Annals of Operations Research 34(1992)163-176

EFFICIENT IMPLEMENTATION OF HEURISTICS FOR THE CONTINUOUS NETWORK DESIGN PROBLEM*

Patrice MARCOTTE and Gérald MARQUIS
Département de Mathématiques, Collège Militaire Royal de Saint-Jean, Richelain, Québec, Canada J0J 1R0

Abstract

In this paper, we present an efficient implementation of heuristic procedures for solving the continuous network design problem where network users behave according to Wardrop's first principle of traffic equilibrium. Numerical results involving a "standard" benchmark problem are given. Also, it is shown that the cost mapping arising in the Iterative-Optimization-Assignment algorithm is integrable if and only if the volume-delay function is of either the BPR or some logarithmic form.

1. Introduction

In this paper, we address a network design problem involving design parameters, referred to as capacities, and private vehicle flows. It is assumed that the network is subject to congestion and that users behave according to Wardrop's first principle of traffic equilibrium, i.e. that, at equilibrium, no user can decrease his/her travel time by a unilateral change of route. This formulation is similar to that used in Abdulaal and LeBlanc [1], Friesz, Suwansirikul and Tobin [2], or Marcotte [8,9]. In the latter papers, several efficient heuristics were compared on networks where all arcs were subject to improvements and had zero capacities. In this paper, we show that those heuristics perform equally well on networks where only a subset of the arcs are considered for improvement, and might have positive initial capacities. The main difference to the previous analysis results from the fact that the capacity-flow relationship is not as simple, and might even be nondifferentiable. The paper is organized as follows. First we give a mathematical formulation of the problem, then we describe the algorithms and their implementation. Next, numerical results on a small network and the benchmark "Sioux Falls" network are presented. Finally, we give necessary and sufficient conditions under which the popular Iterative-Optimization-Assignment (IOA) algorithm yields implicitly a descent direction for a related convex objective function.

*Research supported by the National Sciences and Engineering Research Council of Canada (Grant A5789) and the Academic Research Program of the Department of National Defense (Grant FUHBP).

2. Notation and problem formulation

Consider a transportation network $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a node set and $\mathcal{A}$ an arc set. To each pair of nodes $k = (i, j) \in \mathcal{N} \times \mathcal{N}$ is associated a demand δ^k. A multicommodity flow vector $F = \{f_a^k\}_{a \in \mathcal{A}, k \in \mathcal{N} \times \mathcal{N}}$ is *feasible* if it is nonnegative and satisfies the flow conservation equations:

$$\sum_{j \in \mathcal{N}} f^k_{a=(i,j)} - \sum_{j \in \mathcal{N}} f^k_{a=(j,i)} = \begin{cases} -\delta^k, & \text{if } k = (i, l); \\ \delta^k, & \text{if } k = (l', i); \\ 0 & \text{otherwise,} \end{cases}$$

where $l \in \mathcal{N}$ represents a destination node and $l' \in \mathcal{N}$ an origin node of the network. The polyhedron of feasible flow vectors F will be denoted by the symbol Φ.

We will make use of the following notation:

$f_a = \sum_{k \in \mathcal{N} \times \mathcal{N}} f_a^k$: total flow on arc a;

$f = \{f_a\}_{a \in \mathcal{A}}$: total flow vector;

y_a : capacity of arc a;

$y = \{y_a\}_{a \in \mathcal{A}}$: capacity vector;

$C_a(f, y)$: congestion (delay) function on arc a;

$C = \{C_a\}_{a \in \mathcal{A}}$: congestion vector.

For a given capacity vector y, a multicommodity flow vector $F(y)$, together with the total flow vector $f(y)$, is in equilibrium (see Smith [12] or Dafermos [2]) if it satisfies the variational inequality:

$$\textbf{(VIP)} \quad \langle f(y) - f, C(f(y), y) \rangle \le 0 \qquad \text{for all feasible total flow vectors } f.$$

The continuous network design problem (**NDP**) consists of choosing a capacity vector y in a feasible set Y which is optimal for the generalized bilevel program:

$$\textbf{(NDP)} \quad \underset{y \in Y}{\text{minimize}} \ \langle f(y), C(f(y), y) \rangle \quad \{\text{transportation cost}\}$$
$$+ \lambda g(y) \quad \{\text{investment cost}\},$$

where g represents the cost associated with the capacity vector y, $f(y)$ the equilibrium total flow corresponding to y, and λ a time/money conversion factor. The problem **NDP** is well defined if the total flow f corresponding to the F-solution to **VIP** is unique. A sufficient condition for uniqueness is that the cost function $C(f, y)$ be strictly monotone in f.

In this paper, we will make the following assumptions:

ASSUMPTION 1

C_a is a positive, strictly increasing and continuously differentiable function of the ratio f_a/y_a.

ASSUMPTION 2

g is separable, i.e. $g(y) = \sum_{a \in \mathcal{A}} g_a(y_a)$, where each function g_a is nonnegative, increasing and twice continuously differentiable.

Under the previous assumptions, **NDP** can be written as the bilevel programming problem:

$$(\mathbf{BLP}) \qquad \underset{F \in \Phi, y \in Y}{\text{minimize}} \sum_{a \in \mathcal{A}} f_a C_a(f_a/y_a) + \lambda g_a(y_a) \qquad \{\text{upper level}\}$$

$$\text{subject to } F \in \arg\min_{F \in \Phi} \sum_{a \in \mathcal{A}} \int_0^{f_a} C_a(t/y_a)\, dt \qquad \{\text{lower level}\}.$$

It is also possible to express **NDP** as a mathematical program with nonlinear, nonconvex but finitely many constraints (see Marcotte [9]).

3. Heuristics for BLP

NDP is theoretically a difficult problem. Indeed, its objective is nonconvex and nondifferentiable. Furthermore, each evaluation of the reaction function $F(y)$ requires the *exact* solution of a fixed demand traffic assignment problem. Below, we describe two members of the family of algorithms previously introduced by Marcotte [8,9].

(1) Consider the normative mathematical program:

$$\min_{F \in \Phi, y \in Y} \sum_{a \in \mathcal{A}} f_a C_a(f_a/y_a) + \lambda g_a(y_a). \qquad (\text{NORM})$$

This is a convex program whose optimal value obviously provides a lower bound for **BLP**. Let $(f^{\text{NORM}}, y^{\text{NORM}})$ be optimal for NORM. We can then extract the suboptimal solution:

$$(f^0, y^0) \stackrel{\text{def}}{=} (f(y^{\text{NORM}}), y^{\text{NORM}})$$

by performing a traffic assignment on the network with capacity vector y^{NORM}. A sufficient condition under which (f^0, y^0) is optimal for **BLP** is given in Marcotte [8].

(2) Consider the convex mathematical program:

$$\min_{f \in \Phi, y \in Y} \sum_{a \in \mathcal{A}} \int_0^{f_a} C_a(t/y_a)\mathrm{d}t + \xi\lambda g_a(y_a) \tag{H5}$$

whose solution is (f^5, y^5). By construction of H5, we have that $f^5 = f(y^5)$, i.e. that f^5 is the equilibrium total flow corresponding to the capacity vector y^5. The reason for the code name H5 is historical (see Marcotte [8]).

4. An efficient implementation of NORM and H5

We will concentrate on the implementation of H5. The numerical implementation of NORM is similar, with the exception of an added traffic assignment on the network with capacity vector y^{NORM}.

The constraint $y \in Y$ on the capacity vector will in general assume a simple form, such as $y_a \geq 0$. We make:

ASSUMPTION 3

$Y = \{y \mid y_a \geq 0, a \in \mathcal{A}1, y_a = 0, a \in \mathcal{A}2\}$, where $\mathcal{A}1$ represents the set of arcs subject to capacity improvement and $\mathcal{A}2 = \mathcal{A} - \mathcal{A}1$.

It is then possible, for fixed total flow vector f, to solve optimally for y as a function of f. More precisely, one has, for each $a \in \mathcal{A}1$, to solve the equation:

$$\frac{\partial}{\partial y_a} \phi_a(y_a) = 0,$$

where

$$\phi_a(f_a, y_a) \stackrel{\text{def}}{=} \int_0^{f_a} C_a(t/y_a)\mathrm{d}t + \xi\lambda g_a(y_a). \tag{1}$$

Note that $\phi_a(f_a, y_a)$ is strictly convex. Indeed, the Hessian matrix of $\int_0^{f_a} C(t/y_a)\mathrm{d}t$ is the positive semidefinite matrix

$$C_a'(f_a/y_a)\begin{pmatrix} 1/y_a & -f_a/y_a^2 \\ -f_a/y_a^2 & f_a^2/y_a^3 \end{pmatrix}. \tag{2}$$

Strict convexity then follows from the strict convexity of g_a. We will denote by $y_a(f_a)$ the unique positive solution to the equation:

$$\frac{\partial}{\partial y_a} \phi_a(f_a, y_a) = 0.$$

Remark

From the definition of the objective function, it is implicit that the capacity of an arc can actually be decreased below its present value at no extra cost. This could actually improve the design in certain circumstances such as the Braess paradox situation. □

After substituting expression (2) for y_a in H5, we obtain the network constrained convex program:

$$\min_{F \in \Phi} \sum_{a \in \mathcal{A}} \int_0^{f_a} C_a(t/y_a(f_a))\mathrm{d}t + \xi\lambda g_a(y_a(f_a)) \tag{3}$$

that can be solved by any standard multicommodity convex flow algorithm, most of which are based on the Frank–Wolfe algorithm or some of its variants (PARTAN [4], simplicial decomposition [7], Fukushima's algorithm [5], etc.). The function $y_a(f_a)$ is often, in practice, nondifferentiable, due to possible non-differentiability of the function $g_a(y_a)$; this occurs for instance when improvements

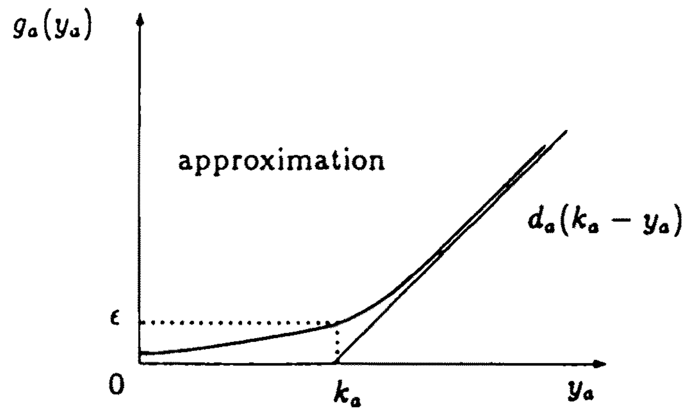

Fig. 1. Linear investment function.

are made on an existing network (see fig. 1). This difficulty is best handled by approximating g_a by a smooth function. The most important case arises when g_a is piecewise linear:

$$g_a(y_a) = \max\{0, d_a(y_a - k_a)\}. \tag{4}$$

In this case, since g_a is not continuously differentiable, we use as an approximating function the branch of hyperbola whose equation is given by:

$$\frac{d_a}{2}\left[y_a - k_a + \sqrt{\varepsilon^2 + (y_a - k_a)^2}\right], \tag{5}$$

where ε represents a small positive number and k_a the initial capacity of arc a (see fig. 1). Henceforth, it is assumed that such an approximation has been performed, and hence that $g_a(y_a)$ is a twice continuously differentiable function. It is to be noted that the nonnegativity constraint $y_a \geq 0$ is automatically enforced.

Remark

An implicit advantage of the approximation scheme is that, whenever given the choice, the algorithm will favor *low* over *high* capacities. For instance, if some sort of Braess paradox phenomenon occurs, the algorithm could prevent it by lowering the capacities of arcs unused in the system-optimal solution. □

The minimization of (3) has been achieved using the linearization algorithm of Frank and Wolfe adapted to the traffic assignment problem (see LeBlanc et al. [11]). When not available in closed form, the function $y_a(f_a)$ was evaluated using the method of the false position ("regula falsi"). The first derivative of $y_a(f_a)$ was obtained by implicit differentiation of ϕ_a:

$$\frac{\partial \phi_a}{\partial f_a}(f_a, y_a(f_a))y_a'(f_a) + \frac{\partial \phi_a}{\partial y_a}(f_a, y_a(f_a)) = 0. \tag{6}$$

One step of the algorithm is summarized below:

(1) *Find a descent direction.* Let

$$\bar{F} \in \arg\min_{F \in \Phi} \psi(f) \stackrel{\text{def}}{=} \sum_{a \in \mathcal{A}} f_a \cdot \frac{\partial}{\partial x}\left[\int_0^x C_a(t/y_a(x))\,dt + \xi\lambda g_a(y_a(x))\right]_{x=f_a} \tag{7}$$

and $\bar{f}$ be the total flow vector corresponding to the multicommodity vector $\bar{F}$.

(2) *Perform linesearch* (regula falsi or safeguarded Newton). Let

$$\alpha \in \arg\min_{\theta \in [0,1]} \psi[f + \theta(\bar{f} - f)],$$

$$f \leftarrow f + \alpha(\bar{f} - f).$$

The algorithm is stopped as soon as the gap function (see Hearn [6]) becomes less than a predetermined small value.

The computational burden of the algorithm consists mainly of two things:

(1) Solving eq. (7) by shortest path methods. This can be achieved in $O(Kn^2)$ or $O(Km \log n)$ operations (where K denotes the number of origins or destinations, n the number of nodes in the network, and m the number of arcs) using Dijkstra's label setting algorithm.

(2) Evaluating $y_a(f_a)$ and $y_a'(f_a)$ for $a \in \mathcal{A}1$ in the linesearch. This implies $O(Lm_1)$ operations, where L denotes the (usually small) number of regula falsi steps.

The overall one-step complexity is therefore $O(Kn^2 + Lm_1)$ or $O(Kn \log n + Lm_1)$ if one implements Dijkstra's algorithm using a heap structure. The latter implementation is more efficient if m is of the order of n, which is usual for urban transportation networks. If the network is large and m_1 small, two things likely to occur in practice, then the term $Km \log n$ dominates, and solving H5 is computationally comparable to performing a *single* traffic assignment on the network.

5. Numerical examples

In the following test problems, the congestion functions are of the BPR type, i.e.:

$$C_a(f_a, y_a) = \alpha_a + \beta_a(f_a/y_a)^p,$$

where α_a is usually referred to as the free flow travel time on arc a. It is worth noting that, in this particular case, heuristic H5 subsumes the Iterative-Optimization-Assignment algorithm (see remark at the end of the appendix), the latter corresponding to $\xi = 1/(p+1)$. In the appendix, it is shown that the *only* functional forms for which the IOA actually solves (implicitly) a convex program are the BPR form and the logarithmic form.

In the test problems, investment functions are either linear or quadratic. As mentioned previously, linear investment functions are approximated by branches of hyperbola. In the quadratic case, we have:

$$g_a(y_a) = \min\{0, d_a(y_a - k_a)^2\}, \qquad (8)$$

which is once but not *twice* continuously differentiable. However, we can replace (8) by the smooth function

$$g_a(y_a) = d_a(y_a - k_a)^2$$

without modifying the optimal solution, since values of y_a less than k_a are dominated by $y_a = 0$ in the solution of (7) (see fig. 2).

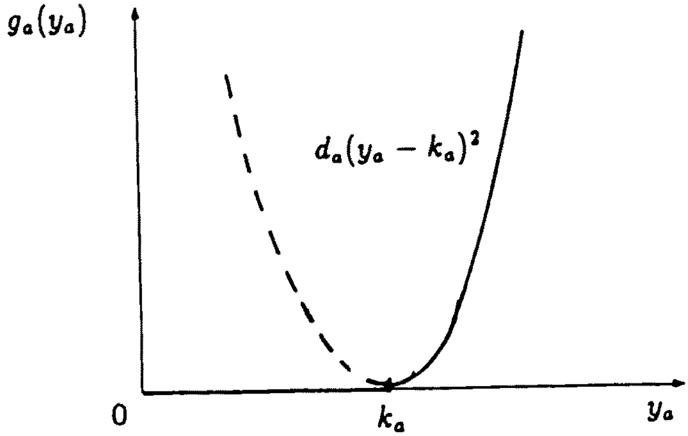

Fig. 2. Quadratic investment function.

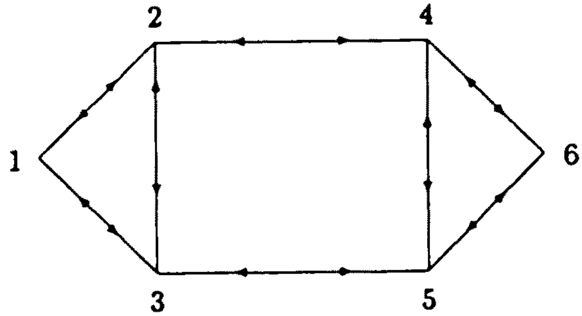

Fig. 3. Network for the first example.

The first example, taken from [3], consists of the network shown in fig. 3. Demand is equal to D from node 1 to node 6, and $2D$ from node 6 to node 1. Coefficients α_a, β_a, k_a and l_a (the investment functions are linear) are given in [3]; λ has been set to 1. The first column (EDO) corresponds to the Equilibrium Decomposed Optimization algorithm of Friesz et al. [3].

The objective function values in table 1 differ slightly from those reported in [3]. Our values have been computed by performing a very accurate assignment, using the y-values given in [3]. The precision with which the assignment has been performed might also explain the (small) discrepancies observed between the IOA results given in [3] and the results obtained by setting ξ to 0.2 in H5. These results should be identical.

Algorithms NORM and H5 were stopped after 50 iterations. At this point, the relative gap value was less than 0.003. On this small example, the best results were

Table 1

Objective function values and number of Frank–Wolfe iterations for the first example

Demand	EDO	IOA	NORM	H5 $\xi = 0.2$	H5 $\xi = \xi^{opt}$
2.5	91.8	100.4	91.07	101.2	93.5 ($\xi = 0.45$)
iter →	(57)	(50)	(50)	(50)	(50)
5	200.89	214.3	199.7	213.3	213.3 ($\xi = 0.3$)
iter →	(71)	(50)	(50)	(50)	(50)
10	540.1	557.8	534.8	559.6	558.7 ($\xi = 0.22$)
iter →	(68)	(50)	(50)	(50)	(50)

consistently obtained by NORM. It is to be noted that the number of Frank–Wolfe steps could be lowered significantly without greatly affecting the quality of the solution.

The second example, first considered by Abdulaal and LeBlanc [1], was also considered in [3], where the complete data may be found. It is the network of the city of Sioux Falls and is composed of 24 nodes, 76 arcs, and the origin–destination matrix is full. A subset of 10 arcs is subject to improvements. Investment functions are quadratic. An absolute lower bound of 79.53 has been obtained by solving the system-optimal problem. Numerical results are summarized in table 2.

Table 2

Results for the Sioux Falls network

	EDO	HJ1 [a]	HJ2 [a]	NORM	H5 $\xi = 0.25$	H5 $\xi = 0.2$
objective	83.68	82.32	82.06	81.61	81.78	81.84
iter →	(89)	(247)	(147)	(56)	(54)	(53)

[a] Results obtained by Hooke and Jeeves algorithm as reported in [3].

The names HJ1 and HJ2 refer to two different settings for the upper bounds of the improvement vectors (see [3]). The best solution was obtained by the heuristic NORM with an objective value very close to the lower bound. The improvement vector $(y - k)$'s entries were: (5.48, 2.08, 5.49, 2.09, 2.85, 4.23, 4.27, 4.24, 4.27) to be contrasted with the solution (4.59, 1.52, 5.45, 2.33, 1.27, 2.33, 0.41, 4.59, 2.71, 2.71) reported in [3]. For heuristic NORM, about one half of the iterations were used to solve the normative problem and the remaining half to obtain a very accurate (0.003 relative gap value) feasible (equilibrium) solution.

To assess the efficiency of algorithm H5, it was decided to check the quality of the solution obtained when stopping the procedure prematurely. For a given

Table 3

Evolution of optimal solution for algorithm H5 ($\xi = 0.25$)

Iter →	5	10	15	20	30	40
y_1	4.41	4.75	4.73	4.83	4.82	4.85
y_2	1.90	1.50	1.39	1.47	1.36	1.33
y_3	4.36	4.60	4.72	4.82	4.87	4.85
y_4	1.45	1.58	1.40	1.44	1.36	1.32
y_5	2.62	2.30	2.48	2.37	2.42	2.39
y_6	2.27	2.47	2.35	2.40	2.40	2.41
y_7	4.73	4.36	4.28	4.21	4.22	4.20
y_8	4.23	4.06	4.11	4.05	4.02	4.23
y_9	4.75	4.47	4.38	4.27	4.21	4.23
y_{10}	4.27	4.07	4.04	4.04	3.98	3.96
objective	82.04	81.81	81.79	81.79	81.81	81.80

iteration count, the objective has been computed to within 0.003 relative gap. The results are given in table 3 for H5 ($\xi = 0.25$).

From the above results, it is clear that a very good solution can be obtained with very little effort (5 to 10 iterations). Actually, the quality of the solution decreases when the number of Frank–Wolfe iterations increases from 20 to 30. This was to be expected, since the problem solved by H5 is related but distinct from the original bilevel program. Similar results were obtained for heuristic NORM. Finally, table 4 illustrates the behavior of the objective as a function of the parameter ξ. The resulting curve is convex-shaped. This feature could be exploited if one is looking for the optimal value of ξ.

Table 4

Evolution of the objective of algorithm H5 as a function of ξ

ξ :	0.15	0.20	0.25	0.30	0.35	0.40	0.45	1.0
objective :	82.38	81.84	81.78	81.95	82.20	82.59	82.83	86.18

Remark

The algorithm was also tested on a five-arc network (Braess paradox network). Heuristic H5, with ξ set to 0.2, converged in three iterations to the optimal solution obtained using the MINOS nonlinear programming code, and reported in [3]. On this same example, slightly suboptimal results were obtained by EDO and Hooke–Jeeves in a number of iterations ranging from 24 to 38. □

6. Conclusions

In this paper, we showed that algorithms initially designed for solving **NDP**, where all arcs are subject to improvement and have zero initial capacities, could also solve very efficiently **NDP**s under more general conditions. The efficiency of the proposed methodology relies on deriving, in explicit or implicit form, the functional relationship relating optimal capacities to flows, thus reducing the initial problem to a convex traffic assignment problem. The heuristics are robust and easy to implement. In particular, no initial lower or upper bounds on the design variables are required. Indeed, the number of Frank–Wolfe steps required to obtain a good solution is so low that, facing a difficult problem, one should solve **NDP** using NORM and H5 with several different values for the parameter ξ.

Appendix

INTEGRABILITY OF THE COST FUNCTION ARISING IN THE IOA ALGORITHM

The IOA algorithm consists of iteratively repeating the following two steps:

(1) *Optimization step*

$$\min_{y \in Y} \sum_{a \in \mathcal{A}} f_a C_a(f_a / y_a) + \lambda g_a(y_a).$$

(2) *Assignment step*

$$\min_{F \in \Phi} \sum_{a \in \mathcal{A}} \int_0^{f_a} C_a(t/y_a)\,\mathrm{d}t.$$

The IOA algorithm can be viewed as a block Gauss–Seidel scheme for finding a solution (y^*, f^*) to the variational inequality

$$\sum_{a \in \mathcal{A}} (y_a^* - y_a)\left[-\left(\frac{f_a}{y_a}\right)^2 C_a'\left(\frac{f_a^*}{y_a^*}\right) + \lambda g_a'(y_a^*)\right] \quad \text{(optimality conditions)}$$

$$+\,\gamma \sum_{a \in \mathcal{A}} (f_a^* - f_a) C_a\left(\frac{f_a^*}{y_a^*}\right) \quad \text{(equilibrium conditions)} \qquad (9)$$

$$\leq 0 \quad \forall y \in Y, F \in \Phi,$$

where γ is an arbitrary positive constant. Indeed, the solution set to (9) remains unchanged, and the IOA iterates are unchanged, if the second inequality is premultiplied by γ. This merely multiplies all path lengths by a positive constant.

We want to show that the only functional forms for which the cost function of the variational inequality (9) is integrable, i.e. is the gradient of some function convex in both y and f, are of the BPR form:

$$\alpha_a + \beta_a(f_a/y_a)^p, \tag{10}$$

where p is a nonnegative constant identical for *all* arcs of the network, and the logarithmic form:

$$\alpha_a + \beta_a \ln (f_a/y_a).$$

A direct consequence of this result is that, if the delay functions are of the form (10), then IOA converges from an arbitrary feasible starting point (y^0, f^0) to the Nash equilibrium solution corresponding to (9).

The cost function of (9) will be integrable if there exist functions $\phi_a(f_a, y_a)$ satisfying the system of partial differential equations:

$$\begin{aligned}\frac{\partial \phi_a}{\partial y_a} &= -\left(\frac{f_a}{y_a}\right)^2 C_a'\left(\frac{f_a}{y_a}\right) + \lambda g_a'(y_a),\\ \frac{\partial \phi_a}{\partial f_a} &= \gamma C_a\left(\frac{f_a}{y_a}\right).\end{aligned} \tag{11}$$

To simplify the notation, we omit the separable term $g_a'(y_a)$ and the arc indices from (11) and obtain the system:

$$\frac{\partial \phi}{\partial y} = -\left(\frac{f}{y}\right)^2 C'\left(\frac{f}{y}\right), \quad \frac{\partial \phi}{\partial f} = \gamma C\left(\frac{f}{y}\right). \tag{12}$$

Let $h(x) \stackrel{\text{def}}{=} \int_0^x C(t)\mathrm{d}t$. We have:

$$\begin{aligned}\phi(f, y) &= \gamma \int_0^f C\left(\frac{t}{y}\right) \mathrm{d}t\\ &= \gamma y\, h\left(\frac{f}{y}\right) + K(y).\end{aligned}$$

Hence:

$$\begin{aligned}\frac{\partial \phi}{\partial y} &= \gamma h\left(\frac{f}{y}\right) - \gamma \frac{f}{y}\, h'\left(\frac{f}{y}\right) + K'(y)\\ &= -\left(\frac{f}{y}\right)^2 h''\left(\frac{f}{y}\right).\end{aligned}$$

From the above identity, we have $K'(y) = 0$. Also, after setting $x = f/y$, we obtain the Euler equation:

$$x^2 h''(x) - \gamma x h'(x) + \gamma h(x) = 0, \tag{13}$$

which admits the two linearly independent solutions:

$$h_1(x) = x, \qquad h_2(x) = x^{1/\gamma}.$$

For $\gamma \neq 1$, the general solution to (13) is:

$$h(x) = \alpha x + \beta x^{\gamma}$$

and we have:

$$C(x) = h'(x) = \alpha + \beta x^{\gamma - 1},$$

which is of the form (10) with $p = \gamma - 1$. Also, p must be nonnegative since $C_a(f_a/y_a)$ has to be an increasing function of f_a over the nonnegative axis.

If $\gamma = 1$, then the general solution to (13) is:

$$h(x) = \alpha x + \beta x \ln x$$

and:

$$C(x) = \alpha + \beta + \beta \ln x,$$

which is of the required logarithmic form. □

Remark

If delay functions are of the BPR type with exponent p, or of the logarithmic form $\alpha + \beta \ln x$ $\gamma = p + 1$, we have $\gamma = p + 1$ and IOA solves the optimization problem:

$$\min \sum_{a \in \mathcal{A}} \int_0^{f_a} C_a(t/y_a)\,\mathrm{d}t + \frac{1}{p+1} \lambda g_a(y_a),$$

which is a member of the H5 family of optimization problems. We therefore conclude that H2 is subsumed by H5 in those cases. Finally, note that the logarithmic form yields negative values for small flows, and is therefore of limited interest.

References

[1] M. Abdulaal and L.J. LeBlanc, Continuous equilibrium design models, Transport. Res. 13B(1979) 19–32.

[2] S.C. Dafermos, Traffic equilibrium and variational inequalities, Transport. Sci. 14(1980)42–54.

[3] T.L. Friesz, C. Suwansirikul and R.L. Tobin, Equilibrium Decomposition Optimization: A heuristic for the continuous equilibrium network design problem, Transport. Sci. 21(1987)254–263.
[4] M. Florian, J. Guélat and H. Spiess, An efficient implementation of the PARTAN variation of the linear approximation method for the network equilibrium problem, Networks 17(1987)319–340.
[5] M. Fukushima, A modified Frank–Wolfe algorithm for solving the traffic assignment problem, Transport. Res. 18B(1984)169–177.
[6] D.W. Hearn, The gap function of a convex program, Oper. Res. Lett. 1(1981)67–71.
[7] S. Lawphongpanich and D.W. Hearn, Simplicial decomposition of the asymmetric traffic assignment problem, Transport. Res. 18B(1984)123–133.
[8] P. Marcotte, Network optimization with continuous control parameters, Transport. Sci. 17(1983) 181–197.
[9] P. Marcotte, Network design problem with congestion effects: A case of bilevel programming, Math. Progr. 34(1986)142–162.
[10] P. Marcotte and J. Guélat, A modified Newton method for solving the asummetric traffic equilibrium problem, Transport. Sci. 22(1988)112–124.
[11] L. LeBlanc, E. Morlok and W. Pierskalla, An efficient approach to solving road network equilibrium traffic assignment problems, Transport. Res. 9B(1975)309–318.
[12] M.J. Smith, The existence, uniqueness and stability of traffic equilibria, Transport. Res. 13B(1979) 295–304.