Design of information points for monitoring traffic conditions

Lídia Montero[1], *Esteve Codina*[2], *Angel Marín*[3]

[1]lidia.montero@upc.es, Statistics and Operations Research Department, Technical University of Catalonia - Barcelona-SPAIN
[2]esteve.codina@upc.es, Statistics and Operations Research Department, Technical University of Catalonia - Barcelona - SPAIN
[3]amarin@dmae.upm.es, Statistics and Mathematics Department, Technical University of Madrid - Madrid - SPAIN

Abstract

This work introduces a particular Stochastic Mathematical Program with Equilibrium Constraints that explicitly incorporates uncertainties in problem data definition through a bilevel formulation. The upper level assumes some network index performance is to be optimized whereas the lower level problem is modelled by an stochastic equilibrium assignment problem with known link cost variance which can be reduced at a set of candidate points where possibly the information to travellers can be send. The algorithmic framework proposed is an heuristic method based on the iterative optimization-assignment method.

Keywords: Stochastic Programming, Bilevel Programming, Transportation Planning, Equilibrium Assignment

AMS: 90C15 90C29 90C90 90B15 90B06

1. Introduction

The information of traffic conditions at a number of points of a traffic network usually has as purpose the enhancement of the quality of service to the users of a traffic network. A key question is to decide which points of the network are critical in order to inform the drivers of their conditions so they can have a chance to avoid congestion. The type and quality of information given is also relevant. In this contribution, the type of information given to the drivers is assumed to be an estimate of link travel times.

Hierarchical decision-making problems are applied in a wide variety of domains in the engineering, regional planning and management. These problems are all defined by the presence of two or more objectives with a prescribed order of priority or information ([14]). We consider in this work a sub-class of these problems having two levels or objectives. We refer to the upper level as the objective having the highest priority and it is defined in terms of an optimization with respect to one set of variables. The lower-level problem, it is an equilibrium problem, which in the most general case is described by a variational inequality, and is then a supplementary problem parameterized by the upper-level variables. These models are known as generalized bilevel programming problems or mathematical programs with equilibrium constraints.

In applications relating to transportation planning and management, a number of the problem inputs will often be subject to uncertainty, in particular with respect of system capacities, demand (in terms of OD matrices) or network costs (impedances). We deal with this last source of uncertainty: the travel time perception on network links. The bilevel formulation we are working with assumes that some network index performance is to be optimized whereas the lower level problem is modelled by an stochastic assignment problem with known link cost variance (which can be related to a set of candidate points where possibly the information to travellers can be send).

In hierarchical models of engineering design external conditions and measurement or manufacturing errors introduce uncertainty into the problems. In both of these cases, the uncertainty can be included explicitly by generalizing some of the problem parameters to random variables. However, this generalization complexifies the model significantly and resolution strategies will require some aproximation methods to solve the resulting stochastic programs. In the simplest case, the expected values of the random variables could be substituted for their distributions and a deterministic model then solved, but in a nonlinear problem subject to constraints, this simplification can be quite costly and indeed, the optimal cost of the expected value solution, not necessarily represent the average of the possible optimal costs, and the solution may not even be feasible with respect to the realized values of the random variables. A robust model should then take into account explicitly the range of possible variable values. Some authors, in order to take into account explicitly the variability of the random inputs, as well as the possible infeasibility, consider stochastic programming extensions of the mathematical programming problem with equilibrium constraints (see [3] for theoretical extensions).

Our approach considers an upper level function that is the classical system equilibrium performance function (Wardrop conditions, see [12]) and the lower level is the solution of an stochastic equilibrium where the random source is the travel time perception (their variance depends on the upper level variables).

In a decission process for allocating detectors for monitoring traffic conditions, each network section is modelled as a directional link and network sections might have or might not have detectors, links without detectors will have higher uncertainty on travel time perceived by drivers, than link with detectors (of any type, missfunctioning is not considered in the present approach).

Let us assume that:

1. according to traffic manager criteria, a certain subset of network links has been selected for allocating traffic detectors.
2. traffic detectors can measure link travel time with a certain error. A higher error measurement applies in a higher variance on the link travel time perceived by drivers.
3. there are several physical devices that are used as traffic detectors with different characteristics and capabilities, let us assume, for simplicity that the expensive ones have lower measurement error than the cheaper ones.
4. Uncertainty on perceived travel time is always greater in network links without detectors than on links with any detection device.

The process of the allocation of detectors might be formulated as decission problem, but we skip the integer formulation and we present an approximation where the optimal values of the upper level variables points to the influence of quality detection on the system equilibrium function (a classical measure of global network performance), i.e. if at the optimal solution $x_a < x_b$ for network links a and b, this means that acccuracy on travel time information submitted to drivers is more critical on link a than on link b, pointing out, in general, to increase the quality on detection to network links where the optimal solution shows the lower values.

Although, the quality of detection does not range on continuous values, since there is a few collection of devices where the traffic manager has to formulated his/her decission, the solutions to the problem that we compute can be a guideline on the required sensibility to the information delivered to drivers, identifying the most critical (and the reciprocal, most uncritical) links for optimal global network performance. Authors have found promising results in the reduction of the average total system cost.

2. Problem Formulation

The traffic assignment problem attempts to find the distribution of the traffic flow throughout a network of routes. It is possible to formulate the problem by means of a network model that represents the physical infrastructure and aims to compute the flows of one or more commodities on the links of the network, each commodity being related to the flows for a particular origin-destination node pair.

In practical applications such an assumption may be too restrictive, e.g. when studying congested junctions. The asymmetric model cannot be reformulated as a convex program by the use of the same arguments as in the separable case.

The traffic assignment problem has received a lot of attention; partly because of its practical importance, partly because the size of real life problems makes it a challenge for algorithmic development. Many specialized strategies have been developed, see Patriksson [9] for an excellent survey on algorithms for urban traffic network equilibria.

The notation used in the mathematical formulation is shown above:

$\mathcal{G} = (\mathcal{N}, \mathcal{A})$	A graph $\mathcal{G}$, defined by a finite set of nodes, $\mathcal{N}$, and a finite set of directed links, $\mathcal{A}$
$\mathcal{C}$	the set of origin destination pairs (i, j), OD pairs
$\mathcal{C}_i$	the set of OD pairs with origin node i each one defines a commodity
$\mathcal{R}_{ij}$	the set of routes between OD pair i and j
d_{ij}	the demand between OD pair (i, j)
$\mathcal{D}$	the set of demand values for OD pairs $(i, j) \in \mathcal{C}$
v_a	total flow on link $a \in \mathcal{A}$ (addition of link flow due to the commodities)
$\mathbf{v}$	link flow vector (bold is used to denote vectors)
$c_a(\mathbf{v})$	measured travel cost on link $a \in \mathcal{A}$ (cost due to total flow on link)
$C_a(\mathbf{v})$	perceived travel cost on link $a \in \mathcal{A}$ (cost due to total flow on link)
$\mathbf{P}_{ij}$	the path choice fraction vector for the (i, j) OD-pair paths ($\mathbf{P}_{ij} = \mathbf{0}$ and $\mathbf{1}^T\mathbf{P}_{ij} = \mathbf{0}$)
P_{ijr}	fraction of users in (i, j) OD-pair on route $r \in \mathcal{R}_{ij}$
$\mathbf{h}$	the set of route flows
$\mathbf{h}_{ij}$	the path flow vector for the (i, j) OD-pair paths
h_{ijr}	flow on route $r \in \mathcal{R}_{ij}$
c_{ijr}	measured total travel cost on route $r \in \mathcal{R}_{ij}$
C_{ijr}	perceived total travel cost on route $r \in \mathcal{R}_{ij}$ it is a random variable
u_{ij}	the cost of the lowest measured cost route between OD pair (i, j) at equilibrium

If we consider a population of drivers who are about to take a trip between a given origin and a given destination, possibly connected by many alternative paths and each associated with some travel cost, then due to variations in perception and exogenous factors (such as weather, lighting, etc.), the path costs are perceived differently by each driver, and thus, it is natural to model the perceived cost of each path or each link (as the proposal currently selected by the authors). Each driver, however, will perceive path time differently and therefore each driver may choose a different path and the perceived travel cost can be modelled as a random variable; its probability density function gives the probability that a driver randomly drawn from the population will associate a given perceived travel cost and thus how many drivers will use each path. In order to make clear the scope of the application for the problem we will formulate, let us fix that C_{ijr} usually refers to the perceived path travel time on route r between OD pair (i,j) and it is assumed to be related to the measured path travel time c_{ijr} by

$$C_{ijr} = c_{ijr} + \xi_{ijr} \; \forall \, r \; \in \; \mathcal{R}_{ij} \tag{1}$$

A much more practical formulation for the algorithmic point of view can be derived from the assumption that link travel costs are inherently random:

$$C_a(\mathbf{v}) = c_a(\mathbf{v}) + \epsilon_a(\mathbf{v}) \; \forall \, a \; \in \; \mathcal{A} \tag{2}$$

In other words, the average perceived cost is the measured or deterministic travel cost at path level $E[\xi_{ijr}] = 0$ and $E[C_{ijr}] = c_{ijr}$. Or at link level, $E[\epsilon_a] = 0$ and $E[C_a] = c_a$ for any $\mathbf{v}$ link flow.

The modelling assumption considered in the deterministic traffic assignment problem was stated by Wardrop [12]. It postulates that the measured journey times on all the routes actually used are equal or less than those which would be experienced by a single vehicle on any unused route. The implication of this principle is that the routes are shortest with respect to the current flow-dependent delays. The traffic flows that satisfy this principle are usually referred to as *"deterministic user optimized flows"*, since each user chooses the route that he considers the best. In contrast *"deterministic system optimized flows"* are characterized by Wardrop's second principle which states that the total measured travel time is minimum [4]. User and system optimal conditions are equivalent when travel times are independent of traffic flow, which it is not realistic in practice.

Assume a transportation network $\mathcal{G}$ with a single mode of transit and a fix demand $\mathcal{D}$. Consider a specific origin destination pair (i,j) and $\mathcal{R}_{ij}$ the set of available paths joining OD pair (i,j). The route flow vector $\mathbf{h}$ induces flows v_a on each link $a \in \mathcal{A}$ given by the expression

$$v_a \; = \sum_{(i,j)\in\mathcal{C}} \; \sum_{r\in\mathcal{R}_{ij}} \delta_{ar} \, h_{ijr} \tag{3}$$

where $\delta_{ar} = 1$ if link a belongs to path r; and $\delta_{ar} = 0$ otherwise. In words, v_a is the sum of flows h_{ijr} on all paths r, over all OD pairs (i,j).

Let $\mathbf{v}$ denote the vector of arc flows and let $\mathbf{\Delta}$ denote the link path incident matrix. Then in vector form, link flows and path flows are related by the following expression

$$\mathbf{v} = \mathbf{\Delta}\,\mathbf{h}$$

Let each unit of flow on link a incur a measured travel cost $c_a(\mathbf{v})$ which depends upon the vector $\mathbf{v}$ of link flows in the network. In the separable traffic assignment problem, the cost on a link depends solely upon the flow v_a on that link, in which case it usually increases with increased levels of the link flow. If we now assume that the measured cost on any path of the network, as a function of path flows, is the sum of measured travel costs on the links of that path, then

$$c_{ijr} = \sum_{a \in \mathcal{A}} \delta_{ar}\, c_a(v_a)$$

We refer to this form of route costs as an *additive model*. Stating this relationship more compactly in vector form, we obtain

$$\mathbf{c}(\mathbf{h}) = \mathbf{\Delta}^{T}\,\mathbf{c}(\mathbf{v}),$$

where ^{T} denotes transposition.

In this expression, $\mathbf{c}(\mathbf{h}) = (\, c_{ijr}(\mathbf{h})\,)$ is a vector-valued function specifying the measured travel costs on each path r and $\mathbf{c}(\mathbf{v}) = (c_a(v_a))$ is a vector valued function whose components specify the measured link travel costs.

An extension generally considered to *non-additive models*, the vector-valued function $\mathbf{c}(\mathbf{h})$ splits into an additive plus a non-additive component,

$$\mathbf{c}(\mathbf{h}) = \mathbf{c}^{\mathbf{ADD}}(\mathbf{h}) + \mathbf{c}^{\mathbf{NADD}}(\mathbf{h}) = \mathbf{\Delta}^{T}\,\mathbf{c}(\mathbf{v}) + \mathbf{c}^{\mathbf{NADD}}(\mathbf{h})$$

The measured route travel costs $c_{ijr}(\mathbf{h})$ on each route r joining an OD pair (i,j) defines the least travel cost u_{ij} over all paths joining that OD pair. This is

$$u_{ij} \equiv \min_{r \in \mathcal{R}_{ij}} c_{ijr}(\mathbf{h})$$

These least measured travel costs certainly provide a reference point against which to measure any route's ability to attract trips. If different routes are used for a given OD pair (i, j), the travel cost of the routes used is u_{ij}, and Wardrop's first principle may be rewritten as

$$\begin{aligned} h_{ijr} &> 0 \quad \Rightarrow \quad c_{ijr} = u_{ij} \\ h_{ijr} &= 0 \quad \Rightarrow \quad c_{ijr} \geq u_{ij} \end{aligned}$$

or equivalently as

$$h_{ijr} \cdot (c_{ijr} - u_{ij}) = 0 \quad \forall\, r \in \mathcal{R}_{ij}\,;\, \forall\,(i,j) \in \mathcal{C} \tag{4}$$
$$c_{ijr} - u_{ij} \geq 0 \quad \forall\, r \in \mathcal{R}_{ij}\,;\, \forall\,(i,j) \in \mathcal{C} \tag{5}$$
$$\sum_{r \in \mathcal{R}_{ij}} h_{ijr} = d_{ij} \quad \forall\,(i,j) \in \mathcal{C} \tag{6}$$
$$h_{ijr} \geq 0 \quad \forall\, r \in \mathcal{R}_{ij}\,;\, \forall\,(i,j) \in \mathcal{C} \tag{7}$$
$$u_{ij} \geq 0 \quad \forall\,(i,j) \in \mathcal{C} \tag{8}$$

Equation 6 reflects that route flow vector $\mathbf{h}$ must be demand feasible, or in other words, the total flow between any OD pair (i, j) satisfies the demand d_{ij} between this OD pair. Negative route flows and route travel costs are not allowed, thus equations (7) and (8).

Consistency between demand and path flows can be alternatively expressed as,

$$\mathbf{h}_{ij} = d_{ij}\mathbf{P}_{ij} \;\forall\,(i,j) \in \mathcal{C}$$

Note that this *deterministic equilibrium condition* must be satisfied by all origin destination pairs in $\mathcal{C}$. Moreover, any shift in user flows from route r to route s would affect the measured travel cost on any route that shares links with these paths. Therefore, the Wardrop's first principle or *deterministic equilibrium condition* requires a route flow vector $\mathbf{h}$ that simultaneously achieves equilibrium for all users.

The additivity property is used in our formulations and is assumed to hold when developing algorithmic approaches. There are two main reasons for this:

1. Under the non additive assumption there does not exist a transformation of the above arc route formulations to arc flow formulations.
2. On working with large networks, as our case, the arc route formulations of traffic assignment models involves very large scale problems, and as far as, it is unknown if an efficient algorithm could be designed for solving

large general problems, it has been preferred to remove the extra size due to arc the route model and concentrate efforts on solving the aggregated arc flow model.

The equivalent optimization problem for the separable traffic assignment problem can be considered as one application of a physical principle, known as the variational or minimal principle. This principle states that equilibrium problems can be stated in one of two equivalent forms:

- A set of equilibrium conditions.
- A related mathematical programming problem.

Beckmann [1] was the first to consider an optimization formulation (convex mathematical programming problem) of the separable traffic equilibrium problem and to present the necessary conditions for the existence and uniqueness of the deterministic user equilibria, where no interactions between the links in the network are present. The equivalent mathematical programming program is

$$\begin{array}{lrcll} \min & \displaystyle\sum_{a\in\mathcal{A}} \int_0^{v_a} c_a(s)ds & & & \\ \text{s.t.} & \displaystyle\sum_{r\in\mathcal{R}_{ij}} h_{ijr} & = & d_{ij} & \forall\,(i,j)\in\mathcal{C} \\ & h_{ijr} & \geq & 0 \quad \forall\, r\in\mathcal{R}_{ij} & \forall\,(i,j)\in\mathcal{C} \end{array}$$

and the definitional constraint

$$\sum_{(i,j)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{ij}} \delta_{ar}\, h_{ijr} = v_a \quad \forall\, a\in\mathcal{A}$$

The above defined problem is a minimization program with nonlinear objective function and linear constraints. The objective function consists of a sum over the link set of the integral of measured link cost functions evaluated in the actual point. The definitional constraint gives the key to convert path flows into link flows, let us observe that the constraints are defined in terms of the path flow variables $\mathbf{h}$ and the objective function is defined in terms of the link flow variables $\mathbf{v}$.

The constraints are the equations 6 and 7 used when defining the demand satisfaction requirements on the mathematical formulation of Wardrop's first principle. The Kuhn-Tucker conditions for the minimum of the Lagrangian function of the former program defines the first order optimality conditions, that can be seen to be the cost of routes with minimum cost condition (5) and thus the equivalence to Wardrop's first principle.

Let $\mathcal{H}$ be the set of all demand feasible path flow vectors

$$\mathcal{H} \stackrel{\text{def}}{=} \{ \mathbf{h} \,|\, \sum_{r\in\mathcal{R}_{ij}} h_{ijr} = d_{ij},\; h_{ijr} \geq 0,\; \forall\, r\in\mathcal{R}_{ij}\,,\, \forall\,(i,j)\in\mathcal{C}\} \tag{9}$$

$\mathcal{H}$ is finite since the number of OD pairs is finite and the number of routes between each OD pair on $\mathcal{G}$ is also a finite number, if routes are defined as simple without cycles.

However, travel cost functions often become nonseparable and asymmetric and a solution to the deterministic Wardrop conditions for user equilibrium can then not be formulated as an optimization problem; instead, they are stated as variational inequality or complementarity models.

The arc flow variational inequality formulation for the traffic assignment problem when travel costs on a route satisfy the additivity property is detailed below (see Patriksson [9] for a deep discussion of alternative formulations):

$$\text{Find } \mathbf{v}^* \in \mathcal{V} \; s.\,t.\; \mathbf{c}(\mathbf{v}^*)^T \cdot (\mathbf{v} - \mathbf{v}^*) \geq 0 \;\; \forall\, \mathbf{v}\in\mathcal{V} \tag{10}$$

and

$$\mathcal{V} \stackrel{\text{def}}{=} \{ \mathbf{v} \,|\, v_a = \sum_{(i,j)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{ij}} \delta_{ar}\, h_{ijr} \;\; \forall\, a\in\mathcal{A},\;\; \mathbf{h}\in\mathcal{H}\, \} \tag{11}$$

According to Sheffi [11], the stochastic network equilibrium conditions are extensions of the deterministic user equilibrium conditions, given a fixed OD demand matrix, the stochastic equilibrium conditions in the routes space can be characterized by the following equations:

$$h_{ijr} = d_{ij} P_{ijr} \qquad \forall\,(i,j)\in\mathcal{C} \tag{12}$$

$$P_{ijr} = Pr(C_{ijr} \leq C_{ijk}, \forall\, k \in \mathcal{R}_{ij} \mid \mathbf{v}) \qquad \forall\,(i,j)\in\mathcal{C} \tag{13}$$

$$C_{ijr} = \textstyle\sum_{a\in\mathcal{A}} \delta_{ar}\, C_a(v_a) \qquad \forall\, r\in\mathcal{R}_{ij}\,;\, \forall\,(i,j)\in\mathcal{C} \tag{14}$$

$$\sum_{r\,\in\,\mathcal{R}_{ij}} h_{ijr} = d_{ij} \qquad \forall\,(i,j)\in\mathcal{C} \tag{15}$$

$$h_{ijr} \geq 0 \qquad \forall\, r\in\mathcal{R}_{ij}\,;\, \forall\,(i,j)\in\mathcal{C} \tag{16}$$

Now $C_{ijr}(\mathbf{v})$ is the random variable representing the perceived travel cost on route r between OD pair (i, j), and the probability that a given route is chosen is the probability that its travel cost is perceived to be the lowest of all the alternative routes. Equation 13 specifies a route-choice probability that can be interpreted in the framework of discrete choice model in transportation analysis (for a comprehensive survey see Ortúzar and Willumsen [8]). For very low variances, the travellers perceive the differences between path travel costs accurately and act accordingly; in fact, if the variance is zero, the behavior is the associated with the deterministic user equilibrium assignment in which all users will be assigned to the shortest travel cost paths from their origin to their destination. If the variance is large, the change in the choice probability is more moderate since the the perception of travel cost is less accurate. At the limit, when the variance is extremely large, the actual travel costs do not affect the perception, which is completely random; anyway, this hypothesis is not realistic in practice, since variances tend to be relatively small, and under high congestion conditions, deterministic equilibrium provides a good approximation to the stochastic one (see [11]).

Note that the choice probability is conditional on the values of the mean link travel cost at equilibrium, but these times are not known and are assumed to be flow dependent: at stochastic user equilibrium, no driver can improve his/her perceived travel cost by unilaterally changing routes and the measures travel cost on all used paths is not going to be equal (as in the deterministic case), instead the travel time will be such that equation (12) is satisfied for the equilibrium path flows, that in turn, will be associated with link flows that satisfy equations (14) and (15) for the equilibrium travel costs.

One of the practical reasons for using stochastic equilibrium models is the sensitivity of the flows in deterministic equilibrium models to small changes in the network, which it is avoided for stochastic equilibrium models.

The various models for stochastic equilibrium differ from each other in the assumed distribution of the perceived travel cost. Classical models developed in literature and used in practice, relay on multinomial logit and probit formulations. The logit formulation is based on the assumption that the error terms (random component) of the perceived path cost for all alternative path joining a given OD pair are identically and independently distributed Gumbel variates. This model may exhibit some unreasonable properties, as discussed largely in literature, resulting from the independence assumption underlying the logit utility choice probability function derived from route overlapping, which is a very common situation in urban networks. Route overlapping means route correlation and means driver perception lower than it really is in overlapped routes. Logit stochastic assignment might seem reasonable if the amount of overlapping between OD paths is relatively small, which might occur in regional planning networks, but it is unacceptable in urban networks (see [11] for a detailed discussion). Also, another possible anomaly is due to the fact that the path choice probabilities are determined solely on the basis of travel cost differences, for example, in practical situations travel cost is travel time, and a 5 minute difference **is not the same** (by common sense) in route travel times (measured) around 120 minutes, that around 10 minutes. Probit formulation models the choice of alternative routes between an OD pair assuming perceived travel cost on paths normally distributed. Probit formulations alleviate model faults associated with logit-based stochastic equilibrium, however computational costs of path based probit formulations are seriously increased.

Once perceived path travel cost distribution is specified, then, **in theory**, the probability of selecting each alternative route can be calculated and the flow assigned accordingly 12 and the link flows can then be calculated by 3. We will see above that it is not so simple. Also, one should not forget the implications of the assumptions of stochastic network loading, path flows are random variables and thus, link flows also become random variables. Stochastic equilibrium models give the mean flow on each path and **the mean flow on each link** of the network. Distribution of path flow variables and joint density function of the flow on all links could be approximated by a multivariate normal distribution, as discussed for some authors ([11]).

For logit-based formulations equation 13 becomes ([8]):

$$P_{ijr} = \frac{\exp\left(-\theta c_{ijr}\right)}{\sum_{\forall\, k \,\in\, \mathcal{R}_{ij}} \exp\left(-\theta c_{ijk}\right)} \qquad (17)$$

where the parameter θ is a constant that scales the perceived path travel cost, in such a way that the random term error becomes a standard random Gumbel variate, this is with null expectation and variance $\frac{\pi^2}{6}$, which implies that the variance of the (Gumbel) perceived path travel cost is $var(C_{ijr}) = \frac{\pi^2}{6\theta^2}$. Thus, the parameter θ is inversely proportional to the standard error of C_{ijr}. Most of the commercial procedures that traffic engineers find available to model logit-based choice in transportation planning models require to fix *global parameter* θ and thus, the variance of C_{ijr} will never depend on currently assigned link flow, nor path flow, nor OD pair characteristics and this is a very serious limitation against realism of many commercial stochastic equilibrium results and in general against logit-based discrete choice models results appearing in transportation planning (due to commercial implementation of procedures): variance of random variables is fixed globally and *a priori*. For example, perception variance of travel time is the same for routes connecting OD pairs at 100 km or 1 km aerial distance!!!.

As mentioned above, the probit model assumes that the perceived travel cost on paths is normally distributed with a mean equal equal to the measured path travel cost and this distribution can be derived from the normal distribution of the perceived link travel costs. Let C_a denote the perceived travel cost on link a, it is a random variable normally distributed with mean equal the measured link travel cost and variance σ_a^2, and non-overlapping road links are assumed to be independent, that is,

$$C_a \sim N(c_a, \sigma_a) \ \ \forall\, a \in \mathcal{A} \tag{18}$$

Several possibilities have been developed by the authors:

1. Standard deviation proporcional to link travel cost: $\sigma_a = \beta c_a(\mathbf{v})\ \forall \mathbf{a} \in \mathcal{A}$.
2. Standard deviation proporcional to link travel cost under free-flow conditions: $\sigma_a = \beta c_a(\mathbf{0})\ \forall\, \mathbf{a} \in \mathcal{A}$.
3. Standard deviation proporcional to link travel cost at link capacity-flow conditions: $\sigma_a = \beta c_a(\mathbf{s_a})\ \forall\, \mathbf{a} \in \mathcal{A}$.
4. Standard deviation proporcional to link length $\sigma_a = \beta c_a(l_a)\ \forall\, a \in \mathcal{A}$.

where β is a proportionality constant that can be interpreted as the variance of the perceived link travel cost over a link of unit travel cost.

The normality assumption may be argued not to be suitable in general since travel costs (travel times, usually) can not be negative; and, from a formal point of view, a nonnegative distribution positive skewness (gamma distribution), could be more appropiated; but, in practice, it is reasonable to believe that the magnitude of the perception error is small with respect to the expected value and thus, it is realistic.

The random vector of perceived path travel cost is a multivariate normal vector with non-diagonal terms in the covariance matrix (in general). Once the perceived path travel cost distribution is specified, neither the probability of selecting each alternative route can be analytically calculated or even approximated since path enumeration is required and thus, nor the flow assigned accordingly 12 and nor the link flows can then be calculated by 3. This is also true if the random perception is defined directly at path level, being less realistic.

From a computational point of view, the number of alternative paths connecting a typical OD pair in real networks is extremely large, consequently it is not possible to enumerate all these paths, calculate probabilities for each one to be chosen under a given distribution, compute the flow assigned to each route, and then, considering all OD pairs, compute the assigned link flow. Thus, algorithms for stochastic equilibrium on large networks, have to be link-based procedures, based on the random perception at link cost level (not path level), which avoid path enumeration and perform the assignment of demand (OD matrix) using only link variables. Feasible methods for large networks are Monte Carlo simulation-based.

If measured link costs are congestion independent, then determining the probabilities P_{ijr} and total expected link flows $\mathbf{v}$ is known as the Stochastic Network Loading (SNL) Problem. If the distribution of the path flow random variables does not depend on link congestion, then the expected total link flows are a monotone non-increasing function of the link costs. For the case of logit based model the SNL problem can be solved rather efficiently, for instance by the classical STOCH method of Dial ([11]). Bell proposes two alternatives to Dial's method for the stochastic loading problem under logit path flow distribution. For the case of a probit based model, Maher developed the SAM stochastic loading method using Clark's approximation for the path-choice probabilities P_{ijr} outperforming the Burrell's method.

For the congestion dependent case and additive models, the formulation of SUE models as a fixed point problem, was done by Daganzo in [2] and by Cantarella:

$$\mathbf{v} = \sum_{(i,j)\,\in\,\mathcal{C}} \{\, d_{ij} \sum_{r \in R_{ij}} P_{ijr} \left(\sum_a \delta_{ar}\, c_a(\mathbf{v}) \right) \delta_{ar} \,\} \tag{19}$$

For the congestion dependent case and non-additive models, the formulation of SUE models as a fixed point problem becomes,

$$\mathbf{v} = \sum_{(i,j)\,\in\,\mathcal{C}} \{\, d_{ij} \sum_{r \in R_{ij}} P_{ijr} \left(\left(\sum_a \delta_{ar}\, c_a(\mathbf{v}) \right) + c_{ijr}^{NADD} \right) \delta_{ar} \,\} \tag{20}$$

Sufficient conditions for the existence of solutions can be derived through Brouwer's theorem requiring continuity of cost functions and path choice functions. Furthermore, assuming that the SNL process is monotone non-increasing with link costs, or equivalently, the link cost functional is monotone strictly increasing, then existence and uniqueness of path flows and costs holds.

Fisk extended the optimization problem of Beckmann for the deterministic and separable user equilibrium to a logit path choice model with dispersion parameter θ for the stochastic, but separable user equilibrium. Fisk's problem is:

$$Min_{\,\mathbf{v}\in\mathcal{V},\,\mathbf{h}\in\mathcal{H}} \quad \sum_{a\in\mathcal{A}} \int_0^{v_a} c_a(x)dx + \frac{1}{\theta} \sum_{(i,j)\in\mathcal{C}} \sum_{r\in R_{ij}} h_r(\ell n(h_r) - 1) \tag{21}$$

It can be shown that the solutions of Fisk's model reproduce the distribution of trips for the logit-based SUE model over the network paths (equation (17)).

The logit model drawback of concentrating too much flow over overlapping paths has been treated by Cascetta, who proposed the C-logit model but explicit path enumeration is required making the proposal not realistic in practice for large networks.

Chen and Alfa proposed two algorithms to solve Fisk's problem in that do not require path enumeration, that was further modified by Bell to avoid inconsistency in flows. These two algorithms are modifications of the flow averaging algorithm or method of successive averages (MSA)used in this work and discussed below. Another algorithm that uses explicit path enumeration is that of Damberg based on the partial linearization algorithm scheme due to Patriksson combined with a column generation phase.

A more general optimization model for any probabilistic distribution of path choice was proposed by Daganzo and Sheffy ([11])for the separable situation:

$$Min_{\,\mathbf{v}\in\mathcal{V}} \quad Z(\mathbf{v}) \,=\, -\sum_{(i,j)\in\mathcal{C}} d_{ij}\tilde{C}_{ij}(\mathbf{v}) + \sum_{a\in\mathcal{A}} v_a c_a(v_a) - \sum_{a\in\mathcal{A}} \int_0^{v_a} c_a(x)dx \tag{22}$$

where $\tilde{C}_{ij}(\mathbf{v}) \,=\, E[\,Min_{r\in R_{ij}}\{C_{ijr}\}\,|\,\mathbf{v}\,]$ is the expected perceived travel time function for trip makers of OD pair $(i,j) \,\in\, \mathcal{C}$. In general the objective function of this model is non-convex with a unique stationary point where it is locally strictly convex in the link flow variables (see, for instance [11], chapter 12), that, additionally is the unique global solution in terms of the link flow variables although the solutions in the path flows may be non-unique. It can be shown that the gradient of $Z(\mathbf{v})$ has the expression:

$$(\nabla_{\mathbf{v}} Z(\mathbf{v}))_a \,=\, (y_a - v_a)\frac{dc_a}{dv_a}(v_a), \text{ where } y_a \,=\, \sum_{(i,j)\in\mathcal{C}} \{\, d_{ij} \sum_{r\in R_{ij}} P_{ijr}(\mathbf{v})\delta_{ar}\,\} \tag{23}$$

Thus, at a point v_a, y_a are the link flows resulting from a stochastic network loading when the measured link travel costs are given by $c_a(v_a)$. It must be noticed that $\nabla_{\mathbf{v}} Z$ vanishes at a point verifying $v_a = y_a$, $\forall\, a \in A$, and this implies that the problem (22) can be solved as an unconstrained problem. There are two basic difficulties in applying a descent method to the problem of minimizing $Z(\mathbf{v})$. The first one is that in order to evaluate $\nabla_{\mathbf{v}} Z$ a stochastic network loading problem must be solved at each iteration. The second difficulty is to evaluate the objective function if a line search is to be used. Maher essays descent algorithms to minimize $Z(\mathbf{v})$ under a logit path choice model, using his SAM loading method, he undertakes heuristically these difficulties and develops a descent algorithm with cubic interpolation for the line searches with better convergence characteristics than the MSA algorithm for a probit based model.

In theory, models incorporating stochastic and equilibrium properties look particularly attractive; there are, however, operational and practical difficulties for applying them as the convergence properties of algorithms and the extension to the nonseparable user equilibrium. A practical algorithm to perform SUE assignment is the iterative loading method known as MSA algorithm or Method of Successive Averages algorithm, that can be described as follows:

STEP 1. Set current measured costs to free-flow travel costs, i.e., $v_a = 0$ and $c_a = c_a(\mathbf{0})\ \forall\, \mathbf{a} \in \mathcal{A}$. Make outer iterator counter $n = 1$ and compute a stochastic network loading based on the set of initial travel times. This generates a set of links flows $\mathbf{v}^{(n)}$.

STEP 2. Update measured link travel costs according to $\mathbf{c^{(n)}} = \mathbf{c(v^{(n)})}$.

STEP 3. Compute a **stochastic network loading** based on the current set of link travel times $\mathbf{c^{(n)}}$. This yields and auxiliary link flow pattern $\mathbf{y^{(n)}}$. *Accuracy controlled by maximum inner iteration parameter.*

STEP 4. Find the new flow pattern by setting $\mathbf{v}^{(n+1)} = \mathbf{v}^{(n)} + \frac{1}{n}(\mathbf{y}^{(n)} - \mathbf{v}^{(n)})$

STEP 5. If convergence is attained, stop. If not, increase the outer iteration counter, $n = n + 1$, and go to Step 2.

This algorithm will always tend to produce small changes in flows and costs as n becomes large, but it converges to the right Stochastic User Equilibrium solution in the long run as has been shown by Sheffi [11], the convergence is not monotonic because the search direction is only a descent direction on the average. The speed of convergence depends on the level of network congestion and the variance of perceived link travel costs. Althought SUE assignment sounds an attractive model alternative, it would only be advantageous in low to medium congested networks, in heavily congested networks, a deterministic user equilibrium gives a fairly good approximation at a faster convergence rate, since very efficient algorithm were developed for UE in the last decades of the twentieth century.

For the deterministic user equilibrium the authors have already implemented under the variational inequality formulation in a restricted simplicial decomposition framework, a very efficient software program written in C++ that uses a variable metric projection scheme to solve the master problem and a LDQUE algorithm for computing trees of shortest paths for the generation of a new feasible flows in the link flow space. This C++ software is an extension of a previous Fortran version developed by Montero ([7]) in her doctoral thesis.

For SUE, a MSA algorithm has been implemented. MSA convergence is not monotonic, fact that supposes a serious problem in the context of practical detection of convergence. In the current implementation, the stopping criteria has been fixed to a maximum number of outer iterations or to satisfaction of flow similarity over the last m parameter iterations, usually being m a fixed parameter between 3 to 5. Let $\bar{\mathbf{v}}^{(\mathbf{n})}$ denote the average flow over the last m iterations,

$$\bar{\mathbf{v}}^{(n)} = \frac{1}{m} \sum_{l=0\ldots(m-1)} \mathbf{v}^{(n-l)} \tag{24}$$

Thus the convergence criteria used is based on moving averages,

$$\frac{\sum_{a \in \mathcal{A}} |\bar{v}_a^{(n+1)} - \bar{v}_a^{(n)}|}{\sum_{a \in \mathcal{A}} \bar{v}_a^{(n)}} < \kappa \tag{25}$$

Observe that Step 3 in MSA Algorithm implies a SNL Subproblem were any algorithm can be applied, but if the choice probabilities are generated according to a Monte Carlo simulation , then SNL generates unbiased estimates of the search direction $\mathbf{y}^{(n)}$ regardless of how many simulation iterations are performed, in particular 1 inner iteration seems to be the most efficient choice (see Sheffi [11] for a deep discussion).

MSA algorithm may be applied to random link perceived travel costs being: normal, uniform (as Burrell original proposal), Gumbel, gamma, etc. It is a very general and flexible scheme were any random cost can be modelled.

Several possibilities have been developed by the authors, up to now interactions between network links are not allowed, and thus, only separable model can be considered:

- Perceived link travel cost Gumbel with standard deviation proporcional to link travel cost: $\sigma_a = \beta c_a(\mathbf{v}) \; \forall \, \mathbf{a} \in \mathcal{A}$.

- Perceived link travel cost normal $C_a \sim N(c_a, \sigma_a)\ \ \forall\, a\ \in\ \mathcal{A}$ and,

  1. Standard deviation proporcional to link travel cost: $\sigma_a = \beta c_a(\mathbf{v})\ \forall \mathbf{a} \in \mathcal{A}$.
  2. Standard deviation proporcional to link travel cost under free-flow conditions: $\sigma_a = \beta c_a(\mathbf{0})\ \forall\, \mathbf{a}\ \in\ \mathcal{A}$.
  3. Standard deviation proporcional to link travel cost at link capacity-flow conditions (vector $\mathbf{s}$): $\sigma_a = \beta c_a(\mathbf{s})\ \forall\, a\ \in\ \mathcal{A}$.
  4. Standard deviation proporcional to link length $\sigma_a = \beta c_a(l_a)\ \forall a \in \mathcal{A}$.

where β is a proportionality constant that can be interpreted as the variance of the perceived link travel cost over a link of unit travel cost.

The extension of the MSA algorithm to compute SUE under nonseparable assumptions is straightforward under a diagonalization scheme, according Sheffi [11], but there are not references of applications and computational results in the asymmetric instances of the Traffic Assignment Problem in the literature. This extension will be discussed in a future work of the authors. The description of the algorithm given above copes with the general (non-separable) TAP link cost functional definition.

3. Bilevel Problem Formulation

In order to assess the decision process of allocating different levels of quality of information at a set of predetermined links $a \in \hat{A}$ in the network, the following hierarchical program can be formulated:

$$\begin{array}{ll} \min_{\mathbf{x}} & \psi(\mathbf{x}) = \mathrm{E}[\phi(\mathbf{V})\,|\,\mathbf{x}] \\ \text{subject to:} & \mathbf{x} \in \mathcal{X} \end{array} \tag{26}$$

In this program the function ϕ that can be considered in a natural way is the expected value of the system performance function, as the expected total travel time spent by the users of the traffic network, i.e., for the deterministic TAP, the system performance function becomes: $\phi(\mathbf{v}) = \sum_{a \in A} c_a(\mathbf{v}) v_a$. ($c_a(\mathbf{v}) = c_a(v_a)$ in a diagonal model), where $\mathbf{v}$ would be the volumes on deterministic user equilibria.

The lower level in the hierarchical scheme is a stochastic user equilibrium traffic assignment model as the one described in section 2 and the link flows $\mathbf{V}$ are random variables distributed accordingly to a probability law with parameters that, in general, are functions of $\mathbf{x}$. Thus, the expected value $\mathbf{v}(\mathbf{x}) = \mathrm{E}[\mathbf{V}|\mathbf{x}]$ of the link flows is in fact a function of the current values $\mathbf{x}$ that the parameters may adopt. Thus, the upper-level objective function ψ

is the expected system equilibrium performance function for fixed upper-level variables $\mathbf{x}$. This value will be estimated by Monte Carlo simulation.

The upper level variables $\mathbf{x}$ must be associated to a measure of uncertainty on link travel time perception (in our approach they are related to the variance of the perceived link travel time). The set $\mathbf{S}(\mathbf{x})$ is the solution set of the stochastic assignment equilibrium parametrized by the upper level variables $\mathbf{x}$. The upper level variables lie in an admissible set $\mathcal{X}$ which we will assume to be of the type:

$$\mathcal{X} \stackrel{\text{def}}{=} \{\sum_{a\in\hat{A}} x_a \geq \delta;\ \underline{\gamma} \leq x_a \leq \overline{\gamma}\} \tag{27}$$

Other functional forms in the upper level could be used such as for example: $\psi(\mathbf{x}) = \mathrm{E}[\phi(\mathbf{V})\,|\,\mathbf{x}] + k \cdot (\mathrm{Var}[\phi(\mathbf{V})\,|\,\mathbf{x}])^{\frac{1}{2}}$ for fixed $k > 0$, and thus $Prob(\phi(\mathbf{V}) \geq \psi(\mathbf{x})) < 1/k^2$ by Chebishev's inequality.

3.1. Monte-Carlo simulation of the lower level model

Once the stochastic equilibrium is attained with enough accuracy, i.e. the mean link flows $\mathbf{v}$ have been computed, the evaluation of the objective ψ can be done only by Monte Carlo simulation in order to obtain an estimate for ψ, say $\hat{\psi}$.

The procedure to obtain a random instance for the function ϕ could be accomplished as follows:

- Take N a sufficiently large integer and set the random link flows to zero: $\mathbf{V} = \mathbf{0}$.

- For $\ell = 1, 2, ..., N$ do:

a) Generate random numbers ϵ_a for the perceived link travel costs.

b) Calculate shortest paths for each O-D pair (i, j) accordingly to the link costs $c_a(\mathbf{v}) + \epsilon_a$ and load the incremental demand d_{ij}/N on them obtaining a set of incremental link flows $u_a^\ell \ \ \forall\, a \in A$ for the ℓ-th step.

c) Accumulate the incremental link flow onto the random link flow:

$$v_a = v_a + u_a^\ell \ \ \forall\, a \in A \tag{28}$$

- Calculate $\phi(\mathbf{v})$.

This procedure should be repeated several times, say m, in order to obtain a sample $\phi^1, \phi^2, ...\phi^m$ to compute the mean value $\bar{\phi} = \frac{1}{m}\sum_{\ell=1}^{m}\phi^\ell$ The following variance reduction technique can be used in order to improve the efficiency of this method.

3.2. A variance reduction method

In order to evaluate the objective function in (26) a variance reduction technique must be used. The one that has been chosen is the control variable technique (see, for instance [10], chapter 4).

Let us note for a sake of simplicity $\phi(\mathbf{v}) = \sum_{a\in A} c_a(\mathbf{v})v_a$ which is the variable for which an estimation of its mean value is required at each evaluation of the objective function in (26). As the link flows $\mathbf{V}$ are random variables, let them write down as $\mathbf{V} = \mathbf{v} + \delta$, with $\mathbf{v} = \mathrm{E}[\mathbf{V}]$ and $\mathrm{E}[\delta] = \mathbf{0}$. Let us define an auxiliary random variable φ as,

$$\varphi = \sum_{a\in A}\left(c_a(\mathbf{v}) + \mathbf{c'}_{\mathbf{a}}^{\mathbf{T}}\delta\right)V_a \tag{29}$$

where $\mathbf{c'_a} = \nabla c_a(\mathbf{v})$ and $\mathbf{c}(\mathbf{v}) + \boldsymbol{\epsilon}$ is the random vector of perceived travel costs on links.

Clearly the expectation $\varphi_0 = \mathrm{E}[\varphi]$ can be easily computed once the mean value $\mathbf{v}$ of the stochastic link flows $\mathbf{V}$ are given by the solution of the stochastic equilibrium lower level problem for $\mathbf{x}$ fixed,

$$\varphi_0 = \mathrm{E}[\varphi] = \sum_{a\in A} c_a(\mathbf{v})V_a \tag{30}$$

The control variable would then be $f_1 = \phi - \eta_1(\varphi - \varphi_0)$, where η is a suitable coefficient. Then $\mathrm{E}[f_1] = \mathrm{E}[\phi]$.

On the other hand it is verified that the link flows $V_a,\ a \in A$ are random variables with an approximate normal distribution, i.e, $V_a = v_a + \delta_a,\ \delta_a \sim N(0, \zeta_a),\ \zeta_a = Var^{1/2}[\delta_a]$.

Both variables φ and ϕ are clearly correlated since,

$$\phi_a(v) = c_a(\mathbf{v}+\delta)(\mathbf{v_a}+\delta_\mathbf{a}) = \varphi_\mathbf{a} + \mathbf{v_a}\mathbf{o_a}(\delta) + \delta_\mathbf{a}\mathbf{c_a}(\mathbf{V}) \tag{31}$$

The method consists of generating a sample for the ϕ's and the φ's, $(\phi^1, \phi^2, ..., \phi^\nu)$, $(\varphi^1, \varphi^2, ..., \varphi^\nu)$ and compute the slope $\hat{b}$ of the linear regression $\phi \leftrightarrow \varphi$:

$$\hat{b} = \frac{\sum_{i=1}^{n}(\phi_i - \bar{\phi})(\varphi_i - \bar{\varphi})}{\sum_{i=1}^{n}(\varphi_i - \bar{\varphi})^2} \tag{32}$$

Then, an estimate for $E[\phi(\mathbf{V})]$ with lower variance is given by,

$$\bar{f_1} = \bar{\phi} - \hat{b}(\bar{\varphi} - \varphi_0) \quad \text{and} \quad \frac{Var[f_1]}{Var[\phi]} = 1 - r^2_{\phi,\varphi} \tag{33}$$

Another auxiliary variable that can be chosen is:

$$\xi = \sum_{a \in A} c_a(\mathbf{v})V_a \tag{34}$$

for which the same computations can be performed along the Monte-Carlo simulation of the lower level problem. Its expectation is also known: $\mathrm{E}[\xi] = \xi_0 = \varphi_0$. The new control variable would then be $f_2 = \phi - \eta_1(\xi - \xi_0)$. Also, $\mathrm{E}[f_2] = \mathrm{E}[\phi]$. And the reduction in the variance would be:

$$\frac{Var[f_2]}{Var[\phi]} = 1 - r^2_{\phi,\xi} \tag{35}$$

As an estimate for $\psi = \bar{\phi}$ take $\bar{f_1}$ or $\bar{f_2}$ accordingly to the one with greater r^2 factor.

4. Algorithmic Scheme

In order to find values for the decision variables $\mathbf{x}$ in (26) that provide a lower level objective function, a method based on the proximal point algorithm has been developed. This method requires a local model of the response surface of the upper level objective function in (26).

The method can be described as follows:

Fix two thresholds δ and $\hat{\delta}$ that bound the proximity of the current $\mathbf{x^k}$ iterate to the points in the set W_k available at iteration k-th in order to obtain an estimation of the function $\psi(\mathbf{x})$ locally:

$$\delta \le \| \mathbf{w}_\ell - \mathbf{x^k} \|_\mathbf{2} \le \hat{\delta}, \ \forall \, \mathbf{w}_\ell \in \mathbf{W_k}$$

Let us denote by $\hat{\psi}_k$ the estimate for $\psi(\mathbf{x}^k)$.

Let $\mathbf{x}^0$ be a feasible point, i.e. $\mathbf{x}^0 \in \mathcal{X}$, and let $W_0 = \{\mathbf{x}^0\}$. Set $k = 0$.

Then at iteration k-th:

(I) Generate a random direction $\mathbf{d^k}$ with $\|\mathbf{d^k}\|_{\mathbf{2}} = \hat{\delta}$.

Then, If $\mathbf{x}^k + \mathbf{d^k}$ is also feasible take $\mathbf{z^k} = \mathbf{x^k} + \mathbf{d^k}$.

If $\mathbf{x}^k + \mathbf{d^k}$ is not feasible solve:

$$\begin{array}{ll} Min_{\mathbf{y}} & \|\mathbf{d^k} - \mathbf{y}\|_{\mathbf{2}}{}^2 \\ & A(\mathbf{x}^k + \mathbf{y}) \geq \mathbf{b} \end{array} \tag{36}$$

- If $\mathbf{y} = \mathbf{0}$ then take $\mathbf{z^k} = \mathbf{x^k} - \alpha^* \mathbf{d^k}$, where

$$\alpha^* = Max\{\, 0 \leq \alpha \leq 1 \mid A(\mathbf{x}^k - \alpha \mathbf{d^k}) \geq \mathbf{0}\, \} \tag{37}$$

- If $\mathbf{y} \neq \mathbf{0}$, then take $\mathbf{z^k} = \mathbf{x^k} + \alpha^* \mathbf{d^k}$, where

$$\alpha^* = Max\{\, 0 \leq \alpha \leq 1 \mid A(\mathbf{x}^k + \alpha \mathbf{y}) \geq \mathbf{0}\, \} \tag{38}$$

- If $\|\mathbf{z^k} - \mathbf{x^k}\|_{\mathbf{2}} < \delta$ then repeat (I).

As an output of step (I) a new point $\mathbf{z^k}$ will be obtained so that $\delta \leq \|\mathbf{z^k} - \mathbf{x^k}\|_{\mathbf{2}} \leq \hat{\delta}$

(II) $W_{k+1} = W_k \bigcup \{\mathbf{z^k}\}$.

(III) Obtain an estimate $\hat{\psi}_k$ of $\psi(\mathbf{z^k})$.

(IV) Let now $W_k = \{\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_{m_k}}\}$ with $m_k = |W_k|$.
Let $W_k' = W_k \setminus \{\mathbf{x}^k\}$ and $\pi_\ell = \mathbf{w}_\ell - \mathbf{x^k}$, for $\mathbf{w}_\ell \in \mathbf{W_k}$

Two cases must be considered:

Case a) $m_k \leq n+1$ or case b) $m_k > n+1$.

If case a), then the restriction of the function ψ to the linear subspace $F_{W_k} \triangleq [\pi_1, \pi_2, \cdots, \pi_{m_k-1}]$ is optimized on that subspace:

$$\psi_k(\lambda) = \psi\left(\sum_{\ell=1}^{m_k-1} \lambda_\ell \pi_\ell + \mathbf{x^k}\right)$$

this function is modelled simply as:

$$\psi_k(\lambda) \simeq \hat{\psi}_0 + \sum_{\ell=1}^{m_k-1} \lambda_\ell(\hat{\psi}_\ell - \hat{\psi}_0) = \hat{\psi}_0 + \sum_{\ell=1}^{m_k-1} u_\ell \lambda_\ell$$

and

$$\mathbf{x}^{k+1} = \sum_{\ell=1}^{m_k-1} \pi_\ell \lambda_\ell^* + \mathbf{x^k}$$

$$\begin{array}{ll} Min_{\ \lambda} & \displaystyle\sum_{\ell=1}^{m_k-1} u_\ell \lambda_\ell + \frac{\rho}{2} \left\| \mathbf{x} - \mathbf{x}^k \right\|_2^{\ 2} \\ s.t. & A\mathbf{x} \geq \mathbf{b} \\ & \left(\mathbf{x} = \displaystyle\sum_{\ell=1}^{m_k-1} \pi_\ell \lambda_\ell + \mathbf{x}^k \right) \end{array} \quad \rightarrow \ \lambda^* \qquad (39)$$

If case b), then using the set of points in W_k a linear regression in order to model the function ψ is carried out. Then $\psi(\mathbf{x}) \simeq a_0 + \mathbf{a}^T\mathbf{x}$:

$$Min_{\ a_0, \mathbf{a}} \ \sum_{\ell=1}^{m_k} (\hat{\psi}_i - (a_0 + \mathbf{a}^T\mathbf{w}_i))^2 \rightarrow \hat{a}_0, \hat{\mathbf{a}}$$

The next iterate $\mathbf{x}^{k+1}$ would be directly provided by the solution of the proximal step (40) as:

$$\begin{array}{ll} Min_{\ x} & a_0 + \mathbf{a}^T\mathbf{x} + \dfrac{\rho}{2} \left\| \mathbf{x} - \mathbf{x}^k \right\|_2^{\ 2} \\ & A\mathbf{x} \geq \mathbf{b} \end{array} \quad \rightarrow \mathbf{x}^{k+1} \qquad (40)$$

(V) Drop from W_k any point $\mathbf{w}_\ell$ so that $\left\| \mathbf{w}_\ell - \mathbf{x}^{k+1} \right\|_2 > \hat{\delta}$, excluding $\mathbf{x}^k$.
Set $k \leftarrow k+1$ and return to step (I)

5. Software Architecture

The application has been coded in the Microsoft Visual C++ Environment.

RODOS Workspace is mainly governed by AMPL, who calls EM2AMPL Project and RSDCA Project (version 5.0), containing an additional MSA algorithm for SUE computation as the main difference from older versions and a modified input/output interaction pattern to allow calls from AMPL.

Directory definitions by means of precompilation directive in the header file RSDCA/GENERIC.H.

Directory tree from RODOS:

- RSDCA: Project to compute general TAP Equilibrium Assignment. Based on the variational inequality formulation.
- EM2AMPL: Converter Program that translates network and demand description from EMME/2 standard batchout/batchin format to AMPL model format.
- AMPL: Upper level AMPL program coding.
- CONTROL: Execution definition, defines network, demand and cost functional and parametrizes the execution of RSDCA and RODOS AMPL (constraints and stopping criteria). Extension files: *.ctl. Default file d111.dat.
- INPUTOUTPUT: Network, demand, turning movements in standard ASCII EMME/2 format. Cost functional is not EMME/2 compatible.
- VINCLES: Transfer files between AMPL and EM2AMPL program to initialize network and demand AMPL model characteristics and interaction files at each major iteration between the upper-level AMPL model and the lower-level RSDCA program for computing SUE.
- RESULTATS: Iteration and final results from upper-level AMPL program that implements a specific bilevel program with equilibrium constraints.

RSDCA VERSION 5.0: C++ Project that uses the following input-output directories: tranfer files between AMPL (high level) and RSDCA (Stoch TAP - lower level) are read/written at RODOS/VINCLES, the description of Stochastic TAP is read from RODOS/CONTROL, compatible with General TAP (non stochastic) and RODOS/INPUTOUTPUT is the directory containing network, demand and functional cost definitions according to control file for a given execution.

EM2AMPL: $C++$ Project that reads a control file from RODOS/CONTROL and according to that file reads network and demand description from
RODOS/INPUTOUTPUT and converts to AMPL model format, at directory RODOS/VINCLES with output file '*filename*'.DAT_AMPL_STRUC.

AMPL Results are writen to RODOS/RESULTATS and transfer files between RSDCA and AMPL are read-written at RODOS/VINCLES.

RODOS AMPL Program initially executes EM2AMPL Project that by default searches RODOS/CONTROL/d111.dat control file, where the names of the files containing the network, the demand matrix are read and the output file name for AMPL model (extension $.DAT\_AMPL\_STRUC$). EM2AMPL converts network and demand description in the AMPL model format that it is written to RODOS/VINCLES. Once the AMPL model file is loaded into RODOS AMPL Program. AMPL model consists on network, demand and global parameters to define upper-level constraint description and stopping criteria (*delta*, *gamma_ci*, *gamma_cs* and *ratio_SUE* parameters). The current implementation is consistent with control files where *Equilibrium Type is 3-SUE AMPL Call* and *SUE model type is 4* implies internal RSDCA link function type 6 - Probit with stdev fixed to link travel cost at capacity flow (meaning cost functional defined thru BPR type functions with standard deviation of perceived link travel cost parametrized by *coe_var* control parameter and $ul1$ link field in transfer file).

```
class ED_PARAM_SUE_MSA { public:
    int     model;
      // 1 - Probit (def) en sigma = coe_var x t(va)
      // 2 - Probit en sigma = coe_var x t0
      // 3 - Logit en sigma=coe_var x t(va)
      // 4 - Probit en sigma=coe_var x gamma_a x t(capacitat)
    int     nparam;
    double  coe_var;
      // coef. de variació en model 1 i escalat en model 2
    double  delta;
      // Només té sentit si param_TAP.model =3 (SUE és 4 i vdf =6 )
      //defineix cota inferior a suma variances
    double  ratio_SUE;
      // Només té sentit si param_TAP.model =3 (SUE és 4 i vdf =6 ) i
      // defineix el tant per u d'arcs amb C(v) aleatori
    double  gamma_ci, gamma_cs; // Cotes inf i sup a gamma_a

    ED_PARAM_SUE_MSA();
    ED_PARAM_SUE_MSA( const ED_PARAM_SUE_MSA & p );
    ED_PARAM_SUE_MSA& operator=( const ED_PARAM_SUE_MSA & p );
    ~ED_PARAM_SUE_MSA();
};
```

The first time that RSDCA program is called from AMPL (situation identified by TAP.model equals 3 and transfer file not existing), it generates randomly the links where a sensor is to be allocated, in such a way that $ratio\_SUE$ (read from control file) fraction is satisfied, while given to ul1 link field a feasible escale factor consistent with $ED\_PARAM\_SUE\_MSA.model$ equal 4 (the average from upper and lower scale constraints is given in the current implementation). This limits $gamma\_ci$ and $gamma\_cs$ are read from the control file. IMPORTANT: Original $ul1$ link network fields are lost and t0 link field is set to 0. It performs STEP 1 in MSA algorithm description. Results are written to transfer file RODOS/VINCLES/$*\_611.DAT$ ($.PIF$ ASCII format file). Random generator seed should be modified each RSDCA execution.

ASCII Transfer File Detailed Format (.pif format)

- First line: Title and brief description of execution object.
- Second line: Mean and Variance of System Equilibrium Objective Function, ($qua\_TAP.fobj$ and $qua\_TAP.gapfun$), MSA iterations ($qua\_TAP.niter$), convergence criteria based on moving averages, κ parameter ($qua\_TAP.gappcent$).
- Third line: $param\_SUE\_MSA.delta$, $param\_SUE\_MSA.ratio\_SUE$, $param\_SUE\_MSA.gamma\_ci$, $param\_SUE\_MSA.gamma\_cs$. Parameters that define upper-level objective function.
- Forward lines: As many lines as network links, each one containing: origin node, destination node, mean SUE link volume, upper-level variable and indicator of sensor allocation state.
- USE: RSDCA reads from fourth to the end of file lines. RODOS AMPL: reads all the lines. On the first call to RSDCA, transfer file should not exist.

ASCII Transfer File Detailed Format: For example for test problem t1.

```
  38220.65      38220.65    240          0.10
      1.00          1.00          0.05            2.00
  1    105      1343.10          1.02      1
  1    106      1656.90          1.02      1
105      3      1343.10          1.02      1
106      3      1656.90          1.02      1
```

ASCII Control File Detailed Format: For example for test problem t1.

```
c RSDVI Module:  User: E032/IOE_FIB...lmm  \\
c Project: Test Network 1 \\
t control init \\
t1_211.dat\\
t1_231.dat \\
t1_311.dat \\
t1_411.dat \\
t1_511.dat \\
t1_611.dat \\
SORTIDA.DAT \\
 0.1 500 0 1 0.00 3
 1 -0.1 2 3 1.0 0.0001
 500.0 0.95 0.75
 0
 4  1  1.0
 1.0 1.0 0.05 2.0

//  File format: 3 initial lines with comments
//  arxius.entrada_xarxa  (default D211.DAT)
//  arxius.entrada_girs (default D231.DAT)
//  arxius.entrada_matriu (default D311.DAT)
//  arxius.entrada_vdf (default D411.DAT)
//  arxius.entrada_interaccions (default D511.DAT)
//  arxius.entrada_pif (default <none>)
//  arxius.sortida_rsdvi (default SORTIDA.DAT)
//  1 Blank line and 6 forward lines containing:
//  1st Line: param_TAP.errorTAP param_TAP.max_iter
              param_TAP.ninf param_TAP.flag
              param_TAP.coef_assim param_TAP.model
//  2on Line: param_RSDVI.flaggap >> param_RSDVI.delta >>
              param_RSDVI.nverini >> param_RSDVI.maxvxtact  >>
              param_RSDVI.error_VIS >> param_RSDVI.min_coo
//  3rd Line: param_PPM.ini_corr >> param_PPM.factor_corr
//  4th Line: param_RSDCA.flag_aprox
//  if param_tap.model is 2 or 3(SUE) then
//    5th Line: param_SUE_MSA.model param_SUE_MSA.nparam
                param_SUE_MSA.coe_var
//  if param_tap.model is 3(SUE) then
//    6tha Line: param_SUE_MSA.delta param_SUE_MSA.ratio_SUE
                 param_SUE_MSA.gamma_ci,param_SUE_MSA.gamma_cs
```

6. Conclusions

Software architecture for solving the bilevel programming approach proposed for the authors to support the design of information points for monitoring the traffic conditions is fully designed and the Lower Level problem is actually solved with a method in the RSDCA environment. RSDCA environment contains a set of methods for solving separable and non-separable instances of the deterministic TAP, assuming a variational inequality formulation, and a MSA method for solving the separable and stochastic TAP version.

The interface between EMME/2 network description files and AMPL is implemented (E2AMPL C++ program) and the protocol for transferring information between the Upper Level (AMPL) and the Lower Level (RSDCA) has been coded and validated. The interface between EMME/2 network description files and RSDCA has been incorporated as input methods in the RSDCA environment and has been also validated.

Perceived link travel costs are assumed to be normal and unbiased variables, centered on the measured link travel costs with standard deviations proportional to the expected link travel costs at saturation flows, i.e., $\sigma_a = \beta x_a c_a(s_a)$. At this point, only separable instances of the stochastic traffic assignment problem can be considered, but the framework will be extended in a near future.

The AMPL upper level implementation is still pending. In preliminary tests on small networks a reduction on the System Performance function has been observed when the perceived link cost variance has been forced to decrease. Results currently available are very promising and show the consistency of the approach proposed by the authors.

7. Acknowledgement

Research Supported by Spanish CYCIT grant BFM2002-04548-C03-01.

8. Bibliography

[1] Beckmann, M. J. , McGuire, C. and Wisten,C. , (1956) Studies in the economics of Transportation.

[2] Daganzo, C. (1983) *" A stochastic Network Equilibrium with Multiple Vehicles Types and Asymmetric, Indefinite Link Cost Jacobians."* Transportation Science, Vol 17 No 2, August.

[3] Evgrafov, A., Patriksson, M. (2004) *On the existence of solutions to stochastic mathematical programs with equilibrium constraints* to appear in Journal of Optimization Theory and Applications.

[4] Florian, M. (1986) *Nonlinear cost network models in transportation analysis* Mathematical Programming Study, vol. 26, pp. 167–196.

[5] Luo, Z.Q., Pang, J.S. and Ralph, D. (1996). *Mathematical Programs with Equilibrium Constraints.*, Cambridge University Press.

[6] Marcotte, P. (1986) *Network Design problems with congestion effects: a case of bilevel programming*, Mathematical Programming 34, 142-162.

[7] Montero, L. and Barceló, J. (1996)“A simplicial decomposition algorithm for solving the variational inequality formulation of the general traffic assignment problem for large scale networks,” TOP, vol. 4, no. 2, pp. 225–256.

[8] Ortúzar, J.D., Willumsen, L.G. (2001) "Modelling Transport", John Wiley and Sons.

[9] Patriksson, M. (1994) ” The Traffic Assignment Problem: Models and Methods”, VSP BV, The Netherlands.

[10] Ross, S.M. (1997) ”Simulation”, Academic Press. Second Edition. Statistical Modeling and Decision Science Series. 1997

[11] Sheffi, Y. (1985) ”Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods”, Prentince Halls.

[12] Wardrop, J. G. (1952) *Some theoretical aspects of road traffic research* in Proceedings of the Institute of Civil Engineers, Part II, vol. 1, pp. 325–378.

[13] Wets, R.J.-B. (1989) ”Stochastic Programming”, Handbooks of Operations Research and Management Science, Volume 1, G.L.Nemhauser, Rinnooy and Todd eds. North-Holland, pp. 573-629.

[14] Zhang, R. (1994) *Problems of hierarchical optimization in finite dimension*, SIAM Journal on Optimization 4, 521-536.