A Gradient Projection Algorithm for Side-constrained Traffic Assignment

Joseph N. Prashker* and Tomer Toledo**
*Technion - Israel Institute of Technology
Department of Civil Engineering
Haifa
Israel
Email: prashker@netvision.net.il

**Massachusetts Institute of Technology
Center for Transportation and Logistics
Cambridge, MA
USA
Email: toledo@mit.edu

EJTIR, **4**, no. 2 (2004), pp. 177-193

Received: October 2003
Accepted: May 2004

Standard static traffic assignment models do not take into account the direct effects of capacities on network flows. Separable link performance functions cannot represent bottleneck and intersection delays, and thus might load links with traffic volumes, which far exceed their capacity. This work focuses on the side-constrained traffic assignment problem (SCTAP), which incorporates explicit capacity constraints into the traffic assignment framework to create a model that deals with capacities and queues. Assigned volumes are bounded by capacities, and queues are formed when capacity is reached. Delay values at these queues are closely related to Lagrange multipliers values, which are readily found in the solution. The equilibrium state is defined by total path travel times, which combine link travel times and delays at bottlenecks and intersections for which explicit capacity constraints have been introduced.
This paper presents a new solution procedure for the SCTAP based on the inner penalty function method combined with a path-based adaptation of the gradient projection algorithm. This procedure finds a solution at the path level as well as at the link level. All intermediate solutions produced by the algorithm are strictly feasible. The procedure used to ensure that side-constraints are not violated is efficient since it is only performed on constrained links that belong to the shortest path.

1. Introduction

The static traffic assignment problem (TAP) deals with predicting traffic flows on the links of a transportation network, given the travel demand between origins and destinations. The effects of traffic flows on travel times are represented by link performance functions. User equilibrium (UE) traffic assignment is found by solving a mathematical minimization problem (Beckmann et al. 1956).

The solution of the standard TAP formulation has been thoroughly investigated in the literature. The Frank-Wolfe (F-W) algorithm is widely used due to its simplicity and minimal computer memory requirements. However, the method suffers from slow convergence to the optimum, especially in the vicinity of the optimal solution.

In recent years, considerable attention has been given to path-based solution algorithms for the TAP. Path-based algorithms were previously considered infeasible for large networks. However, Jayakrishnan et al. (1994) and Chen and Lee (1999) showed that large-scale problems could be efficiently solved using path-based algorithms with existing computing capabilities. Moreover, the path-based solution is useful in cases that require knowledge of used paths and path flows, such as ATMS\ATIS applications. In particular, two path-based solution algorithms were introduced in the literature: Larsson and Patriksson (1992) proposed the Disaggregate Simplicial Decomposition algorithm (DSD) and Jayakrishnan et al. (1994) used the Gradient Projection (GP) algorithm. Both algorithms were shown to be superior to the F-W algorithm in terms of convergence and computing time (Jayakrishnan et al. 1994, Tatineni et al. 1998). Chen and Lee (1999) compared DSD and GP on several realistic networks. They found that GP performed better in most aspects.

A class of problems closely related to TAP is the side-constrained traffic assignment problem (SCTAP). A side-constrained problem is created when additional constraints are added to the TAP. These constraints may represent capacity limitations at bottlenecks and intersections or external constraints on the transportation system.

The purpose of this paper is two-fold: first, we discuss the SCTAP formulation and highlight its advantages in terms of modeling realism compared to standard traffic assignment. Secondly, we present a new development of the GP algorithm to efficiently solve the SCTAP. The new algorithm exploits the characteristics of the GP algorithm to combine it with a penalty function method.

The rest of the paper is structured as follows: The next section motivates the use of SCTAP. The extension of the TAP to include side-constraints and solution algorithms to the SCTAP are presented in section 3. A new path-based solution algorithm for the SCTAP and assignment results of the proposed algorithm are presented in section 4. A summary of the results is presented in section 5.

2. Motivation

Static traffic assignment has been the main tool for transportation network analysis within the four-step planning paradigm. Over the years, many deficiencies of TAP have been raised, pointing at the over-simplicity and unrealistic aspects of the results produced by this model. Research efforts in recent years have led to the development of dynamic assignment models that relax some of the simplifying assumptions made in static TAP. However, dynamic

assignment models are still far from being able to replace static models in practice. One important reason for that is the ability of TAP models to efficiently handle large-scale applications that may be beyond the capabilities of more complex and detailed models. Furthermore, given the long-term focus of some of the applications with the associated uncertainty in inputs and the data intensive and computation-heavy nature of dynamic models it is likely that static models will continue to be an important transportation modeling tool in the foreseeable future. Thus, efforts to improve the realism of static models are very important (FHWA 2002). As we will discuss next, the addition of side-constraints to the standard TAP model has a potential to significantly improve the realism of the solution, while making relatively simple modifications in the TAP that do not overly complicate the solution process.

TAP solutions, especially in congested networks, often assign links with flows that exceed their capacity. For example, approximately 15% of the links in the ADVANCE network solution (Chen and Lee 1999) are over-saturated. Such assignment results cannot be used for realistic engineering analysis: Not only that traffic flows on over-saturated links are unrealistically high, but the assignment on the alternative paths is also distorted since these links will often be under-utilized. This behavior is typical to models that use link performance functions that do not provide an upper bound on flows and so tend to under-estimate queuing delays, such as the BPR function (BPR 1964). Daganzo (1977) proposed to use link performance functions that are asymptotic to the capacity, such as Davidson's function (Davidson 1966) in order to restrict assigned flows. However, application of these functions produces very high travel times in saturated links (Boyce et al. 1981), and so does not improve the realism of the solution. The introduction of explicit capacity constraints to the TAP model is a simple and intuitive alternative to the use of asymptotic link performance functions to prohibit over-saturated links.

Another shortcoming of standard static assignment is that separable link performance functions (in which the cost of a link is a function of the flow on that link only) cannot capture delays caused by the interaction between flows on two or more links. Non-separable cost functions, which may capture these interactions, require complicated and less efficient solution techniques (Patriksson 1994). Moreover, existence and uniqueness of equilibrium are not always ensured (Smith 1979, Sheffy 1985). Similar to capacities, it may be simpler to represent the limitations imposed by link interactions by introducing side-constraints in the problem formulation, rather than through the link performance functions. For example, in the next section we will discuss a simple constraint that can guarantee that the total flow entering a merging area is smaller or equal to the saturation flow of the merged link. However, it may be much more difficult to calibrate a link performance function that captures the merging effect.

3. Side-constrained traffic assignment (SCTAP)

3.1 Formulation

We begin by briefly reviewing some of the basic definitions of the assignment model. Let $G=(N, A)$ be a directed graph, where N and A are the sets of nodes and links, respectively.

Denote q^{rs} the demand for trips from origin $r \in R$ to destination $s \in S$. R and S are subsets of N. A set, K^{rs}, of paths that connect r to s is defined for each OD pair. The user equilibrium (UE) principle states that for each OD pair, all used paths have equal and minimal travel times:

$$\begin{cases} f_k^{rs} > 0 \quad \Rightarrow \quad t_k^{rs} = t_{\min}^{rs} \\ f_k^{rs} = 0 \quad \Rightarrow \quad t_k^{rs} \geq t_{\min}^{rs} \end{cases} \qquad \forall\, k,\ \forall\, rs \tag{1}$$

f_k^{rs} is the flow on path $k \in K^{rs}$, t_k^{rs} and $t_{\min}^{rs}$ are travel times on path k and on the shortest path from r to s, respectively. Path travel times are the sum of travel times on all the links that comprise the path.

Assuming separable link performance functions, the solution of the following optimization problem, defined in the space of path flows variables, corresponds to the UE principle:

$$\min\ Z = \sum_a \int_0^{V_a} t_a(w)\, dw \tag{2}$$

Subject to:

$$\sum_k f_k^{rs} = q^{rs} \qquad \forall\, rs \tag{3}$$

$$f_k^{rs} \geq 0 \qquad \forall\, k,\ \forall rs \tag{4}$$

$$V_a = \sum_{rs} \sum_k f_k^{rs} \delta_{ak}^{rs} \qquad \forall\, a \tag{5}$$

t_a and V_a are travel time and flow on link a, respectively. δ_{ak}^{rs} is an indicator variable, which takes a value of 1 if link a is on path k for OD pair rs and 0 otherwise.

We consider the addition of explicit side-constraints to the TAP formulation in Equations (2)-(5) to represent various conditions and control measures in the network. The extended formulation was first introduced in Thompson and Payne (1975), which considered capacity constraints and flow-independent link travel times. Smith (1987) proved their results. Patriksson (1994) extended them to flow-dependent travel times and general side-constraints. The theoretical development below considers a set of general side-constraints:

$$g_i(V) \leq 0 \qquad \forall i \in I \tag{6}$$

$g_i(V)$ is some function of the vector of link flows in the network, V. The subscript i denotes the constraint index within the set I.

The optimal solution to the SCTAP is characterized by the Karush-Kuhn-Tucker (KKT) first order optimality conditions (see, for example, Ravindran et al. 1987):

$$\sum_a t_a(V_a)\, \delta_{ak}^{rs} + \sum_i \lambda_i \sum_a \frac{\partial g_i(V)}{\partial V_a} \delta_{ak}^{rs} \geq \tau^{rs} \qquad \forall\, k,\ \forall\, rs \tag{7}$$

$$\sum_k f_k^{rs\,*} = q^{rs} \qquad \forall\, rs \tag{8}$$

$$f_k^{rs*} \geq 0 \qquad \forall\, k,\ \forall\, rs \tag{9}$$

$$g_i(V) \leq 0 \qquad \forall\, i \tag{10}$$

$$\lambda_i \geq 0 \qquad \forall\, i \tag{11}$$

$$\tau^{rs} \geq 0 \qquad \forall\, rs \tag{12}$$

$$f_k^{rs*}\left[\sum_a t_a(V_a)\,\delta_{ak}^{rs} + \sum_i \lambda_i \sum_a \frac{\partial g_i(V)}{\partial V_a}\delta_{ak}^{rs} - \tau^{rs}\right] = 0 \qquad \forall\, k,\ \forall\, rs \tag{13}$$

$$\lambda_i g_i(V) = 0 \qquad \forall\, i \tag{14}$$

λ_i and τ^{rs} are Lagrange multipliers associated with the side-constraints and trip demands, respectively.

Equations (10), (11) and (14), and can be summarized jointly by:

$$\begin{aligned} g_i(V) < 0 \quad &\Rightarrow \quad \lambda_i = 0 \qquad \forall\, i \\ g_i(V) = 0 \quad &\Rightarrow \quad \lambda_i \geq 0 \qquad \forall\, i \end{aligned} \tag{15}$$

Assuming that side-constraints capture limited capacities of various road facilities, the above equations can be interpreted as defining the delays caused by these limitations. When flows are such that capacities are not reached, there are no delays associated with the corresponding constraint. When flows reach the capacity, queues form and drivers experience delays.
From Equations (7), (9) and (13) we get:

$$\begin{cases} f_k^{rs*} > 0 \quad \Rightarrow \quad \sum_a t_a(V_a)\,\delta_{ak}^{rs} + \sum_i \lambda_i \sum_a \frac{\partial g_i(V)}{\partial V_a}\delta_{ak}^{rs} = \tau^{rs} \\ f_k^{rs*} = 0 \quad \Rightarrow \quad \sum_a t_a(V_a)\,\delta_{ak}^{rs} + \sum_i \lambda_i \sum_a \frac{\partial g_i(V)}{\partial V_a}\delta_{ak}^{rs} \geq \tau^{rs} \end{cases} \qquad \forall\, k,\ \forall\, rs \tag{16}$$

This is a generalization of the UE principle stated in Equation (1). The network equilibration is defined over generalized path travel times that include link travel times and delays at queues. The delays are captured by the magnitude of the Lagrange multiplier of the side-constraint:

$$d_{ai} = \lambda_i \frac{\partial g_i(V)}{\partial V_a} \qquad \forall\, a,\ \forall\, i \tag{17}$$

d_{ai} is the delay on link *a* caused by constraint *m*.

It is important to note that, unlike the link flows solution, the path flows and the Lagrange multipliers are generally not unique. An optimal solution to the SCTAP problem will yield one out of the possibly infinite such solutions. It is therefore important to use path flow solutions and interpret Lagrange multipliers with caution. Larsson and Patriksson (1999) discuss this issue in detail and provide sufficient conditions for uniqueness.

3.2 Side-constraints

Different facilities in the network may be represented in the assignment model through a simplification of the mechanism that operates them to a single (or set) of constraints. The most common side-constraints are link capacity constraints:

$$V_j \le C_j \tag{18}$$

C_j is the capacity of link *j*. The subscript *j* denotes links in the subset $J \subseteq A$ for which capacity constraints are defined.

Capacity constraints represent link geometry and bottleneck capacities created by lane drops, lane closure, or road incidents. Facilities that limit usage time of links (e.g. fixed-time traffic signals, portals and ramp controls) can also be modeled with capacity constraints. Other constraints may also be used. Bell (1995) modeled the operations of traffic-actuated signals in which green proportions change according to the relative demands on all links approaching the intersection with the constraint:

$$\sum_{j \in IN} \frac{V_j}{S_j} \le 1 - \frac{W}{c} \tag{19}$$

IN is the set of all links approaching the intersection. S_j is the saturation flow of link *j* approaching the intersection. *c* is the cycle time and *W* is the lost time. Thus, $\frac{W}{c}$ is the proportion of lost time.

In merging situations, the total flow entering the merging area is limited by the capacity of the common outgoing link. This can be represented by:

$$\sum_{j \in IN} V_j \le C_{out} \tag{20}$$

IN denotes the set of all links entering the merge. C_{out} is the capacity of the outgoing link.

The above constraint assumes non-priority merging. However, it may also be used as an approximation for priority merges, such as an on-ramp merging into a freeway or two-way stop or yield controlled intersections. While in these situations the major stream is supposed to have absolute priority over the minor stream in the allocation of available capacity, there is ample empirical evidence (e.g. Bunker and Troutbeck 2003, Bonneson and Fitts 1999) that behaviors such as yielding, gap forcing and intersection blockages by minor stream vehicles result in capacity allocation that is more similar to that of a non-priority merge. To further simplify the constraints, the merging constraint can be replaced by simple capacity constraints for each merging link under the assumption that all the merging links are saturated in congested conditions. In this case the exit capacity from each one of the links entering the merge will be proportional to its saturation flow:

$$V_j \le C_j = \frac{S_j}{\sum_{l \in IN} S_l} C_{out} \tag{21}$$

S_j and S_l are the saturation flows of links *j* and *l* that enter the merge area, respectively.

3.3 A simple example

The following example illustrates the ability of the SCTAP model to produce more realistic assignment results compared to standard traffic assignment. Consider the network presented in Figure 1Figure1, which consists of a central intersection that may represent a city center surrounded by two ring roads. Travel times on each link are given by the BPR formula with parameters $\alpha = 0.15$ and $\beta = 4$. The free-flow travel times (t_0) and capacities of the links in this network are presented inTable 1. Trip demands are for passing traffic (1→2, 2→1) and for trips to the center (1→7, 2→7). These demands are presented inTable 2.

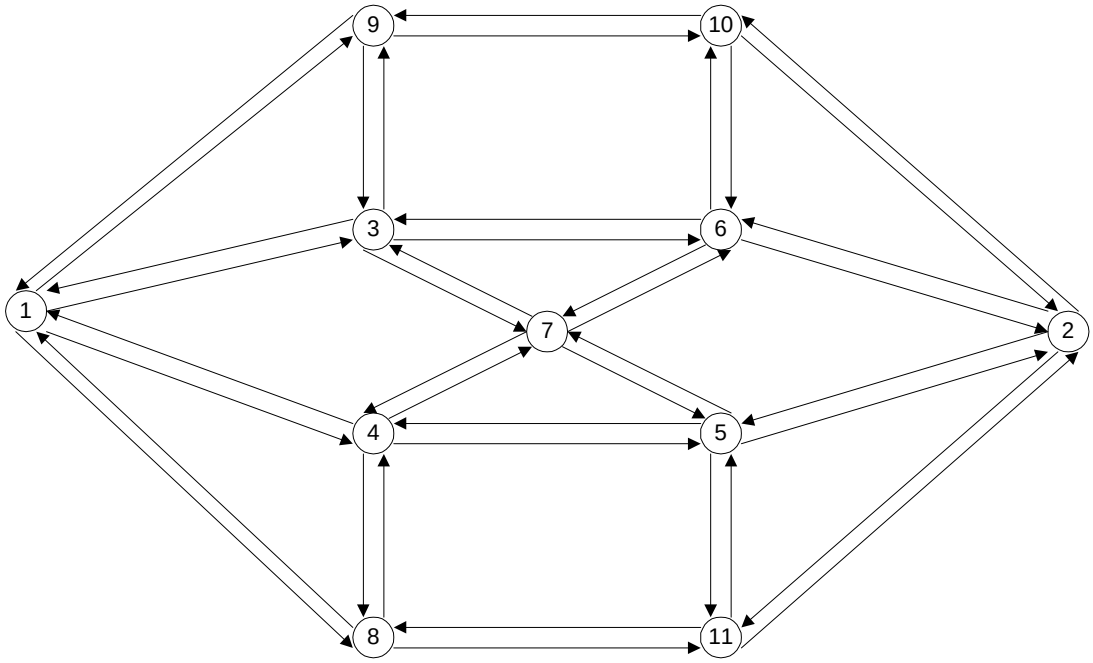

Figure1. Example network

User equilibrium flows on this network were found using two models: the standard model and the side-constrained model. In the latter, constraints on intersection flows, given by Equation **Fout! Verwijzingsbron niet gevonden.**, were imposed on intersections 3, 4, 5 and 6. Both solutions are presented in Figure 2. The unconstrained solution shows high levels of traffic passing through the center and the inner ring. The outer ring is not used at all. As a result the flows on four links exceed capacities. The flows through intersections 3, 4, 5 and 6 are more than double the capacity of these intersections. It is more realistic to assume that limited capacities and delays near the center will divert traffic to other paths bypassing the center. This is apparent in the SCTAP solution: only center-bounded traffic goes into the center. Passing traffic is diverted from the center to the two ring roads. The resulting link

flows are such that the flows through all intersections satisfy the capacity constraints of these facilities. This example shows that the introduction of side-constraints can significantly alter traffic assignment patterns and provide more realistic and plausible predictions. The impact on decision making may not only be in terms of the conclusions that will be drown from the assignment results, but also in terms of the fidelity decision-makers associate with the assignment results.

Table 1. Link parameters

Up Node	Down Node	t_0	C	Up Node	Down Node	t_0	C
1	3	20	3000	6	2	20	3000
1	4	20	3000	6	3	15	3000
1	8	30	4000	6	7	6	2000
1	9	30	4000	6	10	20	4000
2	5	20	3000	7	3	6	2000
2	6	20	3000	7	4	6	2000
2	10	30	4000	7	5	6	2000
2	11	30	4000	7	6	6	2000
3	1	20	3000	8	1	30	4000
3	6	15	3000	8	4	20	4000
3	7	6	2000	8	11	30	4000
3	9	20	4000	9	1	30	4000
4	1	20	3000	9	3	20	4000
4	5	15	3000	9	10	30	4000
4	7	6	2000	10	2	30	4000
4	8	20	4000	10	6	20	4000
5	2	20	3000	10	9	30	4000
5	4	15	3000	11	2	30	4000
5	7	6	2000	11	5	20	4000
5	11	20	4000	11	8	30	4000

Table 1 Trip demands

Origin	Destination	Demand
1	2	4000
1	7	3000
2	1	4000
2	7	3000

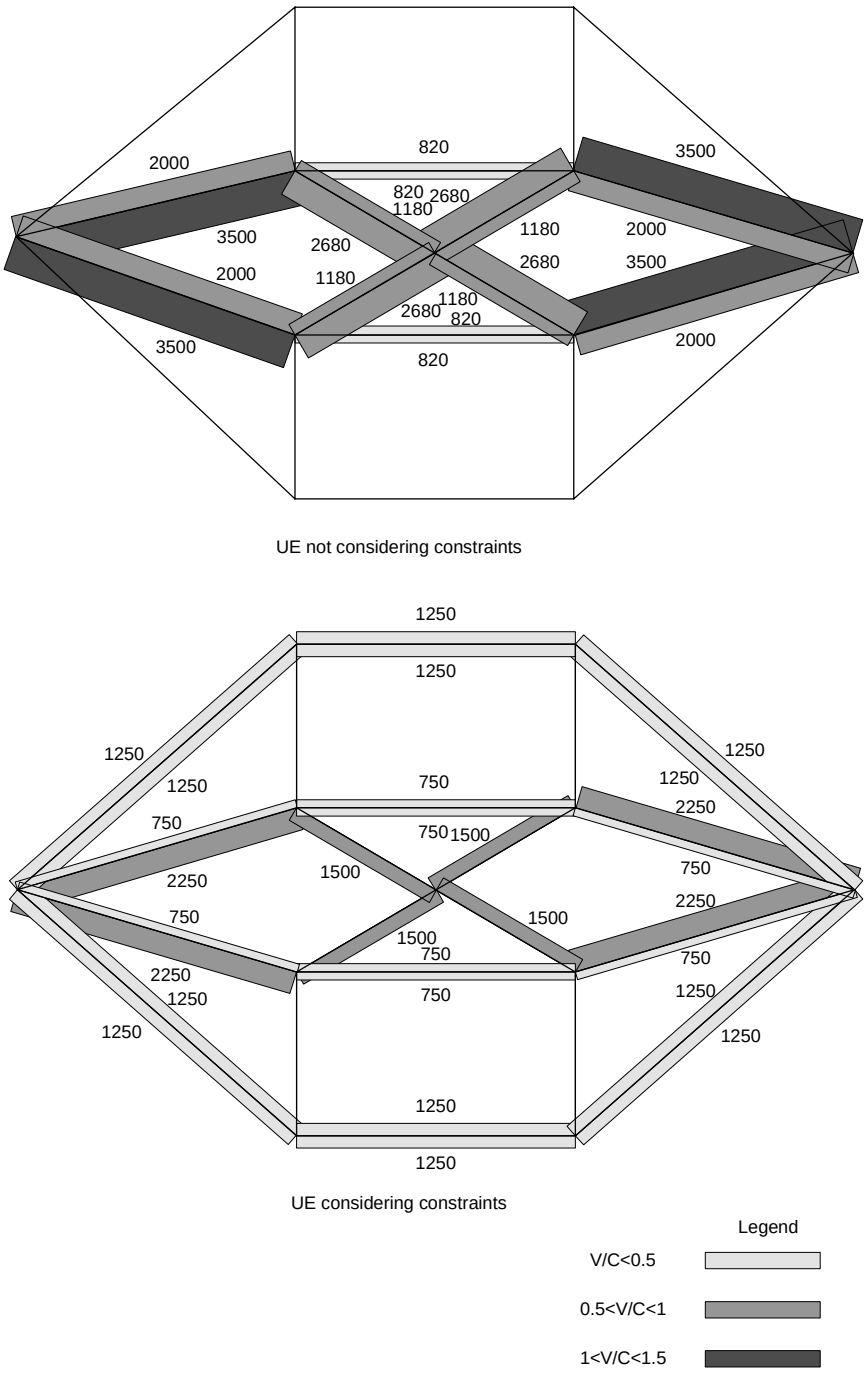

Figure 2. Equilibrium traffic flows with and without capacity constraints

3.4 Solution algorithms

Several solution algorithms for the SCTAP were proposed in the literature. Inouye (1986), for the case of capacity side-constraints, and Yang and Yagar (1994, 1995) for linear side-constrains, used the inner penalty function method to generate a sequence of standard traffic

assignment sub-problems, with penalty terms added to the objective function for each side-constraint. Penalty factors were decreased in consecutive iterations. The sequence of sub-problem solutions approaches the optimal solution of the original problem. The F-W algorithm was used to solve assignment sub-problems. However, these methods exhibit the same slow convergence properties of the F-W algorithm. Larsson and Patriksson (1995) considered a simpler version of the problem with only simple capacity constraints. They used an augmented Lagrangean dual algorithm to solve the problem. The objective function is augmented to include a Lagrangean exterior penalty term expressing capacity constraints. At every iteration, a new solution is found and the Lagrange multipliers estimates are updated. The unconstrained dual problems are solved with the Disaggregated Simplicial Decomposition (DSD) method proposed by Larsson and Patriksson (1992). Lawphongpanich (2000) and Larsson et al. (2004) propose modified variants of the basic algorithm that improve convergence and stability. These methods are applicable to general side-constraints. Other Lagrangean dualization approaches were proposed by Inouye (1986) and Hearn and Lawphongpanich (1990).

4. The GPCAP algorithm

As discussed earlier the GP algorithm (Jayakrishnan et al. 1994) is an efficient method to solve the standard TAP problem. We propose a solution algorithm adapted from the GP algorithm for the SCTAP. The algorithm is a path-based penalization method. Our algorithm consists of two levels: an inner penalization step and a gradient projection step. In the former, the SCTAP is replaced with an unconstrained sub-problem. The sub-problem is similar in structure to TAP with an additional penalty term in the objective function for each side-constraint. The sub-problem is solved using the GP algorithm. Next, a new sub-problem is generated with smaller penalty terms. The optimal solution of the previous sub-problem is used as an initial solution to the new one. The sequence of sub-problems solutions converges to the optimal solution of the original problem when the penalty terms approach zero. The algorithm is named GPCAP (Gradient Projection with CAPacity constraints). Its details are described next.

4.1 Inner penalty function

In the inner penalty function method, a penalty term for every side-constraint is added to the objective function. The penalty term is such that when the solution nears a feasibility border its value rises rapidly. This ensures feasibility of the solution at every iteration. The penalty term contains a parameter γ. This parameter is decreased in consecutive iterations through multiplication by a scaling factor $\sigma(0<\sigma<1)$ thus creating the sequence of sub-problems to be solved. A suitable penalty function is (Yang and Yagar 1994):

$$P_i(V)^n = -\gamma^n \ln\left(\frac{-g_i(V)}{H_i}\right) \qquad \forall\, i \tag{22}$$

$P_i(V)^n$ is the penalty term associated with side constraint i at iteration n of the inner penalty function. γ^n is the penalty parameter at iteration n. H_i is a scaling factor given by:

$$H_i = \sup_V \left(-g_i(V)\right) \qquad \forall\ m \tag{23}$$

Hence, at the n^{th} iteration, the sub-problem is given by:

$$\min \sum_a \int_0^{V_a} t_a \left(\sum_{rs} \sum_k f_k^{rs} \delta_{ak}^{rs} \right) dV + \sum_i P_i(V)^n \tag{24}$$

Subject to the constraints in Equations (3), (4) and(5).

An important feature of this method is that the optimal values of Lagrange multipliers associated with the penalized constraints (λ_i) are given by the derivative of the penalty function when $\gamma \to 0$. Equation (17) relates λ_i to delays in the network. Hence, an explicit expression for delays is given by:

$$d_{ai} = \lim_{\gamma^n \to 0^+} \frac{\gamma^n}{-g_i(V)} \frac{\partial g_i(V)}{\partial V_a} \qquad \forall\ a,\ \forall\ i \tag{25}$$

4.2 Gradient projection

The sub-problems created through inner penalization are solved by the GP method. In this method, given a feasible solution, a gradient related search direction for the new solution is chosen. The step size in this direction can be determined in various ways. When the resulting new solution is not feasible (i.e. the step size was too large) a projection function transforms the solution into the feasible area.

This method is very efficient for problems with many simple constraints. It allows the solution to advance along arcs defining the feasible area rather than along a straight line inside the feasible region as does the F-W algorithm. Feasibility borders do not slow down the progress of the solution, since the projection function will ensure feasibility. This allows for faster convergence, especially in cases (such as TAP) in which the optimal solution lays on the boundaries with many binding constraints.

Adaptation of the GP algorithm to this problem is based on ensuring the validity of the demand constraints [Equation (3)]. Flow-carrying paths are separated in two groups: shortest paths at the current solution, one for each OD pair, and all other used paths, which are non-optimal at the current solution. The problem is defined in terms of the non-optimal paths only. Flows on the shortest paths are calculated such that trip demands will be satisfied:

$$f_{\bar{k}}^{rs} = q^{rs} - \sum_{k \neq \bar{k}} \tilde{f}_k^{rs} \qquad \forall\ rs \tag{26}$$

$f_{\bar{k}}^{rs}$ and $\tilde{f}_k^{rs}$ are the flows on the optimal and non-optimal paths, respectively.

According to the UE principle, the optimal solution will be reached when travel times on all paths carrying flow will be equal, and not greater than the travel time on any unused path. In order to advance towards such a solution, flows are transferred from currently non-optimal paths to the shortest one for every OD pair. The gradient of the objective function in the space of non-optimal path flows is given by:

$$\frac{\partial \tilde{Z}(\tilde{f})}{\partial \tilde{f}_k^{rs}} = \frac{\partial Z(f)}{\partial f_k^{rs}} - \frac{\partial Z(f)}{\partial f_{\bar{k}}^{rs}} = \sum_a \left[t_a(V_a) + \gamma^n \sum_i \frac{1}{-g_i(V)} \frac{\partial g_i(V)}{\partial V_a} \right] (\delta_{ak}^{rs} - \delta_{a\bar{k}}^{rs}) \tag{27}$$
$$= t_k^{rs} - t_{\bar{k}}^{rs} + d_k^{rs} - d_{\bar{k}}^{rs} = c_k^{rs} - c_{\bar{k}}^{rs} \qquad \forall\, k,\ \forall rs$$

Several choices of search direction and step size choices may be used. The simplest option is to use the gradient direction with a predetermined step size. The step size may also be based on exact or inexact (e.g. Armijo rule) line search. Bertsekas and Gallager (1982) and Jayakrishnan et al. (1994) found that for network problems, a variation of the GP algorithm that uses a diagonally scaled Newton's search direction gives best results. The Hessian matrix is approximated by its diagonal elements only:

$$H_k^{rs} = \left[\frac{\partial^2 Z(\tilde{f})}{\partial \tilde{f}_k^{rs2}} \right] \tag{28}$$

The resulting GP iteration is given by:

$$\tilde{f}_k^{rs,\,m+1} = \tilde{f}_k^{rs,\,m} - \left[H_k^{rs,\,m} \right]^{-1} \frac{\partial Z\left(\tilde{f}^m \right)}{\partial \tilde{f}_k^{rs}} \qquad \forall\, k,\ \forall rs \tag{29}$$

The superscript *m* denotes the GP iteration counter.
$H_k^{rs,\,m}$ can be explicitly written as:

$$H_k^{rs,\,m} = \sum_{a \in L_k^{rs,\,m}} \left[\frac{\partial t_a(V_a)}{\partial V_a} + \gamma^n \sum_i \frac{1}{g_i(V)^2} \frac{\partial g_i(V)}{\partial V_a} + \frac{1}{-g_i(V)} \frac{\partial g_i{}^2(V)}{\partial V_a{}^2} \right]_{V^m} \tag{30}$$

$L_k^{rs,\,m}$ is the set of all links that belong to either path *k* or the shortest path $\bar{k}$ but not to both.
Validity of the non-negativity constraints [Equation **Fout! Verwijzingsbron niet gevonden.**] is ensured by a projection function:

$$\tilde{f}_k^{rs,\,m+1} = \max\left\{ 0,\ \tilde{f}_k^{rs,\,m} - \left[H_k^{rs,\,m} \right]^{-1} \left(c_k^{rs,\,m} - c_{\bar{k}}^{rs,\,m} \right) \right\} \qquad \forall\, k,\ \forall rs \tag{31}$$

After performing the projection, flows on all non-optimal paths are smaller or equal to the previous iteration. Shortest paths flows are increased according to Equation (26).

4.3 Overall structure

The GPCAP algorithm requires efficient combination of the GP algorithm with the inner penalty function steps. In doing so, it is necessary to ensure that side-constraints are not violated when performing the GP iteration. Each iteration of the GP algorithm diverts flows from non-optimal paths to currently shortest paths. Therefore, only side-constraints that involve links that belong to the optimal path $\bar{k}$ (and do not belong to some non-optimal path *k*) may be violated. To ensure that the new flows on these links do not violate constraints, the flow diverted in each iteration is bounded by the smallest slack in side-constraints:

$$\Delta\tilde{f}_k^{rs,\,m} = \inf_i \left[\sum_a \left. \frac{\partial g_i(V)}{\partial V_a} \right|_{V^m} \left(\delta_{ak}^{rs} - \delta_{a\bar{k}}^{rs} \right) \right]^{-1} \left(-g_i(V) \right) - \omega \qquad \forall\, k,\ \forall rs \tag{32}$$

$\Delta\tilde{f}_k^{rs,\,m}$ is an upper bound on the flow diverted from non-optimal path *k* to the shortest path $\bar{k}$. ω is a small positive constant that ensures a strictly feasible solution at each iteration.
The number of checks required to perform this step is relatively small. For example, in the case of link capacity constraints [Equation (18)] it is at the most equal to the number of constrained links on the shortest path. The bound on the flow that can be diverted would in this case simplify to:

$$\Delta\tilde{f}_k^{rs,\,m} = \inf_{\substack{j\in\bar{k} \\ j\notin k}} \left(C_j - V_j^m \right) - \omega \qquad \forall\, k,\ \forall rs \tag{33}$$

The GP algorithm is easily adapted for these checks, hence allowing it to efficiently cope with side-constraints. The only modification required is the addition of a term to the projection function corresponding to Equation (32). The modified projection function is given by:

$$\tilde{f}_k^{rs,\,m+1} = \max\left\{ 0,\ \tilde{f}_k^{rs,\,m} - \left[H_k^{rs,\,m} \right]^{-1} \left(c_k^{rs,\,m} - c_{\bar{k}}^{rs,\,m} \right),\ \tilde{f}_k^{rs,\,m} - \Delta\tilde{f}_k^{rs,\,m} \right\} \ \forall\, k,\ \forall rs \tag{34}$$

Given a feasible initial solution, the GPCAP algorithm is summarized in these steps:

1. Initialization: Set σ and γ^0 values and iteration counters *m:=0*, *n:=0*.
2. Inner penalty function iteration: Set *n:=n+1*, $\gamma^n = \sigma\gamma^{n-1}$
3. GP iteration:
   - Update link traffic flows and costs. *m:=m+1*.
   - Calculate the shortest path for every OD pair. If it is a new path, add it to the path list.
   - Update path flows according to equations (34) and (26).
4. GP convergence: If GP converged go to step 5. Otherwise, go to step 3.
5. Overall convergence: If overall convergence holds calculate link flows and travel times, stop. Otherwise, set *m:=0*, go to step 2.

Next, we present a procedure for finding a feasible initial solution based on a technique proposed by Daganzo (1977). The procedure is based on the GPCAP algorithm itself and retains the same structure. Given an infeasible solution, we define a temporary feasible area that contains this solution. Based on this feasibility area, a new solution is calculated by performing an iteration of the GPCAP algorithm. The procedure is repeated until the temporary feasibility area is contained in the original one. The feasible area is narrowed by temporarily changing the right-hand side of side-constraints to $g_i(V) \le \hat{C}_i$:

$$\hat{C}_i = \begin{cases} 0 & g_i(V) < 0 \\ g_i(V)(1+\varepsilon) & g_i(V) \ge 0 \end{cases} \tag{35}$$

$\hat{C}_i$ is the temporary right-hand side value of constraint *i*. ε is a small positive constant.
At each iteration constraints that are violated will have very high travel costs (due to the penalty value). This will cause flows on links associated with these constraints to be smaller

in the next iteration. The use of the inner penalty function in the procedure ensures that once a constraint is satisfied it cannot be violated again. The starting solution for this process can be an all-or-nothing (AON) assignment or an incremental assignment.
Given an infeasible starting solution, the initializing algorithm proceeds as follows:

1. Initialization: Set σ and γ^0 values and iteration counters *m:=0, n:=0.*
2. Inner penalty function iteration: Set *n:=n+1*, $\gamma^n = \sigma\gamma^{n-1}$.
3. GP iteration:
   - Update link flows.
   - If all side-constraints are satisfied, stop. Otherwise, calculate temporary right-hand side values according to equation (35) and costs. *m:=m+1.*
   - Calculate the shortest path for every OD pair. If it is a new path, add it to the path list.
   - Update path flows according to equations (34) and (26).
4. GP convergence: If GP converged go to step 2. Otherwise, go to step 3.

4.4 Numerical results

The convergence of GPCAP algorithm was compared to Yang and Yagar's algorithm. Their algorithm is based on the F-W algorithm. The two algorithms were tested on the Sioux Falls network (Leblanc 1973). This network contains 24 nodes, 76 links and 550 OD pairs. Capacity constraints were imposed on 12 links. The algorithm was run to achieve a very accurate solution. The progress of the algorithms was measured by the deviation from this solution. Convergence is measured by the maximum absolute percentage error (MAPE) between the optimal and current flows over all links:

$$MAPE = \max_{a \in A} \left| \frac{V_a^n - V_a^{opt}}{V_a^{opt}} \right| \tag{36}$$

V_a^{opt} and V_a^n are the flows on link *a* in the optimal and n^{th} iteration solutions, respectively.
Convergence results are presented in Figure . GPCAP iterations are more complex because of the need to calculate second derivatives of the objective function and perform path comparisons to identify common links. However, the additional information used allows the solution to converge after many fewer iterations. The GPCAP algorithm converged faster and in a more stable manner. The gap between the algorithms widens as they approach the optimum. Overall, Yang and Yagar's algorithm took approximately 27%, 108% and 7820% time longer than GPCAP to get to levels of accuracy within 10%, 5% and 1% of the optimal solution, respectively.

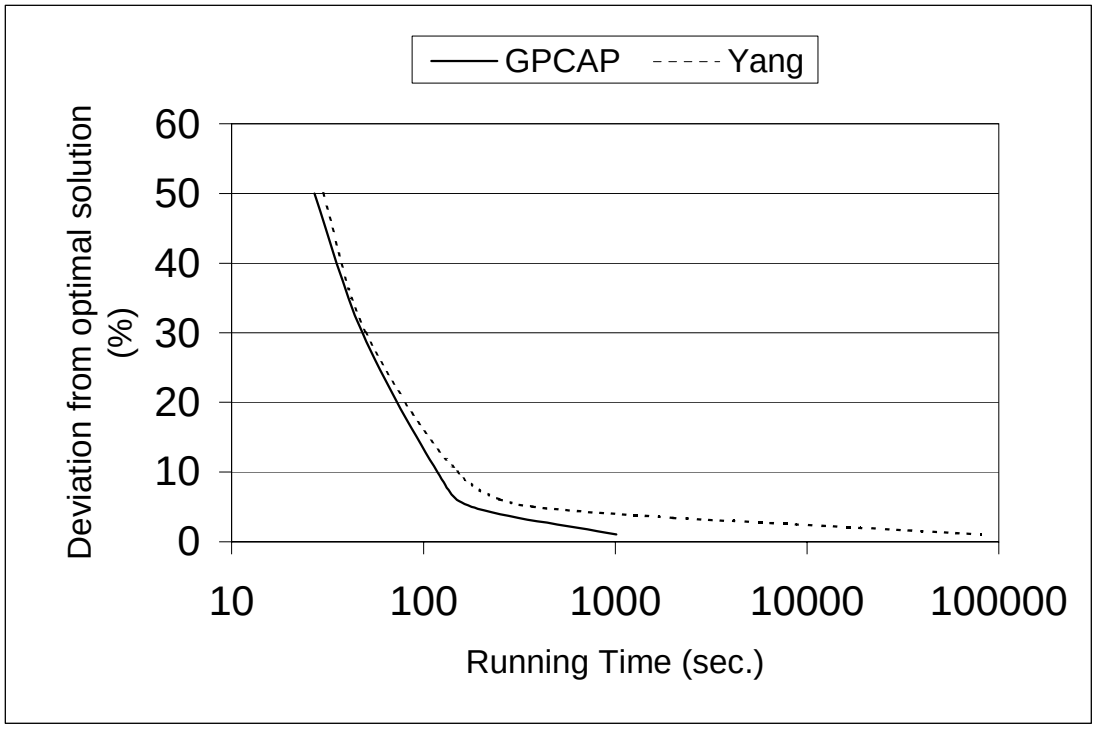

Figure 3. Convergence in Sioux-Falls network

5. Summary

This paper presented a model and solution algorithm for the static traffic assignment problem with side-constraints. The SCTAP is created by adding side-constraints to the standard static TAP. Side-constraints may be used to represent various bottlenecks in the network as well as external factors that affect traffic flow. Several such constraints were described. Queuing delays are created when side-constraints are satisfied as equalities, i.e., when traffic flows equal the capacity of the facilities the constraints represent. These delays are closely related to the optimal values of the corresponding Lagrange multipliers. The resulting UE state is defined by total path travel times comprised of link travel times and queuing delays.

The SCTAP solution may significantly enhance the realism of the assignment, as the demonstrated by a simple example. This may not only improve the reliability of long-term planning decisions, but also allow assignment results to be used for short-term policy and operational analysis, such as evaluation of demand management and congestion pricing strategies, estimation of environmental impacts and design of traffic plans for special events.

We introduced the GPCAP algorithm to solve the SCTAP. The algorithm is based on combining an internal penalty function method with a path-based adaptation of the GP algorithm. All intermediate solutions produced by the algorithm are strictly feasible. The procedure used to ensure that side-constraints are not violated is efficient since it is only performed on constrained links that belong to the shortest path. The algorithm was tested against Yang and Yagar's algorithm with favorable convergence results.

The GPCAP algorithm provides a solution that includes path flows. These are useful for estimating turning flows and for route identification. While the path flow solution is not unique and therefore needs to be interpreted with caution, it is useful for practical analysis, as in the case of link of interest problems.

References

Beckmann M., McGuire C.B. and Winsten C.B. (1956), *Studies in the Economics of Transportation*, Yale University Press, New Haven CT.

Bell M.G.H. (1995), Stochastic User Equilibrium Assignment in Networks with Queues, *Transportation Research Part B*, 29, pp. 125-137.

Bertsekas D.P. and Gallager R.G. (1992), *Data Networks* (2nd ed.), Prentice-Hall, Englewood Cliffs NJ.

Bonneson J.A. and Fitts J.W. (1999), Delay to Major Street Through Vehicles at Two-Way Stop-Controlled Intersections, *Transportation Research Part A*, 33, pp. 237-253.

Boyce D.E., Janson B.N. and Eash R.W. (1981), The Effect on Equilibrium Trip Assignment of Different Link Congestion Functions, *Transportation Research Part A*, 15, pp. 223-232.

BPR (1964), *Traffic Assignment Manual*, Bureau of Public Roads, US Department of Commerce, Washington DC.

Bunker J. and Troutbeck R. (2003), Prediction of Minor Stream Delays at a Limited Priority Freeway Merge, *Transportation Research Part B*, 37, pp. 719-735.

Chen A. and Lee D.H. (1999), Path-Based Algorithms for Large Scale Traffic Equilibrium Problems: a Comparison between DSD and GP, *Paper presented at the 78th Transportation Research Board Annual Meeting*, Washington DC.

Daganzo C.F. (1977), On the Traffic Assignment Problem with Flow Dependent Costs - I and II, *Transportation Research*, 11, pp. 433-441.

Davidson K.B. (1966), A Flow Travel Time Relationship for Use in Transportation Planning, *Proceedings of the Australian Road Research Board*, 3, pp. 183-194.

FHWA (2002), *2002 Status of the Nation's Highways, Bridges and Transit: Conditions and Performance*, US Department of Transportation, Washington DC.

Hearn D.W. and Lawphongpanich S. (1990), A dual Ascent Algorithm for Traffic Assignment Problems, *Transportation Research Part B*, 24, pp. 423-430.

Inouye H. (1986), Traffic Equilibria and its Solution in Congested Road Networks, *Proceedings of the IFAC Conference on Control in Transportation Systems*, Gensher R., ed., Pergamon Press, Oxford, pp. 267-272.

Jayakrishnan R., Tsai W.K., Prashker J.N. and Rajadhyaksha S. (1994), A Faster Path Based Algorithm for Traffic Assignment, *Transportation Research Record*, 1443, pp. 75-83.

Larsson T. and Patriksson M. (1992), Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem, *Transportation Science*, 26, pp. 4-17.

Larsson T. and Patriksson M. (1995), An Augmented Lagrangean Dual Algorithm for Link Capacity Side Constrained Traffic Assignment Problems, *Transportation Research Part B,* 29, pp. 433-455.

Larsson T. and Patriksson M. (1999), Side Constrained Traffic Equilibrium Models – Analysis, Computation and Applications, *Transportation Research Part B,* 33, pp. 233-264.

Larsson T., Patriksson M. and Rydergren C. (2004), A Column Generation Procedure for the Side Constrained Traffic Equilibrium Problem, *Transportation Research Part B,* 38, pp. 17-38.

Lawphongpanich S. (2000), Simplicial with Truncated Dantzig-Wolfe Decomposition for Nonlinear Multicommodity Network Flow Problems with Side Constraints, *Operations Research Letters*, 26, pp. 33-41.

LeBlanc L.J. (1973), *Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems*, PhD Thesis, Department of Industrial and Management Sciences, Northwestern University, Evanston IL.

Patriksson M. (1994), *The Traffic Assignment Problem: Models and Methods*, VSP, Utrecht, the Netherlands.

Thompson W.A. and Payne H.J. (1975), Traffic Assignment on Transportation Networks with Capacity Constraints and Queuing, *Paper presented at the 47th National ORSA/TIMS North American Meeting,* Chicago IL.

Ravindran A., Phillips D.T. and Solberg J.J. (1987), *Operations Research: Principles and Practice* (2nd ed.), John Wiley & Sons, New York NY.

Sheffi Y. (1985), *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, Englewood Cliffs NJ.

Smith M.J. (1979), The Existence, Uniqueness and Stability of Traffic Equilibria, *Transportation Research Part B,* 13, pp. 295-304.

Smith M.J. (1987), *Traffic Control and Traffic Assignment in a Signal-Controlled Network with Queueing, Transportation and Traffic Theory*, Gartner N.H., Wilson N.H.M., eds., Elsevier Science, New York NY, pp. 61-77.

Tatineni M., Edwards H. and Boyce D. (1998), Comparison of the Disaggregate Simplicial Decomposition and Frank-Wolfe Algorithms for User-Optimal Route Choice, *Transportation Research Record,* 1617, pp. 157-162.

Yang H. and Yagar S. (1994), Traffic Assignment and Traffic Control in General Freeway Arterial Corridor Systems, *Transportation Research Part B,* 28, pp. 463-486.

Yang H. and Yagar S. (1995), Traffic Assignment and Signal Control in Saturated Road Networks, *Transportation Research Part A,* 29, pp. 125-139.