A successive linear approximation approach to the design of congested urban traffic networks

Les R. Foulds
Hugo A. D. do Nascimento*
Humberto Longo
Bryon R. Hall†

Instituto de Informática – INF
Instituto de Matemática e Estatística – IME †
Universidade Federal de Goiás – UFG
Goiânia – GO, Brasil

{lesfoulds,hadn,longo}@inf.ufg.br
bryon@mat.ufg.br

Resumo

Este trabalho trata o problema de encontrar um conjunto de restrições de conversões e de direções de vias que incremente o fluxo de tráfego em uma rede urbana congestionada. É apresentado um método de aproximação linear sucessiva para a identificação de uma solução aproximada para um modelo não-linear do problema. O método visa ajustar o conjunto atual de restrições de conversões e direções de vias em uma determinada rede, a fim de minimizar o custo total de viagem do usuário quando a escolha de rota é dirigida pelo princípio de equilíbrio do usuário. Uma discussão sobre o modo como o método pode ser aplicado usando a geração de colunas para resolver problemas práticos eficientemente também é incluído.

Palavras-Chave: Problema de projeto de redes de tráfego urbano, Restrições de conversões, Direção de ligações, Aproximação sucessiva linear, Geração de colunas.

Área principal: L&T – Logística e Transportes.

Abstract

This paper is concerned with the problem of finding a set of turning restrictions and link directions to promote flow in a congested urban traffic network. We present a successive linear approximation method for identifying a heuristic solution to a nonlinear model of this problem. The method aims to adjust the current set of turning restrictions and link directions in a given network in order to minimise total user travel cost when route choice is driven by user equilibrium principles. A discussion of how the method can be applied using column generation to solve practical problems efficiently is included.

Keywords: Urban network design problem, Turning restrictions, Link directions, Successive linear approximation, Column generation.

Main area: L&T – Logistics and Transportation.

*Hugo do Nascimento is partially sponsored by a Scholarship of Research Productivity from CNPq (309463/2009-2).

1 Introduction

Recently, there has been increasing interest in the system control approach to improving the performance of congested traffic networks through operations research. In general, a common objective is to make investment decisions that ease congestion by reducing the total travel cost of all users while accounting for individual route choice behaviour. The *Urban Network Design Problem* (*UNDP*) involves finding improvement strategies comprising non-physical adjustments to characteristics of the network, such as link directions, turning restrictions, signal settings, parking prohibition and tolls. These strategies are usually lower in cost and easier to implement than strategies that involve physical interventions such as link addition or link improvement. The *UNDP* is a combinatorial optimisation problem that is very expensive to solve exactly and many of the articles in the literature concentrate on either approximate solutions to exact *UNDP* models or exact solutions to approximate models. The approach in the present article falls into the former case.

Two particular low-cost effective non-physical adjustment strategies for the *UNDP* are that of deciding on the direction of each link (street) of the network and which restrictions of maneuvers (turns) at intersections should be imposed on network users. Regarding link directions, it must be decided which links are to become two-way, one-way, or to have vehicle travel completely banned (and are thus transformed into pedestrian malls). Regarding turning restrictions, certain travel maneuvers (left turn, right turn or drive straight ahead or any other exit option for a non-standard intersection) may be prohibited at each particular intersection. The question is how to select link directions and which maneuvers at intersections to restrict (if any) in order to enhance a given system performance measure. The present article is concerned with only these two adjustment strategies of the *UNDP*.

We have constructed a *Sequential Linear Approximation* (*SLA*) method for this problem that starts with a given set of link directions and intersection maneuver restrictions specified, corresponding to the present situation in a given network. It aims to identify which additions or subtractions (allowing all practical possibilities) should be made in order to create the link direction and maneuver restriction regime that minimises user equilibrium-based total travel cost. The well-known *SLA* methodology (Palacios-Gomez et al., 1982, Bazaraa et al., 1993) has been used with success in the petroleum industry. More recently, Sherali et al. (2003) and Foulds et al. (2011, 2012) have used *SLA* to estimate origin-destination (O-D) travel demand matrices and to specify turning restrictions in traffic networks.

We also discuss how *Column Generation* (*CG*) (Nemhauser, 2012) can be applied within the proposed *SLA* method to solve practical problems efficiently. *CG* is an implicit pricing mechanism that was originally devised to solve Linear Programming (*LP*) problems with a huge number of variables compared to the number of constraints. It performs the Simplex method step of establishing optimality or finding a variable to enter the basis by generating columns (nonbasic variables) that correspond to unsatisfied constraints in the dual *LP*. This is done, not by enumeration, but rather by optimisation, with the columns being introduced only as needed. This approach has been embedded in branching schemes, resulting in branch-and-price algorithms that have been used to solve huge, difficult, Mixed Integer Programming (*MIP*) problems. This success for *MIP* models of transport problems in particular has come about mainly due to the development of very efficient dynamic programming (*DP*) algorithms for effective pricing, and also branching and cutting schemes that force integer solutions.

The main contributions of the present article are: (i) the presentation of a generalised single-level model of the *UNDP* specialised to link directions and maneuver restrictions at intersections; (ii) a demonstration of how *SLA* can be used to improve existing link direction and maneuver restriction regimes; and (iii) a discussion of how *CG* can be applied to solve practical problems efficiently. The remainder of this article is organised as follows. In the next section we survey the relevant literature and in Section 3 we formulate a specialised model for the *UNDP*. In Section 4 we develop a more detailed model and a solution algorithm that is based on successive linear approximation. In Section 5 we discuss how to use column generation to efficiently solve numerical problems of practical size. We draw some conclusions and present suggestions for future work in Section 6.

2 Literature Review

The earlier articles in the literature on the *UNDP* are concentrated mainly on exact optimisation strategies and the later ones on meta-heuristics. Foulds (1981) was one of the first to develop a branch and bound approach for the special case of the *UNDP* where the only improvement strategy available is to specify link directions. More recently, Gallo et al. (2010) built on the approach of Cantarella and Vitetta (2006) by using stochastic, rather than deterministic assignment, to solve a particular *UNDP* optimisation model using a meta-heuristic technique. The model has the objective of specifying link directions and traffic signal settings at intersections. A non-linear constrained optimisation model for solving this problem was formulated which adopts a bi-level approach in order to reduce the complexity of solution methods and the computation time. A scatter search algorithm (see Laguna, 2002; and Martí et al., 2006) based on a random descent method was proposed and tested on a practical network. Initial results showed that the proposed approach allows local optimal solutions to be obtained in reasonable computation time.

The advantages of making restrictions on maneuvers at intersections have been discussed by Chen and Luo (2006). Long et al. (2010) have introduced and defined the *Turning Restriction Design Problem* (*TRDP*) as a special case of the *UNDP*. The *TRDP* involves determining the optimal set of turning restrictions to be imposed in order to minimise the total user equilibrium-based travel cost. The authors developed a bi-level model of the *TRDP* in which at the lower level, a path choice set generation method is applied to establish stochastic user-equilibrium flows. At the upper level, the resulting flows are used within a sensitivity algorithm-based branch and bound scheme to solve a relaxed version of the *TRDP* model. The method progressively identifies a set of restrictions that are selected from a relatively limited subset of all possibilities, until some termination criterion is met. Only some so-called "crucial" intersections can be considered for restrictions and only left-turn restrictions are allowed. The method is based on a nonlinear *MIP* model. Foulds et al. (2012) presented a bi-level *SLA* algorithm for identifying a heuristic solution to a nonlinear model of the *TRDP* that is based on link capacity adjustment. The algorithm compares favourably with the method of Long et al. (2010) when comparative numerical tests are made on standard network examples from the literature. The present article expands on the previously mentioned algorithms by including link directions (as well as all feasible maneuver restrictions). A single level method based on link cost adjustment is proposed.

3 Developing a *UNDP* model

We now explain a single level method for the *UNDP* that is based on a model to be solved by column generation within an *SLA* scheme. The reader is referred to Foulds (1991) for the necessary graph theory notation and terminology. We shall construct a digraph $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ based on a transformation due to Potts and Oliver (1972) that models a given traffic network with node set $\mathcal{N}$ and arc set $\mathcal{A}$. Each node in $\mathcal{N}$ represents either the beginning or the end of a street, usually an entrance or departure point of an intersection, and each arc in $\mathcal{A}$

represents either a street travel direction or a feasible maneuver at an intersection. Multiple arcs connecting the same pair of nodes are not permitted. The digraph construction process with a particular example has been presented by Foulds et al. (2012), where a particular intersection of two two-way streets is shown along with the digraph $\mathcal{D}$ that results from the application of the process.

Note that each intersection of the original network is not represented by a unique node in $\mathcal{D}$, for that would not permit inclusion of alternative connections between nodes. Rather, each intersection of two streets is represented by a set of eight nodes, connected by twelve arcs that represent all possible alternatives of feasible maneuvers. Streets that may be one-way or two-way are represented by appropriate arcs. Certain potential maneuvers and link directions can never be allowed for reasons of safety or physical considerations. Arcs representing such maneuvers and link directions are never created in $\mathcal{D}$ and play no further part in the discussion. Conversely, we establish which maneuvers or link directions are permanently allowed and flow is always possible in them. Let $\mathcal{A}' = \mathcal{A}_P^T \cup \mathcal{A}_P^L$ denote the set of such corresponding arcs, where $\mathcal{A}_P^T$ is the set of arcs representing links with permanent maneuvers and $\mathcal{A}_P^L$ is the set of arcs representing links with permanent link directions. We also establish which maneuvers and link directions are presently allowed but flow in them could be prohibited. Let $\mathcal{A}_I^T$ denote the set of such corresponding arcs whose maneuvers could be restricted and $\mathcal{A}_I^L$ denote the set of such corresponding arcs for which link direction could be restricted. Finally, we establish which maneuvers and link directions are presently restricted but the restriction could be lifted. That is, flow in them is at present prohibited, but could possibly be allowed. Let $\mathcal{A}_R^T$ denote the set of arcs for which the current maneuver restriction could be lifted and $\mathcal{A}_R^L$ denote the set of arcs for which the current link direction restriction could be lifted. The procedure is repeated analogously for all the intersections irrespective of how many incident streets there are. $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ is the resulting digraph, where $\mathcal{A} = \mathcal{A}' \cup \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L$. Note that $\mathcal{A}'$ comprises arcs whose status cannot be changed, whereas $\mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L$ comprises arcs representing the link direction and maneuver restriction possibilities whose status can be changed.

Let the travel cost of link a be denoted by $t_a(\cdot)$, $\forall a \in \mathcal{A}$. The following popular separable travel cost function for any link $a \in \mathcal{A}$ in a regional network has been provided by the USA Bureau of Public Roads (1964):

$$t_a(f_a) = t_a^F \cdot \left[1 + \theta \cdot \left(\frac{f_a}{u_a}\right)^{\gamma}\right], \tag{1}$$

where f_a is the flow, t_a^F is the congestion-free travel cost, u_a is the effective capacity and θ and γ are parameters that must be calibrated according to the actual network being studied. The values $\theta = 0.15$ and $\gamma = 4.00$ have been used in the United States and $\theta = 2.62$ and $\gamma = 5.00$ in Holland and Japan (Steenbrink, 1974). The separable link cost function (1) has been generalised by Horowitz (1997) to allow for nonseparable link costs (where the cost of each link is not based only on flow in the link). This nonseparable cost function has been further generalised by Long et al. (2010) to capture the effects of opposing and turning flows on link costs in urban networks. We have made yet a further generalisation of the cost function and have also included the effects of link direction specification on link costs. The actual functions used in the proposed method are given later in (7), (8) and (9).

Alternatively, software is available that calculates non-separable costs for each link. As an example, PETGYN (Jradi et al., 2009) is software that models urban traffic flow in Brazil, taking into account the characteristics of the common urban traffic structure existing in many developing countries. With PETGYN, the non-separable cost function for each link depends not only on its own flow, but also on its traffic signal settings, on local physical structure such as the number of lanes and on flow in neighbouring links.

The purpose of the version of the *UNDP* studied here is to identify which intersection maneuver restrictions and link directions should be imposed when traffic flow is based on user equilibrium principles. The decision to restrict a particular maneuver at a certain intersection or to specify a certain link direction can be represented by a binary variable. Furthermore, it is often desirable in practice to introduce a budgetary consideration because changing the network design comes at a cost. To this end, it is assumed that for each design adjustment, a known cost is incurred. The objective is to minimise the sum of the total (user equilibrium) travel cost, subject to a budget on the total cost of all adjustments in status of the maneuvers and link directions.

We now turn to selecting a theme for modelling the *UNDP*. The most common models of problems of this type are either link-based or route-based. Link-based models are sometimes used because they obviate the need to enumerate explicitly all the routes than could possibly be chosen by users of the network. The number of such routes grows exponentially in network size. Link-based models avoid this difficulty by adopting instead only an implicit set of route choices, such as the so-called "efficient" routes. The problem is that the resulting traffic assignment often does not correspond even remotely to actual user equilibrium behaviour (Bekhor and Toledo, 2005).

For this reason, although there may be computational difficulties, we construct route-based models that usually lead to more realistic assignments. Also, route-based models are more naturally aligned with Wardrop's First Principle (Wardrop, 1952) and they can be used to provide route flow information which is essential for network design adjustment. Of course, link flow information can be easily deduced from route-based information.

The *UNDP* can be formulated as the following optimisation model:

$$\underset{y\in\mathcal{Y}}{\text{Minimise}}\ v(y,x) \tag{2}$$

subject to

$$x \text{ are user equilibrium route flows,} \tag{3}$$

where $\mathcal{Y}$ is the set of all feasible network design combinations, y is the binary vector representing a specific combination, x is the matrix of O-D route flows (which is an implicit function of y) and $v(y,x)$ is the total user cost when the design decisions are specified by y and the route flows are specified by x.

Note that x can be obtained by an assignment procedure. If y is given and the link cost functions are continuous, twice differential and increasing then the user equilibrium route flows $x(y)$ exist uniquely (Cascetta et al., 2006). Thus, once the network is fixed by specifying y there is just one equilibrium route flow matrix x. Therefore, the *UNDP* involves finding the binary values of a feasible y that minimises the total user cost represented by (2). Because the route flow matrix x is a unique outcome once y is fixed, x may appear only in the objective function (2), as shown below:

$$\underset{y\in\mathcal{Y}}{\text{Minimise}}\ v(y,x) \tag{4}$$

The interplay between the route flow matrix x and the decisions vector y can be viewed as a game between two players: the network manager and the network users. The manager attempts to minimize the objective (4) over the complete network whereas the users attempt to minimize their individual route costs (Fisk, 1984). As the objective is to identify the constrained minimum of (4) corresponding to y^* and x^*, the model (4) is, in terms of game theory, a Stackelberg game where the leader-player is the manager and the follower-player is the population of users. Even though the leader (manager) may know how the follower (users) may respond to the leader's interventions, the leader cannot directly influence the follower's responses.

The model (4) is a nonlinear mixed binary optimisation problem subject to user equilibrium flow. To reduce the computational burden in solving the model, a linear approximation of the objective function (4) is proposed, as discussed in the next section. We now discuss the model (4) in more detail. The decision variables are denoted by y_a, $\forall a \in \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L$, where y_a is set to unity if flow is allowed in a and to zero otherwise. We begin by discussing the possible maneuvers at intersections and their restrictions. The discussion is based on the premise that the alteration in the flow restriction status of a particular maneuver may affect the flow in other network arcs. For all arcs $a \in \mathcal{A}$ and arcs $b \in \mathcal{A}_I^T \cup \mathcal{A}_R^T$, let s_a^b denote the fraction of the change in f_a when the flow restriction status of b is changed. That is, the quantity $s_a^b \cdot f_b$ is added to f_a whenever either (i) $b \in \mathcal{A}_I^T$ and flow in b is prohibited (with y_b set to zero) or (ii) $b \in \mathcal{A}_R^T$ and flow in b is allowed (with y_b set to unity). Note that s_a^b can be positive or negative, depending upon the relationship between a and b. Furthermore, let $\mathcal{E}_a = \{b \mid b \in \mathcal{A}_I^T \cup \mathcal{A}_R^T,\ s_a^b \neq 0\}$. We introduce a record keeping parameter δ_b, where δ_b is set to unity if arc b is currently a member of $\mathcal{A}_I^T$ and to zero if b is in $\mathcal{A}_R^T$.

We next discuss the possible link directions and their restrictions. Links often occur in oppositely directed pairs, denoted by $(a, \overline{a})$, where $a, \overline{a} \in \mathcal{A}' \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L$. The proposed model deals with the case where both $a, \overline{a} \in \mathcal{A}$, and because $a \in \mathcal{A}' \cup \mathcal{A}_I^L$ (flow is possible in a), and $\overline{a} \in \mathcal{A}_R^L$ (flow is prohibited in $\overline{a}$), consequently the capacity u_a can be increased by $u_{\overline{a}}$. Additional constraints could be added to any proposed models as cuts to reflect cases where (i) flow restrictions make flow in ongoing arcs infeasible, and (ii) flow is permitted in at most one of a and $\overline{a}$. As will be seen in the nonseparable cost functions developed later as part of the proposed method, link costs are affected by adjustments to intersection maneuvers and link directions.

Let c^I (c^R) denote the cost of changing the flow restriction status of any arc in $\mathcal{A}_I^T \cup \mathcal{A}_I^L$ (in $\mathcal{A}_R^T \cup \mathcal{A}_R^L$), respectively. Let $\mathcal{OD}$ denote the set of origin-destination node pairs that define the rows and columns of a given trip table matrix $T = (T_{ij})$, where T_{ij} is the travel demand from origin node i to destination node j. For all $(i, j) \in \mathcal{OD}$, let n_{ij} be the number of routes from node i to node j and let p_{ij}^k denote the k^{th} such route, for $k = 1, 2, \ldots, n_{ij}$. The route p_{ij}^k is represented by a binary vector of elements $(p_{ij}^k)_a$, corresponding to the links $a \in \mathcal{A}$, where $(p_{ij}^k)_a$ is unity if link a belongs to p_{ij}^k and is zero otherwise. Let x_{ij}^k be the (integral) number of users of p_{ij}^k.

Let $t_{ij}^k(y, x^*)$ be the cost of the k^{th} route from node i to node j for design decision $y = \{y_a \mid a \in \mathcal{A}_I^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^T \cup \mathcal{A}_R^L\}$ and the corresponding user equilibrium flow pattern x^*. A constrained objective function that can be used in models that produce solutions tending towards user equilibrium flow principles and takes design adjustment costs in account is:

$$\text{Minimise } v(y, x) = \sum_{(i,j) \in \mathcal{OD}} \sum_{k=1}^{n_{ij}} t_{ij}^k(y, x^*) \cdot x_{ij}^k, \tag{5}$$

subject to

$$\sum_{a \in \mathcal{A}_I^T \cup \mathcal{A}_I^L} c^I \cdot (1 - y_a) + \sum_{a \in \mathcal{A}_R^T \cup \mathcal{A}_R^L} c^R \cdot y_a \leq B, \tag{6}$$

where B is a budget representing the maximum total cost that can be spent on changing the flow restriction status of all arcs in $\mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L$.

4 An algorithmic solution

We propose a single level iterative solution scheme that is described in Algorithm SLATLIDD (***S**ucessive **L**inear **A**pproximation **T**urning and **L**ink **D**irection **D**esign*) given later in pseudocode. It is assumed that the flow pattern in the traffic network under study is in user equilibrium. That is, according to Wardrop's First Principle (Wardrop, 1952), all routes actually used between any origin-destination pair of nodes should have close to equal travel costs and this cost must not exceed the cost of any unused route between this pair. Initially, a traffic assignment process is applied to establish used routes and a user-equilibrium flow pattern in the given network. Alternatively, link counts (traffic flow volumes) can be used instead, if they are available for sufficiently many of the links. In this case, the initial flows of links without counts are set to zero. The resulting link flows and link travel costs are then used to calculate the relevant route costs $t_{ij}(\overline{y}, x^*)$, using (17) and (18) below, where $\overline{y} = \{y_a = 1 \mid a \in \mathcal{A}_I^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^T \cup \mathcal{A}_R^L\}$, that are temporarily fixed. This creates an initial numerical *UNDP* problem.

In general, the link flows $\boldsymbol{f}$ that result from solving (4) at each iteration are used to establish route costs for the next iteration as follows:

$$t_a(\boldsymbol{f}, \boldsymbol{y}) = \begin{cases} t_a^F[1 + \theta \cdot ((f_a + \sum_{b \in \mathcal{E}_a} (s_a^b \cdot (y_b - \delta_b) \cdot f_b)/u_a)^\gamma], & \forall\, a \in \mathcal{A}_P^T \cup \mathcal{A}_I^T \cup \mathcal{A}_R^T; \quad (7) \\ t_a^F[1 + \theta \cdot ((f_a/(u_a + (1 - y_{\overline{a}}) \cdot u_{\overline{a}}))^\gamma], & \forall\, a \in \mathcal{A}_P^L \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L,\ \forall\, \overline{a} \in \mathcal{A}_R^L. \quad (8) \\ t_a^F[1 + \theta \cdot (f_a/u_a)^\gamma], & \text{otherwise}, \quad (9) \end{cases}$$

where parameters θ and γ have been introduced in (1). Recall that y_a is defined only for arcs for which design decisions can be made. That is, for arcs in $\mathcal{A}_I^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^T \cup \mathcal{A}_R^L$. Thus y_a is not defined for $a \in \mathcal{A}_P^T \cup \mathcal{A}_P^L$.

The necessary route costs are modified and then held constant in the current iteration. This provides the objective function coefficients of a linear *Integer Problem* (*IP*). A relaxed version of this *IP* is solved to identify a promising regime of design decisions. Updated route costs are inputted to establish a new design regime at the next iteration. The r^{th} iteration is based on the current link flows $\boldsymbol{f}^r$ together with link costs $t_a(\boldsymbol{f}^r)$, $\forall\, a \in \mathcal{A}$, found by using $\boldsymbol{f}^r$ in some suitable processes that permits link costs to be nonseparable. Let $\boldsymbol{t}^r$ denote the vector of link costs $t_a(\boldsymbol{f}^r)$, $\forall\, a \in \mathcal{A}$.

Let c_{ij}^k denote the cost of p_{ij}^k, which is calculated by using (7), (8) and (9). Let $c_{ij}^* = \min\{c_{ij}^k \mid k = 1, \ldots, n_{ij}\}$, $\mathcal{K}_{ij} = \{k \mid k \in \{1, \ldots, n_{ij}\},\ c_{ij}^k = c_{ij}^*\}$ and $\mathcal{K}'_{ij} = \{1, \ldots, n_{ij}\} \backslash \mathcal{K}_{ij}$. At the r^{th} iteration the cost of the k^{th} (i, j) route is denoted by $(C_{ij}^k)^r$. It is calculated as the cost of the shortest (i, j) route and is multiplied by a large constant M_2 only if $k \in \mathcal{K}'_{ij}$. The purpose of $(C_{ij}^k)^r$ is to guide the algorithm towards user equilibrium.[1] Using these ideas leads to the following model that represents the r^{th} iteration of an iterative process for identifying an adjusted link direction and turning restriction design scheme that tends towards user equilibrium flows.

Model $\boldsymbol{SLA(TLD)^r}$:

$$\text{Minimise} \quad \sum_{(i,j) \in \mathcal{OD}} \sum_{k=1}^{n_{ij}} (C_{ij}^k)^r \cdot x_{ij}^k, \tag{10}$$

subject to

[1]This approach was introduced by Sherali et al. (2003) as a part of an *SLA* O-D matrix estimation process.

$$\sum_{k=1}^{n_{ij}} x_{ij}^k = T_{ij}, \qquad \forall\,(i,j) \in \mathcal{OD}, \tag{11}$$

$$\sum_{a\in\mathcal{A}_I^T\cup\mathcal{A}_I^L} c^I \cdot (1-y_a) + \sum_{a\in\mathcal{A}_R^T\cup\mathcal{A}_R^L} c^R \cdot y_a \leq B, \tag{12}$$

$$\sum_{(i,j)\in\mathcal{OD}} \sum_{k=1}^{n_{ij}} (p_{ij}^k)_a \cdot x_{ij}^k \leq M_1 \cdot y_a, \quad \forall\, a \in \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L, \tag{13}$$

$$x_{ij}^k \in \mathbb{Z}^+, \qquad \forall\,(i,j) \in \mathcal{OD},\ k = 1,\ldots,n_{ij}, \tag{14}$$

$$y_a \in \{0,1\}, \qquad \forall\, a \in \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L, \tag{15}$$

where

$$M_1 = \sum_{(i,j)\in\mathcal{OD}} \sum_{k=1}^{n_{ij}} T_{ij}, \tag{16}$$

$$(C_{ij}^k)^r = \begin{cases} (c_{ij}^*)^r, \quad \forall\,(i,j) \in \mathcal{OD}, \quad \forall\, k \in (\mathcal{K}_{ij})^r, & (17)\\ M_2 \cdot (c_{ij}^*)^r, \quad \forall\,(i,j) \in \mathcal{OD}, \quad \forall\, k \in (\mathcal{K}'_{ij})^r, & (18) \end{cases}$$

M_2 is a suitably chosen positive real number and $(c_{ij}^*)^r$, $(\mathcal{K}_{ij})^r$ and $(\mathcal{K}'_{ij})^r$ are the versions of c_{ij}^*, $\mathcal{K}_{ij}$ and $\mathcal{K}'_{ij}$ at the r^{th} iteration. The function (10) represents the objective of minimising the user equilibrium assignment cost. Relationship (11) is constraint for all travel demand and (12) is the arc status change budget constraint. Next, (13) prevents travel in any arc with a flow restriction. Finally, (14) and (15) are the usual integer and binary conditions.

Consider the particular case where a user equilibrium assignment can be found that is a feasible solution to (10)–(15). Next, consider the model $SLA(TLD)^r$ at the r^{th} iteration, whose objective function coefficients have been calculated by substituting the link flows of this assignment into (17) and (18). Then an optimal solution to $SLA(TLD)^r$ can be found that is a user equilibrium solution.

It should be observed that the above model is not designed to be the basis of an algorithm to be used to solve the nonlinear model (5), (11)-(15) obtained by using nonlinear link cost functions t_{ij}^k. Instead, the model has been constructed to identify a collection of O-D routes whose costs can be substituted into (17) and (18) to compute a set of temporarily constant objective function coefficients. These coefficients are substituted into (10) to create a *linear* integer model. The optimal solution to this model is not necessarily optimal for the nonlinear model and thus, like many *UNDP* approaches, the proposed solution technique is an approximating, iterative, heuristic procedure.

In general, as with many previous methods, including Sherali et al. (2003), we avoid solving the difficult *IP* problem (10)-(15) by linear relaxation. The procedure solves the linear model $SLA(TLD)^r$ with (14) and (15) relaxed, with revised link flows f_a^r, the link travel costs $t_a(\boldsymbol{f}^r, \boldsymbol{y})$ and the revised route costs $(C_{ij}^k)^r$ calculated using (17) and (18). The column generation technique can be used to generate O-D routes as they are needed as part of a Simplex method step through the implicit pricing of nonbasic variables to generate new columns or to prove *LP* optimality. This is discussed later.

When a stopping criterion is satisfied the procedure is terminated and the latest computed solution is outputted. Several types of stopping criteria can be defined for the method. For instance, it can be established whether or not the link flows or the total user equilibrium cost of the current solution did not change significantly after a number of iterations, or whether a given number of iterations have been executed, the number being based on the network size. Whichever stopping criteria are collectively adopted, the important idea here is to provide sufficient iterations to enable the link flows and their associated route travel costs to converge towards the user-equilibrium assignment. If no termination criterion is met, updated link capacities that are part of the solution to $SLA(TLD)^r$, the outputted link flows, link costs and route costs $(C_{ij}^k)^{r+1}$ are used to define model $SLA(TLD)^{r+1}$ which is then formulated and solved.

Instead of simply replacing the link flow values with the new values at each iteration, we have found it more effective to update them with an exponentially smoothed value of the previous values. This approach, also applied by Sherali et al. (2003) and by Foulds et al. (2011, 2012), promotes a better transition of the link flows from one iteration to another, avoiding drastic jumps back in forth in flow values.

Algorithm SLATLIDD:

input : $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, $\mathcal{OD}$, $\mathcal{A}'$, $\mathcal{A}_I^T$, $\mathcal{A}_R^T$, $\mathcal{A}_I^L$ and $\mathcal{A}_R^L$.
output: a set of turning restrictions and link directions.

`// Pre-Processing: steps 1-2.`

1 $r \leftarrow 1$.
2 Use a traffic assignment algorithm or given link counts to identify an initial link flow pattern for $\mathcal{D}$. Use this pattern to calculate initial link flows f_a^r, $\forall\, a \in \mathcal{A}$.

`// Main process: steps 3-9.`

3 Calculate the link travel costs $t_a(\boldsymbol{f}^r, \boldsymbol{y})$, $a \in \mathcal{A}$, using (7), (8) and (9).
4 Use (17) and (18) to calculate the new cost of each relevant O-D route.
5 Solve a relaxation of $SLA(TLD)^r$ by column generation.
6 Establish if a stopping criterion has been met.
7 If so, output the current set $\boldsymbol{y}$ of turning restrictions and link directions as the solution and terminate the process.
8 If not, set $r \leftarrow r + 1$. Obtain $\boldsymbol{f}^r$ as a smoothed combination of $\boldsymbol{f}^{r-1}$ and the flows identified in Step 5.
9 Go to step 3.

5 Efficient solution of problems of practical size

A numerical instance of the model $SLA(TLD)^r$ must be solved at the r^{th} iteration (step 5) of the algorithm SLATLIDD. This can be problematic, as an explicit statement and processing of such an *IP* problem (with integer x variables and binary y variables) requires the enumeration of all possible routes between each O-D pair, which is computationally infeasible for problems of practical size. Indeed, for such problems, there is a huge number of such routes compared to the number of constraints. Indeed, in some practical situations it is quite difficult to even state all the route variables in any *IP* model that can be solved by any conventional method in reasonable time. For this reason, an efficient approach that avoids explicit route enumeration is required.

What is done in *CG* is to begin with a so-called *restricted master problem* (*RMP*). The attractive concept of *CG* is to create an *RMP* that has only a relatively small subset of the x_{ij}^k's but that is sufficiently large to be meaningful. Additional x_{ij}^k's are included only as required. At Step 5 of SLATLIDD, subproblems denoted by SP_{ij}^r and $\overline{SP}_{ij}^r$, for each (i, j) pair in $\mathcal{OD}$, are used to price the extreme route flows not yet identified that could profitably enter the basis $\boldsymbol{B}$ say, in the *RMP*.

Let $\mathcal{R}_{ij} = \{(x_{ij})^1, (x_{ij})^2, \ldots, (x_{ij})^p\}$, for each OD pair $(i,j) \in \mathcal{OD}$, be a finite set of p points of the limited convex domain $\{x_{ij} \in \mathbb{Z}_+^{n_{ij}} \mid \sum_{k=1}^{n_{ij}} x_{ij}^k = T_{ij}\}$ and let Q_{ij} be an $n_{ij} \times p$ matrix, such that each column of Q_{ij} represents an element of the set $\mathcal{R}_{ij}$. All the other points of this domain can be represented as a convex linear combination of the p points as $\mathcal{X}_{ij} = \{Q_{ij} \cdot \lambda_{ij} \,|\, \mathbf{1} \cdot \lambda_{ij} = 1,\, \lambda_{ij} \in \mathbb{Z}_+^p\}$. Substituting the x_{ij} variables in model $SLA(TLD)^r$ by their equivalent λ expressions given by $\mathcal{X}_{ij}$, leads to a *master problem.*

Due to the large number of λ_{ij} variables in the master problem it is often impractical to solve this problem directly. Consequently, the RMP created here has only a subset Q'_{ij} of the columns of the corresponding Q_{ij} for each O-D pair $(i,j) \in \mathcal{OD}$. The relaxed version of RMP at the r^{th} iteration is denoted by $(RMP)^r$, which is stated as:

The Restricted Master Problem (RMP)^r:

$$\text{Minimise} \sum_{(i,j)\in\mathcal{OD}} \sum_{k\in Q'_{ij}} (C_{ij}^r)^T \cdot \lambda_{ij}, \tag{19}$$

subject to

$$\sum_{a\in\mathcal{A}_I^T\cup\mathcal{A}_I^L} c^I \cdot (1-y_a) + \sum_{a\in\mathcal{A}_R^T\cup\mathcal{A}_R^L} c^R \cdot y_a \le B, \tag{20}$$

$$\sum_{(i,j)\in\mathcal{OD}} \sum_{k\in Q'_{ij}} (P_{ij}^T)_a \cdot \lambda_{ij} \le M_1 \cdot y_a, \quad \forall\, a \in \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L, \tag{21}$$

$$\sum_{k\in Q'_{ij}} \lambda_{ij} = 1, \qquad \forall\,(i,j) \in \mathcal{OD}, \tag{22}$$

$$0 \le \lambda_{ij} \le 1, \qquad \forall\,(i,j) \in \mathcal{OD}, \tag{23}$$

$$0 \le y_a \le 1, \qquad \forall\, a \in \mathcal{A}_I^T \cup \mathcal{A}_R^T \cup \mathcal{A}_I^L \cup \mathcal{A}_R^L. \tag{24}$$

Here Q'_{ij} denotes some subset of all the feasible (i,j) routes (obeying (20) and (21)), including those generated by solving the pricing problem so far. The purpose of CG here is to find a minimum cost set of routes for all O-D pairs that complement the design decision $\boldsymbol{y}$ associated with the minimum possible total user equilibrium cost. To achieve this we must invoke a CG pricing process at each step of the Simplex method when solving $(RMP)^r$ in step 5 of SLATLIDD.

In (11) the variables represent route flows that can be thought of as "commodities", one for each O-D pair and the flows here are integral. The number of routes and hence the number of x_{ij}^k variables, $n = \sum_{(i,j)\in\mathcal{OD}} n_{ij}$ say, grows exponentially in the size of D. Even so, at each Simplex method step the least reduced cost for each O-D pair can be determined by solving a shortest path problem (Ford and Fulkerson, 1958). For each O-D pair, the extreme points of (11) are the simple routes (with no node repetition) for that pair. Thus, feasible solutions to $(RMP)^r$ are convex combinations of these extreme route flows that also satisfy the constraints (12), (13) and (15).

The pricing subproblems for each $(i,j) \in \mathcal{OD}$ are to find the least reduced cost as:

$$SP_{ij}^r : \underset{k\in\mathcal{K}_{ij}^r}{\text{Minimise}}\ (M_3 \cdot \boldsymbol{t}^r - \boldsymbol{\pi}) \cdot p_{ij}^k - \alpha_{ij} - \beta. \tag{25}$$

$$\overline{SP}_{ij}^r : \underset{k\in(\mathcal{K}_{ij}^r)'}{\text{Minimise}}\ (M_2 \cdot \boldsymbol{t}^r - \boldsymbol{\pi}) \cdot p_{ij}^k - \alpha_{ij} - \beta. \tag{26}$$

where M_2 is introduced in (18), $M_3 = \sum_{a\in\mathcal{A}} |\pi_a| + 1$ and β, $\boldsymbol{\pi}$ and α_{ij} are the dual values for (20), (21) and (22), respectively.

The subproblems (25) and (26) are essentially *simple* path problems in the sense that no node repetition is allowed. Note that the elements of $(M_2 \cdot \boldsymbol{t}^r - \boldsymbol{\pi})$ may be negative in sign as they correspond to the arcs of a k^{th} route in $\mathcal{D}$ when $k \in (\mathcal{K}_{ij}^r)'$. Thus the arc costs may be positive or negative. Hence a shortest path algorithm that can detect and prevent negative cost cycles is needed. The Bellman-Ford algorithm (Bellman, 1958) can be used to detect whether negative cycles exist in the network $\mathcal{D}$. If any negative cycles are detected, all negative-cost arcs are assigned a very small positive value. Alternative modifications to $\mathcal{D}$ to deal with negative costs cycles have been provided by Sherali et al. (1994). After $\mathcal{D}$ has been modified to account for negative cycles if necessary, the Floyd-Warshall algorithm (Floyd, 1962) can be applied to find the shortest routes between all O-D pairs. Or more efficiently, the shortest routes from any i to all j can be found in $\mathcal{O}(\mathcal{N}^2)$ time using the NETFLOW suite of Kennington and Helgason (1980). Equally efficient methods can be found in Bazaraa et al. (1990).

We can employ the above ideas to implicitly price the reduced costs of the nonbasic route flow variables x_{ij}^k. If at least one reduced cost is negative, we can then establish which route to add to the basis, being the one with the lowest reduced cost value, over all O-D pairs, creating a new column in $\boldsymbol{B}$. Then we must determine the flow on the entering route and simultaneously establish which variable must leave $\boldsymbol{B}$. The leaving variable is the one that maximises the flow on the incoming route $(x_{ij}^k)^*$ say, while maintaining current solution feasibility (keeping all basic route flows nonnegative). Suppose the shortest route of SP_{ij}^r is $(p_{ij}^k)^*$ with $k \in K_{ij}^r$ and let $h_{ij} = (M_3 \cdot \boldsymbol{t}^r - \boldsymbol{\pi}) \cdot (p_{ij}^k)^* - \alpha_{ij} - \beta$. If $h_{ij} < 0$, $(x_{ij}^k)^*$ has a reduced cost of h_{ij} and is entered into basis $\boldsymbol{B}$. Otherwise, no route in K_{ij}^r can be entered into $\boldsymbol{B}$. In this case, suppose the shortest route of $\overline{SP}_{ij}^r$ is $(p_{ij}^k)^\dagger$ with $k \in (K_{ij}^r)'$ and let $\overline{h}_{ij} = (M_2 \cdot \boldsymbol{t}^r - \boldsymbol{\pi}) \cdot (p_{ij}^k)^\dagger - \alpha_{ij} - \beta$. If $\overline{h}_{ij} < 0$, $(x_{ij}^k)^*$ has a reduced cost of $\overline{h}_{ij}$ and is entered into $\boldsymbol{B}$. Otherwise, the current solution is optimal for $\boldsymbol{t}^r$ and a check is made to see if any stopping condition is met.

6 Conclusions and summary

We have presented a successive linear approximation method, termed SLATLIDD, for identifying a heuristic solution to a nonlinear model of the *UNDP*. The method aims to adjust the current intersection maneuver restriction and link direction regime in a given network in order to minimise the total cost when user route choice is driven by user equilibrium principles. It has been demonstrated how the method can be applied to solve efficiently numerical problems of practical size using column generation. The authors are in the process of refining the SLATLIDD, conducting numerical experiments on large-scale Brazilian city networks and investigating the convergence properties of the method.

References

Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (1990). *Linear Programming and Network Flows*. John Wiley and Sons, New York, NY, USA.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (1993). *Nonlinear Programming, Theory and Applications*. John Wiley & Sons, New York, 2nd edition.

Bekhor, S. and Toledo, B. (2005). Investigating path-based algorithms for the stochastic user equilibrium problem. *Transport. Res. A-Pol.*, 39(3):279–295.

Bellman, R. E. (1958). On a routing problem. *Q. Appli. Math.*, 16(1):87–90.

Cantarella, G. E. and Vitetta, A. (2006). The multi-criteria road network design problem in an urban area. *Transportation*, 33(6):567–588.

Cascetta, E., Gallo, M., and Montella, B. (2006). Models and algorithms for the optimization of signal settings on urban networks with stochastic assignment models. *Ann. Oper. Res.*, 144(1):301–328.

Chen, K. and Luo, Z. Z. (2006). Analysis of feature of left turn traffic at level crossing and countermeasures. *Technol. Highw. Transport*, 22(2):114–118.

Fisk, S. C. (1984). Game theory and transportation systems modelling. *Transport. Res. B-Meth.*, 18(4):301–313.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345.

Ford, L. R. and Fulkerson, D. R. (1958). A suggested computation for multi-commodity network flows. *Manage. Sci.*, 5(1):97–101.

Foulds, L. R. (1981). A multicommodity flow network design problem. *Transport. Res. B-Meth.*, 15(4):273–284.

Foulds, L. R. (1991). *Graph theory applications.* Springer-Verlag, New York.

Foulds, L. R., do Nascimento, H. A. D., Calixto, I. C. A., Hall, B., and Longo, H. (2011). A fuzzy set approach to estimating OD matrices in congested brazilian traffic networks. In *Proc. XLIII Simp. Brasil. Pesq. Oper. (SBPO2011)*, pages 1386–1397, Brazil.

Foulds, L. R., Duarte, D. C. S., do Nascimento, H. A. D., Longo, H., and Hall, B. (2012). Turning restriction design for congested urban traffic networks. In *Proc. XVI Cong. Latino-Iberoamericano de Investigación Operativa / XLIV Simp. Brasil. Pesq. Oper. (CLAIO/SBPO2012)*, Brazil.

Gallo, M., D'Acierno, L., and Montella, B. (2010). A meta-heuristic approach for solving the urban network design problem. *Eur. J. Oper. Res.*, 201(1):144–157.

Horowitz, A. J. (1997). Intersection delay in regionwide traffic assignment: Implications of 1994 update of the highway capacity manual. *Transp. Res. Record*, 1572:1–9.

Jradi, W., do Nascimento, H. A. D., Longo, H., and Hall, B. R. (2009). Simulation and analysis of urban traffic the architecture of a web-based interactive decision support system. In *Proc. 12th Int. IEEE Conf. Intelligent Transport. Systems (ITSC'09)*, pages 1–6.

Kennington, J. L. and Helgason, R. V. (1980). *Algorithms for network programming.* John Wiley and Sons, New York, NY, USA.

Laguna, M. (2002). Scatter Search. In Pardalos, P. M. and Resende, M. G. C., editors, *Handbook of Applied Optimization*, pages 183–193. Oxford University Press.

Long, J., Gao, Z., Zhang, H., and Szeto, W. Y. (2010). A turning restriction design problem in urban road networks. *Eur. J. Oper. Res.*, 206(3):569–578.

Martí, R., Laguna, M., and Glover, F. (2006). Principles of scatter search. *Eur. J. Oper. Res.*, 169(2):359–372.

Nemhauser, G. L. (2012). *Extra Volume: Optimization Stories*, chapter Column Generation for Linear and Integer Programming, pages 65–73. Documenta Mathematica.

Palacios-Gomez, F., Lasdon, L., and Engquist, M. (1982). Nonlinear optimization by successive linear programming. *Manage. Sci.*, 28(10):1106–1120.

Potts, R. B. and Oliver, R. M. (1972). *Flows in Transportation Networks.* Academic Press.

Sherali, H. D., Narayanan, A., and Sivanandan, R. (2003). Estimation of origin-destination trip-tables based on a partial set of traffic link volumes. *Transport. Res. B-Meth.*, 37(9):815–836.

Sherali, H. D., Sivanandan, R., and Hobeika, A. G. (1994). A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transport. Res. B-Meth.*, 28(3):213–233.

Steenbrink, P. A. (1974). *Optimization of Transport Networks.* John Wiley and Sons.

USA Bureau of Public Roads (1964). Traffic assignment manual. Technical report, US Department of Commerce, Urban Planning Division, Washington DC, USA.

Wardrop, J. (1952). Some theoretical aspects of road traffic research. *P. I. Civil Eng. PT. 2*, 1(36):352–362.