Comparison of Toll Pricing Strategies and Analysis of Toll Variation with Demand

Course project: CE 392C Transportation Network Analysis (15635)

Aupal Mondal
Suyash Chandra Vishnoi

TRANSPORTATION ENGINEERING
DEPARTMENT OF CIVIL, ENVIRONMENTAL AND ARCHITECHTURAL ENGINEERING
THE UNIVERSITY OF TEXAS AT AUSTIN

Abstract

In this study we look at the various tolling strategies in transportation network theory and compare three of them namely Marginal Social Cost Pricing (MSCP), Minimum System Revenue tolls (MINSYS) and Minimum toll booths (MINTB) on networks of different sizes. The intricacies of implementing these strategies on larger networks, like the Sioux Falls, are also discussed. This is followed by an empirical analysis of the effect of demand variation on the MINSYS tolling outcomes. We discuss how MSCP tolls tend to be unrealistically large, making MINSYS and the MINTB better suited strategies for real-world implementations. The empirical study on the demand variation does not reveal any clear pattern in the impact of varying demand on the tolls, alluding to the fact that tolls are primarily network-specific.

Contents

1 Introduction

Users of transportation networks are believed to follow the principle of user equilibrium. This means that they always only try to minimize their own travel time when traveling from an origin to a destination and they do not care about other people's travel times. System administrators/planners on the other hand are less concerned about minimizing the travel time of any individual user, instead what they care about is reducing the overall travel time of the system such that the average cost to each user is minimized. However, merely preaching ideals to the greedy users seldom does the trick for administrators of the transportation system. Tolling is what some administrators go to when trying to achieve this objective. Over the past many years different types of tolling strategies have been studied by researchers based on a variety of planning priorities. Some of them are computationally cheap while the others require tedious mathematical formulations and reasonably good computational power to evaluate. However, the benefits of implementing them computationally heavy schemes easily surpass the former's.

In this study, we focus on some of these strategies as presented in the toll pricing literature. Beginning with some theory behind the idea of toll pricing and the need for the computationally demanding techniques, we compare some of the pricing strategies on networks of varying sizes. This is followed by an analysis of variations in the tolls generated with different demands on the same network. Simultaneously we explore the intricacies involved in the implementation of these strategies on larger networks as compared to smaller networks.

2 Background and Theory

This section briefly introduces the theory behind toll pricing, some mathematical concepts and the types of toll strategies that are implemented on transportation networks.

2.1 Principle of Toll Pricing

Tolls are essentially costs that are put on some (or all) of the links in the network such that when the users follow in user equilibrium on the tolled network, they end up moving as per the system optimal flow of the un-tolled system. This is also defined in the principle of toll pricing which in the exact words of [1] can be stated as follows:

" The tolls imposed should be such that the resulting tolled user equilibrium problem has at least one solution and every such solution is an untolled system optimal solution."

2.2 Marginal Social Cost Pricing (MSCP) Tolls

The MSCP toll is a well-known toll vector that allows us to calculate the system optimum solution of any network using the methods to find user equilibrium. Its mathematical expression is given as:

$$\beta_{MSCP} = t'(x)x$$

where $t(x)$ is the link performance function of a link and x is the flow on the link. The MSCP toll on a link is therefore the product of the derivative of the link performance function at flow x with the flow itself.

2.3 Mathematical equivalence

As mentioned above, MSCP tolls allow us to obtain SO solutions from UE formulations. This is achieved by adding the MSCP toll cost to the link cost of each link in the user equilibrium optimization formulation. The UE objective function then becomes

$$\int_0^{x_{ij}} \left(t_{ij}(x) + t'_{ij}(x)x\right)dx$$

Which on integrating by parts turns out to be equal to $t_{ij}\left(x_{ij}\right)x_{ij}$ which is the same term used for link (i,j) in the system optimal optimization formulation. This shows that the two problems become mathematically equivalent when MSCP tolls are added to the link costs. This property of the formulations is exploited later in the study.

2.4 Limitations of MSCP

From the discussion in section 2.3, we can say that calculating the MSCP tolls is reasonably easy for any kind of network as it does not require any special formulations for computation. However, the MSCP tolls are unrealistic when it comes to implementation in the network. This is primarily because of the expression for these tolls which requires every used link to have a toll. This has the following limitations:

(1) large infrastructure cost involved in setting up of tollbooths on every used link, and

(2) large revenue in tolls collected from travellers on network

This has forced planners to investigate more realistic and implementable tolling strategies. Some of them are discussed in the sub-section below.

2.5 Types of Tolls

Some tolling strategies studied by researchers and discussed in [2] are defined below in brief:

1. MINSYS – We try to minimize the total revenue collected in the system through tolls

2. MINTB – We minimize the total infrastructure cost in setting up of tollbooths by minimizing the number of tollbooths in the system

3. MINMAX – We minimize the maximum toll that is implemented on any link

4. ROBINHOOD – We constrain the total revenue collected from the system to be zero while allowing tolls to be negative as well as positive. So, on some links users get subsidies while on the others they get tolled bringing the total collected revenue to 0

3 Mathematical Formulation

3.1 Feasibility Equations

The following are the feasibility conditions that define a valid toll vector:

$$Z^T\left(\, t(x^*) + \beta \,\right) \geq A^T \rho_{rs}$$

$$(x^*)^T\left(\, t(x^*) + \beta \,\right) = d_{rs}^T \rho_{rs}$$

Here, Z is the arc-path incidence matrix, A is the OD-path incidence matrix, and ρ_{rs} is a vector of unknown constants specific to each OD pair.

The feasibility conditions of toll pricing define the polyhedron of possible toll vectors that can be implemented in the network to ensure user equilibrium as well as system optimum flows. This means that when implementing any of the different toll pricing strategies, the feasibility conditions must be satisfied. According to [1] these conditions are obtained by performing straightforward manipulations on the KKT conditions of the system optimal linear optimization program.

3.2 Linear Programming Formulations

The tolling strategies being studied in this project are MINSYS and MINTB corresponding to minimum system toll revenue and minimum number of toll booths considerations respectively. Both strategies can be formulated as optimization problems with a linear objective function and linear constraints making them linear programming problems. The exact mathematical forms of the two LPs are given below. Being LPs they can be easily solved using standard LP solvers like the one in the MATLAB linear programming toolbox. However, when the network size is large, LP solvers sometimes fail to provide a valid solution in which case heuristic algorithms are used with some modifications in the formulation. This problem however was not encountered in this project and LP solver provided valid solutions to the problems.

3.2.1 MINSYS

The LP for this problem [2] can be written as:

$$
\begin{aligned}
min_{(\beta,\rho)} \quad & \beta^T x^* \\
s.t. \quad & Z^T(\, t(x^*) + \beta \,) \geq A^T \rho \\
& (x^*)^T(\, t(x^*) + \beta \,) = d^T \rho \\
& \beta \geq 0
\end{aligned}
$$

The objective is to minimize the product of the toll vector and the SO flow vector, that is the total revenue of the system. The first two constraints are the feasibility conditions while the third one is simply a non-negativity constraint on tolls ensuring that there are no subsidies on any link.

3.2.2 MINTB

The LP for this problem [2] can be written as:

$$
\begin{aligned}
min_{(y,\beta,\rho)} \quad & \sum y_{ij} \\
s.t. \quad & Z^T(\, t(x^*) + \beta \,) \geq A^T \rho \\
& (x^*)^T(\, t(x^*) + \beta \,) = d^T \rho \\
& \beta \leq M y_a \\
& \beta \geq 0 \\
& y_{ij} \in \{0,1\}
\end{aligned}
$$

Here y_a signifies the presence or absence of tollbooth on a link. It is 1 if a tollbooth is present and 0 if it is absent. The objective is to minimize the sum of y_a on all links, that is the number of tollbooths in the entire system. The first, second and forth constraints are same as the MINSYS problem. The third constraint links β with y_a. M is a positive constant which stores the upper limit for tolling on a link.

4 Methodology (Programming Implementation)

Figure 1 presents a flow chart of the steps involved in the applied methodology. The following is an explanation of the intricacies involved in each of the steps. The first three steps are performed using Python programming language while Step 4 and 5 are performed using the MATLAB linear programming toolbox.

Step 1. Change the link cost functions to a sum of their original cost function plus the MSCP tolls functions and solve the network for User Equilibrium. As per the Mathematical Equivalence proved in section 1.3, this solution gives us the un-tolled SO solution for the given network. The SO flows and costs will be used in solving the Linear programs for the different tolling strategies.

Step 2. Enumerate all paths between each individual OD pair in the network using a recursive approach. The paths are stored as lists of nodes forming the paths. They are then converted into lists of links forming the paths for ease in computing the Z matrix.

Step 3. Using the lists of paths obtained in Step 2, we compute the Z and A matrices. The Z matrix consists of as many rows as the number of links and as many columns as the number of total paths between all OD pairs. The links that are present in the path have 1 and the others have 0 stored in the matrix. The A matrix consists of as many columns of 1 as the number of paths.

The ρ matrix is also created while creating the Z and A matrices. It contains as many rows as the number of OD pairs and columns equal to the number of total paths including all OD pairs. The columns corresponding to paths for an OD pair are stored as 1 in the row corresponding to that OD pair and the rest are kept as 0.

Step 4. The matrices are imported into MATLAB and manipulated according to the linear programming toolbox requirements. The function used is given as:

Y=linprog(objective, A, b, Aeq, beq, ub, lb);

Where ‘objective’ stores the objective function, A and b store the inequalities, Aeq and beq store the equalities, and ub and lb store the upper and lower bounds of unknown variables. Y stores the result obtained after minimizing the objective function considering all the constraints. For the MINSYS problem, the equalities and inequalities correspond to feasibility constraints. For MINTB problem, they correspond to the feasibility constraints as well as the additional constraint between β and y_a.

The lower bound for tolls is set as 0, while the upper bound is set as positive infinity. The lower and upper bound for rho values is kept as negative and positive infinity as no bounds are proposed for it in the literature. The rest of the formulation can be taken from the section on mathematical formulations.

Step 5. We run the linear programming solver to obtain the β(toll vector) and ρ (unknown constant) values.

In the following sections we implement this methodology on different networks namely Braess, Nine-Node and Sioux Falls, and discuss the results obtained.

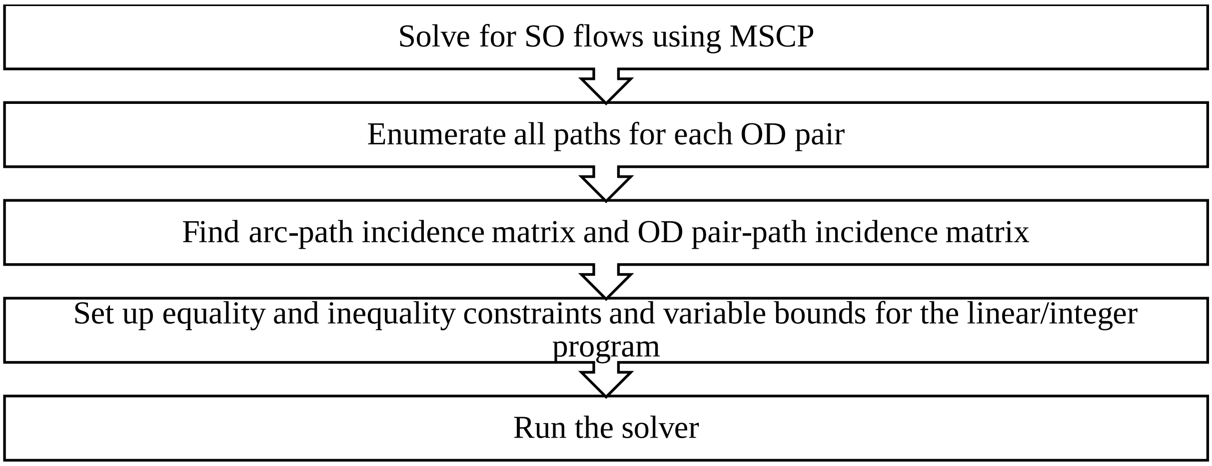

FIGURE 1 Flowchart of applied methodology

5 Small network implementation

This section discusses the implementations of MINSYS and MINTB strategies on small networks primarily for the purpose of demonstration. The networks chosen for the implementations are the Braess network and the Nine-node network. The MINSYS and the MINTB implementations on these networks have already been accomplished by researchers in the past [1]. The motivation behind re-estimating the models is to recreate the results and validate the optimization technique undertaken in the present work for comparisons with the established results. Section 5.1 talks about the Braess network followed by section 5.2 which deals with the Nine-node network.

5.1 Braess Network

The Braess network is 4-node 5-link network historically used for the purpose of illustrating the classical Braess paradox. The network is shown below in Figure 2. The link performance functions are linear and are shown along the links. The demand for the network is given as 6 vehicles from node 1 to node 4.

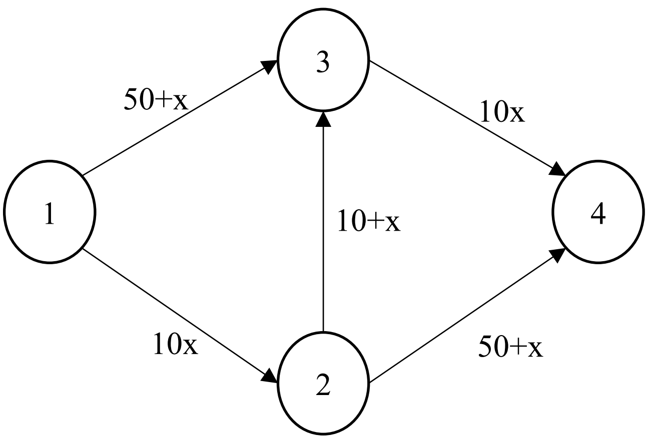

FIGURE 2 Braess Network with Link performance functions

The number of possible paths between the OD pair 1-4 are three (1-2-4, 1-3-4, 1-2-3-4). At user equilibrium (UE) all the three paths are used with 2 vehicles on each of them. The travel time on each path is 92 minutes. The flow and link travel times for the UE solution is shown below in Figure 3.

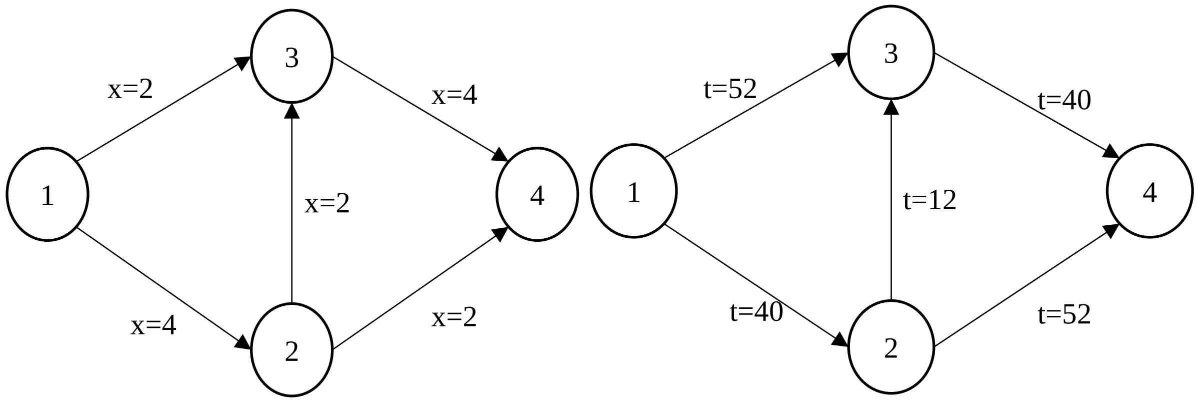

FIGURE 3 UE solution of the Braess Network

The system optimal flows for the network occurs when the each of the paths 1-2-4 and 1-3-4 has a flow of 3 vehicles. The path travel times for each of the two used paths turns out to be 83 minutes. The link flows and travel times are shown in the following Figure 4

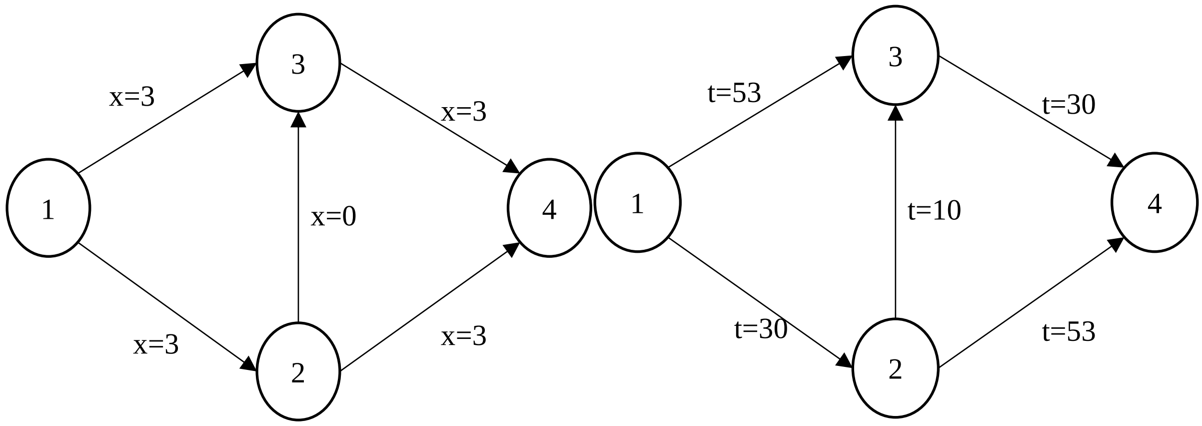

FIGURE 4 SO solution of the Braess Network

However, one might observe that the system optimal flows do not satisfy the principle of user equilibrium (since the path 1-2-3-4 has a travel time lower than the other two used paths). Here is where tolling plays an important role. By introducing appropriate tolls, the system optimal flows can be made to satisfy the user equilibrium (tolled user equilibrium) as well. The following sections describes the MSCP, MINSYS and MINTB tolling strategies on the Braess network.

5.1.1 MSCP tolls on Braess network

MSCP or Marginal Social Cost Pricing tolling strategy provides a very simple mechanism for achieving the equivalent tolled user equilibrium to the system optimal flows. As mentioned in section 2.2, this type of tolling involves introducing the toll of value equivalent to the product of the link flows and the derivative of the link performance function on each of the links of the network. Therefore, mathematically speaking, all links with non-negative flows and increasing link performance function will have tolls. Nevertheless, the tolled user equilibrium for the network conforms to the system optimal flows. The MSCP toll values obtained for the Braess network is shown in the following Figure 5.

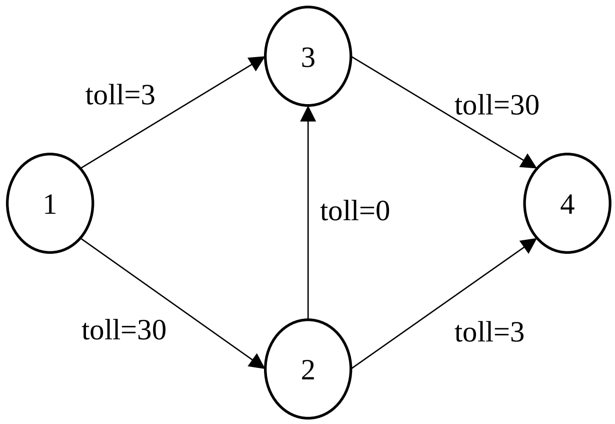

FIGURE 5 MSCP tolling on Braess Network

Under the MSCP tolls, considering the system optimal flows, the path cost for path 1-2-4 and 1-3-4 is 182 minutes each (here, we have considered the generalized travel cost in minutes). On the other hand, the path cost for path 1-2-3-4 is 250 minutes! Therefore, this tolling strategy satisfies the user equilibrium as all used paths between 1 and 4 have equal and minimal travel cost, and hence no driver can improve her/his cost by switching to another route. However, the MSCP strategy involves tolling all the used links, and thereby collecting a significant amount of revenue from the system, which appears to be unreasonable. Even for a small network like the Braess, four out of five links have been tolled with a revenue of 198 units! Hence, we look at a more sophisticated tolling mechanisms like the MINSYS and MINTB in the following two sections.

5.1.2 MINSYS tolls on Braess network

The MINSYS implementation minimizes the total revenue collected from the total as explained in section 3.2.1. The figure below presents the result of the MINSYS strategy on the Braess network in terms of the tolls imposed on each link.

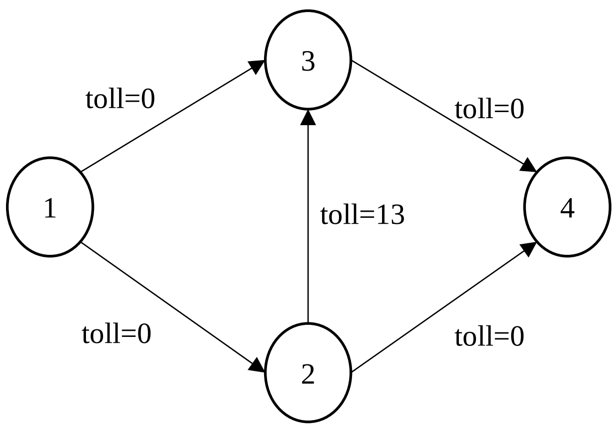

FIGURE 6 MINSYS tolling on Braess Network

The MINSYS strategy results in just the one toll equivalently to 13 minutes on link 2-3. If we evaluate the path travel times under system optimal flows and MINSYS implementation, we see that all the paths (1-2-3, 1-3-4 and 1-2-3-4) have travel time/cost equal to 83 minutes. Hence, this system of flow satisfies the user equilibrium principle under this MINSYS strategy. The total revenue collected under this configuration is zero as no users use the link 2-3; this toll acts as an impeding factor discouraging the use of the middle link. Another popular strategy is the MINTB strategy which is discussed in the next section.

5.1.3 MINTB tolls on Braess network

The MINTB implementation minimizes the total number of toll booths to be imposed on the network to make the system optimal flows equivalent to the tolled user equilibrium flows. The MINTB implementation on the Braess network is trivial and it reproduces the exact same result obtained from the MINSYS implementation as shown in Figure 6.

The summary of the three type of tolling implementations on the Braess network is given in the following table, Table 1.

Table 1 Tolling implementation results for Braess Network

	MSCP	**MINSYS**	**MINTB**
Total revenue ($\beta^T x^*$) in minutes	198	0	0
Average toll per user in minutes	33	0	0
Number of toll booths	4	1	1

5.2 Nine-node network

The nine-node network is a 18-link 9-node network which has been used by several researchers earlier for various implementations. The network is shown in Figure 7. Node 1 and node 2 are the origin nodes and node 3 and node 4 are the destination nodes. The demand are as follows: 10 between node 1 and 3, 20 between node 1 and 4, 30 between node 2 and 3, and 40 between node 2 and 4. The link performance functions follow the standard BPR functions with α=0.15 and β=4. The labels along the links in the figure denote the free flow time and the capacity of the links in the form (free flow time, capacity).

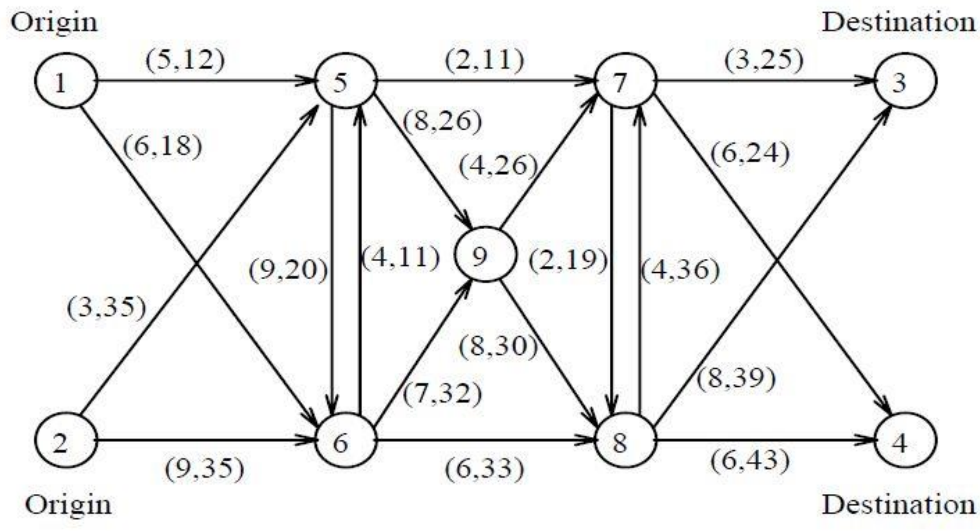

FIGURE 7 The Nine-Node Network

There are a total number of 24 possible paths between each O-D pair in this network. We attempt to implement the MSCP, MINSYS and the MINTB strategies on this nine-node network.

5.2.1 MSCP tolls on Nine-node network

MSCP tolling is a naïve way of tolling a network to achieve the system optimal flows. As discussed earlier, MSCP strategy impose a toll on all the links that have positive flow value, hence sometimes unreasonable. The figure below represents the location of tolls for the MSCP implementation on the nine-node network.

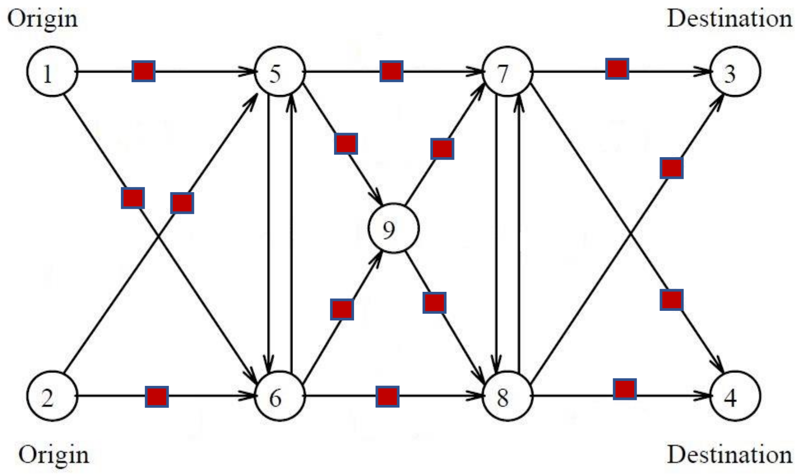

FIGURE 8 MSCP tolling on Nine-Node Network

It is observed that 14 out of the 18 links are imposed with a positive toll for the MSCP implementation. The average toll per user was found to be close to 15 minutes. The four links that are not tolled are actually the ones with zero flows. Since such an 'unsparing' tolling scheme is highly unlikely, we look into more parsimonious tolling strategies like the MINSYS and MINTB for the Nine-node network.

5.2.2 MINSYS tolls on the Nine-node network

As discussed earlier, the objective of the MINSYS implementation is to minimize the total revenue collected by the tolls. The figure below shows the links on which tolls have been imposed to achieve the minimum revenue model.

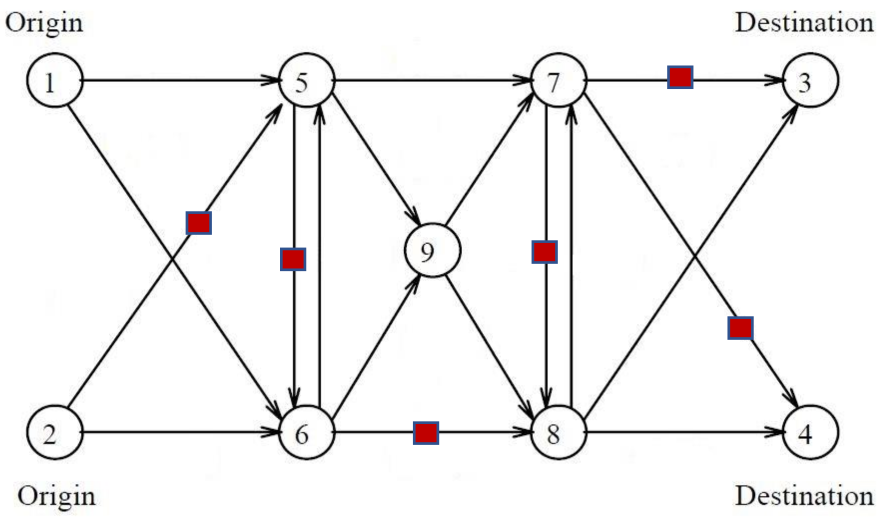

FIGURE 9 MINSYS tolling on Nine-Node Network

It is observed that six links are tolled when the MINSYS strategy is implemented on the Nine-node network with total revenue collected as 887.562 minutes. It was interesting to note that the MINSYS results obtained from the current work slightly differed from the MINSYS implementation by [2]. The MINSYS results obtained by [2] reports to have five toll booths with a total revenue of 887. 574. Although the difference in the total toll revenue obtained from the two works may be negligible, the corresponding number of toll booths obtained in the current work is one more than the other. Since the objective of the MINSYS implementation is to minimize the total toll revenue, it implies that converging to a lower revenue value is the highest priority, hence the number of toll booths imposed are inconsequential. The slight difference in the results could be due to the different computational capacity or optimizing software methods applied by [2] and the current study.

5.2.3 MINTB on the Nine-node network

The focus of the MINTB implementation, as discussed earlier, is to minimize the total number of toll booths in obtaining the tolled user equilibrium equivalence of the system optimal flows. The results obtained for this implementation in terms of the links that are tolled is shown in the figure below.

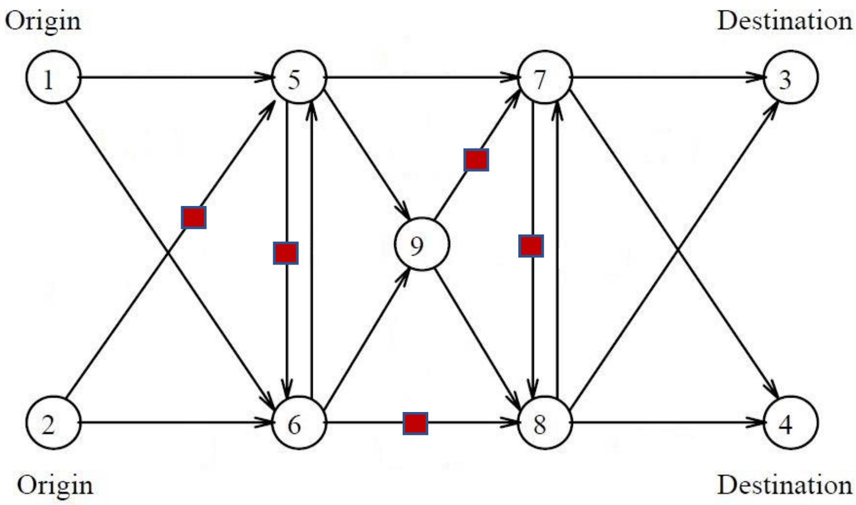

FIGURE 10 MINTB tolling on Nine-Node Network

The results indicate that a minimum of 5 toll booths are required to be imposed on the network to achieve system optimal flows. The links are marked in the figure. The total revenue collected by this configuration is 887.57, which is marginally higher than that of the MINSYS. The MINTB result obtained here, however, conform with the work of [2], unlike the MINSYS implementation. The summary of the MSCP, MINSYS and MINTB results are presented in Table 2.

Table 2 Tolling implementation results for Nine-Node Network

	MSCP	**MINSYS**	**MINTB**
Total revenue ($\beta^T x^*$) in minutes	1493.478	887.56	887.57
Average toll per user in minutes	14.93	8.88	8.88
Number of toll booths	14	6	5

The smaller network implementation involving the Braess and the Nine-node network were presented for a reasonable explanation of the working of the various toll strategies tested in this present work. A real-world implementation of this methodology would demand an implementation on a larger real-life network. The section after following section extends this methodology to a larger network, describing the subtle changes that are required to be made when such tolling strategies are implemented on a larger network. However, before moving on to the larger network implementation, we slightly deviate to see the importance of convergence in tolling.

5.3 Importance of convergence

As an aside, we would also like to bring to the notice of the readers, the importance of converging traffic assignment to a ‘sufficient’ level. The MINSYS implementation on the nine-node network was executed several numbers of times for various convergence level to observe its impact on the tolling outcomes. As an experiment, the system optimal traffic assignment on the nine-node network was executed to three different relative gaps: $10^{-3}, 10^{-4}$ and 10^{-6}. The MINSYS optimization was then performed on these three different networks, the resulting system travel times and toll revenue values are presented in the following Table 3.

Table 3 Tolling Implementation results variation with level of convergence

Relative gap	**E-3**	**E-4**	**E-6**
TSTT	2254.033	2253.918	2253.916
Total revenue ($\beta^T x^*$) in minutes	1281.92	887.57	887.56
Average toll per user in minutes	12.82	8.88	8.88
Number of toll booths	**8**	**5**	**6**

The difference in the total system travel time (TSTT) between the convergence level of E-3 and E-4 may appear to be trivial, however, it has far reaching implications as far as the tolls are concerned. The average toll per user observed for the E-3 assignment is about 12.82 minutes whereas for the E-4 and E-6 assignments, they were about 8.88 minutes. The number of toll booths required, in turn, for the E-3 assignment was much higher. Even at E-4 and E-6 convergence level, we observe a difference in the number of toll booths required to obtain the desired minimization objective. Hence, converging to a ‘sufficiently’ low level of relative gap is crucial while implementing the tolling strategies so that the resulting outcomes are stable and reliable. Having made our point regarding the vitality of convergence, we now move on to the large network implementation section.

6 Large network Implementation

This section describes our implementation on a large realistic network. The Sioux Falls network is used to implement the various toll strategies for our study. This network consists of 24 nodes and 76 links and a total demand of 360,600 vehicles. Although the overall methodology is similar to the small network implementation, there are two slight modifications that are required in order to run the implementations on larger networks. The two caveats in the original methodology and their corresponding modifications for larger networks are described below.

i. The total number of paths increases exponentially with the size of the network. Hence, enumerating all the paths in the network and using them in the optimization step becomes tedious. The arc-path incidence matrix (Z) and the OD pair-path incidence matrix (A) both requires enumeration of paths, and the order of such matrices explodes with the size of the network. For example, for the Sioux Falls network, the total number of acyclic paths considering all OD pairs come up to be around 1.71 million with an average close 2,950 paths per OD pair! With a total of possible 576 OD-pairs, the A matrix would then consists of close to 1 billion elements! This would make data handling tedious and computational time enormous.

One way to avoid this situation is to consider only a set of reasonable paths between each OD pair instead of all the paths. Since the set of all possible paths would include a bulk of unreasonable paths and irrational de-routes, truncating the number of paths to a reasonable few would not affect the outcome of the results to a great extent. Practically speaking, the total number of paths that an user would consider while going from an origin O to a destination D would not be more than 5,10 or may be 20 odd paths at the most. To avoid using all the possible paths, the method that is utilized in this study to decide a set of reasonable paths is the k-shortest paths strategy. Instead of considering all the possible paths between an OD pair, only the top ‘k’ shortest acyclic paths are taken as the set of paths that are used between a certain OD pair. This method involves enumerating all the possible paths between an OD pair and then selecting the top ‘k’ shortest paths for the rest of the implementation.

ii. The optimization formulation (say, for the MINSYS strategy) is given by the following set of objective function and constraints.

$$\begin{aligned} & min_{(\beta,\rho)} && \beta^T x^* \\ s.t. \quad & && Z^T(\, t(x^*) + \beta\,) \geq A^T \rho \\ & && (x^*)^T(\, t(x^*) + \beta\,) = d^T \rho \\ & && \beta \geq 0 \end{aligned}$$

The equality constraint used to define the set of feasible tolls becomes a very strict constraint for larger networks. Often for larger networks, the flows and tolls are described with precision up to 4 or 5 decimal places, and with the enormous number of variables that such networks have, it becomes very unlikely that this equality constraint would always be exactly satisfied. To overcome this problem, the difference between the left-hand side and the right-hand side of the equality constraint is shifted to the objective function with a high penalty multiplier ‘μ’. The modified optimization formulation is presented below,

$$\begin{aligned} min_{(\beta,\rho)} \quad & \beta^T x^* + \mu[(x^*)^T(\, t(x^*) + \beta\,) - d^T \rho] \\ s.t. \quad & Z^T(\, t(x^*) + \beta\,) \geq A^T \rho \\ & \beta \geq 0 \end{aligned}$$

Similarly, for the MINTB optimization formulation for larger networks is modified as below,

$$min_{(y,\beta,\rho)} \sum y_{ij} + \mu[(x^*)^T(\, t(x^*) + \beta\,) - d^T \rho]$$

$$\begin{aligned} s.t. \quad & Z^T(\, t(x^*) + \beta\,) \geq A^T \rho \\ & \beta \leq M y_a \\ & \beta \geq 0 \\ & y_{ij} \in \{0,1\} \end{aligned}$$

6.1 MSCP, MINSYS and MINTB tolls on Sioux Falls network

Since the detail working of these three tolling strategies (MSCP, MINSYS and MINTB) are already explained under the small network implementation, the results for the Sioux Falls network for these strategies are summarised in this section together. Before implementing the tolls, we need to set the 'k' value for the k-shortest path method and the 'μ' value for the penalty multiplier. For the MINSYS and MINTB implementations on the Sioux Falls network, the top 100 shortest acyclic paths are considered to be as the reasonable set of paths between each OD pair (for the OD pair that had less than 100 possible acyclic paths between them, all of them were included in the reasonable set). For the optimization formulation, a penalty value of 1 million was used for 'μ'.

Additionally, in order to report the tolls in monetary units (dollars), a value of travel time (VOTT) value of $10/hr has been assumed for this study. This value is also assumed in an implementation in [3]. The summary of the MSCP, MINSYS and the MINTB implementations are presented in the following Table 4.

Table 4 Tolling implementation results for Sioux Falls network

	MSCP	MINSYS	MINTB
Total revenue ($\beta^T x^*$) in minutes	14.93 E6	2.08 E6	2.23 E6
Toll revenue ($)	$2488K	$346K	$368K
Average toll per user	$6.9	$0.95	$1.0
Number of toll booths	76	39	38

Expectedly, the MSCP strategy is seen to have significantly higher toll revenue collection than the MINSYS and MSCP with an average toll per user of about $7! Also, since all the links with a positive flow are assumed to have tolls, the total number of toll booths for the MSCP strategy would be 76 for the Sioux Falls network. Since, such a tolling strategy is grossly unreasonable, the MINSYS and MINTB strategies provide better alternatives.

Since the objective of the MINSYS implementation is to minimize the total revenue collected, we obtain the least average toll per user value for this which is just under one dollar. The number of toll booths required to minimize the total revenue is found to 39. We observe that the total toll revenue is brought down to below $350K for the MINSYS implementation from about $2.5M for MSCP strategy.

Since construction and establishment of a toll involves considerable investment, minimizing the total number of toll booths for obtaining system optimal flows may be the most economical option from the point of view of the planners. The MINTB implementation on the Sioux Falls network reveals that a minimum of 38 toll booths are required to obtain the system optimal flows. By minimizing the total number of tolls, the total revenue goes up by a small margin to an average toll per user of about $1. The following figure represents the tolled link of the Sioux Falls network for the MINTB implementation (the red blocks indicate on which links are the toll booths required to be imposed).

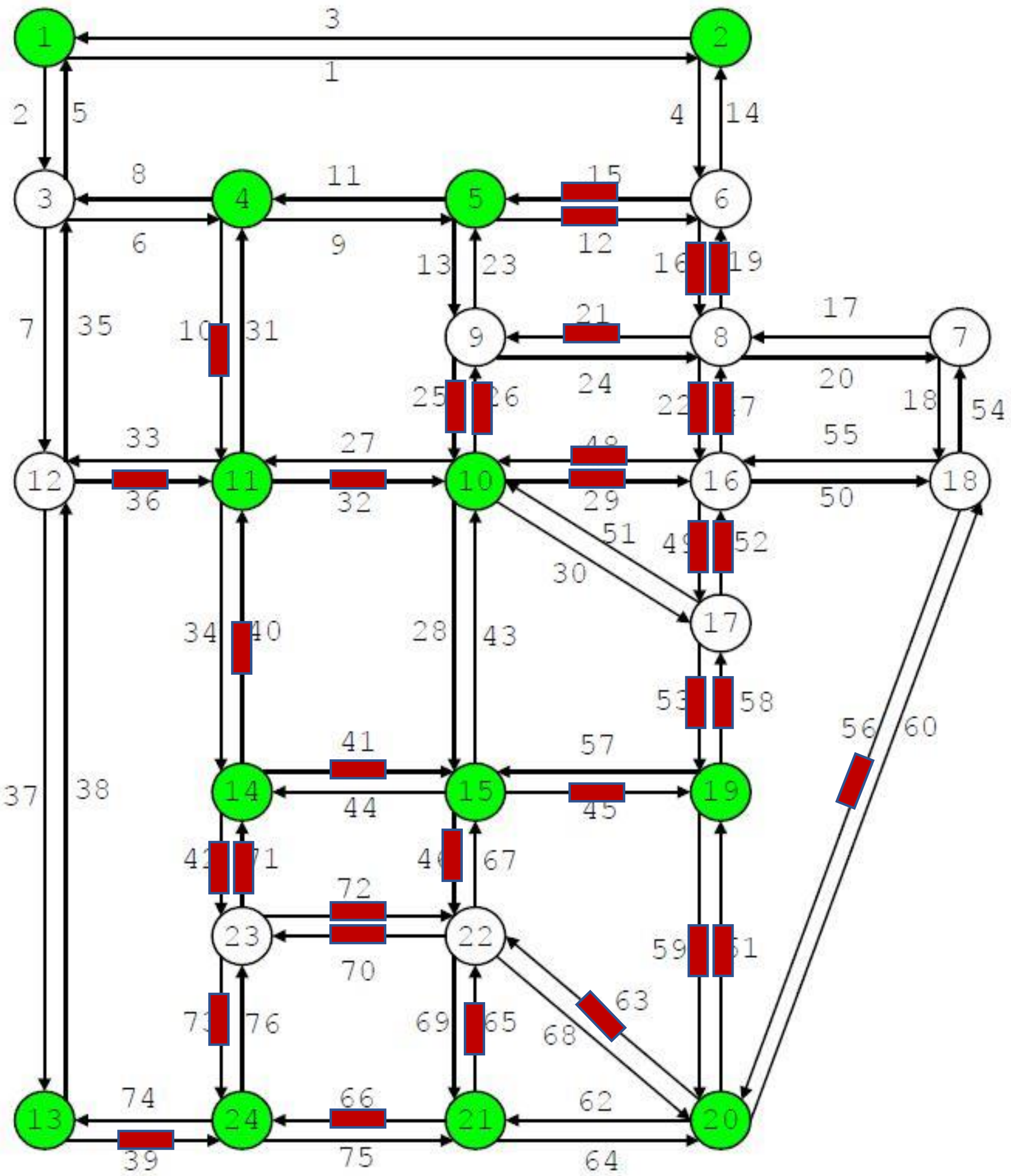

FIGURE 11 The Sioux Falls Network – MINTB implementation (Red blocks indicate tolls)

The pattern of the tolling suggests that more tolls are required towards the central part of the network (where the congestion might be more in effect) than the peripheral links. This is intuitively obvious if we draw an analogy to the congestion pricing concept. Users are discouraged to use already congested links and are encouraged to take a de-route to non-congested links with the idea to minimize the total system travel time. Similar to the Braess and the nine-node networks, once again we observe that the MINSYS and MINTB solutions are not extravagantly different from each other. The MINTB happens to reduce the total number of booths by just one unit for a relatively large network like the Sioux Falls. The average toll per user obtained from both the implementations were also very similar.

7 Toll variation with the change in demand

This study also makes an attempts to empirically observe the variation in toll with the change in demand. Specifically, the variation of the MINSYS/MINTB tolls obtained for different overall demand levels of a network. Firstly, we looks at the Nine-node network and perform the MINSYS implementation for various demand levels. This is followed by the demand variation experiment on the Sioux Falls network for the MINTB formulation.

7.1 Effect of demand variation on toll for the Nine-node network

To study the effect of demand variation on the tolling outcomes, the overall demand for the nine-node network is varied at three different levels and the outcomes of the MINSYS strategy are compared (this could also have been done for the MINTB implementation alternatively). The three other demand levels other than the base demand were – 50% decrease in demand, 50% increase in demand and 100% increase in demand. The MINSYS results obtained for the different demand levels are shown in Table 5.

Table 5 Effect of Demand Variation in Nine-Node Network

	Base demand	**50% decrease**	**50% increase**	**100% increase**
Demand	100	50	150	200
Total revenue ($\beta^T x^*$) in minutes	887.56	143.20	1591.50	2060.90
Average toll per user in minutes	8.88	1.43	10.61	10.30
Number of toll booths	6	2	8	8

From the results, it is observed that for the reduced demand, the tolling requirement for the MINSYS implementation falls significantly, with just two tolls and the average toll per user at just 1.43 minutes. However, for the increased levels of demand (50% and 100%), the results were very similar. Of course the total revenue of the 100% increased demand level is higher due to the sheer increase in the users, but both the increased levels have same number of tolls. In fact the average toll per user for the 50% increase demand level was slightly lower than that of the 100% increased demand level. Although a general increase in the number of tolls is observed as we increase the demand, there is no conclusive trend regarding the average toll per user, especially on the higher demand levels.

7.2 Effect of demand variation on toll for the Sioux Falls network

A similar analysis is done for the Sioux Falls network for the MINSYS implementation. The demand levels were increased and decreased by 50% for two separate scenarios and the results obtained for each of the MINSYS implementation is given below in Table 6.

Table 6 Effect of Demand Variation in Sioux Falls network

	Base demand	**50% decrease**	**50% increase**
Demand	360,600	180,300	540,900
Total revenue ($\beta^T x^*$) in minutes	2.08 E6	0.22 E6	2.66 E6
Toll revenue (in $)	$346K	$37K	$444
Average toll per user (in $)	$0.95	$0.20	$0.82
Number of toll booths	39	31	34

The results indicate that when the demand is decreased by 50%, it leads to a significant reduction in the total revenue collected (about 90% reduction) as well as the average toll per user (about 80% reduction). The number of tolls for the MINSYS strategy also goes down from 39 to 31. However, the increased demand scenario provides interesting outcomes. Although the toll revenue shoots up, it is primarily because of the direct increase in the total number of users. What is interesting is the total number of toll booths required to achieve the MINSYS optimization is seen to decrease from 39 to 34. Additionally, the average toll per user is also observed to be reasonably lesser than the original case.

A clear pattern between demand and tolls could not be observed by this empirical experiment of varying the demand for each MINSYS implementation. However from the results obtained, few intuitive deductions can be made. It appears that the tolling requirements (number of toll booths required, average toll per user) generally increases when the demand increases from a state in which the entire network operates at a fairly uncongested level to a state wherein the network is just about to be congested. For a change in demand for any higher demand levels, the tolling requirement does not seem to follow a general pattern and it depends upon several other factors like the link performance functions, demand distribution and the toll values themselves. It is rather safer to state that the tolling requirements are network-specific even for varied demand levels.

7.3 Effect of partial demand variation on MINSYS tolls

In this section we empirically analyse the impact of partial demand variation on the MINSYS tolls for the Sioux Falls network. We experiment with two scenarios which are given as follows,

(i) Vary the demand among some OD pairs along the periphery of the network.
(ii) Vary the demand among some OD pairs in the interior of the network.

7.3.1 Scenario 1: Demand variation along peripheral OD pairs

For this scenario, the demand between OD pairs 1-2, 2-1, 1-13 and 13-1 are artificially increased by 2,500 each (resulting in an increase of a total of 10,000 vehicles on the network). These nodes lie on the periphery of the Sioux Falls network, hence chosen for this specific scenario. The rest of the network is kept as it is in the original case. The MINSYS strategy is implemented, and the resulting network (with tolls) is shown in Figure 12. The resulting network obtained is the exact same configuration as the MINSYS implementation on the original network. The revenue increase is also observed to be less than 1% from the original network.

7.3.2 Scenario 2: Demand variation along internal OD pairs

For this scenario, the demand between OD pairs 11-14, 14-11, 11-19 and 19-11 are artificially increased by 2,500 each (resulting in an increase of a total of 10,000 vehicles on the network). These nodes lie on the internal part of the Sioux Falls network, hence chosen for this specific scenario. The rest of the network is kept as it is in the original case. The MINSYS strategy is implemented, and the resulting network (with tolls) is shown below in Figure 13.

The green boxes in Figure 13 indicate that the tolls at these locations are removed (3 in number) which were previously present in the original network, whereas, the yellow boxes indicate the new tolls that are added (six in number) when the demand in the internal OD pairs are increased. The table below, Table 7 summarizes the results of the 2 scenarios of the partial demand variation experiments.

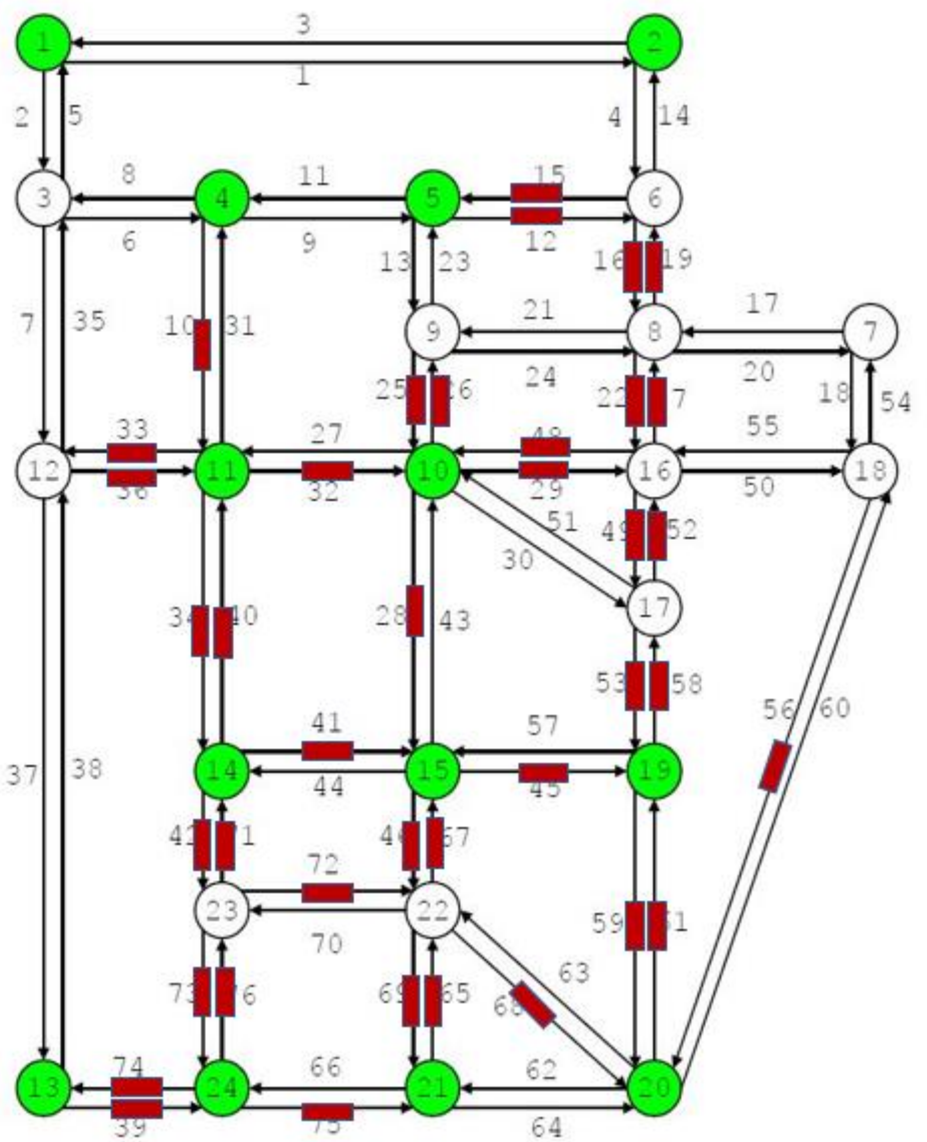

FIGURE 12 MINSYS implementation for partial demand variation - Scenario 1

Table 7 MINSYS implementation results for scenario 1 and 2 on Sioux Falls Network

	Base network	**Peripheral demand increase**	**Internal demand increases**
Demand	360,600	370,600	370,600
Total revenue ($\beta^T x^*$) in minutes	2.080 E6	2.086 E6	2.62 E6
Toll revenue (in $)	$346K	$347K	$437
Average toll per user (in $)	$0.95	$0.94	$1.18
Number of toll booths	39	39	42

The comparative analysis of the two scenarios reveal interesting observations. The increase I demand on the peripheral nodes did not have any significant impact on the tolling configuration of the network; in fact, it resulted in the exact MINSYS toll locations as that of the original network, with a marginal increase in the total revenue. The number of toll booths remained the same. On the other hand, when the demands for some of the OD pairs at the internal part of the network were increased by the same amount, the impact on the overall tolling scheme was profound. For the MINSYS implementation, the total revenue went up by more than 25%. The number of toll booths required to achieve the MINSYS strategy also increased by 3. What is more thought-provoking observation is that the effect of the increased demand on the network was not localized; all the additional tolls were actually located away from these nodes where the demand increased. Such findings reinforces the fact that network dynamics are not always intuitive, an alteration in any part of the network may have a significant impact on some other part as well.

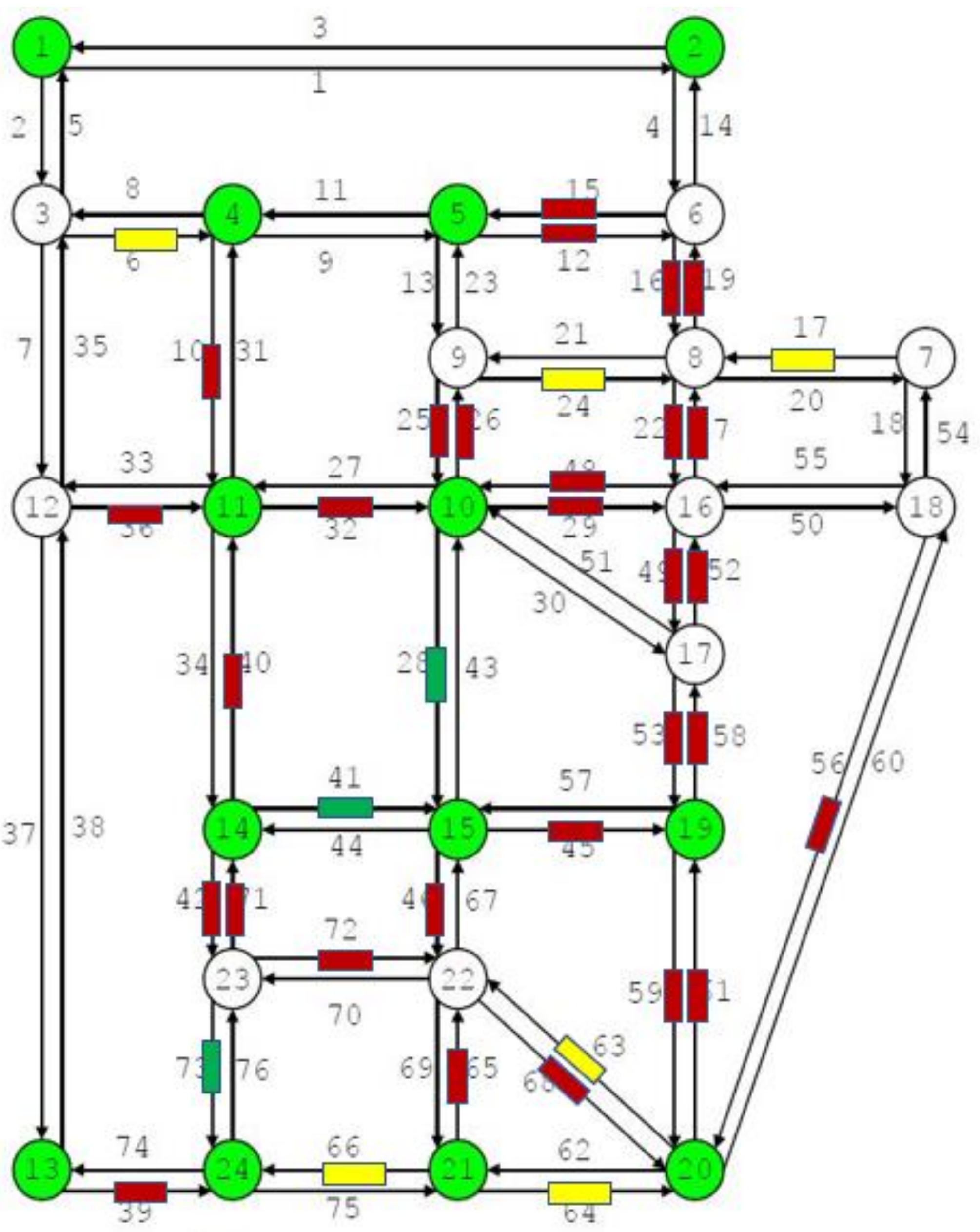

FIGURE 13 MINSYS implementation for partial demand variation - Scenario 2 (yellow blocks indicate new toll locations as compared to the original network, green blocks indicate removal of the tolls which were present in the original network)

8 Summary and conclusions

This study implements the MSCP, MINSYS and the MINTB tolling strategies on networks of various sizes and compares their outcomes. Smaller networks like the Braess network and the nine-node network were utilized to show how each of these tolling strategies achieve the equivalence between the system optimum flows and the tolled user equilibrium flows. The implementations were then extended to a larger and more realistic network, the Sioux Falls network. This was followed by an empirical analysis on the impact of overall and partial demand variation on the tolling configuration of the entire network.

There are a number of conclusions and inferences that can be drawn from this study, they are listed below point-wise.

i. Since the MSCP tolling strategy directly imposes the product of the link flows and the derivative of the performance functions as the link tolls, it often produces unrealistically large toll values. Moreover, this strategy introduces a toll on every link that has a positive flow. On the other hand, MINSYS and MINTB tolls are observed to produce more realistic toll values, with the former focussing on minimizing the total toll revenue, whereas the later focuses on minimizing the total number of toll booths to obtain system optimal flows.
ii. Comparative analyses of the tolling strategies on the various network sizes reveal that the results of the MINSYS implementations are not very different from the MINTB implementations. A possible reason for this could be that the total toll revenue collected is somewhat linked with the total number of toll booths, hence while minimizing the one, the other is partially minimized as well. Since establishment of toll booths requires considerable investment, transport planners might be incentivized to prioritize MINTB over MINSYS, especially when the eventual outcomes are not significantly different.
iii. Convergence of the relative gap measure to a sufficient low level is absolutely crucial for any toll strategy implementation. Insufficient convergence leads to unstable and erroneous toll schemes.
iv. Implementing these tolling optimization strategies on larger networks require handling two major issues – consideration of all possible paths in the network, and satisfying the equality constraint of the feasible toll set condition. Using the k-shortest paths method to truncate the total number of paths to a reasonable set, and using a penalty multiplier by introducing the equality constraint to the objective function respectively were found to be suitable in handling these two issues.
v. The empirical study on the effect of overall demand variation on the tolls (for the MINSYS implementation) indicates that although the total toll revenue increases with the increase in total demand, the average toll per user or the number of toll booths required for achieving the MINSYS objective do not follow a particular trend. However, an intuitive deduction from the results suggests that the number of toll booths required and average toll per user generally increases when the overall demand of the network increases from a state in which the entire network operates at a fairly uncongested level to a state wherein the network is just about to be congested. For an increase in demand at any higher demand levels, the tolling aspects do not seem to follow a general pattern and it depends upon several other factors like the link performance functions, demand distribution and the toll values themselves. It is rather safer to state that the tolling trends are network-specific even for varied demand levels.
vi. An empirical analysis on the impact of partial demand variation on tolling schemes disclose that a change in demand on the peripheral OD pairs for the Sioux Falls network did not have a significant impact on the overall tolling configuration of the network. On the other hand, an equivalent change in the OD pair demand values for nodes in the internal

part of the network was seen to have a profound effect on the overall tolls (for MINSYS implementation), even at links far away from the OD pairs where the demand was increased

References

[1] P. Bergendorff, D. W. Hearn and M. V. Ramana, 1997, Congestion Toll Pricing of Traffic Networks, Network Optimization 51-71.

[2] D. W. Hearn, M. V. Ramana, 1998, Solving Congestion Toll Pricing Models. Equilibrium and advanced transportation modelling, 109-124.

[3] S.D. Boyles, K. M. Kockelman, S. T. Waller, 2010, Congestion pricing under operational, supply-side uncertainty. Transportation Research Part C 18 pg. 519–535.

[4] Transportation Networks for Research Core Team. *Transportation Networks for Research*. https://github.com/bstabler/TransportationNetworks. (Accessed: November, 14, 2018)