17

An Algorithm for the Combined Distribution and Assignment Model

Jan T. Lundgren

Department of Mathematics, Linköping University

S-581 83 Linköping, Sweden

Michael Patriksson

Department of Mathematics, Chalmers University of Technology

S-412 96 Göteborg, Sweden

Abstract

We consider the problem of simultaneously determining the distribution of trips between origins and destinations in a transportation network and the assignment of trips to routes in each origin–destination pair. We consider a model for such a problem where the distribution follows a gravity model and the assignment a user equilibrium model. The most well-known algorithm for this model is that of Evans (1976). Although this algorithm has been shown to be more efficient than the Frank–Wolfe method, which it generalizes, it builds the sequence of link flow solutions based on the same underlying algorithmic principle, and therefore also is subject to slow convergence. We propose to combine Evans' approach with the disaggregate simplicial decomposition (DSD) algorithm for updating the link flows; this leads to faster convergence, as well as other improvements. For example, the algorithm is less sensitive to inexact solutions of the entropy maximization subproblems; it further provides explicit route flows in each of the origin–destination pairs. Numerical results are presented for four small and medium scale network models.

1 Introduction

We consider the problem of simultaneously determining the distribution of trips between origins and destinations in a transportation network and the assignment of trips to routes in each origin–destination (OD) pair. We consider a model for such a problem, first formulated by Evans (1973 and 1976), where the distribution follows a gravity model with negative exponential deterrence function, and the assignment follows a classical user equilibrium model, as described by Wardrop (1952) and formalized in a mathematical programming form by Beckmann et al. (1956).

Evans (1976) presents an iterative algorithm for this combined distribution and assignment (CDA) problem. One iteration proceeds as follows. Given a feasible demand and link flow, together with the associated link travel costs, a shortest route and its cost is computed for each OD pair. Given these costs, a doubly constrained entropy maximization problem is then solved to obtain an auxiliary OD flow, and an auxiliary link flow is obtained by assigning the auxiliary demands onto the shortest routes. A one-dimensional minimization is then performed on the line segment between the current and the auxiliary demand and link flow solutions, resulting in a new feasible solution, and the procedure is repeated until the solution is near-optimal.

Evans' algorithm generalizes the method of Frank and Wolfe (1956) applied to the CDA problem, in that the auxiliary problem is the result of an approximation of the objective function by means of a *partial* linearization only, and has been observed to be more efficient (e.g., Frank, 1978; and LeBlanc and Farhangian, 1981). The difference in the behaviour of the two methods is, however, often rather small, for reasons that will be explained in more detail in the next section, and Evans' algorithm therefore also suffers from slow convergence.

As a means to improve the convergence rate, we propose to combine Evans' algorithm with the disaggregate simplicial decomposition (DSD) method of Larsson and Patriksson (1992), which was originally proposed for the solution of user equilibrium models. In this algorithm, all the routes generated in the shortest route computations are retained and, instead of as in the Frank–Wolfe and Evans algorithms rerouting flow towards one route flow pattern only, it selects the optimal flow among all possible combinations of route flows, given the set of routes generated. Its established efficiency in applications to user equilibrium models suggests the possibility of a successful application also to the CDA model.

The rest of the paper is organized as follows. In the next section, we present the CDA model, describe Evans' algorithm, and discuss some its characteristics. In Section 3, the new algorithm is presented. Numerical experiments conducted with the algorithm are reported in Section 4. The paper is concluded with a summary of our experience with the method, and a discussion on possible extensions of it to more general models.

2 The combined distribution and assignment model

The urban area under study is represented by a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. The set $\mathcal{N}$ of nodes contains subsets corresponding to the origin zones (denoted by $\mathcal{P}$) and destination zones ($\mathcal{Q}$), in addition to the nodes representing the road intersections. Each (directed) link $a \in \mathcal{A}$ is associated with a generalized travel cost, $c_a(f_a)$, representing the disutility of using link a as a function of its flow f_a. This cost may include several additive components, perhaps most importantly the travel time on the link, and it is assumed that, as a result of congestion, c_a is a positive, continuous and strictly increasing function of the flow on link a.

Between a subset of nodes in $\mathcal{P}$ and $\mathcal{Q}$ there is a potential demand for transportation; for notational simplicity we will however assume that *all* pairs $(p,q) \in \mathcal{P} \times \mathcal{Q}$ define potential OD pairs. The (marginal) total flow emanating from an origin node $p \in \mathcal{P}$ is known, and equals O_p; correspondingly, the (marginal) total flow terminating at a node $q \in \mathcal{Q}$ equals D_q. For each OD pair (p,q), we denote the set of simple (loop-free) routes from p to q by $\mathcal{R}_{pq}$—this is a set which in general is not known explicitly—and the flow on route r from p to q by h_{pqr}. By defining a link–route incidence matrix (δ_{pqra}) for $\mathcal{G}$, that is, $\delta_{pqra} = 1$ if route $r \in \mathcal{R}_{pq}$ contains link a, and $\delta_{pqra} = 0$ otherwise, link flows may be calculated from route flows according to the relation

$$f_a = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr}.$$

We assume that the assignment of travellers to routes in each OD pair (p,q) satisfies the user equilibrium conditions of Wardrop (1952),

$$h_{pqr} > 0 \implies c_{pqr} = \pi_{pq}, \qquad \forall r \in \mathcal{R}_{pq}, \tag{2.1a}$$
$$h_{pqr} = 0 \implies c_{pqr} \geq \pi_{pq}, \qquad \forall r \in \mathcal{R}_{pq}, \tag{2.1b}$$

where $c_{pqr} = \sum_{a\in\mathcal{A}} \delta_{pqra} c_a(f_a)$ is the cost on route r, and π_{pq} is the least route cost; the Wardrop conditions (2.1) state that only the routes of least cost are used.

We also assume that the distribution of travellers between OD pairs follows a gravity model with negative deterrence function, according to

$$d_{pq} = a_p O_p b_q D_q \exp(-\gamma \pi_{pq}), \qquad p \in \mathcal{P}, \quad q \in \mathcal{Q}, \tag{2.2}$$

where $\gamma > 0$ is the dispersion parameter, and a_p and b_q are positive balancing factors, determined so that (2.2) satisfies the marginal total constraints.

The combined distribution and assignment model can be formulated as the following convex optimization problem:

[CDA]

$$\nu^* := \min \; T(d,f) = \sum_{a\in\mathcal{A}} \int_0^{f_a} c_a(s)\, ds + \frac{1}{\gamma} \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} d_{pq} \log d_{pq} \tag{2.3a}$$

subject to

$$\sum_{r\in\mathcal{R}_{pq}} h_{pqr} = d_{pq}, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{Q}, \tag{2.3b}$$
$$\sum_{q\in\mathcal{Q}} d_{pq} = O_p, \quad \forall p \in \mathcal{P}, \tag{2.3c}$$
$$\sum_{p\in\mathcal{P}} d_{pq} = D_q, \quad \forall q \in \mathcal{Q}, \tag{2.3d}$$
$$h_{pqr} \geq 0, \quad \forall r \in \mathcal{R}_{pq}, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{Q}, \tag{2.3e}$$
$$\sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} = f_a, \quad \forall a \in \mathcal{A}. \tag{2.3f}$$

This formulation was first proposed by Evans (1973 and 1976), combining works of Beckmann et al. (1956), Murchland (1966) and Wilson (1967). Its optimal solution satisfies the conditions (2.2) for a gravity model, as well as the user equilibrium conditions (2.1), for all OD pairs. In an application, the value of the parameter γ must be calibrated through some estimation procedure (e.g., through a maximum likelihood estimation); this estimation issue has been considered by Erlander et al. (1979), Fisk and Boyce (1983) and Boyce et al. (1983).

Evans (1973 and 1976) proposes and establishes the convergence of a solution algorithm for the problem; the algorithm is described in Table 2.1.

Table 2.1: Evans' algorithm

---

0. (*Initialization*): Choose an initial feasible solution (d^0, f^0), and let $k = 0$.

1. (*Search direction generation*):

- **a**. Given are the fixed link costs $c_a(f_a^k)$. For each OD pair (p, q), compute a shortest route and its corresponding route cost, π_{pq}^k.
- **b**. Solve the doubly constrained entropy maximization problem of minimizing $\underline{T}(d) := \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} (\pi_{pq}^k d_{pq} + \frac{1}{\gamma} d_{pq} \log d_{pq})$ subject to (2.3c)–(2.3d) and $d \geq 0$. The resulting auxiliary demand is $\overline{d}^k$. Assign this demand onto the shortest routes computed in **a**; the resulting auxiliary link flow is $\overline{f}^k$.

  The resulting search direction is $(\overline{d}^k, \overline{f}^k) - (d^k, f^k)$.

2. (*Line search*): Solve $\min_{\ell\in[0,1]}\{T[(d^k, f^k) + \ell(\overline{d}^k - d^k, \overline{f}^k - f^k)]\}$. The resulting step length is ℓ_k.

3. (*Update*): Let $(d^{k+1}, f^{k+1}) := (d^k, f^k) + \ell_k(\overline{d}^k - d^k, \overline{f}^k - f^k)$. Let $k := k + 1$.

4. (*Termination criterion*): If some termination criterion is satisfied, then stop. Otherwise, go to Step **1**.

Remark 2.1 (Notes on Evans' algorithm) **(i)** An initial OD matrix satisfying (2.3c)–(2.3d) and $d \geq 0$ can be determined through a simple feasibility heuristic for linear transportation problems, or be based on a predicted or historical OD matrix. An initial feasible link flow can be given by an all-or-nothing assignment of this initial OD matrix based on the link costs at zero flow, $c_a(0)$, $a \in \mathcal{A}$.

(ii) The subproblem in Step **1** is the result of a partial linearization of the original objective. The problem in **1.b** is traditionally solved using a dual algorithm, which can be interpreted as a balancing procedure with respect to the marginal constraints (2.3c)–(2.3d); see, for example, Lamond and Stewart (1981). Being a dual algorithm, the procedure does not provide a feasible demand matrix if the problem is not solved exactly. The accuracy by which this subproblem is to be solved is a crucial point in the implementation of the algorithm, since errors made in the calculation of the demand vector result in a primal infeasibility, which will accumulate both in the demand and link flow vectors.[1]

[1] An infeasible OD matrix calculated in the balancing procedure can be adjusted to a feasible one by

(iii) In an implementation of the Frank–Wolfe algorithm, the subproblem in **1.b** is a linear transportation problem; see Florian et al. (1975) for an application to the CDA model.

(iv) The solution to the subproblem provides a lower bound, $\underline{T}(\overline{d}^k)$, on the optimal value ν^* of [CDA]; this bound is strictly better than the corresponding one for the Frank–Wolfe algorithm. A lower bound is also obtained when the subproblem is solved inexactly, if it is defined by the dual functional value at termination.

(v) Feasible solutions to [CDA] provide upper bounds on ν^*, and hence an interval of values, the (relative) length of which,

$$[T(f^{k+1}, d^{k+1}) - \underline{T}(\overline{d}^k)]/|\underline{T}(\overline{d}^k)|, \tag{2.4}$$

can be utilized in a termination criterion in Step **4**. □

As noted above, the approximation utilized in the construction of the auxiliary solution in Evans' algorithm is better than that of the Frank–Wolfe algorithm, since only the term associated with the link flows is linearized. Notwithstanding this principal improvement, the two algorithms suffer from a similar, slow asymptotic convergence behaviour, which manifests itself in a rapidly diminishing step length ℓ_k. The reason for this is the following. Most of the formulas in use for the travel costs are such that the first term in the objective function T of [CDA] is highly nonlinear, as compared to the second term of T; to the relatively flat shape of the second term contributes the fact that d is at least an order of magnitude smaller than f in general. Since both algorithms use a linear approximation of the highly nonlinear term and update the link flow in the same manner in a simple line search, both algorithms suffer from slow convergence; moreover, Evans' algorithm behaves similarly to the Frank–Wolfe algorithm, since the second term of T is comparatively unimportant in the minimization of Step **2**.

Based on this observation, improvements have been made, especially in the line search step. Horowitz (1989) proposes to perform the line search in the link flows only, and Huang and Lam (1992) analyze the convergence characteristics of this heuristic. This algorithm is however based on the same updating formula, and the improvement over Evans' algorithm therefore minor.

The unfavourable properties of Evans' algorithm discussed above have lead us to consider utilizing developments made in recent years in the computation of user equilibrium flows in the construction of more efficient and robust methods for the CDA model. The next section is devoted to the description of one such scheme.[2]

3 A column generation algorithm for CDA models

3.1 Disaggregate simplicial decomposition

The disaggregate simplicial decomposition (DSD) algorithm was developed by Larsson and Patriksson (1992) for the solution of the traffic assignment problem

[TAP]

$$\text{minimize } T(f) = \sum_{a \in \mathcal{A}} \int_0^{f_a} c_a(s)\, ds, \tag{3.1a}$$

means of a primal feasibility heuristic for network flows (e.g., Marklund, 1993), which however, has not been considered so far.

[2] An alternative approach, provoked by the above discussion, is to utilize a nonlinear approximation of the first term of [CDA]; this is, however, left as a possible topic for future research.

subject to

$$\begin{aligned}
\sum_{r \in \mathcal{R}_{pq}} h_{pqr} &= d_{pq}, \quad \forall (p,q) \in \mathcal{P} \times \mathcal{Q}, && (3.1\text{b}) \\
h_{pqr} &\geq 0, \quad \forall r \in \mathcal{R}_{pq}, \quad \forall (p,q) \in \mathcal{P} \times \mathcal{Q}, && (3.1\text{c}) \\
\sum_{(p,q) \in \mathcal{P} \times \mathcal{Q}} \sum_{r \in \mathcal{R}_{pq}} \delta_{pqra} h_{pqr} &= f_a, \quad \forall a \in \mathcal{A}. && (3.1\text{d})
\end{aligned}$$

Suppose that, at some iteration k, subsets $\mathcal{R}_{pq}^k$ of $\mathcal{R}_{pq}$ are known, and f^k solves the restriction of [TAP] obtained when replacing $\mathcal{R}_{pq}$ with $\mathcal{R}_{pq}^k$. Given this link flow, the shortest routes are calculated for all OD pairs. If all the routes thus generated are included in the corresponding subsets $\mathcal{R}_{pq}^k$ already, then the current link flow is optimal, since (2.1) then is fulfilled. Otherwise, each new route obtained is added to its corresponding subset $\mathcal{R}_{pq}^k$, and a new, larger, restriction is defined; solving this new restriction (approximately) yields a solution f^{k+1}, and the process is repeated with $k := k+1$. Convergence of this scheme is finite in the number of restricted problems solved, and the output is a set of routes with flows satisfying the user equilibrium conditions (2.1). See Larsson and Patriksson (1992) and Patriksson (1994) for further details on this and other column generation and simplicial decomposition algorithms for [TAP].

Numerical experiments performed with this method on many known test networks have established that it is robust and much more efficient than the Frank–Wolfe algorithm, while also being at least competitive with state–of–the–art codes for [TAP]. It also has very favourable reoptimization capabilities; thanks to the route information stored, it is easy to reoptimize [TAP] if some model components should change, such as the travel cost functions or the demand matrix.

The DSD algorithm was, for these reasons, considered for use in solving [CDA].

3.2 The new algorithm

The combination of the DSD algorithm with Evans' algorithm works as follows. Assume that, at iteration k, subsets $\mathcal{R}_{pq}^k$ are known, and that (d^k, f^k) is the current iterate. Solving the same subproblem as in Evans' algorithm yields a set of shortest routes and an auxiliary demand, $\overline{d}^k$. Instead of assigning this demand matrix onto the shortest routes and performing a line search, we continue by storing those routes generated that were not stored already. Each new route, say $r \in \mathcal{R}_{pq}$, receives a designated variable, λ_{pqr}, which denotes the portion of the total demand of OD pair (p,q) that is allocated to route r. The relation $h_{pqr} = \lambda_{pqr} d_{pq}$ can be used as the definition of the route flow given the weights λ and current demand. (Choosing this variable definition in place of the route flow variable h_{pqr} has the advantage that the value of λ_{pqr} needs not be scaled when the demand changes.)

The algorithm operates as a column generation algorithm only in the link flow space. The restricted master problem, generalizing that of the DSD algorithm for [TAP], will hence solve the original problem [CDA] over the same line segment in the demand space as Evans' algorithm, while simultaneously optimizing the objective over all possible combinations of weights among the routes currently stored. The restricted master problem has the following form:

[RMP]

$$\nu^k := \min \ T(d,f) = \sum_{a \in \mathcal{A}} \int_0^{f_a} c_a(s)\, ds + \frac{1}{\gamma} \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} d_{pq} \log d_{pq} \tag{3.2a}$$

subject to

$$\sum_{r \in \mathcal{R}_{pq}^k} \lambda_{pqr} = 1, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{Q}, \tag{3.2b}$$

$$\lambda_{pqr} \geq 0, \quad \forall r \in \mathcal{R}_{pq}^k, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{Q}, \tag{3.2c}$$

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} \sum_{r \in \mathcal{R}_{pq}^k} \delta_{pqra} \lambda_{pqr} d_{pq} = f_a, \quad \forall a \in \mathcal{A}, \tag{3.2d}$$

$$d_{pq} = d_{pq}^k + \ell(\overline{d}_{pq}^k - d_{pq}^k), \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{Q}, \tag{3.2e}$$

$$\ell \in [0,1]. \tag{3.2f}$$

Provided that d^k and $\overline{d}^k$ are feasible, the value of ν^k is an upper bound on ν^*. The solution to this problem defines the new iterate (d^{k+1}, f^{k+1}), and if the algorithm is not terminated due to some termination criterion being satisfied, then the process is repeated, with $k := k+1$. We summarize the algorithm in Table 3.1.

Table 3.1: The Evans/DSD algorithm

---

0. (*Initialization*): Choose initial sets $\mathcal{R}_{pq}^0 \subset \mathcal{R}_{pq}$, $p \in \mathcal{P}$, $q \in \mathcal{Q}$, a feasible solution (d^0, f^0) consistent with the sets $\mathcal{R}_{pq}^0$, and let $k = 0$.

1. (*Subproblem*):

a. Given are the fixed link costs $c_a(f_a^k)$. For each OD pair (p,q), compute a shortest route and its corresponding route cost, π_{pq}^k; if the shortest route, $r \in \mathcal{R}_{pq}$, is not contained in $\mathcal{R}_{pq}^k$, then let $\mathcal{R}_{pq}^k := \mathcal{R}_{pq}^k \cup \{r\}$.

b. Solve the doubly constrained entropy maximization problem of minimizing $\underline{T}(d) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} (\pi_{pq}^k d_{pq} + \frac{1}{\gamma} d_{pq} \log d_{pq})$ subject to (2.3c)–(2.3d) and $d \geq 0$. The resulting auxiliary demand is $\overline{d}^k$.

2. (*Restricted master problem*): Solve [RMP]. Let the solution be (d^{k+1}, f^{k+1}).

3. (*Update*): Let $k := k+1$.

4. (*Termination criterion*): If some termination criterion is satisfied, then stop. Otherwise, go to Step **1**.

Remark 3.1 (Notes on the Evans/DSD algorithm) (i) In Step **0**, an initial set of routes can be defined by the routes generated when performing an all-or-nothing assignment given zero flows.

(ii) Step **1** is essentially identical to that of Evans' algorithm, with the exception that routes generated are stored if not known already, and that a one-dimensional search direction is only defined in the demand space.

(iii) Because the number of routes in the network is finite, the column generation phase in Step **1.a** will finitely cease to generate new routes.

(iv) In contrast to Evans' algorithm, the availability of explicit route information enables the use of termination criteria in Step **4** based on detailed measures of the failure of a solution to satisfy Wardrop's conditions (2.1). □

Remark 3.2 (Primal feasibility at termination) The algorithm generates route flows explicitly, in contrast to Evans' algorithm which only provides link flows. The consequences of this fact are several.

In contrast to Evans' algorithm, errors in the computation of the demand subproblem solution are not transfered to the link flows, since they are defined implicitly by the weights λ_{pqr} whose values are calculated independently of the demands. As an example of the consequences of this fact, consider a scenario in which we have a near-optimal, but primal infeasible, demand matrix d^k, and wish to terminate the algorithm with a pair of near-optimal but primal feasible vectors (d, f). We then first either solve the subproblem of Step **1.b** exactly (if possible) or apply a conservative, primal feasibility heuristic on d^k to convert it into a (still near-optimal) feasible matrix, d. Fixing this feasible demand matrix in the restricted master problem (which then is equivalent to a master problem for a traffic assignment problem) then yields a near-optimal and primal feasible set of route and link flows. Evans' algorithm can not perform this task, for the simple reason that the link flow solution f^k, which is the only flow information stored, is contaminated by the errors made in all the previous computations of demand matrices.

An immediate effect of this is that the subproblem Step **1.b** can be solved with almost arbitrary accuracy throughout the algorithm, since primal feasibility can be restored at any iteration. □

Remark 3.3 (Schittenhelm's algorithm) Schittenhelm (1990) presents an algorithm for the CDA problem which is related to ours. He improves upon Evans' algorithm by replacing the auxiliary link flow solution $\overline{f}^k$, which in her algorithm is an all-or-nothing flow, by the link flow solution obtained when solving [TAP] with the demand matrix fixed at $\overline{d}^k$. This subproblem is solved using a method similar to the DSD algorithm. However, he only utilizes the flow solution obtained in a one-dimensional search as in Evans' algorithm, whereas we make use of all information generated. As such, our algorithm can be expected to converge faster. Also, errors in the solution to the demand subproblems propagate in the link flow solution in Schittenhelm's algorithm as in Evans' algorithm. □

Although the restricted master problem [RMP] is more difficult to solve than the one-dimensional problem in Evans' algorithm, the Evans/DSD method is expected to converge in much fewer iterations; further, each new restricted master problem is efficiently reoptimized starting from the solution to the former master problem.

It still needs to be shown that the much fewer iterations needed in the Evans/DSD method amortizes the additional work performed in each iteration due to the more complex updating steps. This is the purpose of Section 4.

3.3 Solution of the restricted master problem

The problem [RMP] is solved with a coordinate-wise (Gauss–Seidel) search, in which alternately the vector λ and the step length ℓ is fixed at its current value while optimizing with respect to the other variable(s). This scheme is globally convergent, since the problem is strictly convex in f and ℓ, and the respective feasible sets are nonempty, convex and bounded.

When fixing ℓ, that is, when fixing the OD matrix, the problem [RMP] reduces to a traffic assignment problem. For solving this problem, we utilize the same method as for the restricted master problem in the implementation of the DSD method by Larsson and Patriksson (1992), that is, a method based on reduced gradients combined with a Newton method, where line searches are performed using the Armijo step length rule.

When fixing λ, that is, when fixing the portion of the demands to allocate to each route, [RMP] reduces to a simple one-dimensional problem in the demand space. Preliminary tests indicated that golden section search was better than the Armijo step length rule for this problem, and was subsequently used in our implementation. The iterate d^k and the auxiliary demand matrix $\overline{d}^k$ define an interval in which an improved matrix is sought. In the first iteration of the solution of [RMP], this matrix is found in a sufficiently smaller interval than the original one; in latter iterations, the interval of search is from the current matrix to one of the endpoints of the original interval, according to the sign of the derivative.

4 Numerical experiments

4.1 Test networks and implementational considerations

The starting point for creating test networks for our experiments was four networks used previously in applications of [TAP]. Their sizes are found in Table 4.1.

Table 4.1: Traffic network models.

Network	# Nodes	# Links	# OD pairs	γ
Sioux Falls	24	76	528	1.0
Hull	501	798	142	1.0
Winnipeg	1052	2836	4344	0.1
Barcelona	1020	2522	7922	0.2

To create instances of [CDA], we used the marginal totals corresponding to the respective network's fixed demand matrix. (Hence, the fixed demand matrix is a feasible, but most probably non-optimal, demand matrix.) The initial solution used in our experiments is the matrix resulting from using the balancing procedure on a problem where the link costs are uniform (equal to 1). (From our experience, different choices of initial matrices did not result in a much different behaviour of the algorithms.)

The value of the dispersion parameter γ was not available, but was instead calculated, for each network, such that the respective optimal demand matrix has a structural similarity to the original, fixed demand matrix. Their respective values are found in Table 4.1.

The Evans/DSD and Evans algorithms were coded in Fortran-77, compiled using the `-fast` switch, and executed on a SUN Ultra SPARC 170E. The main termination criterion used was that of a low relative objective error (2.4); it was checked twice during each main iteration, namely after the solution of the balancing problem, when the lower bound is updated, and after each restricted master problem (Evans/DSD) or line search (Evans), when the upper bound is updated.

Common to the two methods are the solution of shortest route and doubly constrained entropy maximization subproblems. The shortest route calculations were performed using the `lqueue` implementation of Dijkstra's algorithm (the same as the one used in the original DSD code). The procedure for solving the demand subproblems is a standard dual coordinate search (balancing) procedure (e.g., Lamond and Stewart, 1981). The procedure was terminated when the maximal relative difference between any left- and right-hand side in the marginal total constraints fell below 10^{-5}.

For any fixed value of ℓ in the restricted master problem in the Evans/DSD algorithm, we have implemented three methods; a reduced gradient algorithm, a diagonalized Newton method, and a combined method which switches from the former to the latter when the relative error (2.4) in the objective falls below 1%. The line searches in these methods were performed using the Armijo step length rule, with acceptance parameter 0.2. More details on these methods are found in Larsson and Patriksson (1992).

Initial tests performed with the Winnipeg network on the Evans/DSD algorithm showed that each restricted master problem should be solved rather roughly only; a generally good strategy is to perform only a few iterations of the Gauss–Seidel procedure, and, for a fixed value of ℓ, utilize only a few steps of the procedure for the fixed demand problem, regardless of which of the three methods were used. Within reasonable limits of these choices, we found that the total computational effort was roughly constant, so the proper choice is not very crucial. (More work performed on [RMP] results in less main iterations, and hence less work performed on the shortest route and balancing calculations.) In all the numerical experiments reported we have applied 2 iterations of the Gauss–Seidel procedure per main iteration, and 5 iterations of the fixed demand algorithm in each of these. (Hence, in each restricted master problem, a total of 2 (10) line searches are performed in the demand (link flow) space.)

The golden section method was used such that the initial interval is reduced to 10% of the previous one.

4.2 Numerical experience

Detailed experiments with the Winnipeg network. To illustrate the performance of the Evans/DSD algorithm compared to Evans' algorithm, we first present results for the algorithms applied to the Winnipeg network. In Evans/DSD, we used the combined version for the restricted master problems. At each iteration of the two algorithms, we recorded the respective lower and upper bounds on the optimal value, and CPU times.

In Figure 1, upper and lower bounds are plotted against iteration, while Figure 2 plots the bounds against CPU time (in seconds). In both figures, dashed (dotted) lines correspond to the Evans/DSD (Evans) algorithm.

The appearance of Figure 1 is the expected one, since more effort is spent in each iteration of the Evans/DSD algorithm. Figure 2 is therefore more revealing of the performance of the algorithm, which shows that it is indeed more efficient than Evans' algorithm. The reader should note that in the Evans/DSD method, upper bounds are generated throughout the procedure for [RMP]; only the last iterate in each such problem is reported in the figures, however. (This explains why Evans' algorithm, from Figure 2, appears to generate better upper bounds initially.)

An experiment was made to investigate the computational effort of the algorithms when demanding different levels of accuracy. In Table 4.2, we report the number of iterations and CPU time needed to solve the Winnipeg network model with the three versions of the Evans/DSD method, and with Evans' method.

In this case, the three versions of the Evans/DSD algorithm have similar behaviour. It is also clear that Evans' algorithm is efficient in the first few iterations, but much less so when approaching higher accuracies. Note that in the Evans/DSD algorithm, the accuracy

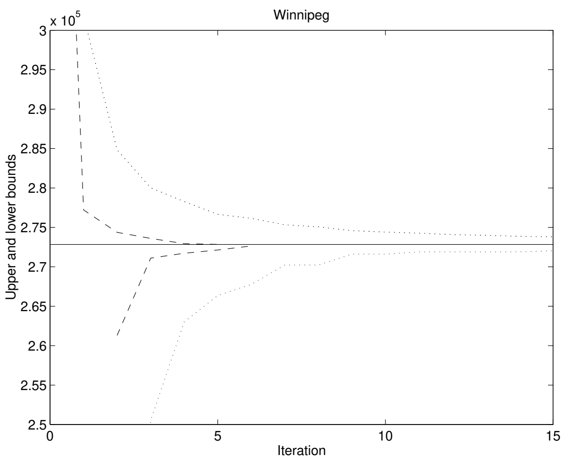

Figure 1: Upper and lower bounds versus iteration.

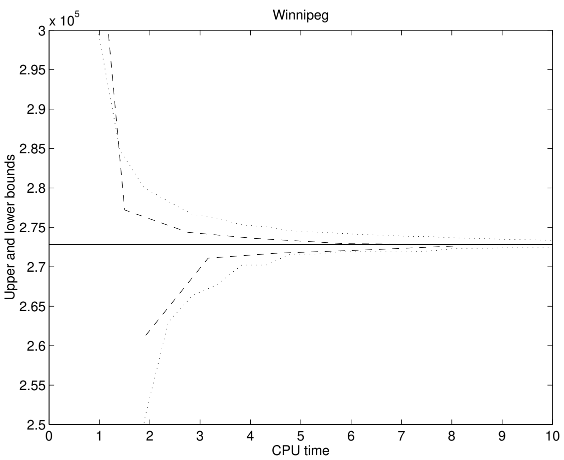

Figure 2: Upper and lower bounds versus CPU time.

Table 4.2: Computational effort for obtaining various accuracies.

	Evans/DSD						Evans	
Rel acc	Red grad		Diag Newton		Combined			
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
10 %	1	1.5 s	1	1.8 s	1	1.5 s	3	1.9 s
5 %	2	2.8 s	2	3.3 s	2	2.7 s	4	2.4 s
1 %	3	4.1 s	2	3.3 s	3	4.1 s	10	5.3 s
0.5 %	4	5.5 s	3	5.0 s	4	5.8 s	16	8.1 s
0.1 %	7	9.9 s	5	9.0 s	5	8.0 s	42	20.6 s

level required is reached earlier than the times reported, since the termination criterion is only checked at termination of the restricted master problems.

When solving to the highest accuracy reported in the table, in the Evans/DSD algorithm, the portions of the total time spent on the shortest route, gravity and restricted master problems were (on average over the three method versions), roughly, 25%, 10%, and 75%, while the corresponding figures for Evans' algorithm were 65%, 20%, and 15%. The Evans/DSD algorithm generated, on average, $14,700$ routes, of which $9,800$ were actually used. (By comparison, Evans' algorithm generated $24,700$ routes during the execution.)

An experiment was made wherein the balancing was made to a lesser accuracy. As indicated earlier, Evans' algorithm should be more sensitive to the errors generated through the inexact computations. This was indeed observed: when the accuracy was decreased from 10^{-5} to 10^{-3}, Evans' algorithm was terminated prematurely due to numerical problems associated with the generation of search directions at infeasible points.

As was noted in Section 2, the objective function is relatively flat with respect to the demand variables. A possible consequence of this fact is that a small difference between the upper and lower bounds on the optimal value may wrongly indicate near-optimality of the link flows and demands. Therefore, we also investigated the behaviour of the two algorithms with respect to their convergence in terms of the link flows and demands. Using the combined algorithm version in the Evans/DSD algorithm, and the same algorithm parameters in both methods as in the experiments above, we calculated the Euclidean distance from each link flow and demand iterate to the corresponding optimal solution vector. (The "optimal solution" is, in fact, a very accurate solution obtained from the use of the Evans/DSD algorithm, executed with a tolerance in relative objective error (2.4) set to 10^{-7}.) For illustration purposes, we terminated the two algorithms after a longer period of time (25 CPU seconds) than in the previous experiment (which resulted in a relative objective error (2.4) of $2.0 \cdot 10^{-5}$ for the Evans/DSD algorithm and $5.0 \cdot 10^{-4}$ for Evans' algorithm). In Figure 3, we plot the Euclidean distance from a link flow iterate, normalized by the Euclidean length of the optimal link flow vector, against the CPU time (in seconds). Figure 4 contains the corresponding information for the demand iterates. (As in Figures 1 and 2, dashed (dotted) lines correspond to the Evans/DSD (Evans) algorithm.)

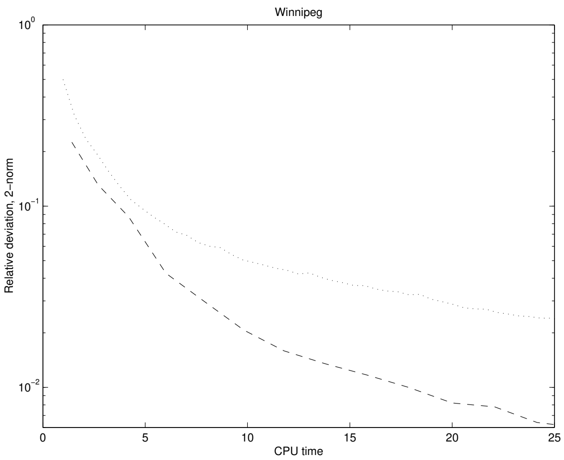

Figure 3: Relative deviation to optimal link flows versus CPU time.

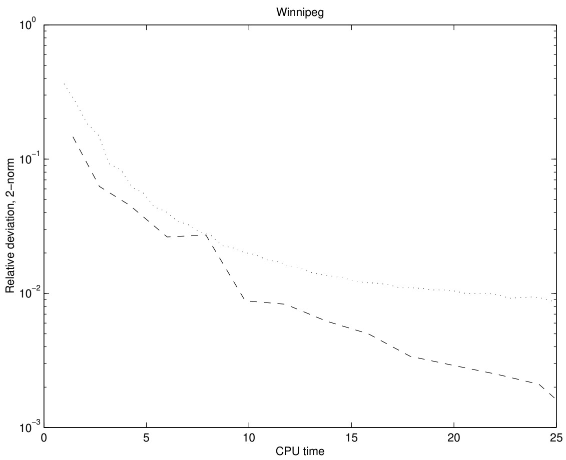

Figure 4: Relative deviation to optimal demands versus CPU time.

The figures show that the Evans/DSD algorithm converges faster than the Evans algorithm with respect to link flows and OD demands; they also indicate that, not surprisingly, convergence is faster in the demand space for both methods. We remark that for any given level of accuracy [defined in terms of the relative objective error (2.4)], the deviations from optimality in the link flow space were similar in the two methods, and the same is true for the demands.

To complement Figures 3 and 4, for both methods we also calculated the maximal deviation from the optimal flow among the links, and from the optimal demand among the OD pairs; these values are shown in Figures 5 and 6.

The result is in many ways similar to that of Figures 3 and 4, with the exception that the deviation fluctuates more in Evans' algorithm (although this can partly be explained by the fact that the value of the deviation is registered more often in Evans' algorithm than in the Evans/DSD algorithm), and, more importantly, that the maximal deviation from the optimal solution is, relatively speaking, even higher in Evans' algorithm than in the proposed method, compared to the measure used in Figures 3 and 4. To assist in the interpretation of Figures 5 and 6, we remark that the average equilibrium flow on a link is about 450, while an average demand is around 15; thus, from Figure 5, we can note that at termination, the maximal deviation in the link flow is in the order of 10 % in the Evans/DSD algorithm while it is in the order of 30 % in Evans' algorithm; in the case of OD demands, the corresponding figures, taken from Figure 6, are 4 % and 50 %, respectively. Clearly, convergence is not only faster in the Evans/DSD algorithm, but it is also more stable.

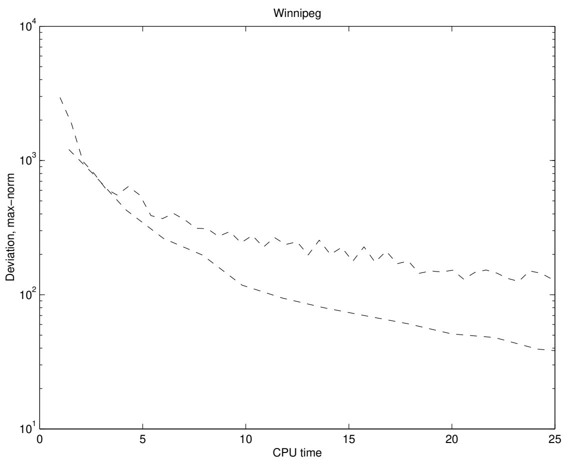

Figure 5: Maximal deviation to optimal link flows.

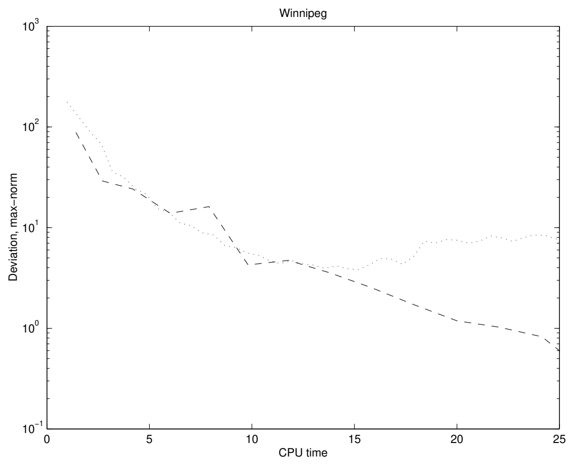

Figure 6: Maximal deviation to optimal demands.

An important conclusion that can be drawn from the above tests is that the relative objective error (2.4) can be used as a convergence measure in practice, although the tolerance chosen must be orders of magnitude smaller than the average or maximal errors accepted in the flows and demands; we base this conclusion on the observation that the difference in magnitude between the relative objective error presented in Figures 1 and 2 and the deviations from the optimal solution presented in Figures 3–6 is roughly the same, regardless of the accuracy level studied, that is, the relative error and the deviation from the optimal solution decreases to zero at about the same rate.

Further experiments. With the same parameter settings as in the initial experiment, we ran the other three network models, with the results shown in Table 4.3.

Table 4.3: Computational effort for different network models.

Network	Rel acc	Red grad		Evans/DSD Diag Newton		Combined		Evans	
		Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
Sioux Falls	0.1 %	13	0.7 s	16	1.2 s	9	0.6 s	104	1.3 s
Hull	0.1 %	5	0.3 s	4	0.2 s	4	0.2 s	48	0.7 s
Barcelona	0.1 %	7	7.0 s					24	10.9 s

With regards to the results for Evans' algorithm when applied to the Sioux Falls network model, an important comment must be made. In order to reach the accuracy demanded, we were forced to increase both the accuracy of the line search (so that the length of each consecutive interval was 1% of the former) and of the balancing procedure (to a relative accuracy of 10^{-7}). This is due to the combined effect of generating search directions with small directional derivatives and the propagation of errors in the computations.

The Barcelona network model is extreme, in that the network really is capacitated; the polynomial travel time formulas reflect this by being flat for moderate flows and highly nonlinear (with powers in the 10–20 range) for large flows. It was the experience in an earlier work (Larsson and Patriksson, 1992) that second-order methods did not work very well on this model, and our conclusion is the same. The (first-order) reduced gradient method worked well, as did Evans' algorithm, but the two second-order ones did not converge; the combined algorithm will in the near future be modified so that the second-order option is never used for such problems.

5 Conclusions

The experiments conducted with the proposed algorithm show that it is an interesting alternative to Evans' algorithm, and that it indeed has some advantages over it. For example, we encountered problems with numerical instabilities with Evans' algorithm, while the most robust version of the Evans/DSD approach so far—the one using the reduced gradient approach only—always worked satisfactorily. This method was, on average, twice as fast as Evans' algorithm. We then note that the parameter choices in the algorithm was not calibrated for the different networks, whence one most probably can construct a specialized version with a better performance when using it in a practical application. Further, demanding a higher accuracy poses no problems with the Evans/DSD algorithm, while Evans' algorithm tends to suffer from numerical difficulties. We also experimented with different convergence criteria on a sample network, and found that the use of the relative objective error (2.4) seems to be useful in practice, but that it must be used with caution, in particular since there is a large difference (orders of magnitude, in fact) between this measure and the deviation from an iterate to the optimal solution.

The CDA model can be extended to include modal split (see e.g. Florian and Nguyen, 1978). Also, there exists a class of models for location, destination and modal split choices in combination with user equilibrium (see, e.g., Boyce et al., 1983) which are founded on the combined model framework of Evans. The proposed algorithm can be applied also to such extended models.

Acknowledgements

The authors were sponsored by the Swedish Transport and Communications Research Board (Grants TFB 92-128-63 and KFB 95-118-63).

References

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

Boyce, D. E., Chon, K. S., Lee, Y. J., Lin, K. T. and LeBlanc, L. J. (1983). Implementation and computational issues for combined models of location, destination, mode and route choice. *Environment and Planning*, **A15**, 1219–1230.

Erlander, S., Nguyen, S. and Stewart, N. F. (1979). On the calibration of the combined distribution–assignment model. *Transportation Research*, **13B**, 259–267.

Evans, S. P. (1973). Some applications of optimisation theory in transport planning. PhD thesis, Research Group in Traffic Studies, University College London, London.

Evans, S. P. (1976). Derivation and analysis of some models for combining trip distribution and assignment. *Transportation Research*, **10**, 37–57.

Fisk, C. S. and Boyce, D. E. (1983). Alternative variational inequality formulations of the network equilibrium-travel choice problem. *Transportation Science*, **17**, 454–463.

Florian, M., Nguyen, S. and Ferland, J. (1975). On the combined distribution-assignment of traffic. *Transportation Science*, **9**, 43–53.

Florian, M. and Nguyen, S. (1978). A combined trip distribution and modal split and trip assignment model. *Transportation Research*, **12**, 241–246.

Frank, C. (1978). A study of alternative approaches to combined trip distribution-assignment modeling. PhD thesis, Department of Regional Science, University of Pennsylvania, Philadelphia, PA.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**, 95–110.

Hearn, D. W., Lawphongpanich, S. and Ventura, J. A. (1987). Restricted simplicial decomposition: computation and extensions. *Mathematical Programming Study*, **31**, 99–118.

Horowitz, A. J. (1989). Tests of an ad hoc algorithm of elastic-demand equilibrium traffic assignment. *Transportation Research*, **23B**, 309–313.

Huang, H.-J. and Lam, W. H. K. (1992). Modified Evans' algorithms for solving the combined trip distribution and assignment problem. *Transportation Research*, **26B**, 325–337.

Marklund, J. (1993). A study of Lagrangian heuristics for convex network flow problems. Master's thesis, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden.

Lamond, B. and Stewart, N. F. (1981). Bregman's balancing method. *Transportation Research*, **15B**, 239–248.

Larsson, T. and Patriksson, M. (1992). Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science*, **26**, 4–17.

LeBlanc, L. J. and Farhangian, K. (1981). Efficient algorithms for solving elastic demand traffic assignment problems and mode-split assignment problems. *Transportation Science*, **15**, 306–317.

Murchland, J. D. (1966). Some remarks on the gravity model of traffic distribution, and an equivalent maximization formulation. Report LSE-TNT-38, Transport Network Theory Unit, London School of Economics, London.

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

Schittenhelm, H. (1990). On the integration of an effective assignment algorithm with path and path flow management in a combined trip distribution and traffic assignment algorithm. In: Proceedings of the 18th PTRC Summer Annual Meeting on European Transport and Planning.

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

Wilson, A. G. (1967). A statistical theory of spatial distribution models. *Transportation Research*, **1**, 253–269.