Planning Brazilian Urban Traffic with a Geographic Application Software

JULIANO LOPES DE OLIVEIRA[1]
ANDRÉ CONSTANTINO DA SILVA[1]
BRYON RICHARD HALL[2]
[1]Instituto de Informática, Universidade Federal de Goiás, CP 131, Campus II, CEP 74001-970, Goiânia-GO
{juliano, andrecons}@inf.ufg.br
[2]Instituto de Matemática e Estatística, Universidade Federal de Goiás, Campus II, CEP 74001-970, Goiânia-GO
bryon@mat.ufg.br

Abstract. Planning and optimizing urban traffic is a difficult problem, with considerable economic and social impacts. The development of suitable software tools to aid municipal governments to control and to improve the traffic flow is urgent due to the increasing number of vehicles in urban areas. However, the development of this kind of software represents a great challenge since the problems to be dealt with in this domain are both computationally and mathematically complex. This article presents a geographic application software as a feasible solution for this problem. The solution is geared towards Brazilian medium to large cities and is based on a mathematical model of the urban traffic. This model is stored in a geographic database and allows the user of the software (typically a traffic engineer) to search for alternative solutions for traffic problems. The software can provide optimal solutions for an important family of traffic problems, namely the Traffic Equilibrium Problem (TEP). This family involves the common hashing problems (e.g. traffic bottlenecks) as well as more sophisticated problems like vehicles' pollution minimization. We have developed a prototype of the software and its solutions for the TEP were empirically validated against real traffic data from Goiânia, capital of the Brazilian state of Goiás.

1 Introduction

Geographic information systems (GIS) deal with geo-referenced information to describe entities and phenomena that are associated to the Earth. GIS allow the development of user application software to improve access to the knowledge embedded in geographic databases. Therefore, a *Geographic Application* is a software built on top of GIS to manipulate geo-referenced data according to a user's specific requirements.

For example, *Environmental Geographic Applications* include software that allows users to understand, control and rationally use natural resources (such as mineral resources). These applications also deal with prevision and prevention of immediate environmental problems (such as storms and floods).

Another category of geographic applications, called *Urban Applications*, has a basic purpose of organizing, administrating and distributing adequately the infra-structure and public service in urban zones. This text presents a geographic application software in this last category.

Specifically, the problem we deal with is planning of, control of, and optimization of urban traffic in (medium to large size) Brazilian cities. In this context we examine the Traffic Equilibrium Problem (TEP) with its variations [1,6]. The solution of this problem involves, besides other aspects, the mathematical modeling of traffic flow in the urban zone [8,9] and optimization of certain variables with respect to this model [10,14].

The tremendous complexity of the mathematical modeling of traffic flow in a large Brazilian city turns impossible any modeling without computer support. However, in spite of the economic, social and technological relevance of the TEP, there are few references in the literature that treat the subject in the perspective of Computation. Moreover, to the best of our knowledge, there is no available software in the market which allow city administrators and traffic engineers to successfully model Brazilian urban structure and thus solve their problems. Existing software, such as Emme/2 [7], are biased to the traffic conditions of North America, which are very different from Brazilian traffic conditions.

This text presents a proposal which deals with the TEP from a computational point of view, through a geographic application software. The basic idea of this software is to define an adequate database for the complexity of mathematical modeling the Brazilian urban structure of streets and intersections, with traffic lights, restrictions on permission to maneuver and interaction between different flows of traffic.

In the prototype of the software we have developed, the concept of network is the basis of the mathematical model of the TEP. Streets, avenues and highways are represented by a digraph with each arc associated to a function $\mathbf{t_a}$ which expresses the necessary time to travel between its endpoints in function of various factors such as distance, intensity of traffic flow, number of lanes, and duration of traffic lights. In the proposed graph, the nodes between which each arc is placed will normally be intersections with other flows of traffic. This graph model was mapped into a geographic database, and by associating geographic coordinates to the nodes we have allowed a graphical visualization and an easier comprehension of the solutions obtained by applying the mathematical model.

The remainder of this paper presents a general vision of the geographic application software that was developed, showing how this software could aid in planning and control of urban traffic, with the varied uses of the TEP. Section 2 formally defines the TEP. Section 3 presents a computational view of the TEP, seeking a correspondence between the mathematical model and the database information with respect to the implementation of the software for its solution. Section 4 describes the software prototype that was developed and Section 5 deals with a few conclusions.

2 The Traffic Equilibrium Problem (TEP)

The Traffic Equilibrium Problem (TEP) is defined as the problem of predicting the urban traffic flow that will result from numerous vehicles of various types trying each one to minimize its time of travel from an origin to a given destination [13].

The solution of the TEP involves simulation of the traffic flow in the city, being either the present traffic flow or that which would result from some proposed urban reform. In this way the solution of the TEP aids in planning and control of urban traffic.

The flow of vehicles modeled in the TEP needs to satisfy two rules so as to correctly model the real conditions of traffic:

1. The flow is a non-negative real number;
2. The flow should obey Wardrop's condition [4]: "Every motorist travels by a route from which no one may unilaterally alter, reducing his/her time of journey."

Wardrop's condition implies that each driver intends, individually, to minimize his/her travel time. Thus we suppose that the majority of the motorists are familiar with the existing options as to travelling from a certain origin to some destination, and that they will use that knowledge to minimize their travel time.

In this sense, the TEP is a problem of simultaneous minimization of time or cost of travel by thousands of drivers, with interaction among these people [6]. It is not, therefore, a simple minimization problem. It is classified as a variational inequality on a large scale [14], and does not satisfy the mathematical conditions which guarantee a unique solution.

The mathematical solution of the TEP consists of utilizing a digraph with the associated functions $\mathbf{t_a}$ mentioned in Section 1 to simulate the existing (or predicted) conditions of urban structure. So we consider a digraph $\mathbf{D} = (\mathbf{V},\mathbf{A})$ with **m** nodes and **n** arcs. The streets, avenues, etc. are arcs and their connections and intersections are nodes. We associate to this graph a vector **d** of demands, representing the vehicles which intend to travel from origin to destination (O-D) in the prefixed time interval. To each arc will be associated a flow **x**, that is, the number of vehicles to transit on that arc in the time interval. The time to travel over the arc **a** will be governed by a function $\mathbf{t_a}$.

The solution to the TEP consists of information as to how the traffic will flow as the community of drivers seek to individually minimize their travel times (total times). The calculated flows will converge to certain limit values after a number of iterations, and the values found will serve to help take decisions as to how to best organize the conditions of traffic flow, including decisions as to whether certain streets should be one way or two way, whether or not traffic lights should be installed, whether certain maneuvers should be permitted or not, among others [15].

2.1 Mathematical Definition of the TEP

Consider then a digraph $\mathbf{D} = (\mathbf{V},\mathbf{A})$ with **m** nodes and **n** arcs. As said above, the arcs represent blocks of streets, avenues, etc. and the nodes their intersections.

Associated to the digraph is a vector **d** of demands, $\mathbf{d} = \{(r_i, s_i), 1 \le i \le k\}$, representing the number of vehicles to drive from a certain point of origin to a certain destination (in **k** pairs) in the time interval for study.

To each arc **a** will be associated a traffic flow $\mathbf{x_a}$. Finally, the time to travel over each arc will be determined by the function $\mathbf{t_a} = \mathbf{t(x, d, \Gamma)}$, where Γ corresponds to various factors such as length of arc, number of lanes, and characteristics of traffic lights (if they exist), and **x** is the traffic flow as a vector in R^m_+ and $\mathbf{d} \in R^k_+$. We collect the functions $\mathbf{t_a:R^n \Rightarrow R}$ in one non-linear application alone, $\mathbf{t:R^n \Rightarrow R^n}$.

Let **K** be the number of routes $\mathbf{C_j}$ from all origins to all corresponding destinations on the digraph **D**. Δ is a $n \times K$ binary matrix of arc-routes such that:

$$\Delta_{ij} = 1 \quad \text{if the arc } a_i \in \text{(route) } \mathbf{C_j}$$
$$\Delta_{ij} = 0 \quad \text{otherwise}$$

We group together the routes which refer to a certain origin-destination pair (r_i, s_i) as part of the matrix, $\Delta^{r_i s_i}$ so that $\Delta = [\, \Delta^{r_1 s_1} \mid \ldots \mid \Delta^{r_k s_k} \,]$.

We associate to each route $\mathbf{C_j}$ a traffic flow f_j and write $\mathbf{f} = (f_1, \ldots f_k)$. Define the $k \times K$ matrix Λ so that:

$$\Lambda_{ij} = 1 \quad \text{if } \mathbf{C_j} \text{ is a route of the } i^{th} \text{ demand O-D;}$$
$$\Lambda_{ij} = 0 \quad \text{otherwise.}$$

The TEP has four conditions imposed:

(1) $\mathbf{x = \Delta.f}$
(2) $\mathbf{d = \Lambda.f}$
(3) $f_i \ge 0, \quad 1 \le i \le K$
(4) $x_j \ge 0, \quad 1 \le j \le n$

being **d** initially a fixed vector of demand. Given **d** and Λ, a vector of flux on routes **f** is viable if the conditions (2) and (3) hold. The matrix Δ determines **x**. In these conditions, and satisfying (4), **x** is said to be a viable flux, which means that it attends demand and is non-negative. The set of all viable fluxes is compact and convex in R^n_+ and is thus a limited set κ. (1) and (2) imply that this set is a convex polytope, the equivalent to a convex polyhedron of dimension (large) r. Notice that if x_a and x_b are both viable, then all fluxes λx_a + (1- λ)x_b are viable, for all $\lambda \in [0, 1]$, so that the set is in fact identical in behavior to a polyhedron.

We call $T^{rs} = (T^{rs_1}, \ldots T^{rs_h})^T$ the cost vector for the O-D pair (r, s). Therefore:

(5a) $T^{rs} = [\Delta^{rs}]^T.\mathbf{t}$

(5b) $T = [\Delta]^T.\mathbf{t}$

Considering the components T^{rs}, determined for certain viable f^{rs}, Wardrop's condition will be satisfied for **f** if $T^{rs_i} > T^{rs_j}$ implies that $f^{rs}_i = 0$ for all i, j, (r,s). Given a viable flux **f**, let u_{rs} be the minimum time on the corresponding O-D pair (r,s). **f** satisfies Wardrop's condition if

(6a) $T^{rs_i} - u_{rs} \geq 0$ for all i, (r,s)

(6b) $f^{rs}_i.(T^{rs_i} - u_{rs}) = 0$ for all i, (r,s)

Each vector **f** determines uniquely a cost vector **T(f)**. Let **F** be a viable vector which satisfies Wardrop's condition, i. e., that is a solution of the TEP. The total cost will be **T(F).F**. If we fix the vector **T(F)** and consider the conditions (6a) and (6b) satisfied, the total cost **T(F).F** will be less than that of any other product **T(F).f**, where **f** is a viable flow on routes. Thus:

(7) $\mathbf{T(F).f} \geq \mathbf{T(F).F}$ for all viable **f,** or

(8) $\mathbf{T(F).(f - F)} \geq \mathbf{0}$ for all viable **f**

If we apply the relations of (1) and (5b), the inequality (8) can be expressed as

(9) $\mathbf{t(x^*).\ (x - x^*)} \geq \mathbf{0}$ for all **x** in κ.

With this, we have therefore shown that the TEP may be expressed as the problem of finding a flux **x*** in κ such that

$$\mathbf{t(x^*).(x - x^*)} \geq \mathbf{0} \quad \text{for all } \mathbf{x} \text{ in } \kappa$$

s.t. $\mathbf{x = \Delta.f}$

$\mathbf{d = \Lambda.f}$

(10) $x_j \geq 0$ for $1 \leq i \leq n$

The solution for this problem is based on Sheffi's algorithm [13], which proposes one iteration of the Frank-Wolf algorithm [4] in each of its iterations (approximate solutions) with subsequent correction of values of **x**. We have κ as the set of viable fluxes; *y* as a viable flux which minimizes the traffic times with time per arc fixed; and α as the parameter in which there is minimization.

Step 1 i:= 0. Find $\mathbf{x_i} \in \kappa$

Step 2 Calculate $\mathbf{t_i = t(x_i)}$

Step 3 Attribute these times and find minimum time routes for each demand (O-D)

Step 4 Solve for α_i in

$$\min \quad z(\alpha) = \sum_{a=1}^{n} \int^{x_a^i + \alpha(y_a^i - x_a^i)} t_a(x_1^i, \ldots, x_{a-1}^i, \omega, x_{a+1}^i, \ldots, x_n^i) d\omega$$

obtaining $\alpha_\mathbf{i}$ with $0 \leq \alpha_\mathbf{i} \leq 1$

Step 5 Put $\mathbf{x_{i+1} := x_i + \alpha_i(y_i - x_i)}$

Step 6 Test for convergence. If not satisfied, let i:= i+1 and go to Step 2.

Step 3 uses Dijkstra's algorithm for finding minimal path in valued digraphs, with modifications for penalizing certain maneuvers or prohibiting other maneuvers, typical of traffic flow in Brazil. Step 4 uses as method the Golden Section Search. As the function to be minimized changes in each step, Step 6 is in fact mathematically badly defined and in practice various methods of measuring convergence are used.

For engineers who plan traffic, the complexity of the TEP is enormous. The conditions of traffic flow translate into non-convex functions, which imply that there is no formal mathematical guarantee of convergence or even existence of a solution. To bypass these difficulties we developed a geographic application software which encapsulates the mathematical complexity of the problem. The corner stone of this software is a geographic database model, described in the next section.

3 A Geographic Database Model for the TEP

The mathematical model described in the previous section formally solves the TEP. The solutions given by the model were compared to samples of real traffic flows in Goiânia, capital of the state of Goiás, which confirmed the correctness of the solutions. However, a software solution for this problem has to deal with three additional questions:

- the implementation of the mathematical model;
- the visualization and manipulation of the geographic data associated to the urban network and to the flow of vehicles in this network;
- the efficient storage and retrieval of these data.

Although these questions may be considered independently, this is not advisable since there is a strong correlation among their possible solutions. Our approach was to deal with the three issues in a common framework: a geographic database containing all the mandatory information to solve the TEP. The schema of this database defines the relationships between the mathematical algorithms that solve the TEP and those algorithms that aim to offer a suitable visualization of the data and the correspondent solutions.

Our software was designed to be a geo-processing software tool to solve variants of TEP, taking into account the requirements of users who work in the management and planning of vehicle traffic in Brazilian urban areas. The goal of the software is to aid in the detection and prevention of problems in the traffic system through the comparison of different TEP scenarios in a given city region.

3.1 Modeling Traffic Flow in Brazilian Networks

The mathematical model of TEP, defined in Section 2, associates to each street segment a traffic function. However, empirical tests showed that idiosyncrasies of each population have influence on the definition of this function. The traditional modeling approach is based on generic hypothesis, such as that the traffic is defined in terms of preferential ways and semaphores. This hypothesis may hold for many countries, but in Brazil the reality of most cities is more complex. Indeed, in Brazilian cities there are two main groups of traffic controls, which may be used to define the traffic functions in our mathematical model for TEP:

- Group 1: controls based on the direction of the traffic flow, that can be one-way or two-way. In the latter case, another factor is related to the separation of the two directions of the traffic flow. Therefore, there are three subgroups:
  - one-way;
  - disjoint two-way, where traffic lanes with opposite directions are mutually isolated;
  - overlapping two-way, where there is no physical separation between opposite lane directions.
- Group 2: defines the kind of influence that one street has on other streets in an intersection, which may be:
  - preferential;
  - based on semaphore;
  - not preferential;
  - based on rotator (e.g., a square).

The function associated to each road segment is influenced by both groups of controls. The combination of these groups give rise to twelve types of streets, although some types may share the same traffic function. Table 1 shows the equations for computing the time to travel in each type of street. The equations are based on five parameters:

- length, the extension of the street segment;
- volau: the traffic flow in the street;
- ul2: the standard maximum velocity in the street;
- ul3: the traffic flow in the opposite direction in overlapping two-way streets;
- lanes: the number of lanes of the street.

Type 1 – Disjoint preferential two-way and **Type 2 –** One-way preferential
$fd1=\frac{6*length}{ul2}+\frac{6*length}{ul2}.\left(\frac{50+2.5*ul2}{7500*faixas*ul2}\right)^{3,8}*volau^{3,8}$
Type 3 – Overlapping two-way preferential
$fd3=\frac{6*length}{ul2}+\frac{6*length}{ul2}.(\left(\frac{50+2.5*ul2}{7500*faixas*ul2}\right)^{3,8}*volau^{3,8}+\frac{1,5x10^{-8}*ul3*volau}{faixas-0,5})$
Type 4 – Disjoint two-way with semaphore and **Type 5 –** One-way with semaphore
$fd4=\left(\frac{6*length}{ul2}+\frac{ul1^2}{2}\right)\left(1+\left(\frac{50+2.5*ul2}{7500*faixas*(1-ul1)*ul2}\right)^{3,8}*volau^{3,8}\right)$ where *ul1* is the time interval when the semaphore is closed.
Type 6 – Overlapping two-way with semaphore
$fd6=\left(\frac{6*length}{ul2}+\frac{ul1^2}{2}\right).(1+\left(\frac{50+2.5*ul2}{7500*faixas*(1-ul1)*ul2}\right)^{3,8}*volau^{3,8}+\frac{1,5x10^{-8}*ul3*volau}{faixas-0,5})$ where *ul1* is the time interval when the semaphore is closed.
Type 7 – Disjoint two-way without preference, **Type 8 –** One-way without preference, **Type 10 –** Disjoint two-way based on rotator, and **Type 11 –** One-way based on rotator.
$fd7=\frac{6*length}{ul2}+\frac{6*length}{ul2}.\left(\frac{50+2.5*ul2}{7500*faixas*ul2}\right)^{3,8}*volau^{3,8}+2x10^{-9}*ul1^{2.6}$ where *ul1* is the traffic flow on the preferential street.
Type 9: Overlapping two-way without preference and **Type 12 –** Overlapping two-way based on rotator
$fd9=\frac{6*length}{ul2}+\frac{6*length}{ul2}.(\left(\frac{50+2.5*ul2}{7500*faixas*ul2}\right)^{3,8}*volau^{3,8}+\frac{1,5x10^{-8}*ul3*volau}{faixas-0,5})+2x10^{-9}*ul1^{2.6}$ where *ul1* is the traffic flow on the preferential street.

Table 1 Functions to estimate the flow in the different types of Brazilian streets.

The parameters of these equations were empirically adjusted according to real traffic data collected in Goiânia-GO. This is a representative city with more than one million inhabitants and having the second highest relation of vehicle per habitant among the Brazilian states capitals.

3.2 The Geographic Database Schema

The schema of the geographic database designed for the geographic application software that implements the TEP solution results from the mapping of the mathematical model described in Section 2, taking into account the idiosyncrasies of the Brazilian cities discussed in the previous section. The schema has four main sources of information:

i) Street data: the database stores not only the name of the street, but also the properties of each segment of the street, such as semaphores, number of lanes, origin, destination, type (according to the taxonomy described in Section 3.1, which is essential to calculate the travel time for each demand), length, allowed speed, and the set of streets that influence the given street;

ii) Demand data: all the possible demands of flow are modeled and stored in a data structure composed of origins, destinations, and the number of vehicles that are willing to travel from a given origin to each destination;

iii) Parameters data: the values which adjust the model and are used in the optimization functions, such as the number of times that the minimization function will be applied, the exponent to the road segments' functions, and the degree of convergence to the stopping criteria;

iv) Penalization data: the rules that are applied to each scenario of traffic modeling. The database stores, for each travelling vehicle, the previous node, the current node and the next node that it will access in its way from its origin to its destination. Through these parameters the software may not only define constraints, but also penalizations for the travelling vehicles, depending on the path they are following.

All these data are necessary and sufficient to solve the TEP in the context of Brazilian cities. Besides that, the geo-referenced perspective of the traffic network allows the construction of sophisticated graphic interfaces to help the user understand the results produced by the software as solutions for the TEP.

Figure 1 presents a simplified version of the schema of the geographic database, using an Entity-Relationship notation. This schema stores traffic data for a single city, which is divided into Districts. A Street is a weak entity since in Brazilian cities distinct districts may have streets with the same name. Moreover, a street may have alternative names (often there is an official name and a popular name for a given street).

Streets are divided into Segments, which are the basic building block of our geographic database. A street segment has a Node as origin and destination, according to the directed graph of out mathematical model. The schema defines the possible paths via the "allow access to" relationship, which also defines penalizations for moving from one segment to another.

The twelve types of street segments are represented in the schema as an specialization hierarchy (partially shown in Figure 1, due to space limitations), and the mutual influence among segments are defined through the "influence" recursive relationship.

The demand matrix of the mathematical model is represented in the database schema by the recursive relationship "have demand to" of the Node entity. An instance of this relationship forms a pair of nodes (origin, destination). The "minimal path" relationship shows the street segments that belong to the minimal path for this pair (that is, the minimal path to go from the given origin to the given destination).

Parameters data are stored in the Information entity. The identifier for this entity is the number of the current iteration of the TEP resolution algorithm.

4 The Implementation of the TEP Software

We developed a prototype to verify and to validate the viability of the computational solution proposed for the TEP. In this section we describe the architecture behind the prototype and show how this architecture was used on the prototype's implementation.

4.1 The Geographic Software Architecture

The four major components of the software architecture are shown in Figure 2, where an arrow indicates a dependency of the pointed component .

Our Geographic Application Software is split into two components: the User Interface and the TEP Resolution. Moreover, in our architecture the GIS is logically separated from the Geographic DBMS. Both partitions follow the same principle: modularity and independence improve maintainability.

Our architecture is based on the approach defined in [11] for decoupling the user interface from the application in geographic software. The main advantage of this architecture is that it allows the independent definition and evolution of the user interface and the main application components. This is a very important factor for the maintainability of geographic applications since both components are typically complex.

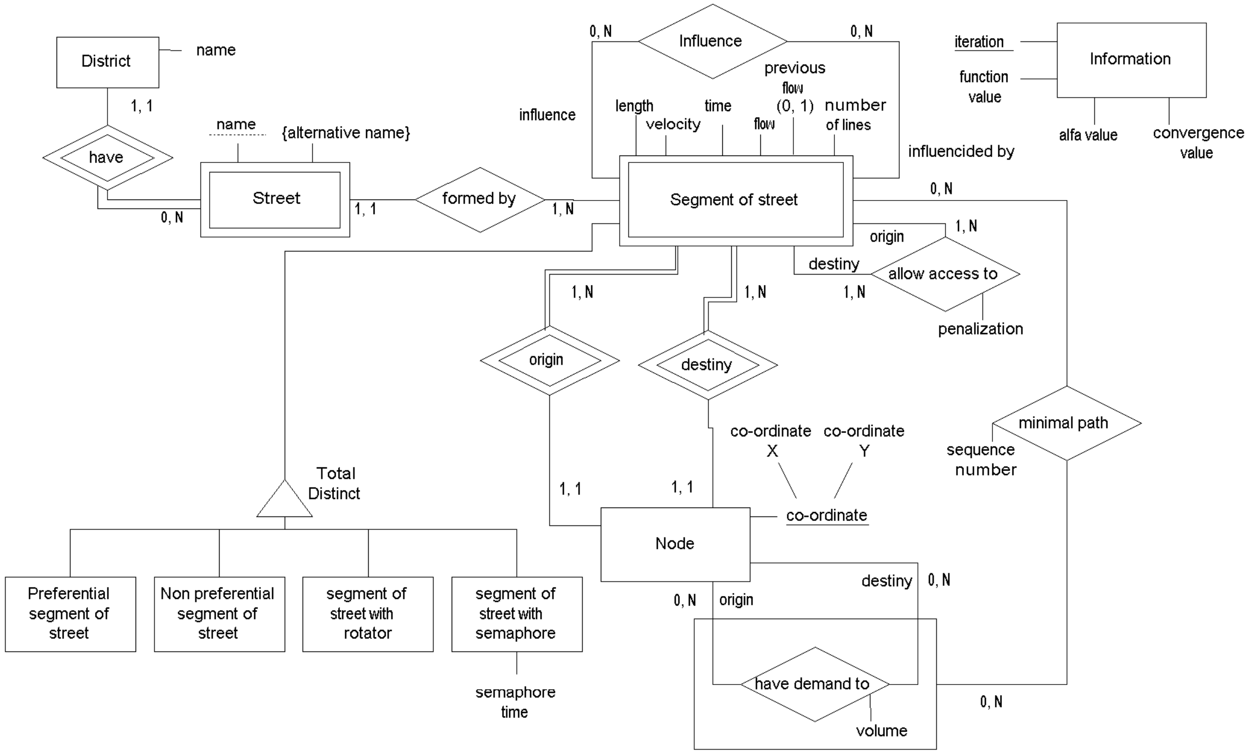

Figure 1 Geographic Database Schema for Planning Urban Traffic Flow

By default, our architecture demands only conventional (i.e., relational) capabilities from the DBMS, which should act as a conventional database server for the other components. However, many existing GIS provide their own proprietary DBMS, and in this case the two components may be collapsed in the architecture.

The TEP Resolution is completely independent from the GIS, since it only deals with the mathematical model presented in Section 2. Thus, it relies only on the relational capability of the Geographic DBMS. The TEP Resolution retrieves the (conventional) data from the DBMS, solves the TEP, and stores the solution data back on the DBMS, using the database schema described in Section 3.

According to the GIS architecture, it may use the GDBMS as a server for its own geographic data, or as an external storage (in this case, the GIS have to import the solution data from the DBMS in order to make it available for the TEP User Interface). The GIS allows the user to start the TEP Resolution from the TEP User Interface Component. The latter relies on the sophisticated graphic manipulation capabilities of the GIS for the visualization and interaction with the TEP solution data. There is, therefore, a strong dependency between the TEP User Interface and the underlying GIS.

It is worth noting that the independence between the components does not compromise the logical unity of the geographic application system because all the components share a common database schema.

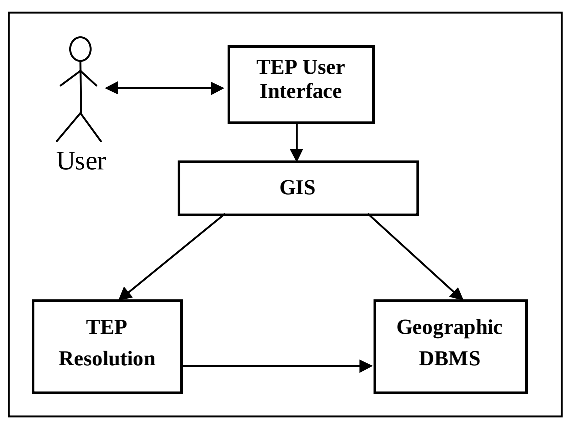

Figure 2 Architecture for the TEP Software

4.2 Using the Geographic Software Prototype

In our prototype, the user interacts only with the TEP User Interface, as it is prescribed in the geographic software architecture. This user interface component was implemented with the Avenue language [2] on top of the ArcView GIS. In our prototype we used the PostgreSQL DBMS [12] to store both the conventional and the geographic data since, in the prototype, the geographic data were limited to geographic coordinates for nodes and simple line segments for streets. The ArcView GIS facilities were used only to the interactive manipulation of maps (zooming and panning, for instance).

As the user updates the geographic data (traffic network or demand vectors, for instance), the TEP User Interface asks the GIS to forward the modifications to the underlying DBMS, via PostgreSQL's ODBC API.

After inserting the geographic model's data, the user may start the TEP Resolution component, which in our prototype implements the TEP solution using the C programming language. The ArcView GIS allows the execution of external applications, defining an API to exchange data with these external applications. The TEP Resolution component accesses the geographic database in the PostgreSQL DBMS using embedded SQL statements and the ODBC protocol.

As we mentioned before, the main advantage of this architecture is the independence of the components. For example, if one wishes to replace the DBMS in our prototype, the only requirement is that the new DBMS must provide a C compiler and must support ODBC.

Figure 3 presents the main window of the TEP User Interface prototype, showing real data from a district of Goiânia-GO. This window allows the user to analyze the results generated by the TEP Resolution component. The window presents two distinct views of the solution for the TEP on the given traffic network. The top view shows the flow of vehicles in each segment for a given solution of the TEP (calculated by the TEP Resolution component). The bottom view shows the time needed to traverse each segment in this solution. These views are essential for the planning activities, and are used simultaneously by the experts in traffic planning. Since it is an ArcView application, the prototype automatically inherits the GIS abilities and tools to manipulate graphics and maps.

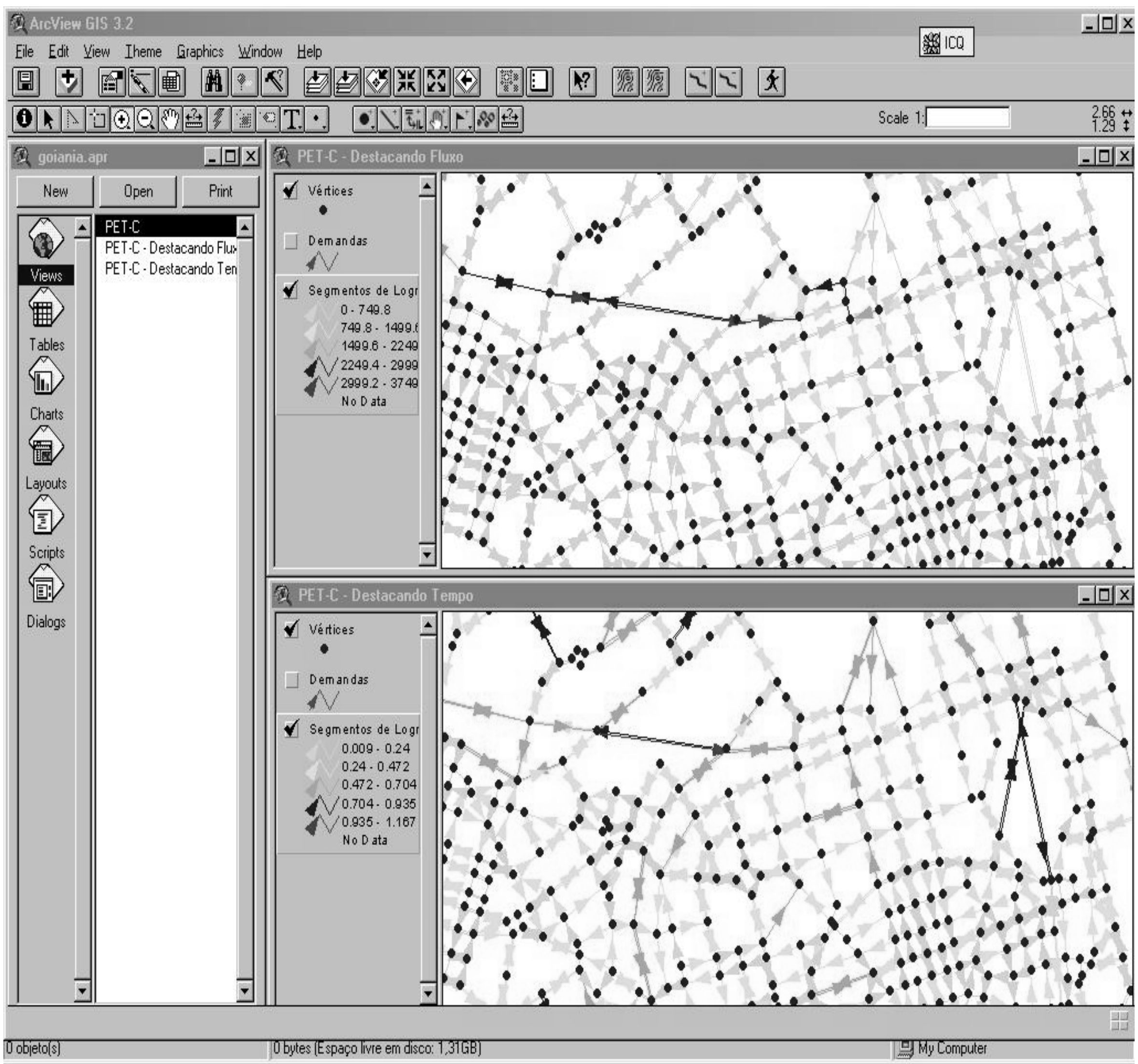

Figure 3 The TEP User Interface with two views of the solution for a TEP scenario.

5 Conclusions

The Traffic Equilibrium Problem (TEP), defined over an urban street network, aims at the simulation of the vehicles' flow, given certain demand conditions and constraints. The solution for the TEP deals with a complex mathematical model and is of great importance for medium and large cities. These cities have to control their traffic flow continuously to avoid bottlenecks and to improve the traffic conditions.

This paper presented a formal definition for the TEP and a corresponding mathematical solution. This solution is geared towards the idiosyncrasies of Brazilian cities and is in the kernel of a geographic application software that makes it easy for the users to solve different instances of the TEP.

A prototype of the designed software was implemented [3] using a geographic database schema as a framework. We believe that this work is of great interest for the Brazilian cities administrators, since our solution takes it account the idiosyncrasies of the traffic in these cities.

Possible extensions to the work described here includes the improvement of the definition of the demands data. Instead of being inserted by the user, the demands matrixes could be derived from other data, such as the number of vehicles in a given region, and the points of interest along the city.

Another possible extension to the software we proposed would be the inclusion of solutions for other variations of TEP, such as those described in [5,13,14].

References

[1] H. Z. Aashtiani, T. L. Magnanti. *Equilibria on a Congested Transportation Network*. SIAM Journal on Algebraic and Discrete Methods, Vol. 2, No. 3, pp. 213-226, 1981.

[2] ESRI - Environmental System Research Institute. *Avenue: Customization and Application Development for ArcView GIS*. Redlands, 1996.

[3] A. C. da Silva. *Otimização do Software para Modelagem do Tráfego Urbano de Goiânia*. (in Portuguese). Technical Report PIBIC/CNPq. Instituto de Informática, Universidade Federal de Goiás, December/2002.

[4] M. Fukushima. *A Modified Frank-Wolfe Algorithm for Solving The Traffic Assignment Problem*. Transportation Research, vol. 18B, pp. 169-177, 1984.

[5] D. Bernstein, S. A. Gabriel. *Solving the nonadditive traffic equilibrium problem*. Lecture Notes in Economics and Mathematical Systems Network Optimization, P. M. Pardalos, D. W. Hearn, W. W. Hager, eds., Springer, pp. 72-102, 1997.

[6] B. R. Hall. *Métodos de Resolução do Problema de Equilíbrio de Tráfego*. (in Portuguese). XVII CNMAC – Congresso Nacional de Matemática Aplicada e Computacional, Vitória-ES, 1994.

[7] INRO Consultants. *The EMME/2 software on-line documentation*. Available in: http://www.inro.ca/products/e2_doc.html. Last visited: September/2003.

[8] K. Ashok, M. E. Ben-Akiva. *Estimation and Prediction of Time-Dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows*. Transportation Science, vol. 36, pp. 184-198, 2002.

[9] M. Shigeno, I. Takahashi, Y. Yamamoto. *Minimum Maximal Flow Problem: An Optimization over the Efficient Set*. Journal of Global Optimization, vol. 25, no. 4, pp. 425-443, April/2003.

[10] S. Lawphonpanich**,** D. W. Hearn. *Simplicial Decomposition of the Asymmetric Traffic Assignment Problem*, Transportation Research, vol. 18B, pp. 123-133, 1984.

[11] J. L. Oliveira, C. B. Medeiros. *A Software Architecture Framework for Geographic User Interfaces*. Proceedings of the International Workshop on Emerging Technologies for Geo-Based Applications. Ascona, Switzerland, pp. 233-248, 2000.

[12] The PostgreSQL Global Development Group. *PostgreSQL Programmer's Guide*. Available in: http://www.postgresql.org. Last visited: September/2003.

[13] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming*. Prentice-Hall, 1985.

[14] P. T. Harker. *Multiple equilibrium behaviors on networks*. Transportation Science, vol. 22, pp. 39-46, 1988.

[15] A. S. Kenyon, D. P. Morton. *Stochastic Vehicle Routing with Random Travel Times*. Transportation Science, Vol. 37, no. 1, pp. 69-82, February/2003.