

A Scatter Search Algorithm for Solving a Bilevel Optimization Model for Determining Highway Tolls

José Luis González Velarde¹, José-Fernando Camacho-Vallejo², and Gabriel Pinto Serrano¹

¹ Tecnológico de Monterrey, Mexico

² Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Mexico

gonzalez.velarde@itesm.mx, jose.camachovl@uanl.edu.mx, gabrielpinto_s@hotmail.com

Abstract. The problem of determining optimal tolls established on a subset of arcs in a multicommodity capacitated transportation network is presented. The problem is formulated as a bilevel optimization problem where the upper level consists of an administrator who establishes tolls in some arcs of a network, while the lower level is represented by a group of users who travel along the shortest paths with respect to the travel cost. The objective is not only to increase the tolls, but also to maintain an optimal flow on the arcs of the network in order to maximize the leader's profit. If the leader sets very high toll values, the followers will be discouraged from using the tolled arcs, so the profit obtained from that decision is not going to be convenient for the leader. A methodology to solve this problem using optimization software at the lower level and the metaheuristic Scatter Search at the upper level is proposed.

Keywords. Bilevel programming, scatter search, toll optimization problem.

1 Introduction

The importance of transportation for the economic and productive growth of any organization or country is unquestionable. An area that has acquired a great relevance within this subject is the one related to highway tolls. Tolls help to reduce congestion and lessen the cost of maintenance for the organization in charge of a particular highway because users of this infrastructure are the ones who pay for it.

Existing literature related to highway tolls is very extensive. Nevertheless, a great majority of papers focuses basically on congestion reduction and demand regulation, however, without

considering studies on toll setting for maximizing benefits.

It is evident that congestion is an important and increasingly growing problem in any transportation network. On the other hand, the efforts made so far to attack this issue have not been successful mainly because once a strategy for congestion reduction is proposed, it is not guaranteed that the new road capacities would be appropriately used. Efforts to motivate people to stop using their vehicles and start using alternative transportation means have also been ineffective [1]. Most of the papers dedicated to congestion reduction in transportation networks solve the problem by balancing the flow of vehicles in highways as in [2]; the problem with this approach is that there is just one administrator who makes the decision based only on his own criteria, and in most cases this fact affects network users. In [3] a review of the evolution of the models and methods for the problem of estimating traffic flow balance in urban zones is presented, and at the same time this paper shows the scope and limitations of the current traffic models.

Also, much attention has been given to demand regulation. For example, in [4, 5] negative effects entailed by traffic jams like contamination are highlighted. In order to decrease pollution levels, researchers have considered the importance of proposing new regulations which address this fact.

Moreover, highway pricing is a very simple concept that extends a common practice of using prices to reflect shortage of a certain resource

and to assign this resource for its more efficient use. This idea is very well-known and practically applied in any sector of economy. A case study of the lines concession in Singapore in 1975 is the pioneering application of toll setting in transportation networks. But in spite of its success, in [6] it is explained that due to big structural differences, the Singapore example is not seen as good for cities in Western Europe or North America. Nevertheless, the Singapore scheme offers evidence of highway pricing effects. The initial policy was to collect a fee from vehicles entering downtown between 7:30 and 9:30 a.m. This policy demonstrated to be inadequate since the traffic simply moved outside the periods of concession. Very soon this period was extended until 10:15 a.m. including as well a period in the afternoon. The impact of this new scheme was impressive from the beginning because it obtained a reduction of 24700 automobiles during rush hours.

Putting together the ideas for setting tolls in highways and not affecting network users in an indiscriminate way, the toll optimization problem arises. This problem was introduced in [7], where an authority set tolls in some arcs of a congestion-free network aiming to maximize the revenue considering that the users would select their routes based on the minimum cost. In that problem, both the authority and the users make their decisions in a sequential non-cooperative way.

The toll optimization problem is NP-hard [7, 8] which motivates the development of heuristics and metaheuristics algorithms for solving it. A path-based heuristic algorithm is proposed in [9, 10], where the authors developed four algorithms and compared the obtained results in order to determine which algorithm had better performance. These algorithms are based on a penalization method, a gradient approximation method, Quasi-Newton, and a direct resolution method. Referring to the metaheuristics algorithms proposed to solve this problem, we found only two papers: [11], which present a Tabu search algorithm, and [12], where a Genetic algorithm is introduced. These algorithms try to allocate highway tolls on a transportation network

aiming to maximize profit without considering congestion in the network.

By noting the importance of considering congestion in the arcs of a transportation network, we analyze the problem according to this approach. Only four algorithms presented in [10] have been developed to solve arc capacitated toll optimization problem. In this paper, we solve the optimization model introduced in [10] by developing a metaheuristic procedure based on Scatter Search by using the implementation presented in [13] interacting with the commercial optimizer CPLEX, obtaining better results than those presented in [10].

The lower level considered in the problem tackled in [10] will be modeled as a minimum cost flow problem that, as the rest of the models for transportation problems, has been widely studied, for example, in [14, 15].

The rest of the paper is organized as follows. Section 2 shows the structure of a bilevel program and the problem to be solved. In Section 3 the formulation used to solve the problem is stated. Section 4 gives a general description of the optimization tools used to solve the problem. In Section 5 the computational experiments and results are presented. Finally, Section 6 gives conclusions and outlines future work.

2 Bilevel Programming

Bilevel programming problems are hierarchical optimization problems that involve two non-cooperative decision makers. In this kind of problems, a decision maker may be able to influence the behavior of the other one without taking a complete control on his actions, see [16]. The importance of bilevel programming is based on the fact that decision making in any large organization rarely is done from a single point of view.

Multilevel systems share the following characteristics: the existence of interactive decision making between different hierarchical levels, each subordinated level carries out its policies only after a superior level makes its decisions, each level optimizes its own benefit independently, but they are affected by the

actions of the other levels, the external effects on the problem of a decision maker are reflected in the objective function and in a feasible set of solutions.

Multilevel programming was defined for the first time in the 1970s by Bard [17] as a generalization of mathematical programming. In that work he showed how two levels of decision can be used to analyze the dynamics of a regulated economy with a case study of agricultural development of the north of Mexico [18].

If only two levels of decision are considered, this case receives the name of bilevel programming. The foundations are presented in [19] and an overview is given in [20]. In order to be able to mathematically formulate the problem, it is assumed that the leader has control on the vector $x \in X \subseteq R^n$ and the follower has control on the vector $y \in Y \subseteq R^m$. The leader starts selecting a vector x trying to minimize $F(x, y(x))$, which can be subject to certain constraints. The component $y(x)$ indicates that the leader's problem is implicit in y variables. After observing the decision made by the leader, the follower reacts by selecting the y that minimizes his objective function $f(x, y)$ satisfying a set of constraints.

This problem is defined as follows:

$$\begin{aligned} \min \quad & F(x, y) \\ \text{subject to} \quad & G(x, y) \leq 0 \text{ for } x \in X \\ & \min f(x, y) \\ & \text{subject to } g(x, y) \leq 0 \text{ for } y \in Y, \end{aligned}$$

where $F, f: R^n \times R^m \rightarrow R^1$, $G: R^n \times R^m \rightarrow R^p$, and $g: R^n \times R^m \rightarrow R^q$. The sets X and Y may have additional constraints on their variables, such as no negativity or integrality of the variables. This formulation may be handled to generate new forms for the problem, for example, changing the **min** operator by **max** according to the specifications of a particular problem.

The problem of highway toll allocation considering congestion in the arcs may be formulated as a bilevel optimization problem. In this problem the upper level consists of an administrator that establishes tolls for the network aiming to maximize its income, whereas the lower level is represented by a group of users who travel in the shortest paths with respect to a

generalized cost. In order to be able to obtain a bilevel model, a constraint of the type **argmin** has been added to the objective function of the follower that optimizes the flows in the arcs of the transportation network once tolls are allocated. By considering this, we try to find a balance between the established tolls and the number of users who travel on these roads. It is important to mention that the flows are restricted due to the capacity corresponding to an arc; it does not matter if the arc is tolled or toll free, it will have a maximum capacity associated to it.

It is important to remark that multilevel programming, in particular, bilevel optimization is different from bi-objective optimization. Thus, the concepts related to the Pareto front are not valid for the problem studied here. In fact, there are many papers that discuss that optimal bilevel solutions are not in the Pareto front of the bi-objective version of the same problem. On the other hand, bilevel solutions are given in the Inducible Region. Some references supporting the idea described above are [21-25].

3 Problem Formulation

As mentioned above, the highway toll optimization problem can be approached from the point of view of a bilevel programming problem with a leader and a follower which interact in a multicommodity network $G = (K, N, A)$ defined by a set of commodities K (given in the origin-destination form), a set of nodes N and a set of arcs A . The set of arcs is partitioned in subset A_1 which represents the arcs with tolls and its complementary subset A_2 , representing the toll free arcs. Each arc $a \in A$ is provided with a generalized cost c_a which represents the associated cost of traveling through each arc. Therefore, we consider that each arc $a \in A$ in the network has a limit q_a in its capacity simulating congestion on that arc. There is also a parameter n_k that represents the existing demand for each commodity between the origin and destination nodes associated with the commodity $k \in K$. Finally, the decision variable t_a associated to the toll arcs $a \in A_1$ represents an extra cost to be determined to travel through these arcs. The

other set of decision variables are x_a^k which represent the flows between the arcs to fulfill the demands of the existing commodities.

Considering that the tolls t_a cannot exceed a pre-established maximum value of t_a^{max} and that the flows x_a^k must be an optimal solution of the lower level which is parameterized by the toll vector t_a decided at the upper level, this problem can be formulated as a bilevel program with linear constraints as follows:

$$\max_{t,x} \sum_{k \in K} \sum_{a \in A_1} t_a x_a^k \quad (1)$$

$$s. t. \quad t_a \leq t_a^{max} \quad \forall a \in A_1, \quad (2)$$

$$t_a \geq 0 \quad \forall a \in A_1, \quad (3)$$

$$x^k \in \varphi(t) \quad \forall k \in K, \quad (4)$$

$$\sum_{k \in K} x_a^k \leq q_a \quad \forall a \in A, \quad (5)$$

where $\varphi(t)$ defined in (4) represents the solution set given by the following family of equations:

$$\varphi(t) = \operatorname{argmin}_{\bar{x}} \sum_{a \in A_1} (c_a + t_a) \bar{x}_a + \sum_{a \in A_2} c_a \bar{x}_a \quad (4.1)$$

$$s. t. \quad - \sum_{a \in i^-} \bar{x}_a + \sum_{a \in i^+} \bar{x}_a = b_i^k \quad \forall i \in N, \quad (4.2)$$

$$\bar{x}_a \geq 0 \quad \forall a \in A. \quad (4.3)$$

Equation (1) is the leader's objective function which tries to maximize the total profit obtained from the sum of the products of the tolls t_a and the users' flow x_a^k through the arc a for all the commodities. However, the set of lower constraints given by the family of equations (4) enforce the follower to assign flows to the shortest paths with respect to the current tolls, that is, the follower's objective is to minimize the total cost of the paths selected by the users, and it is defined by (4.1). The constraints (4.2) at the lower level are in charge of the flow conservation and the non-negativity in the flows given by (4.3). Finally, the constraint (5) is added to prevent exceeding

the capacities on the arcs with the assigned flows considering all the commodities.

The inequality defined by (5) complicates the problem even more. In the absence of this constraint the lower level problem can be solved by considering all the commodities separately, this is, we can solve exactly $|K|$ minimum cost flow problems independently and then add the travel costs. But now, this constraint correlates all the commodities. This important issue is analyzed in [10].

4 Solution Methodology

Scatter Search is an evolutionary metaheuristic that has been successfully applied to solve hard optimization problems. It is based on formulations and strategies developed in the 60s, but it was not until 1977 when it was officially proposed by Glover as a method in its own right. In [26], Scatter Search is described as "a method that uses a succession of coordinated initializations to generate solutions".

As described in [26, 27] and some other implementations based on this framework, the methodology includes the following basic elements:

- Generation of a population P ,
- Extraction of a reference set R ,
- Combination of elements from R and update of R .

The dimension and structure of the solution set in different evolutionary algorithms may differ. Genetic Algorithms, for example, handle the whole created population (typically 100 solutions), Memetic Algorithms work with a small (and sometimes structured) population, see [28], and Scatter Search works with a subset of 10 to 20 solutions from the set of created solutions. This subset called reference set R is built from the population P (generated by the diversification generation method) with only a few solutions from P . The way the reference set is initialized, updated and rebuilt is a crucial aspect in Scatter Search performance. If the construction of a reference set is made based only on the solution quality, the reference set will be formed by

selecting the best b solutions in P . Nevertheless, a desired characteristic in general search procedures and particularly in Scatter Search is an adequate balance between intensification and diversification.

The next step in the Scatter Search methodology is combination of elements in the reference set. In order to accomplish this, two or more elements from R are chosen in a systematic way with the purpose of creating new solutions. This is achieved through the construction of subsets of solutions from R by applying the combination method to the solutions contained in each one of these subsets. This combination is intended to be intelligent trying to create a better solution than those in the subset. At this point, it is possible that as a result of the combination method an infeasible solution is created. In this case, the combination method must have a procedure to restore feasibility. As new solutions are being created, these will gain membership to the reference set not only by their quality, but by their degree of diversity. The general procedure may iterate several times to achieve a better quality in the created solutions. The way Scatter Search combines solutions and updates the set of reference solutions used for the combination establishes the difference among this methodology and other population-based approaches.

In order to solve the problem, in this paper we analyzed the implementation developed in [13] of the metaheuristic Scatter Search which was taken as a basis to solve the upper level problem where the leader seeks to determine the optimum toll combination to maximize the benefits in a transport network. The optimizer CPLEX was used to solve the problem at the lower level, where the optimal flows in the transport network are determined with the objective of minimizing the costs of traveling through the network.

The considered algorithm started by generating P , an initial set of diverse solutions for the toll variables of the upper level, in a random but controlled form in order to guarantee that the generated initial solutions are within the pre-established ranks (in this case between 0 and t_a^{max}). For generating these solutions, the algorithm starts dividing the ranks of each

variable (between 0 and t_a^{max}) in 4 sub-ranks of same size. The next one of these sub-ranks is randomly selected and a solution within the selected sub-rank is generated.

Once the initial set of solutions has been generated, the method proceeds to improve these solutions. Considering that the solution generation step constructs solutions that are within the allowed ranks only, the improvement method will always start to work with a feasible solution. The improvement procedure consists in the Nelder-Mead Simplex Method proposed in [29], which is a classic optimizer for unconstrained nonlinear problems. This method requires an input parameter that specifies the number of evaluations of the objective function. A main feature of this method is that the bigger the number of evaluations, the better is the quality of solutions.

The next step consists in extracting the best generated solutions in order to form the reference set. The 50% of this set must be formed by solutions selected attending to their quality with respect to the leader's objective function, and the other 50%, by their diversity. For obtaining this, the solutions are sorted in a decreasing order and the first half of the desired amount for the reference set is selected (for example, if it is desired to form a sub-group of reference of size 10, the top 5 solutions of the list are selected) and they are deleted from the initial set P . Next, the minimum Euclidean distance between the remaining solutions in the initial set P and the solutions selected for the reference set is computed. The solution with the maximum of the minimum computed distance is selected, added to the reference set, and deleted from the initial set P . Once this is done, the process is repeated until it completes the desired size of the reference set which, as a result of this construction, will contain the solutions with the greater quality and diversity.

Once this has been done, subsets of the reference set are generated and then the combination method is applied to all these subsets. This combination method consists in creating 3 test solutions for each pair of solutions of the reference set. The first test solution is computed as

$$C1. x = x' - d,$$

Table 1. Objective function values obtained by each method for instances of Network 1

Inst	Gradient approx.	Quasi-Newton	Direct Method	SS-CPLEX
1	162.850	162.880	162.360	180.990
2	274.890	274.890	274.340	274.950
3	109.850	109.850	108.870	199.990
4	150.860	150.840	150.320	146.550
5	112.860	112.860	112.040	133.840
6	203.950	203.960	202.820	199.973
7	41.970	41.960	41.650	41.976
8	104.950	104.950	104.110	125.990

Table 2. Objective function values obtained by each method for instances of Network 2

Inst	Gradient approx.	Quasi-Newton	Direct Method	SS-CPLEX
1	1342.24	1342.76	1341.84	1328.98
2	7184.85	7184.80	7184.42	7112.13
3	1577.95	1577.91	1577.66	1686.03
4	420.70	420.77	420.26	1022.00
5	764.93	764.96	763.89	1514.00
6	2350.86	2350.88	2,350.36	3004.49

Table 3. Objective function values obtained by each method for instances of Network 3

Inst	Gradient approx.	Quasi-Newton	Direct Method	SS-CPLEX
1	1456.80	1456.83	1456.03	1631.64
2	2247.87	2247.85	2246.89	2611.99
3	3891.83	3891.87	3891.45	3263.98
4	5621.80	5621.82	5621.11	11339.61
5	3433.72	3433.79	3432.77	3419.90
6	544.87	544.89	543.91	868.59

the second test solution is computed as

$$C2. x = x' + d,$$

and the third test solution is computed as

$$C3. x = x'' + d,$$

where $d = r \frac{x'' - x'}{2}$ and r is a random number within the interval $(0, 1)$.

After the new solutions have been generated by the combination method, they are processed by the improvement method and the best resulting solution is selected. This solution will replace the worst quality solution in the reference set because it has better quality. This cycle is repeated until the reference set no longer

changes and all the solutions in the reference set have been processed by the combination method.

At this point the diversification method is used to reconstruct half of the reference set and the search continues. This procedure is repeated for a pre-established number of iterations and once this number is reached, the method stops. Some related papers revised by us for this research are [30, 31].

It must be mentioned that each time a new solution is generated for the variables at the upper level (toll values), these variables are introduced as parameters to the lower level and the optimal flows are computed using these toll values and the original costs associated to the arcs of the network (keeping feasibility for the solutions for the bilevel problem). These new flows are then used to measure the quality of the generated solutions when the upper level objective value is computed.

5 Computational Results

In order to evaluate the algorithm proposed for solving the bilevel model considered in this paper, three different multicommodity transportation networks were considered. For the numerical experimentation eight examples for the first network and six examples for the second and third network, respectively, were solved. Finally, with the objective to show that the proposed methodology is able to solve larger instances for the problem without any complications, the last example consisted in solving a variation from the second network where an additional commodity was considered.

The first graph represents a transportation network with 7 nodes and 12 arcs (7 toll arcs and 5 toll free arcs). The second graph is composed by 20 nodes and 35 arcs (15 toll arcs and 20 toll free arcs). The third graph is formed by 25 nodes and 40 arcs (20 toll arcs and 20 toll free arcs). The parameters of the graphs considered in each example are the same as the ones used in [10], listed in the appendices of the referenced paper.

The program that we developed to solve this problem was coded in C language and compiled with Microsoft Visual Studio 2010. All the

Table 4. Percentage of increase for the examples considered for Network 1

Instance	Increase
1	11.12%
2	0.02%
3	82.06%
4	-2.86%
5	18.59%
6	-1.95%
7	0.01%
8	20.05%

Table 5. Percentage of increase for the examples considered for Network 2

Instance	Increase
1	-1.03%
2	-1.01%
3	6.85%
4	142.89%
5	97.92%
6	27.80%

Table 6. Percentage of increase for the examples considered for Network 3

Instance	Increase
1	12.00%
2	16.20%
3	-16.13%
4	101.71%
5	-0.40%
6	59.41%

examples were executed on a personal computer HP Compaq 6000 Pro with a Pentium Dual-Core processor at 3.00 GHz and 2.00 GB RAM under the operating system Windows 7.

As mentioned in the previous section, the methodology designed to solve the upper level problem is the metaheuristic Scatter Search. In [22], a C implementation for this methodology was

developed and taking advantage of this, we considered it for this research. The CPLEX 11.1 software optimization libraries from ILOG were used in order to exactly solve the lower level problem.

In this research, 21 instances were solved in order to validate the analyzed methodology. To simplify the analysis we divided the examples as follows:

- Network 1 conformed by 12 arcs: 8 instances.
- Network 2 conformed by 35 arcs: 6 instances.
- Network 3, conformed by 40 arcs: 6 instances.
- Network 4 is a variation of Network 2 considering four commodities: 1 instance.

The purpose of considering the first three sets of instances (not the last case) is to compare directly the results against the ones obtained in [10] where the performance of four different algorithms was measured and the corresponding results were presented. The four algorithms were the penalization method, the algorithm based in the gradient's approximation, the Quasi-Newton algorithm and the direct algorithm based on the Nelder-Mead's optimization method.

The results obtained in this paper as well as the results obtained by the Scatter Search methodology described in Section 4 are shown in Tables 1, 2, and 3. It is worth mentioning that the numbers marked in bold correspond to the best known value for that specific instance.

It can be seen from the tables presented above that the leader's objective function values were improved in 14 of 20 cases; and in 5 of 6 cases the proposed algorithm did not improve the best known result, but it obtained a close leader's objective function value compared against the results obtained in [10], this is a gap less than 3% for those 5 instances.

In Tables 4 to 6 the percentages of increase obtained by the bilevel method based on Scatter Search and CPLEX (for each of the different examples considered) with respect to the best result obtained in the previous work are shown. A negative value implies that the value we reached does not improve the one presented in [10].

The average obtained for the increase of the leader's objective function value is **28.66%** for the

examples tested in this paper with respect to the best existing results. We emphasize that for instance 4 of Network 2 we got the maximum increase by improving the current best leader's objective function value by 142.89%.

As it was mentioned at the beginning of this section, we also measured the time required to obtain the objective function value for the capacitated toll optimization problem for each of the considered examples. Also, the time consumed by the optimizer for solving the lower level problem at each step in the developed algorithm is presented. These results are shown in Tables 7, 8, and 9. Furthermore, the computational time required by the methods proposed in [10] is also presented in these tables.

A notorious increase in time required for solving the examples by the proposed methodology based on Scatter Search and CPLEX can be expected due to the nature of the metaheuristic for solving bilevel optimization problems. It can be appreciated from the last column of Tables 7, 8, and 9 that obtaining an optimal solution for the follower's problem demands on average the 53.7% of the total time. We computed the average of the differences between the times required for the tested examples by using the methodology proposed here against the results obtained in [10], and our result is **1137.08** seconds (18.95 minutes).

As it was mentioned before, the problem studied in this paper is NP-hard. Hence the behavior of exact methods will be affected as the size of instances increases. Particularly, the time required by the algorithm described in this paper is increased in a polynomial way as the size of an instance increases. On the other hand, the other three algorithms do not show this important characteristic due to a substantial increase of their times.

Finally, for Network 4 (a variation of Network 2 with one extra commodity) the obtained leader's objective function value was 1368.657 requiring a total computational time of 1935.79 seconds and 909.82 seconds in CPLEX, which shows that our considered algorithm is not very sensitive to the number of commodities as the other algorithms seem to be.

Table 7. Required time (in seconds) to solve the instances considered for Network 1

Inst	Gradient approx.	Quasi-Newton	Direct Method	Scatter Search	CPLEX time
1	2.107	2.023	2.950	924.02	498.96
2	2.449	2.433	3.235	910.80	500.94
3	1.573	1.426	2.035	961.92	557.91
4	3.414	3.173	3.415	900.93	531.55
5	2.006	1.940	2.002	876.60	525.96
6	2.178	2.156	3.782	879.62	395.83
7	1.308	1.277	1.259	1057.77	560.62
8	3.065	2.899	1.959	1101.60	638.93

Table 8. Required time (in seconds) to solve the instances considered for Network 2

Inst	Gradient approx.	Quasi-Newton	Direct Method	Scatter Search	CPLEX time
1	601.00	517.00	723.00	1026.11	584.82
2	781.00	748.00	917.00	2904.06	1335.87
3	492.00	437.00	714.00	1726.40	1018.58
4	278.00	220.00	439.00	1315.63	710.44
5	644.00	591.00	835.00	1173.20	574.87
6	831.00	806.00	980.00	1685.45	859.58

Table 9. Required time (in seconds) to solve the instances considered for Network 3

Inst	Gradient approx.	Quasi-Newton	Direct Method	Scatter Search	CPLEX time
1	1004.00	985.00	1054.00	2964.25	1659.98
2	525.00	498.00	893.00	1635.63	801.44
3	439.00	421.00	922.00	2227.21	1224.96
4	1723.00	1755.00	1480.00	1666.80	883.40
5	764.00	785.00	761.00	1939.22	1047.17
6	89.00	84.00	244.00	2428.83	1262.98

6 Conclusions and Future Work

As a result of this research, we can conclude that it is possible to maintain a profitable toll road system against competition from toll free roads, as long as right tools and good planning are used.

As we observed, although for each origin-destination pair there is at least one toll free path, benefits were always obtained; this indicates that users are motivated to use a tolled infrastructure.

It was also observed that the resolution method (based on the metaheuristic Scatter Search and the CPLEX optimizer) proposed in

this paper to solve the problem of determining tolls for the roads gave significant improvements over the methods proposed in previous research which was taken as the basis for our research (an increase of 28.56% among all examples tested). Moreover, it was demonstrated that Scatter Search is a very robust and efficient optimization technique, free from complications for larger-size instances.

An important point to consider is the determination of the upper bound for rates in toll arcs, because if this value is established in a wrong way, there could be a case where the model becomes very restricted and does not get the maximum benefit from the network (in a case when a very low toll is set) or that the network is underutilized forcing customers to go through toll free roads (in a case when a very high toll is set).

On the other hand, if it is true that the benefits (in terms of the value achieved by the objective function) when using this methodology are significantly higher, an increase in the time to obtain the results must be considered (around 19 minutes in average for each example). It is recommended to perform a cost/benefit analysis for each particular situation: in some occasions for solving this problem it might be necessary to sacrifice some benefits in favor of getting results in a less time period or to sacrifice time to obtain greater benefits.

It is important to mention that the problem considered in this research is more complicated to solve than the original toll optimization problem due to the capacity constraint which relates all the commodities (it can be seen as a bilevel problem with multiple followers). This issue motivated us to propose a metaheuristic algorithm that performs analysis of neighborhoods in order to find better solutions for the capacitated problem.

Concerning scalability of the developed method for solving real instances, the information on the web page of Capufe (Mexican agency in charge of highways) indicates that the real Mexican highway system consists of 52 tolled roads. Hence, Scatter Search will have a similar performance than the one showed in Network 3, due to the fact that the size of instances does not increase very much.

For further research, we propose the following improvements and recommendations:

- To use data from real cases to observe the behavior of the model proposed in this paper.
- To use other metaheuristics to compare their performance against Scatter Search.
- To optimally solve the lower level problem with another equivalent technique. This approach has been successfully applied in other bilevel problems, e.g. [32].
- To reduce the time required by this method by trying to streamline the code or parallelize a particular method in Scatter Search.
- To develop new models to solve the problem of assigning toll roads to compare them with the model considered in this research and in previous work.

Acknowledgements

This research has been supported by Tecnológico de Monterrey – Research Group in Industrial Engineering and Numerical Methods 0822B01006, the Mexican National Council for Science and Technology (CONACyT) through grant SEP-CONACyT CB-2011-01-166397 and the Secretary of Public Education (SEP) within the Consolidation of the Academic Groups Program with the project PROMEP/103.5/12/4953 and the UANL within the PAICYT support for the project CE960-11. The authors also gratefully acknowledge the comments and suggestions of the reviewers.

References

1. **Button, K. (2004).** *Road Pricing*. Fairfax: Center for Transportation Policy Operations and Logistics.
2. **Mong-Sim, K. & Hong-Sun, W. (2003).** Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, Vol. 33, No. 5, pp. 560–572.
3. **Patriksson, M. (1994).** *The Traffic Assignment Problem: Models and Methods*. VSP International Science Publishers.

4. **Bergendor, P., Hearn, D., & Ramana, M. (1997).** *Congestion in Toll Pricing of Traffic Networks*. Gainesville, FL, Center for Applied Optimization.
5. **Cropper, M. & Oates, W. (1992).** Environmental economics: a survey. *Journal of Economic Literature*, Vol. 30, No. 2, pp. 675–740.
6. **Button, K. (2010).** *Transport Economics*. MPG Books Group, UK.
7. **Labbé, M., Marcotte, P., & Savard, G. (1998).** A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, Vol. 44, pp. 1608–1622.
8. **Roch, S., Savard, G., & Marcotte, P. (2005).** An approximation algorithm for Stackelberg network pricing. *Networks*, Vol. 46, No. 1, pp. 57–67.
9. **Didi-Biha, M., Marcotte, P., & Savard, G. (2006).** Path-based formulations of a bilevel toll setting problem. *Optimization with Multivalued Mappings Theory: Theory, Applications and Algorithms*, S. Dempe, V. Kalashnikov (eds.), Springer, Vol. 2, pp. 29–50.
10. **Kalashnikov, V., Camacho, F., Askin R., & Kalashnikova, N. (2010).** Comparing various algorithms performance: application to bilevel toll setting problem. *International Journal of Innovating Computing, Information and Control (IJICIC)*, Vol. 6, No. 8, pp. 3529–3549.
11. **Brotcorne, L., Cirinei, F., Marcotte, P., & Savard, G. (2012).** A tabu search algorithm for the network pricing problem. *Computers & Operations Research*, Vol. 39, pp. 2603–2611.
12. **Dimitriou, L., Tsekeris, T., & Stathopoulos, A. (2008).** Genetic computation of road network design and pricing Stackelberg games with multi-class users. *Applications of Evolutionary Computing*, Springer, M. Giacobini et al. (Eds.), Vol. 4974, pp. 669–678.
13. **Laguna, M. & Martí, R. (2003).** *Scatter Search: Methodology and Implementations*. Kluwer Academic Publishers.
14. **Zadeh, N. (1973).** A bad network problem for the simplex method and other minimum cost flow algorithms. *Mathematical Programming* 5.
15. **Edmonds, K. (1972).** Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, Vol 19, No. 2, pp. 248–264.
16. **Anandalingam, G. & Friesz, T. (1992).** Hierarchical optimization. *Annals of Operations Research*, Vol. 34, No. 1, pp. 1–11.
17. **Bard, J.F. (1998).** *Practical Bilevel Optimization. Algorithms and Applications*. Kluwer Academic Publishers.
18. **Chandler, W. & Norton, R. (1977).** Multi-Level Programming and development policy. *Working Paper No. 258*. World Bank.
19. **Dempe, S. (2003).** *Bilevel Programming: A survey*. Preprint Technical University Bergakademie, Freiberg, Germany.
20. **Colson, B., Marcotte, P., & Savard, G. (2007).** An overview of bilevel optimization. *Annals of Operations Research*, Vol. 153, pp. 235–256.
21. **Calvete, H.I. & Galé, C. (2010).** A Multiobjective Bilevel Program for Production-Distribution Planning in a Supply Chain. *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Lecture Notes in Economics and Mathematical Systems*, Vol. 634. Springer-Verlag, pp. 155–165.
22. **Candler, W. (1988).** A linear bilevel programming algorithm: A comment. *Computers & Operations Research*, Vol.15, pp. 297–298.
23. **Clarke, P. & Westerberg, A. (1988).** A note on the optimality conditions for the bilevel programming problem. *Naval Research Logistics Quarterly*, Vol.35, pp. 413–418.
24. **Haurie, A. Savard, G., & White, D. (1990).** A note on: an efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, Vol.38, pp. 553–555.
25. **Marcotte, P. & Savard, G. (1991).** A note on the Pareto optimality of solutions to the linear bilevel programming problem. *Computers & Operations Research*, Vol.18, pp. 355–359.
26. **Glover, F. (1998).** A template for scatter search and path relinking. *Artificial Evolution: Third European Conference, Lecture Notes in Computer Science*, Vol. 1363, pp. 13–54. Springer, Heidelberg, Germany.
27. **Martí, R. & Laguna, M. (2003).** Scatter Search: Basic Design and Advanced Strategies. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, Vol. 19, pp. 123–130.
28. **Moscato, P. (2000).** Memetic Algorithms. *Handbook of Applied Optimization*, P. M. Pardalos and M. G. Resende (Eds.), Oxford University Press, USA.
29. **Nelder, J.A. & Mead, R. (1965).** A simplex algorithm for function minimization. *The Computer Journal*, Vol. 7, No. 4, pp. 308–313.
30. **González-Velarde, J.L. & Martí, R. (2008).** Adaptive memory programming for the robust capacitated international sourcing problem. *Computers & Operations Research*, Vol. 35, pp. 797–806.

- 31. González-Velarde, J.L., Alvarez, A.M., & De Alba, K. (2005).** Grasp Embedded Scatter Search for the Multicommodity Capacitated Network Design Problem. *Journal of Heuristics*, Vol. 11, No. 3, pp. 233–257.
- 32. Camacho-Vallejo, J.F., Cordero, A.E., & González-Ramírez, R.G. (2014).** Solving the Bilevel Facility Location Problem under Preferences by a Stackelberg Evolutionary algorithm. *Mathematical Problems in Engineering*, Vol. 2014, 14 p.

José Luis González Velarde is a Professor and Researcher at the Monterrey Institute of Technology (Tec de Monterrey), Monterrey Campus. He received his Ph.D. from the University of Texas, Austin, in 1990. Also, he obtained a Master degree in Industrial Engineering and Operations Research from the University of California, Berkeley, in 1978, a Master degree in Mathematics from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico, in 1973. His Bachelor degree is in Mathematics from Tec de Monterrey in 1971. He has several papers published in the most important journals of industrial engineering such as *European Journal of Operations Research*, *Journal of Heuristics*, *IIE Transactions*, *Annals of OR*, *Computers & OR*, among others. He is recognized by the National System of Researchers (SNI), Mexico, with Level II. His principal research areas include computational optimization and algorithm design for logistics and manufacturing.

José Fernando Camacho Vallejo is a full time Professor and Researcher at the Department of Physics and Mathematics of the Autonomous University of Nuevo León (UANL). He received a Ph.D. in Industrial Engineering from the Monterrey Institute of Technology (Tec de Monterrey) in 2009, his M.Sc. in Engineering with major in Industrial Engineering from Arizona State University in 2007 and a Bachelor degree in Mathematics from UANL in 2005. He is a member of the National System of Researchers (SNI), Mexico, with Level C. He is responsible for the academic group for Bilevel Programming and Applied Statistics, he is also a coordinator of the Master and Doctoral programs in Mathematics at UANL. His main areas of interest are solution of operational research problems, in particular, the ones that can be modeled as bilevel programming problems, and the design of numerical methods for solving them.

Gabriel Pinto Serrano studied Industrial Engineering at the Universidad del Valle, Mexico, Hermosillo Campus, from which he obtained his B.Sc. in 2007. Then he obtained a Master degree in Science with specialty in Manufacturing Systems from the Monterrey Institute of Technology (Tec de Monterrey), Monterrey Campus, in 2010. His areas of interest are mainly in the automotive manufacturing industry and production planning.

*Article received on 27/11/2013, accepted on 08/11/2014.
Corresponding author is José Fernando Camacho Vallejo.*