSP) (Fortz and Thorup, 2004) which attributes an integer weight to each link in a telecommunication network such that when flow is sent by the shortest path (calculated considering the weights) the traffic in the network is minimized. Metaheuristics were successfully applied for solving the WSP (Fortz and Thorup, 2004; Buriol et al., 2005).

In this paper we approach the toll booth problem with routing through shortest paths, first studied in Buriol et al. (2010). In this problem the objective is to allocate a fixed number \mathcal{K} of toll booths and set their tariffs so that the users travel through shortest paths between origin and destination, reducing network congestion. Untolled links are considered to have zero weight (no tariff). The least cost paths are calculated based only on the tariffs of the tolled arcs, and flow is sent by these shortest paths, in analogy to the OSPF protocol. In this work we present models of variations of the toll booth problem. We further propose two piecewise linear functions that approximate the convex travel cost function of Bureau of Public Roads (1964) for measuring the congestion on each link. Finally, we extend the work in Buriol et al. (2010) presenting a larger set of experiments.

This paper is organized as follows. In Section 2 we present the mathematical models for minimal average link usage and the average link travel time, and the two approximate piecewise linear functions. These mathematical models provide some information about the network and are intermediate steps for constructing the mathematical model for the *toll both problem*. The genetic algorithm with local search proposed in Buriol et al. (2010) is presented in Section 3. Computational results are reported in Section 4. Finally, conclusions are drawn in Section 5.

2. Problem formulation

The road network is represented as a directed graph $G = (V, A)$ where V represents the set of nodes (i.e., vertices or points of interest), and A the set of arcs (i.e., links or roads). Each arc $a \in A$ has an associated capacity c_a , and a time t_a spent to traverse the unloaded arc (i.e., the free flow time). For calculating the congestion on each link, a cost Φ_a is calculated as a function of the load ℓ_a (or flow) on the arc, plus two tuning parameters p_a (power), and β_a (cost). In addition, $K \subseteq V \times V$ is the set of commodities or Origin-Destination (OD) pairs, where $o(k)$ and $d(k)$ represent the origination and destination nodes for each $k \in K$. Each commodity $k \in K$ has an associated demand of traffic flow d_k , i.e., for each OD pair $\{o(k), d(k)\}$, there is an associated amount of flow d_k that emanates from node $o(k)$ and terminates at node $d(k)$.

Before presenting the models, some notation is introduced. We denote by $IN(v)$ the set of incoming links to node v , $OUT(v)$ the set of outgoing links from node v , $T_K = \sum_{k \in K} d_k$ is the total sum of demands d_k . Moreover, we denote by Φ_a the traffic congestion of arc a , whilst φ_a and φ_a^l are approximations of traffic congestion cost on arc

a given by piecewise linear functions. We note that throughout the text, flow and load are synonymous, as are commodity and demand.

The subsections 2.1 and 2.3 introduce the mathematical models and the piecewise linear functions proposed in this work, while subsection 2.2 presents the model proposed in Buriol et al. (2010).

2.1. First Mathematical Model (MM1): Minimal Average Link Usage

This problem aims at minimizing the maximum utilization of links, where the usage of a link to is defined by $\frac{\ell_a}{c_a}$, the ratio of load allocated to a link and the maximum load the link is able to accommodate.

Decision Variables:

x_a^k = proportion of commodity k on arc a ;

ℓ_a = load or flow of arc a ;

Θ = maximum utilization;

$$\text{minimize } \Theta \quad (1)$$

$$\text{subject to: } \ell_a/c_a \leq \Theta \quad \forall a \in A \quad (2)$$

$$\ell_a = \sum_{k \in K} d_k x_a^k \quad \forall a \in A \quad (3)$$

$$\sum_{a \in IN(v)} x_a^k - \sum_{a \in OUT(v)} x_a^k = \begin{cases} 1, & \text{if } v = d(k) \\ -1, & \text{if } v = o(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall v \in V, k \in K \quad (4)$$

$$x_a^k \in [0, 1], \ell_a \geq 0 \quad \forall a \in A, \quad \forall k \in K. \quad (5)$$

Objective function (1) represents minimizing the maximum utilization among all links. Constraint set (2) defines the maximum value of Θ . Constraint set (3) defines total flow on arc a considering all commodities. Constraint set (4) guarantee flow conservation, and the last constraint set defines the domain of the variables (5).

This model provides important information about the network traffic, which is the ability to distribute the demands without exceeding the capacity of links (case of $\Theta > 1$).

2.2. Second Mathematical Model (MM2): Average Link Travel Time

The evaluation costs of a route can be defined in different ways according to specific goals. The cost function $\Phi_a = \ell_a t_a [1 + \beta_a (\frac{\ell_a}{c_a})^{p_a}]$ represents the average link travel time for the system. We normalized this value dividing it by T_K . The function Φ_a is convex and nonlinear and we assume that Φ_a is a strictly increasing function.

The model presented below aims at defining the flows on each link such that the overall cost is minimized. Decision variables x_a^k and ℓ_a are defined as in MM1.

$$\text{minimize } \Phi = \sum_{a \in A} \ell_a t_a [1 + \beta_a (\ell_a/c_a)^{p_a}] / T_K \quad (6)$$

$$\text{subject to: } (3)-(5)$$

The objective function (6) represents the average link travel time for the system.

2.3. Mathematical model 3 (MM3): Linear Approximate Average Link Travel Time

The performance of commercial mixed integer linear programming solvers has improved considerably over the last few years, whose efficiency cannot be disregarded by the scientific community. This current state motivates the creation of linear mathematical models (approximating the original mathematical models) so that these resources can be utilized. Thus, we propose the model below which describes a piecewise linear approximation of function Φ .

Decision Variables:

φ_a = cost on arc a ;

$$\text{minimize } \sum_{a \in A} \varphi_a \quad (7)$$

subject to: (3) and (4)

$$(m_a^i/c_a)\ell_a + b_a^i \leq \varphi_a \quad \forall a \in A, i = 1, \dots, n \quad (8)$$

$$x_a^k \in [0, 1], \varphi_a \geq 0, \ell_a \geq 0 \quad \forall a \in A, \quad \forall k \in K. \quad (9)$$

Where:

$$\begin{aligned} m_a^i &= (Y_i - Y_{i-1})/(X_i - X_{i-1}) \\ b_a^i &= Y_i - X_i m_a^i \\ \text{with } Y_i &= X_i c_a t_a (1 + \beta_a (X_i)^{p_a}) / T_K \\ \text{for } X_0 &= 0 < X_1 < \dots < X_n. \end{aligned}$$

The objective function (7) minimizes link costs. Constraint set (8) evaluates the partial cost on each arc, while the last constraint set (9) defines the domain of variables.

The cost Φ_a is approximated by a piecewise linear function φ_a defined in constraint set (8). The linearized function and the values of X and Y are described in the next subsection.

2.3.1. Piecewise linear functions φ_a and φ_a^l

The piecewise linear function is an estimation of the cost function Φ . This linearization requires a balance between the accuracy of the solution found and computational time. A lower value of n (number of line segments composing the approximation) can lead to a poor approximation of the original objective function, while a large value of n can lead to a considerable increase in computational time since each element entails $|A|$ additional constraints to the model.

We describe two linearizations of the cost function Φ for $a \in A$. In both cases it is necessary to define a vector X whose values are computed as a function of $(\frac{\ell_a}{c_a})$. This vector can be arbitrarily defined according to the accuracy required for the linearization of the cost function, or according to characteristics of the set of instances. The linearization φ is an overestimation of Φ . The second linearization, denoted φ^l , is an underestimation, which gives a lower bound of Φ . A representation of these three functions is depicted in Figure 1. It shows the behavior of the cost function Φ (solid line) and the linear piecewise cost functions φ and φ^l for an arc with $t_a = 5$, $c_a = 200$, $p_a = 4$, $\beta_a = 0.15$, and

$T_K = 1000$ to vector X with $n = 6$ for a given instance. The filled circle dots are defined by vector X and the square filled dots by vector X^l .

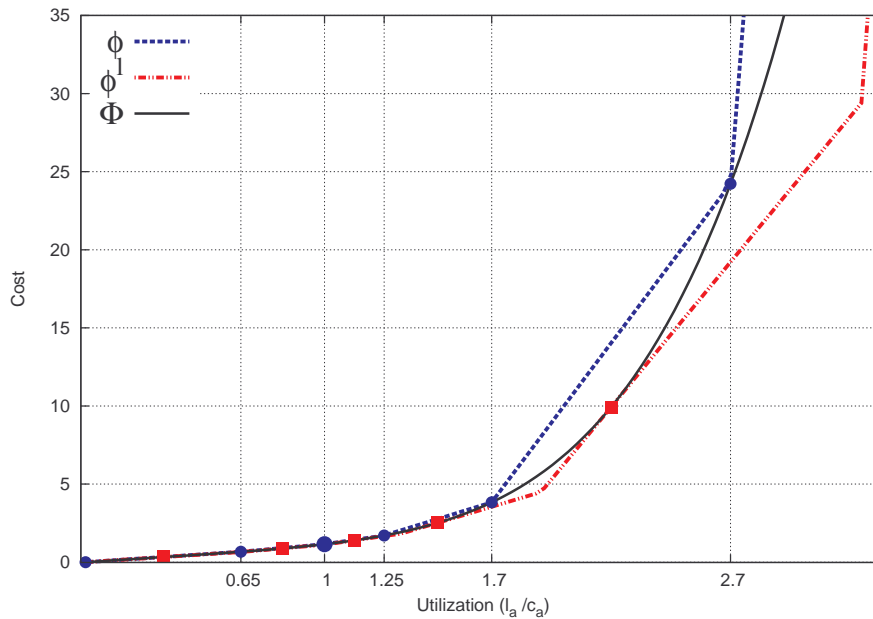


Figure 1. Comparison of the cost function with the linear piecewise cost function.

For φ , the values of Y_i are calculated given the values of X_i as $Y_i = \Phi(X_i)$. For each pair of points $P_{i-1} = (X_{i-1}, Y_{i-1})$ and $P_i = (X_i, Y_i)$ we can write the linear equation of the straight line that links these two points. The slope is represented by m_a^i and the independent term by b_a^i . For φ^l we use $\frac{\partial \Phi}{\partial x} = \frac{t_a}{T_K} + \frac{(p_a+1)t_a\beta_a x^{p_a}}{c_a^{p_a} T_K}$ to calculate the slope in a point x . By convention we use the vector X of the first case to calculate the vector X^l used in this case, where $X_i^l = \frac{X_{i-1} + X_i}{2}$. We empirically defined $X = \{0, 0.65, 1, 1.25, 1.7, 2.7, 5\}$ based on the set of instances tested.

Proposition 1: if $\frac{l_a}{c_a} \leq X_n \forall a \in A$, then $\varphi^l \leq \Phi \leq \varphi$.

By construction $\varphi_a^l \leq \Phi_a$, then $\Phi = \sum_{a \in A} \Phi_a \geq \sum_{a \in A} \varphi_a^l = \varphi^l$. Thus $\Phi \geq \varphi^l$. Furthermore, if $\frac{l_a}{c_a} \leq X_n$ by construction $\varphi_a \geq \Phi_a$, then $\varphi = \sum_{a \in A} \varphi_a \geq \sum_{a \in A} \Phi_a = \Phi$. Thus $\varphi \geq \Phi$. Therefore $\varphi^l \leq \Phi \leq \varphi$.

3. A Hybrid Genetic Algorithm for the Toll Booth Problem

In this section we describe the hybrid genetic algorithm used for solving the toll booth problem proposed in [Buriol et al. \(2010\)](#).

Each solution is represented by two arrays w' and b . Array w' stores the integer arc weights, while b is a binary array indicating the set of tolls. An arc a of the network has weight equal to $w_a = w'_a \cdot b_a$. Each individual weight belongs to the interval $[1, w_{\max}]$. Each demand is routed forward to its destination through the shortest path calculated based on the weights of the tolled arcs. Un-tolled links are considered to have zero weight. Depending on the number of tolls and the network, there can be several shortest paths of zero cost. In this case we evaluate the shortest path through the number of hop counts. Traffic at intermediate nodes is split equally among all outgoing links on shortest paths to

the destination. After the flow is defined, the solution is associated with a fitness value defined by the objective function Φ . The initial population is randomly generated, with arc weights selected uniformly in the interval $[1, w_{\max}]$. Also, the position of \mathcal{K} toll booths is chosen at random.

The population is partitioned into three sets \mathcal{A} , \mathcal{B} , and \mathcal{C} . The best solutions are kept in \mathcal{A} , while the worst ones are in \mathcal{C} . All solutions in \mathcal{A} are promoted to the next generation. $|\mathcal{B}|$ other solutions are generated by crossover of one parent from \mathcal{A} with another from $\mathcal{B} \cup \mathcal{C}$ using the *random keys* crossover scheme. Solutions from class \mathcal{C} of the previous generation are replaced by new randomly generated solutions.

A local search procedure is incorporated into the genetic algorithm to enhance its ability to find better-quality solutions with less computational effort. Local search is applied to each solution generated by the crossover operator. In short, the local search approach works as follows. Given a solution, the five arcs with largest congestion cost are selected. For each of these arcs, in the case there is a toll already installed on it, its weight is increased by one, in order to induce a reduction of its load. The increases are repeated until the weight reaches w_{\max} or Φ does not improve. For each of the arcs that do not currently have a toll installed, a toll is installed with initial weight one, and a toll is removed from some other link (using a circular order). The procedure stops at a local minimum when there is no improved solution changing the weights of the candidate arcs.

4. Computational results

This section presents and discusses the computational experiments performed. First, we present the dataset used in the experiments. Next, we detail the experiments performed. Two sets of experiments were performed: the first explores CPLEX results for the models presented in Section 2, while the second explores the GA results.

For the experiments, we used a computer with an Intel i5-2300 2.80GHz CPU, with 4GB of main memory and Ubuntu 10.10. The genetic algorithms were implemented in the C programming language and compiled with the gcc compiler, version 4.4.5 with optimization flag -O3. The commercial solver CPLEX 12.3 was used to solve the mathematical models presented in Section 2.

Table 1 presents details of the ten real-world instances considered in our experiments and available by Bar-Gera (2011).

4.1. Minimal average link usage MM1 and linear piecewise cost MM3

The first set of experiments tests the performance of CPLEX (set up to use the primal algorithm) applied to a subset of the instances, considering the models introduced in sections 2.1 and 2.3. Table 2 presents, for each instance and model, the objective function values and the computational time (in seconds) needed to solve the problem instance. The objective function is Θ for MM1, while for MM3 the results for both piecewise linearizations are presented. For every solution found, the corresponding Φ (6) value is also calculated for the sake of comparison.

From the results in Table 2, there are four main observations that can be drawn. First, minimizing the maximum utilization (Θ) is not effective in minimizing the link travel time as measured by Φ . However, we were not expecting a close relation among these functions. Second, there are instances where in any given feasible solution, at least one arc

Table 1. Attributes of problem instances. The columns represent the instance name, number of vertices, number of links, number of OD demand pairs, number of nodes that originate demands, and number of nodes that are destinations of demands, respectively.

Instance	Vertices	Links	Demands	Origins	Destinations
SiouxFalls	24	76	528	24	24
Friedrichshain Center	224	523	506	23	23
Prenzlauerberg Center	352	749	1406	38	38
Tiergarten Center	361	766	644	26	26
Mitte Center	398	871	1260	36	36
Stockholm	416	962	1623	45	45
MPF Center	975	2184	9505	98	98
Barcelona	1020	2522	7922	97	108
Winnipeg	1067	2975	4345	135	138
ChicagoSketch	933	2950	93513	387	387

Table 2. Computational results for MM1 and MM3

Instance	MM1			MM3					
	Θ	Φ	Time	φ	Φ	Time	φ^l	Φ	Time
SiouxFalls	1.91	28.03	2.67	21.68	20.52	1.40	18.10	20.92	1.08
Friedrichshain Center	0.40	77.55	12.98	47.61	43.11	3.78	39.57	43.34	2.96
Prenzlauerberg Center	0.66	92.65	60.69	67.21	60.80	34.00	56.98	61.07	29.16
Tiergarten Center	0.41	91.85	15.31	59.68	52.91	53.52	47.63	53.61	45.91
Mitte Center	0.58	120.53	35.52	71.08	63.11	160.24	58.76	63.59	33.67
Stockholm	9,316.94	152,053.61	7.75	108.63	111.03	494.85	79.08	106.84	503.84

carries more flow than its capacity, since $\Theta > 1$. Third, both piecewise linear functions have values close to Φ , which shows that they are good approximations. Finally, for the Stockholm instance, the objective function value is very high. This is expected since node 5 is a destination for a demand of size 9316.94, but has only one incoming link, the capacity of which is 1.

The remaining instances listed in Table 1 cannot be solved because of the size of the problems. The large number of constraints generated by these problem instances increase the memory requirements beyond the capacity of our test machine.

4.2. Computational results of genetic algorithms

This section presents results of the genetic algorithm for the ten test instances. We extend the experimental study performed by [Buriol et al. \(2010\)](#) which presents results for four of these instances (the other six were only available recently). Moreover, we provide an analysis of the best solution for each combination of instance and \mathcal{K} .

In the first experiment we compare the results obtained between the genetic algorithm (GA) and the genetic algorithm with local search procedure (GA+LS). The stopping criteria are a time limit of 1800 seconds and a maximum number of generations (5000 for GA and 1000 for GA+LS). Table 3 shows the results obtained, averaged over five runs with different random seeds and over the seven different values of \mathcal{K} (the number of toll booths used in this experiment are the same as shown in Table 4). Table 3 presents the number of generations and average running times for the GA and the GA+LS. Furthermore, the last column of Table 3 presents the average gap between the fitness value of both methods, showing the superiority of the GA+LS over the GA (a negative average gap means that GA+LS performed better than GA alone). On average, GA+LS spent less time and

found better solutions than GA on the experiments performed. We note that the only cases where the times were comparable was when both approaches reached the time limit of 1800 seconds.

Table 3. Comparison of GA and GA+LS algorithms.

Instance	Number of generations		Run Time (s)		Av. gap
	GA	GA+LS	GA	GA+LS	
SiouxFalls	5,000.00	1,000.00	37.04	48.59	-1.21
Friedrichshain Center	5,000.00	1,000.00	407.28	243.68	-0.56
Prenzlauerberg Center	5,000.00	1,000.00	988.56	527.99	-0.91
Tiergarten Center	5,000.00	1,000.00	722.11	447.31	-0.16
Mitte Center	5,000.00	1,000.00	1,157.31	702.29	-0.18
Stockholm	5,000.00	1,000.00	1,507.18	836.92	-9.57
MPF Center	1,044.43	347.77	1,801.11	1,802.88	-0.83
Barcelona	903.71	300.57	1,801.19	1,802.90	-14.94
Winnipeg	668.43	225.60	1,801.50	1,804.82	-12.94
ChicagoSketch	257.91	86.03	1,803.80	1,809.69	-52.70

For the next experiments we only consider GA+LS. The stopping criteria were set to 1800 seconds or 2000 generations. We also added a third criterium: the number of generations without improvement of the fitness value, set as 5% of the maximum number of generations. Table 4 shows the average over five runs. For each \mathcal{K} (number of installed tolls), in the first vertical half of the table we show the number of generations (gen), best solution value (best Φ) over the five runs, average fitness values (avg Φ), standard deviation (SD), and average running time in seconds. For the best solution found among the five runs, the second half of the table presents the average number of paths for each OD pair (#Path), the average number of hops among all OD shortest paths (#Hops), the average number of tolled arcs among the shortest path (#Toll), the average sum of the tariffs on a path (\sum Tariff), and the number of different arcs used in each OD pair (#Arcs).

As \mathcal{K} increases, the cost tends to decrease and present small oscillations. In most cases, the best solutions were not found with larger \mathcal{K} , which is close to $|E|$ (the number of arcs), but for $\mathcal{K} \approx \frac{|E|}{2}$ or slightly larger than that. Also, we can see that the standard deviation is small in most cases showing that the algorithm is robust.

The column #Path highlights the number of shortest paths on average used between each OD pair. With no tolls, or with few tolls, there are on average about two shortest paths. However, as \mathcal{K} increases, the number of shortest paths decreases. With a small \mathcal{K} , there are alternative paths with no tolls. In this case only the hop count is used to evaluate the shortest path, increasing the possibility of paths with the same cost.

In computer networking, hop count refers to intermediate devices through which a piece of data must pass between source and destination. We extend this concept to road networks where the intermediate devices are vertices which represent intersecting roads or waypoints. Thus, we show in column #Hops the average hop count for all shortest paths between OD pairs. When compared with the situation without tolls ($\mathcal{K} = 0$) we observe that as the number of installed tolls increases, so does the hop counts.

The column #Tolls shows the average number of tolls installed on OD shortest paths. Clearly, this value increases with \mathcal{K} . In the column \sum Tariff we show the average value of tariff in each path. In this problem our objective is to install \mathcal{K} tolls such that

Table 4. Detailed results of GA+LS algorithm.

Instance	GA+LS results						Analysis of the best solution found by the GA+LS				
	\mathcal{K}	gen	best Φ	avg Φ	SD	time (s)	#Path	#Hops	#Toll	\sum Tariff	UArcs
SiouxFalls	0	-	83.97	-	-	-	1.97	1.94	-	-	4.97
	10	167.00	49.59	53.60	3.61	11.87	1.72	2.15	0.22	4.27	4.73
	20	236.20	33.99	38.81	6.37	16.32	1.41	2.24	0.65	2.81	4.25
	30	369.80	26.45	31.70	3.33	22.85	1.20	2.33	0.97	7.50	3.81
	40	461.20	22.49	23.70	0.81	22.12	1.10	2.20	1.45	12.69	3.43
	50	446.80	21.91	22.23	0.42	18.28	1.13	2.18	1.91	12.90	3.50
	60	460.40	21.72	22.08	0.53	13.82	1.09	2.16	2.37	12.31	3.36
70	495.20	21.52	21.90	0.28	11.46	1.08	2.16	2.77	13.21	3.37	
Friedrichshain Center	0	-	94.87	-	-	-	1.96	8.16	-	-	11.94
	10	235.20	56.46	57.32	0.56	90.57	1.98	8.74	0.12	2.00	12.50
	50	700.20	47.71	50.62	2.37	247.87	1.41	9.35	0.50	2.63	12.22
	100	754.80	44.10	45.05	0.99	197.46	1.39	10.46	1.76	10.68	12.74
	200	873.20	43.36	43.76	0.27	204.67	1.18	10.45	3.14	9.95	11.97
	300	1,087.40	43.58	44.03	0.55	209.14	1.06	10.58	5.59	9.93	11.73
	400	945.40	43.80	44.13	0.32	173.23	1.12	10.39	7.65	9.66	11.80
500	936.40	44.93	45.28	0.25	148.49	1.00	10.19	10.51	8.03	11.19	
Prenzlauerberg Center	0	-	149.64	-	-	-	2.75	12.46	-	-	17.11
	10	156.20	78.52	80.77	2.07	136.30	2.58	13.00	0.08	6.00	17.47
	50	593.80	68.28	71.28	2.16	473.40	1.96	13.98	0.35	4.38	17.30
	100	1,199.00	62.80	63.73	0.83	718.60	1.56	14.71	1.09	7.17	16.91
	200	1,201.40	62.27	62.69	0.39	538.35	1.28	15.77	3.84	10.28	17.31
	450	1,376.80	62.22	62.63	0.44	579.74	1.08	14.99	7.82	10.59	16.16
	600	1,626.40	62.88	64.23	1.08	583.70	1.00	14.75	11.00	8.86	15.79
700	1,513.40	65.03	65.29	0.31	517.25	1.00	14.12	14.18	9.17	15.12	
Tiergarten Center	0	-	73.65	-	-	-	1.92	13.42	-	-	17.32
	10	207.40	62.03	62.25	0.15	144.90	2.07	14.41	0.00	0.00	18.57
	50	448.60	56.69	57.29	0.38	284.24	1.35	15.67	0.00	0.00	18.99
	100	806.20	54.67	55.21	0.51	467.96	1.32	16.77	1.34	4.84	18.93
	200	1,161.40	53.74	54.17	0.44	512.54	1.08	16.43	3.54	10.64	17.93
	450	1,103.20	53.30	53.67	0.23	365.15	1.00	15.69	8.34	9.62	16.69
	600	896.20	53.27	54.01	0.60	261.38	1.02	15.96	11.21	9.78	17.10
700	1,450.20	53.65	54.06	0.27	434.97	1.00	15.43	14.43	8.52	16.43	
Mitte Center	0	-	105.98	-	-	-	2.79	12.55	-	-	17.75
	10	170.00	81.34	81.69	0.35	176.96	3.24	13.47	0.00	0.00	19.43
	50	832.60	69.96	71.20	0.99	801.77	1.70	14.02	0.06	2.00	17.56
	100	937.00	69.57	71.13	1.54	869.77	1.51	14.79	0.58	10.70	18.34
	200	1,041.60	65.42	66.34	0.55	694.24	1.19	15.07	3.27	9.94	16.78
	400	1,062.60	64.76	65.39	0.65	665.55	1.02	15.28	6.58	10.91	16.46
	600	1,615.20	64.20	64.60	0.54	833.25	1.03	15.16	8.41	10.56	16.41
800	1,254.00	64.96	65.68	0.49	554.51	1.01	15.08	13.29	9.29	16.18	
Stockholm	0	-	14,933.87	-	-	-	2.62	18.95	-	-	25.45
	10	197.00	491.04	512.80	45.39	248.16	2.07	19.57	0.00	0.00	24.27
	50	715.80	345.09	388.57	27.58	835.91	1.36	20.17	0.19	9.65	22.68
	100	981.20	291.76	342.87	41.82	1,215.07	1.95	21.07	1.66	5.89	25.11
	300	1,494.00	164.79	193.80	20.86	1,159.01	1.25	21.31	6.10	9.16	22.92
	500	1,920.20	145.45	154.21	8.03	1,293.88	1.01	21.30	10.98	9.88	22.36
	700	1,694.00	149.25	156.45	7.83	995.69	1.04	20.29	14.60	10.04	21.53
900	1,615.40	161.59	187.68	21.84	850.32	1.01	20.12	19.36	9.76	21.21	
MPF Center	0	-	108.81	-	-	-	5.09	18.73	-	-	27.54
	10	287.00	92.09	92.41	0.25	1,803.60	5.56	19.29	0.00	0.00	28.30
	100	292.20	82.47	83.79	1.26	1,803.21	2.89	21.48	0.02	1.00	27.53
	250	305.00	83.14	84.07	0.64	1,804.66	3.40	22.13	0.27	6.58	28.86
	500	315.80	83.86	86.45	2.04	1,802.35	2.45	25.16	4.35	8.48	28.48
	1000	320.20	76.63	78.73	1.60	1,802.81	1.20	23.99	8.30	10.68	25.40
	1500	352.20	75.63	76.09	0.53	1,802.31	1.09	23.40	13.77	9.55	24.74
2000	510.40	74.61	75.35	0.70	1,801.53	1.03	22.39	20.07	9.24	23.46	
Barcelona	0	-	687.58	-	-	-	7.37	12.24	-	-	20.82
	10	215.40	15.56	15.80	0.13	1,805.60	7.28	13.88	0.01	1.00	24.92
	100	280.80	9.34	9.51	0.15	1,803.96	6.30	14.82	0.05	7.95	25.13
	500	289.00	9.99	10.31	0.30	1,802.56	3.63	16.22	0.09	3.56	24.97
	1000	249.20	15.50	19.65	6.81	1,802.23	2.88	21.89	5.11	4.11	26.29
	1500	267.40	10.10	10.65	0.68	1,804.63	1.51	19.10	8.28	8.28	21.41
	2000	290.20	8.78	9.17	0.30	1,803.82	1.08	16.37	11.97	8.93	17.76
2500	443.00	8.47	8.64	0.16	1,802.97	1.06	15.39	16.06	8.53	16.54	
Winnipeg	0	-	1,384.34	-	-	-	3.92	17.41	-	-	25.96
	10	172.60	35.60	36.82	1.05	1,806.89	3.84	18.12	0.00	0.00	26.64
	100	176.60	20.97	21.30	0.32	1,807.81	3.88	20.66	0.01	14.76	29.34
	500	189.00	31.82	34.73	3.16	1,805.93	4.36	25.01	0.48	6.28	33.18
	1000	183.20	24.89	28.19	3.40	1,807.95	2.51	24.90	6.49	7.93	29.08
	1500	187.00	20.76	21.59	1.14	1,804.83	1.21	22.92	9.03	9.65	24.38
	2000	206.80	18.05	19.12	0.88	1,804.25	1.08	21.69	13.23	9.20	22.94
2800	335.80	17.41	17.68	0.25	1,803.30	1.05	19.52	20.33	9.66	20.71	
ChicagoSketch	0	-	4,310.54	-	-	-	20.04	10.18	-	-	23.69
	10	70.40	100.33	100.33	0.00	1,815.87	21.59	10.33	0.00	0.00	24.38
	100	75.40	23.59	24.20	0.40	1,807.66	22.84	10.90	0.05	10.80	24.96
	500	80.00	27.45	29.84	2.59	1,808.34	9.78	11.65	0.44	11.03	22.38
	1000	78.80	84.25	119.04	23.41	1,812.11	5.02	13.14	0.84	9.86	20.93
	1500	76.80	330.25	590.58	227.10	1,813.40	2.62	16.05	5.46	5.65	19.22
	2000	74.60	56.81	72.86	11.89	1,810.44	1.27	13.78	7.48	7.40	15.39
2900	106.00	20.02	21.39	1.20	1,811.54	1.08	11.84	12.63	8.97	13.07	

congestion costs are minimized. Since minimizing the tariff is not part of the objective of this problem, that results in more variability in this column. By the last column we conclude that the increase of \mathcal{K} slightly influence the average number of arcs used to send flow for an OD pair.

5. Conclusions

In this paper we proposed various mathematical formulations for different problems of traffic assignment, using two piecewise linear functions to approximate the link travel time. We tested these models on a set of real-world instances, and we applied a previously proposed genetic algorithm for the toll booth problem to all these instances.

The mathematical model MM1 provided information about the network such as the ability to handle the flow without exceeding the capacity. The results demonstrated that in some instances this constraint could not be satisfied. We also observed that the proposed piecewise linear functions closely approximate the original convex function and provided a lower bound for their value.

For the toll booth problem, although being computationally demanding, Also, the procedure produced high quality solutions even for large problem instances.

Finally, considering that users would take the least costly path, toll setting can be used for the distribution of better the flow on the network, thereby reducing traffic congestion.

In future work we intend to include shortest path constraints to the formulation, and to solve larger instances with the GA+LS algorithm.

Acknowledgements

This work has been partially supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) - Brazil, FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul) and PRH PB-17 – Petrobras S.A., Brazil.

References

- Bai, L., Hearn, D., Lawphongpanich, S.** (2004). Decomposition techniques for the minimum toll revenue problem. *Networks* 44 (2), 142–150.
- Bai, L., Hearn, D., Lawphongpanich, S.** (2010). A heuristic method for the minimum toll booth problem. *Journal of Global Optimization* 48, 533–548, 10.1007/s10898-010-9527-7.
- Bar-Gera, H.** (2011). Transportation networks test problems. <http://www.bgu.ac.il/~bargera/tntp>
- Beckmann, M., McGuire, C., Winsten, C.** (1956). *Studies in the economics of transportation*. Yale University Press.
- Bureau of Public Roads, .** (1964). *Bureau of public roads: Traffic assignment manual*. US Department of Commerce, Urban Planning Division.
- Buriol, L., Hirsch, M., Pardalos, P., Querido, T., Resende, M., Ritt, M.** (2010). A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters* 4, 619–633, 10.1007/s11590-010-0226-6.
- Buriol, L. S., Resende, M. G. C., Ribiero, C. C., Thorup, M.** (2005). A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. *Networks* 46, 36–56.
- Dial, R. B.** (1999a). Minimal-revenue congestion pricing part I: A fast algorithm for the single origin case. *Transportation Research Part B* 33, 189–202.
- Dial, R. B.** (1999b). Minimal-revenue congestion pricing part II: An efficient algorithm for the general case. *Transportation Research Part B* 34, 645–665.

- Fortz, B., Thorup, M.** (2004). Increasing internet capacity using local search. *Computational Optimization and Applications* 29 (1), 189–202.
- Hearn, D. W., Ramana, M. V.** (1998). Solving congestion toll pricing models. *Equilibrium and Advanced Transportation Modeling*, 109–124.
- Hearn, D. W., Yildirim, M. B.** (2002). A toll pricing framework for pricing assignment problems and elastic demands. In: Gendreau, M., Marcotte, P. (Eds.), *Transportation and network analysis: current trends*. Vol. 63 of *Applied Optimization*. Springer.
- Schrank, D., Lomax, T., Eisele, B.** (2011). 2011 urban mobility report. Tech. rep., Texas Transportation Institute.
- Tsekeris, T., Voß, S.** (2009). Design and evaluation of road pricing: state-of-the-art and methodological advances. *Netnomics* 10, 5–52, 10.1007/s11066-008-9024-z.
- W., Wen** (2008). A dynamic and automatic traffic light control expert system for solving the road congestion problem. *Expert Systems with Applications* 34 (4), 2370–2381.
- Yang, H., Zhang, X.** (2003). Optimal toll design in second-best link-based congestion pricing. *Transportation Research Record: Journal of the Transportation Research Board* 1857 (1), 85–92.