A STOCHASTIC USER EQUILIBRIUM APPROACH FOR SOLUTION OF THE CONTINUOUS NETWORK DESIGN PROBLEM

Final Report

Prepared by

Gary A. Davis, Ph.D.
Department of Civil and Mineral Engineering
University of Minnesota

Submitted to

Research Administration and Development Section
Office of Materials and Research
Minnesota Department of Transportation

September, 1991

ort represents the results of research conducted by the author and does not ily reflect the views or policy of the Minnesota Department of Transportation. This oes not contain a standard or specified technique.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author would like to thank Patrice Conrath and Shouyan Hua, graduate students at the Univesity of Minnesota, for writing the computer programs and running the tests reported here. Ms. Conrath wrote the implementation of Dial's algorithm used here and obtained the results reported in Experiment 1. Ms. Hua wrote the software which implemented the modification of Dial's assignment for computing derivates and obtained the results reported in Experiments 2 and 3. This research was supported by the Minnesota Department Transportation.

ABSTRACT

The continuous network design problem (NDP) is an idealized mathematical version of a problem frequently faced by highway agencies, namely how to best increase the capacities of various links in the highway network. In particular, the objective in the NDP is to find a set of capacity increases which minimizes a combination of construction cost and the resulting travelling costs to highway users. In order to account for users' reactions to capacity increases, the NDP is contrained so that the minimim total cost is achieved when user route choices are in equilibrium. Characterization of this equilibrium condition is difficult, and the resulting optimization problems tend to be numerically intractable. In the most common formulation of the NDP, it is assumed that users know exactly which routes in the network result in lowest travel costs, and the characteristics of the resulting deteministic user equilibrium (DUE) are the source of the numerical difficulties. If instead we make the more realistic assumption that user perceptions of travel costs are subject to error but that low cost routes are chosen with greater probability than are high cost routes, the resulting stochastic user equilibrium (SUE) is often actually easier to characterize than the DUE, and the resulting NDP is, at least in principle, easier to solve. This project developed software implementing the NDP with SUE assignment using two optimization algorithms, sequential quadratic programming (SQP) and the generalized reduced gradient method (GRG2), and then evaluated their ability to solve NDPs on several hypthotical networks. Generally they performed as well or better than DUE based methods on small networks with low to moderate levels of congestion, but had difficulties as congestion

became artificially high. Both SQP and GRG2 tended to find the same solutions in about the same amount of computer time, with GRG2 showing a slight advantage for the larger networks. On the Sioux Falls network of 24 nodes and 76 links, excessive computer costs prohibited running GRG2 on the available mainframe computers. A version of SQP was however implemented on a Sun Sparcstation computer, and this program was able to generate solutions for the Sioux Falls network that were comparable to those produced by DUE methods, but with fewer assignment iterations. It is concluded that the use of SUE provides both a theoretically and computationally feasible approach to solving the continuous NDP.

INTRODUCTION

The continuous equilibrium network design problem (NDP) is an idealized, mathematical version of a problem commonly faced by highway agencies, namely how to increase the capacities of various roadway segments in a way that both minimizes construction costs and still provides acceptable service to the travelling public. In regions where development and population are on the increase, it is common for the demand for travel on the region's road system to exceed the capacity of at least some roadway segments, causing losses in productivity through excessively long commute and delivery times, excessive fuel consumption and unnecessary air pollution. Since this demand for travel is a natural outcome of both an increasing population and economic growth, many urban areas are likely to require some expansion of roadway capacity in the near future. But roadway expansion projects must compete for limited funds with other equally important social priorities, and responsible use of resources for roadway expansion requires that this expansion be done in the most efficient manner possible.

The efforts to pose and solve NDPs can thus be viewed as first generation attempts to develop optimal methods for allocating resources for roadway expansion. Although the NDP is a well-posed optimization problem, which can in principle be solved by numerical search techniques, currently all available solution methods require inordinantly large amounts of computer time and/or memory, thus leading one prominent reviewer (Friesz, 1985) to conclude that to date no satisfactory solution method for NDPs has been found. The source of the NDP's numerical difficulties lies in how travellers' reactions to capacity increases are characterized, and it is the thesis of this report that by replacing the standard deterministic user equilibrium (DUE)

description of traveller route choice with a stochastic user equilibrium (SUE) characterization, the resulting NDP becomes easier to solve. This report will first review the state of the art concerning solution of NDPs, and then describe the application of SUE methods to the NDP. The results of three computational experiments, two involving solution of NDPs on several small, hypothetical networks and one involving an NDP on the more realistic "Sioux Falls" network will then be described. Finally, conclusions and recommendations will be presented.

Before moving on to a detailed description of this project, it may be helpful to consider the relation of this somewhat academic research effort to the needs of a practice-oriented agency such as the the Minnesota Department of Transportation (MNDOT). As most transportation professionals are well aware, during the past 15 years efforts at alleviating urban transportation difficulties have moved away from solutions involving construction of large amounts of new transportation capacity and towards solutions emphasizing smaller capacity increases and more efficient use of existing facilities. This has generally meant an emphasis on managing the quantity and expression of the demand for travel, that is, on influencing the decisions made by the travelling public so as to produce an efficient roadway use. Unfortunately our ability to devise and implement innovative travel management strategies, such as real-time traffic control, driver information systems, or high occupancy vehicle preferences, has exceeded our ability to accurately describe and predict the effects of these strategies on traveller decsion-making, placing a pressure on researchers and practitioners alike to develop more sophisticated models of the interactions between transportation management actions and travel demand. The general strategy followed in developing these models has been to pose and then solve an increasingly difficult but more realistic sequence of modelling problems, so that the insight gained in solving the simpler

problems can be brought to bear on the more difficult ones. The ultimate goal is the development of the sophisticated dynamic models needed to support the Advanced Traffic Control and Demand Management strategies proposed as components of Intelligent Vehicle and Highway Systems. Examples of early efforts in this process are the work by Horowitz (1984) on the convergence of stochastic traffic assignment for simple two-route networks and the work by Ben-Akiva et al.(1986) on the use of nested logit models to describe traveller selection of departure time and route in simple two-route networks. A more intermediate effort would be the work by Stephanedes et al. (1989) on the use of a dynamic logit model to describe traffic diversion.

Network design problems are arguably the simplest class of transportation management problems which attempt to explicitly account for the interaction between transportation management actions and traveller decision-making. Their simplicity arises from the fact that they are posed as static optimization problems, for which the powerful tools of optimization theory can be brought to bear, yet at the same time many of the analytic and computational issues which arise in dynamic assignment and decision problems are also present in NDPs. Thus an inability to provide satisfactory solutions to NDPs indicates a poor prognosis for solution of more complex dynamic problems. On the other hand, the success encountered in this project in solving NDPs is already providing insight and direction for the developement of dynamic assignment models.

OVERVIEW OF PROBLEM

Network Design Problems.

Basically, the NDP assumes that a region's roadway system can be represented as a network consisting of a set of points, or nodes, connected by directed links. A subset of the network's nodes serve as points of trip origin, while another (possibly different) subset of nodes serve as trip destinations, and it is generally assumed that the total demand for travel between each origin-destination (OD) pair for some planning interval (such as the morning peak period) is known and quantified in the form of an OD matrix. Now suppose the road network has a total of n links, and let the links be indexed by the variable k=1,..,n. To each link there are attached three variables, x_k, y_k and z_k, where x_k denotes the volume of traffic on link k during the planning interval, z_k denotes a measure of the current capacity of link k, and y_k denotes the proposed expansion of link k's capacity, so that the capacity of link k after expansion will be z_k+y_k. In what follows, **x**,**y** and **z** will denote n-dimensional vectors containing the link volumes, capacity expansions and original capacities respectively. Each link is also assumed to have a known travel cost function $c_k(x_k, z_k+y_k)$, which gives the cost of traversing link k as a function of both the volume of travel on link k and the link's capacity. A standard choice for travel cost functions are the Bureau of Public Roads (BPR) relation

$$c_k(x_k,y_k+z_k) = a_k(1+b_k(\frac{x_k}{y_k+z_k})^4) \qquad (1)$$

where a_k denotes the free-flow travel cost on link k and b_k is a coefficient scaling the rate at which congestion increases the cost. Finally, each link is also assumed to have a known constuction cost function $g_k(y_k)$ which gives the cost of increasing link k's capacity by y_k. From a practical standpoint, the appropriate form for these construction cost functions is a question needing additional research, since different forms can produce markedly different solutions (Abdulaal and Leblanc, 1979). Currently researchers tend to employ plausible by somewhat arbitrary functional forms, since the current research focus is on the development of satisfactory solution procedures for any form of cost function. This issue will be touched on again latter in this report.

Since each traveller "pays" a cost of $c_k(x_k,y_k+z_k)$ when traversing link k, the total (nondiscounted) user cost of a given set of capacity increases over a planning horizon of N time intervals would be

$$UC = N\sum_{k=1}^{n} x_k c_k(x_k, z_k+y_k), \tag{2}$$

while the total construction cost is given by

$$CC = \sum_{k=1}^{n} g_k(y_k). \tag{3}$$

Letting α denote the scaling coefficient which converts units of construction cost into units of daily travel cost, (2) and (3) can be combined to give the average daily system cost

$$SC = \sum_{k=1}^{n} [x_k c_k(x_k, z_k + y_k) + \alpha g_k(y_k)]. \quad (4)$$

Since minimizing the quantity SC given in (4) is equivalent to minimizing total system cost, the NDP is solved when one has found appropriate values for x_k and y_k which minimize (4).

If x_k and y_k could both be freely chosen by the highway decision-maker, the NDP would be trivially easy to solve, simply set $x_k = y_k = 0$ for all links and be done. However in addition to minimizing cost, the network must also accomodate the demand for travel, which means that the the link volumes x_k are not directly under the influence of the decision-maker. In order to describe how the link volumes x_k will change in response to capacity increases, it is necessary to introduce a model of how network users select their routes from origin to destination and how these route choices determine the link volumes. The standard assumption in network design (and in much of transportation planning) is that when left to their own devices, travellers independently search for the lowest cost route from origin to destination, and the search terminates when no traveller can decrease his or her travel cost by unilaterally switching to another route. This produces an equilibrium assignment of traffic to the network, and in the case where users can be assumed to know travel costs exactly, the resulting equilibrium assignment is called a deterministic user equilibirium (DUE), or sometimes a Wardrop equilibrium (Wardrop, 1952). Solution of the NDP then requires selecting a set of capacity increases y which when the link volumes are the outcome of a DUE assignment, minimizes the cost given in equation (4).

In order to apply numerical optimization methods to the NDP, one needs to be able to characterize mathematically the DUE. Beckmann et al. (1956) showed that for a given set of

capacities, when the travel cost functions $c_k(.)$ are continuous increasing functions of the link volumes the DUE assignment is unique and can be found by solving a mathematical programming problem, meaning that for each set of capacity increases y there is a unique vector of DUE link volumes, which will be denoted by $x^*(y)$. Although a closed form relation between the capacity increases and the DUE link volumes does not exist, the function can at least be evaluated for a given y by solving the resulting assignment problem. One can then imagine inserting the function $x^*(y)$ for the link volumes in equation (4), giving a cost function which depends only on the capacity increases y. The problem of minimizing (4) would then only be subject to the constraints that the capacity increases be nonnegative, producing a (nearly) unconstrained optimization problem. Unfortunately, the function relating the DUE to the capacity increases is nondifferentiable (Abdulaal and Leblanc, 1979), which means that numerical search methods which do not require derivatives of the cost function must be employed to solve the NDP. This in turn means that numerous DUE assignments must be performed at each iteration of the search routine, resulting in algorithms which require large amounts of computer time. Abdulaal and Leblanc (1979) provide the pioneering work on this approach, and report that of the algorithms they tested, the Hooke-Jeeves algorithm performed best.

Rather than employ the above implicit function methods to produce a bounded but otherwise unconstrained NDP, an alternative would be to derive a set of (hopefully) differentiable constraint relations which are only satisfied by DUE assignments. These could then be attached to the cost function (4) to formulate the NDP as a constrained but differentiable optimization problem, which could be solved by the more efficient search methods of differentiable optimization. In a report by Tan, Gershwin and Athans (1979) a set of such constraints were

constructed, but which required computation of the traffic volume on each route in the network. Since the number of routes in networks of realistic size is very large, this approach can only be employed for small, hypothetical networks. Alternatively, the fact that the DUE assignment can be characterized by a variational inequality (Smith, 1979; Dafermos, 1982) can be used to construct a set of inequality constraints (Marcotte, 1983). Again, the number of these constraints is very large, and Marcotte was forced to employ a constraint accumulation device so that only those constraints which seemed necessary were actually used. Like the Tan *et al.* appoach, this technique has only been tried on small hypthetical networks, and its effectiveness on realistic problems remains to be demonstrated. All of the solution methods described so far guarantee finding a solution which is at least locally optimal, and when algorithms which provide such guarantees prove to be awkward, one may opt instead for a procedure, called a heuristic, which can produce acceptable solutions in practice even though it cannot provide guarantees of optimality. Suwansrikul, Friesz and Tobin (1987) have developed a heuristic method, called equilibrium decomposed optimization (EDO) which does not claim to find locally optimal solutions but does appear to do as well as or better than optimizing techniques such as Hooke-Jeeves, with markedly less computational effort.

The numerical intractability of the NDP originates primarily in the difficulty in characterizing the DUE assignment, which is based on the assumption that travellers have perfect information concerning route travel costs. If we make the more realistic assumption that the perceived travel costs are subject to error, but that users still tend to prefer lower cost routes in the sense that the lower cost routes are chosen with higher probability, it turns out that the resulting NDP becomes, at least in principle, easier to solve. To see this, imagine that the

separate OD pairs in the network are indexed by j=1,..,m, and let d_j denote the demand for travel between OD pair j during the planning interval. Let c_{jr} denote the cost of traversing path number r between OD pair j, and assume that the path cost is simply the sum of the corresponding link costs, *i.e.*

$$c_{jr} = \sum_{k=1}^{n} \delta_{jrk} c_k(x_k, y_k + z_k), \tag{5}$$

where

$\delta_{jrk} = 1$, if link k lies on path r between OD pair j,

0, otherwise.

Assume further that each user's perceived route cost is equal to the actual route cost plus a random error term, and that the random error has zero mean and is independent of the actual cost and also independent across users. In a manner similar to the DUE assignment, users are assumed to search for that route which has lowest perceived travel cost, and that an equilibrium is achieved when no user can lower his or her perceived travel cost by unilaterally switching routes. Let $p_{jr}(\mathbf{c})$ denote the probability that route r between OD pair j is perceived as having lowest cost (i.e. the probability route r is chosen), and this depends on the vector of link costs. Let $q_{jk}(\mathbf{c})$ denote the probability that a trip between OD pair j uses link k, which depends on the route choice probabilities $p_{jr}(\mathbf{c})$ via the equation

$$q_{jk}(c) = \sum_r \delta_{jrk} p_{jr}(c). \tag{6}$$

Under the above conditions, Daganzo (1982) has shown that the stochastic user equilibrium (SUE) link volumes x_k are characterized as solutions to the set of nonlinear equations

$$x_k = \sum_{j=1}^{m} d_j q_{jk}(c), \quad k=1,..,n \tag{7}$$

where the link use probabilities depend on the link volumes via the link cost functions $c_k(x_k, y_k+z_k)$. The NDP with SUE assignment would then be to find a vector of capacity increases y, which minimizes equation (4) subject to the link volumes satisfying equations (7). The number of separate equations in (7) is equal to the number of links in the network and when the route choice probabilities are given by either the logit or probit functions the right hand side of (7) can be evaluated without computing the separate route flows (Sheffi, 1985). Thus if the link use probabilities can be expressed as differentiable functions of the capacity increases y, the NDP with SUE assignment would be a differentiable optimization problem with a manageable number of differentiable constraints. This in turn would allow established algorithms for large-scale differentiable optimization to be employed in solving the NDP.

The use of SUE instead of DUE assignment in NDPs was first employed by Chen and Alfa (1991), who developed a heuristic for solving the discrete version of the NDP. Although the work described in this report was conducted independently of Chen and Alfa, credit for priority of the idea belongs to them. Unlike Chen and Alfa, the focus of this research is on providing optimal solutions to the continuous version of the NDP rather than approximate solutions to the discrete version.

The potential advantage arising from the use of SUE methods comes at a price however, in that one must now determine an appropriate form for the route choice probability functions, $p_{jr}(\mathbf{c})$. Like other aspects of transportation modelling, this is a decision for which a solid base of empirical support does not exist, forcing one to base one's choice on the less satisfactory criteria of plausibility and mathematical tractability. As noted above, the two dominant options for route choice probability functions are the logit and probit functions, obtained when one assumes that the route cost perception errors are distributed as either Gumbel or Normal random variables respectively (Sheffi, 1985). Several authors have argued for the superiority of the probit model, (Daganzo, 1982; Sheffi, 1985), based on the logit model's tendency to treat routes as independent entities, even when two or more routes share common links. On the other hand, the SUE resulting from logit route choice probabilities can be used to approximate DUE assignments (Fisk, 1980) while Sheffi and Powell (1981) report that in a simple test network the difference between probit-based SUE and DUE assignments tended to disappear as congestion increased. More recently, Chen (1991) has provided examples of a similar equivalence between logit and probit assignments. This suggests that for congested networks (which are the networks for which one would generally consider solving an NDP) there is no strong reason to prefer either DUE, probit-based SUE or logit-based SUE as accurate models of traveller behavior. In this case, selection of a model can be based on which is more tractable.

As noted above, DUE assignment methods tend to make the NDP computationally intractable, while employment of an SUE assignment method, for which the equations (7) are differentiable functions of the capacity increases $\mathbf{y}$, could lead to more efficient solution procedures. As is well known, the logit route choice probabilities can be given as explicit

functions of the link costs via

$$p_{jr}(c) = \frac{e^{-\theta c_{jr}}}{\sum_s e^{-\theta c_{js}}} \tag{8}$$

where c_{jr} is the route cost defined in equation (5) and θ is a parameter which is determined by the variance of the route cost perception errors. As long as the link cost function $c_k(.)$ are differentiable functions of the capacity increases y, it is easy to verify that the logit route choice probabilities and the resulting link use probabilities q_{jk} are also differentiable, and explicit formulas for these derivatives are developed in the Appendix to this report. Probit-based route choice probabilities are also differentiable, but no explicit expression such as (8) can be given for them. Thus the evaluation of probit-based probabilities and hence also their derivatives requires numerical approximation, which in turn tends to undermine the hypothesized numerical advantage which SUE methods ought to enjoy. Hence in this project, the focus of effort was on developing and testing methods of using logit-based SUE assignment in the network design problem, with the question of the relative superiority of logit versus probit being left to future efforts.

Implementation of Logit-Based SUE.

Before moving to a discussion of implementation issues arising in using logit-based SUE assignment in solving the NDP, it may be helpful to collect the results discussed above and restate the NDP. Given a representation of a road system as a directed network, a list of OD demand volumes d_j, $j=1,..,m$, a travel cost function $c_k(x_k,z_k+y_k)$ for each link $k=1,..,n$ and a

construction cost function $g_k(y_k)$ for each link k=1,..,n, the problem is then to minimize the system cost function

$$SC=\sum_{k=1}^{n}[x_k c_k(x_k,y_k+z_k)+\alpha g_k(y_k)] \tag{9}$$

subject to the constraints

$$\begin{gathered} h_k(x,y,z) \equiv x_k-\sum_{j=1}^{m} d_j q_{jk}(c)=0, \quad k=1,..,n \\ y_k \geq 0, \quad k=1,..,n. \\ y_k=0, \quad k \in L \end{gathered} \tag{10}$$

where the link use probabilities $q_{jk}(c)$ are determined from the logit route choice probabilities given in equation (8) via the relation given in equation (5). *L* denotes the set of of links which are not candidates for expansion.

Solution of constrained optimization problems such as the one above is usually done numerically, where the algorithm begins with a trial solution, computes a search direction and then performs a line search along this direction to obtain an improved trial solution. When the trial solution satisfies conditions which indicate that it is a local optimum, the algorithm terminates. Numerous algorithms have been built around this general logic, and computer software implementing them are commonly available. In order to specialize these codes to a particular problem, the user usually must provide the algorithm with two types of information. First, for a given trial solution, the user must be able to evaluate the objective function such as (9) and the constraint functions, such as (10). Second, for a given trial solution, the user must

also be able to evaluate the gradient vector of the objective function and the Jacobian matrix of the constraint functions. This information is commonly provided as the output of user-written subroutines which can be called by the algorithm when needed. Thus in order to employ existing differentiable optimization algorithms to the solution of the NDP it is necessary to be able to compute the values of the cost function (9), the partial derivatives of (9) with respect to both the link volumes and the capacity increases, the values of the SUE constraint functions (10) and the partial derivatives of the SUE contraints (10), also with respect to the link volumes and the capacity increases. Evaluation of the link cost and construction cost functions is generally straightforward, while evaluation of their derivatives for the common choices of BPR-type link cost functions and quadratic construction cost functions is also straightforward. Evaluation of the constraint functions is problematic however since the link use probabilities $q_{jk}(\mathbf{c})$ are defined in terms of the route choice probabilities p_{jr} via equation (5). Naive computation of the right hand side of (10) would then require computation of a quantitity p_{jr} for each feasible path in the network, which is a prohibitive task for networks of realistic size. For the logit-based route choice model however, it is well known that Dial's (1971) algorithm can be used to evaluate the right hand side of (10) directly without computation of the route choice probabilities or route flows. In addition, in the Appendix it is shown that the derivatives of the constraint functions are given by

$$\begin{aligned} \frac{\partial h_k}{\partial x_i} &= \delta_{ki} + \theta\left(\frac{\partial c_i}{\partial x_i}\right)\left(\tilde{q}_{jki} - q_{jk}q_{ji}\right) \\ \frac{\partial h_k}{\partial y_i} &= \theta\left(\frac{\partial c_i}{\partial y_i}\right)\left(\tilde{q}_{jki} - q_{jk}q_{ji}\right) \end{aligned} \qquad (11)$$

where $\tilde{q}_{jki}$ denotes the probability a trip between OD pair j uses both link k and link i. Efficient implementation then requires being able to compute the probability terms q_{jk} and $\tilde{q}_{jki}$ without having to resort to computation of the route choice probabilities. Fortunately Van Vliet (1981) describes a modification of Dial's algorithm which can be used to compute the link use probabilities q_{jk} without first having to first compute the route choice probabilities and in the Appendix, a further modification of Dial's algorithm (in fact a modification of Van Vliet's modification) is described which can be used to compute the $\tilde{q}_{jki}$ terms also, again without requiring the route choice probabilities.

To summarize then, if it can be assumed that the link cost and construction cost functions are differentiable and that the route choice probabilities used to generate the SUE assignment constraints (10) are given by the logit function (8), then both the system cost function (9) and the constraint functions (10) are differentiable and modifications of Dial's algorithm can be used to evaluate both the constraint functions and their Jacobian without having to resort to path enumeration. This in turn permits the development of subroutines for computing these quantities, which can be attached to existing numerical optimization codes to provide solution procedures to the NDP described in (9) and (10). The main thesis of this research is that such differentiable algorithms will provide solution procedures which are superior to those currently in use.

COMPUTATIONAL EXPERIMENTS

The computation of the minimum of a differentiable objective functions subject to a set of differentiable constraints is a problem which has interested researchers for at least 50 years, and for which a number of solution algorithms have been implemented (Bazaraa and Shetty, 1979). For problems where the objective function is nonlinear but the constraints are linear equalities, Wolfe (1963) introduced the reduced gradient method, and this algorithm was generalized to handle nonlinear equality constraints by Abadie and Carpentier (1969). The generalized reduced gradient method has been recognized as a feasible candidate for solving large-scale optimization problems, and the most comprehensive computer code implementing the generalized reduced gradient method is GRG2, developed by Lasdon and colleagues at the University of Texas (Lasdon and Waren, 1982). More recently, the MINOS algorithm developed by Murtagh and Saunders at Stanford University (Murtagh and Saunders, 1982) has reported successful solution of some very large problems. Unlike GRG2, MINOS does not require that the constraint functions be satisfied at each iteration of its search procedure, and this feature arguably allows it in some cases to converge more quickly than GRG2 (Murtagh and Saunders, 1982). A further enhancement of the MINOS logic has been the development of sequential quadratic programming (SQP) algorithms, which successively approximate the original optimization with a sequence of problems with quadratic objective functions and linear constraints. SQP is also claimed to enjoy MINOS's superiority over GRG2 (Gill, Murray, Saunders and Wright, 1989) for the same reasons, although currently there are no reports of

successful solution of large-scale problems using SQP.

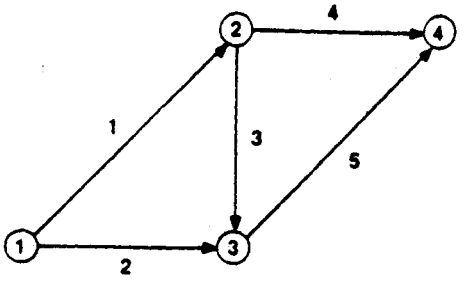
Academic Computing Services at the University of Minnesota (UM) maintains the implementations of GRG2 and MINOS provided by the algorithms' developers, and also maintains an implementation of SQP as part of the IMSL10 subroutine library, so that a "shoot-out" among the candidate procedures GRG2, MINOS and SQP in solving NDP examples was at first seen as feasible option. Unfortunately it was not possible to get all three procedures up and running on the same computer system nor were we able to obtain satisfactory results from MINOS using some simple test problems, so this report will focus on the performance of GRG2 and SQP. Two computer programs, one implementing GRG2 for SUE-based NDP problems and one implementing the SQP routine NCONG contained in the IMSL10 subroutine library were written in FORTRAN for the UM's VAX 11/780 computer system. These programs were used successfully to conduct numerical experiments using small hypothetical transportation networks, but due to excessive computation costs it was not possible to evaluate these routines on networks of more realistic size. Thus a third FORTRAN program implementing the SQP routine E04VDF, which was available on the Principal Investigator's Sun Sparcstation 1+ computer, was written and used to conduct a third computational experiment using the moderately-sized "Sioux Falls" network developed by Leblanc (1975). As noted earlier, the major effort in using these routines is involved in writing subroutines which evaluate the objective and constraint functions and their derivatives. The routines developed here assume that the link cost functions were of the BPR form given in equation (1) and that the construction cost functions were of the quadratic form given in (12) below, which is a common assumption appearing in the literature.

$$g_k(y_k) = e_k y_k^2 \tag{12}$$

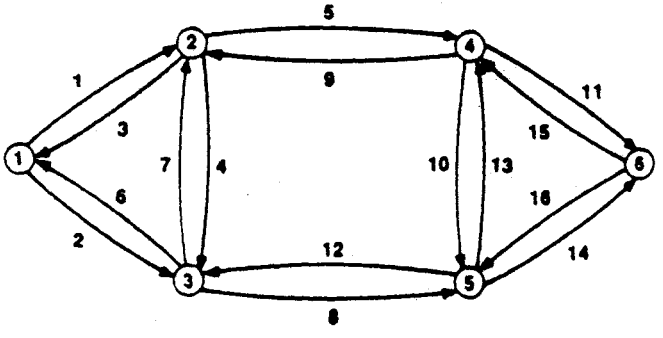
The parameters describing these functions were required to be input by the user. In addition, the user had to provide a representation of the network as a linked list of nodes, and the network's OD matrix. Evaluation of the constraint functions (10) was done using an implementation of Dial's algorithm written for this project, and the Jacobian of the constraints was evaluated using the modifications described above.

Experiment 1: Comparison of SUE and DUE Approaches to the NDP.

The first computational experiment involved a comparison of the relative effectiveness of the SUE approach to solving the NDP to existing DUE approaches. Figure 1 shows two simple example networks used by Suwansrikul et al.(1979) to test several exact approaches to the NDP with DUE assignment, along with their EDO heuristic approach. The upper portion of Figure 1 shows a 4-node, 5-link network with node 1 serving as the only origin and node 4 serving as the only destination. The parameters governing the travel cost and construction cost functions can be found in Suwansrikul et al. (1987). In their paper, Suwansrikul et al. varied to demand for travel between node 1 and node 4 from a low of 100 trips to a high of 300 trips, producing a range of congestion levels. The lower portion of Figure 1 shows a network with 6 nodes and 16 links, with two OD pairs, from node 1 to node 6 and from node 6 to node 1. Again, the parameters govening the travel and construction cost functions can be found in Suwansrikul et al.(1987), and again a range of demand levels was explored.



Test Network 1



Test Network 2

Figure 1. Test Networks 1 and 2 (from Suwansrikul et al., 1987)

As was shown by Fisk (1980), when the logit parameter θ becomes arbitrarily large, the logit-based SUE assignment approaches the DUE assignment. Thus the solution to an NDP with logit-based SUE assignment and with a large enough value for θ should be approximately equal to a solution using DUE assignment. If the SUE solution also requires less computational labor, then this would be evidence for the superiority of the SUE approach. In this first experiment, our program using sequential quadratic programming was compared to the results reported in Suwansirikul et al. (1987). For each problem, an appropriate value of θ was determined by starting the SQP algorithm with the final solutions reported by Suwansrikul et al.(1987), and then adjusting θ until SQP would terminate with a solution very close to the initial one. After determining the appropriate value for θ the SQP program was started with an initial solution of $y_k=0$ for all links and then allowed to run to termination. Suwansrikul et al. (1987) used the number of Frank-Wolfe iterations required by their algorithms as the basic measure of computational labor. Since the SUE-based algorithms do not use the Frank-Wolfe procedure, a direct comparison is not possible. However the major source of computational effort in a Frank-Wolfe iteration is the computation of an all-or-nothing assignment of trips in the network, which requires computation of the shortest paths from each origin to all destinations. Since a Dial assignment also requires computation of these shortest paths, it seems plausible to take one Dial assignment as roughly equivalent to one Frank-Wolfe iteration and use the number of Dial assignments as a rough measure of the computational labor done by the SQP program.

Table 1 compares the performance on Test Network 1 of two algorithms used in Suwansrikul et al. (1987), the Hooke-Jeeves algorithm and their heuristic EDO, to the performance of the SQP routine. The values reported for the Hooke-Jeeves algorithm and for

EDO are taken from Suwansirkul et al. (1987). For the lower demand levels of 100, 150 and 200 total trips between node 1 and node 4, it can be seen that SQP produces solutions equivalent to the DUE algorithms with roughly comparable amounts of computational labor. For instance when demand was set at 100 trips, the Hooke-Jeeves algorithm required 27 Frank-Wolfe iterations, EDO required 24, while SQP with logit-assignment required 16 Dial assignments to produce a comparable solution. When demand was set at 200 trips, the Hooke-Jeeves required 56 Frank-Wolfe iterations, EDO required 32 and SQP required 38 Dial assignments. However when demand was increased to 300 trips, the number of Dial assignments required by SQP jumped to 260, while the number of Frank-Wolfe iterations required by Hooke-Jeeves and EDO only increased to 79 and 38 respectively. Table 2 displays results obtained from Test Network 2. Again, the results reported for Hooke-Jeeves and EDO are taken from Suwansrikul et al. (1987). Here we see that for the lower demand, SQP with logit-assignment performed quite favorably compared to both Hooke-Jeeves and to EDO. However when the demand level was doubled, the SQP algorithm terminated without being able to find a local optimimum.

To summarize the results of Experiment 1, for two small test networks sequential quadratic programming with logit-based SUE assignment tended to find solutions similar to those found by two DUE-based methods with roughly equal or less computation effort, but only when demand for travel on the network was relatively low. On the first network, when demand was increased to 300 trips, the computational effort required by SUE made a marked jump, while in the second network a doubling of demand produced a problem which SQP was unable to solve. The source of these difficulties appears to be a numerical instability caused by exponentiating large negative values, and occurs in hypothetical networks with artificially high congestion levels.

TABLE 1

SOLUTION RESULTS FOR TEST NETWORK 1

Demand	Variable	Hooke-Jeeves	EDO	SQP
100	y_1	1.25	1.31	1.34
	y_2	1.20	1.19	1.20
	y_3	0.0	0.06	0.0
	y_4	0.95	0.94	0.95
	y_5	1.10	1.06	1.11
	SC	1200.6	1200.6	1201.31
	Iterates	27	24	15

Demand	Variable	Hooke-Jeeves	EDO	SQP
150	y_1	5.55	5.98	6.06
	y_2	5.65	5.52	5.45
	y_3	0.0	0.02	0.0
	y_4	4.60	4.61	4.62
	y_5	5.20	5.27	5.29
	SC	3156.4	3156.2	3157.5
	Iterates	37	29	22

Demand	Variable	Hooke-Jeeves	EDO	SQP
200	y_1	13.0	12.9	13.0
	y_2	11.8	12.0	11.7
	y_3	0.0	0.02	0.0
	y_4	10.3	10.3	10.3
	y_5	11.8	11.8	11.8
	SC	7086.2	7086.5	7089.5
	Iterates	56	32	38

Demand	Variable	Hooke-Jeeves	EDO	SQP
300	y_1	28.4	28.1	28.5
	y_2	25.8	26.0	25.7
	y_3	0.0	0.01	0.0
	y_4	23.4	23.4	23.4
	y_5	26.6	26.6	26.6
	SC	21209.9	21209.9	21211.2
	Iterates	79	38	260

TABLE 2

SOLUTION RESULTS FOR TEST NETWORK 2

Demand	Variable	Hooke-Jeeves	EDO	SQP
1,6=2.5	y_1-y_5	0.0	0.0	0.0
6,1=5.0	y_6	0.30	1.84	0.0
	y_7	0.03	0.02	0.0
	y_8-y_{14}	0.0	0.0	0.0
	y_{15}	0.10	0.02	0.0
	y_{16}	0.30	1.84	0.97
	SC	90.1	92.4	91.5
	Iterates	115	57	14

Demand	Variable	Hooke-Jeeves	EDO	SQP
1,6=5.0	y_1-y_2	0.0	0.0	-
6,1=10.0	y_3	1.2	0.13	-
	y_4-y_5	0.0	0.0	-
	y6	3.0	6.3	-
	y_7-y_{14}	0.0	0.0	-
	y_{15}	3.0	0.13	-
	y_{16}	2.8	6.3	-
	SC	215.1	201.8	-
	Iterates	166	71	-

Experiment 2: Comparison of SQP and GRG2.

Since GRG2 is based on a somewhat different algorithmic philosophy than is SQP, it was selected as the candidate against which to assess the performance of SQP. As with SQP, it was necessary to provide subroutines to evaluate functions and compute derivatives, and essentially the some routines as those used by SQP were used with GRG2. In addition to the two test networks described in Experiment 1, two additonal networks were also used. Figures 2 and 3 display these networks, and Tables 3 and 4 show their demands, link cost and construction cost characteristics. Network 3 has 6 nodes, 7 links and 4 OD pairs, while Network 4 is topologically similar to Network 3 but, with 10 nodes, 14 links and 9 OD pairs, is roughly "double" the size of Network 3. For Networks 1-4, both SQP and GRG2 were started with the same initial solution ($y_k=0$ for all links) and allowed to run until termination.

Since the two algorithms, SQP and GRG2 use essentially identical subroutines to evaluate functions and derivatives and since they are both implemented in FORTRAN on the same machine, it is possible to compare them directly using the amount of central processing unit (CPU) time required to obtain a solution. For Network 1, the case where demand equalled 300 trips was selected, since this is the case where SQP had the most difficulty. For Network 2, the case of low demand was selected since this was the only one SQP could solve. Table 5 compares the performance of SQP and GRG2 for Networks 1 and 2. Both algorithms found essentially the same solutions (the solutions found by SQP in Experiment 1) in roughly the same amount of CPU time, although GRG2 did slightly better for the larger Network 2. Table 6 displays the results obtained for the two new networks, 3 and 4. Again, the solutions found by SQP and by

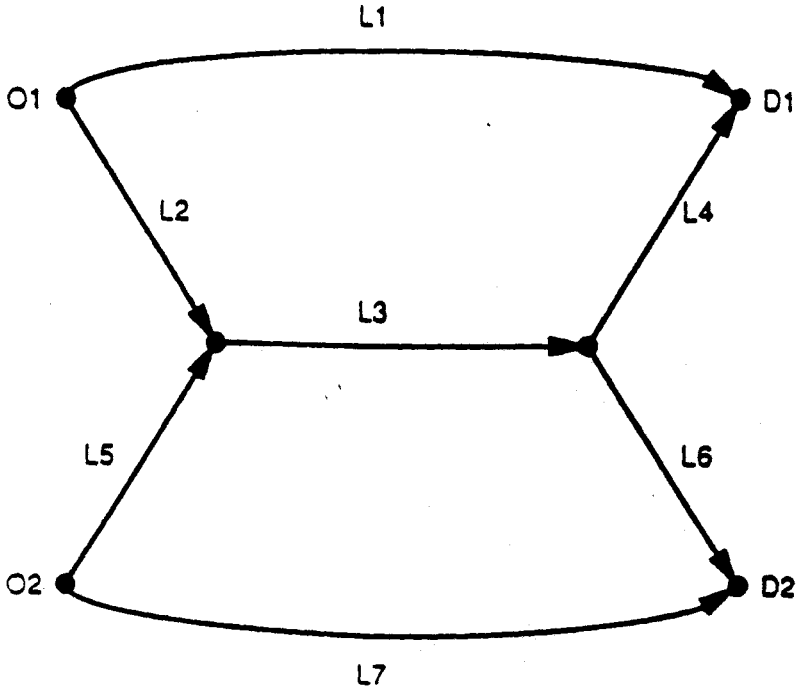


Figure 2. Diagram of Test Network 3.

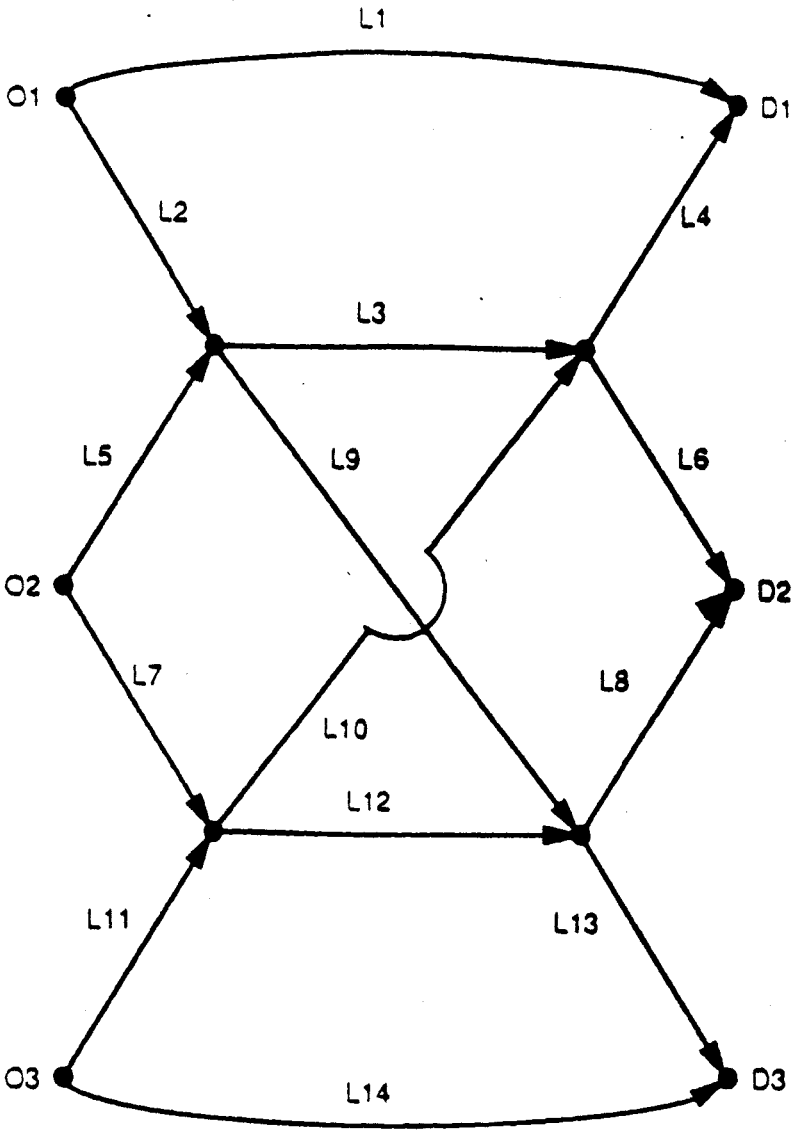


Figure 3. Diagram of Test Network 4.

TABLE 3

DEMAND AND LINK CHARACTERISTICS FOR TEST NETWORK 3

Demand Characteristics

j	OD pair	d_j
1	O1-D1	500.
2	O1-D1	400.
3	O2-D2	400.
4	O2-D2	600.

Link Characteristics

k	a_k	b_k	z_k	e_k
1	1.0	.15	200.	3.0
2	.25	.15	300.	2.0
3	.25	.15	700.	1.0
4	.25	.15	300.	2.0
5	.25	.15	300.	2.0
6	.25	.15	300.	2.0
7	1.0	.15	200.	3.0

TABLE 4

DEMAND AND LINK CHARACTERISTICS FOR TEST NETWORK 4

Demand Characteristics

j	OD pair	d_j
1	O1-D1	500.
2	O1-D2	400.
3	O1-D3	350.
4	O2-D1	400.
5	O2-D2	700.
6	O2-D3	200.
7	O3-D1	200.
8	O3-D2	250.
9	O3-D3	1000.

Link Characteristics

k	a_k	b_k	z_k	e_k
1	1.0	.15	200.	4.0
2	.25	.15	600.	2.0
3	.25	.15	500.	3.0
4	.25	.15	600.	2.0
5	.25	.15	500.	3.0
6	.25	.15	500.	3.0
7	.25	.15	500.	3.0
8	.25	.15	500.	3.0
9	.25	.15	600.	2.0
10	.25	.15	700.	1.0
11	.25	.15	500.	3.0
12	.25	.15	600.	2.0
13	.25	.15	600.	2.0
14	.25	.15	200.	4.0

TABLE 5

COMPARISON OF SEQUENTIAL QUADRATIC PROGRAMMING AND GENERALIZED REDUCED GRADIENT

	Network 1			Network 2	
Variable	SQP	GRG2	Variable	SQP	GRG2
y_1	28.48	28.50	y_1-y_{15}	0.0	0.0
y_2	25.69	25.67	y_{16}	0.96	0.96
y_3	0.00	0.00			
y_4	23.26	23.36			
y_5	26.60	26.60			
SC	21213.7	21213.7	SC	91.45	91.45
CPU	16.9	17.1	CPU	11.7	8.8
iterates	233	165	iterates	15	15

TABLE 6

COMPARISON OF SEQUENTIAL QUADRATIC PROGRAMMING AND GENERALIZED REDUCED GRADIENT

	Network 3			Network 4	
Variable	SQP	GRG2	Variable	SQP	GRG2
y_1	27.8	27.8	y_1	6.20	6.21
y_2	7.15	7.15	y_2	6.32	6.32
y_3	8.20	8.14	y_3	3.02	3.02
y_4	2.53	2.53	y_4	3.38	3.38
y_5	18.95	18.94	y_5	6.01	6.01
y_6	18.95	18.94	y_6	1.81	1.80
y_7	20.78	20.77	y_7	1.33	1.33
			y_8	1.74	1.74
			y_9	2.37	2.37
			y_{10}	7.83	7.83
			y_{11}	1.86	1.86
			y_{12}	16.75	16.75
			y_{13}	13.83	13.84
			y_{14}	15.73	15.73
SC	3992.7	3992.7	SC	6416.0	6416.0
CPU	6.1	7.0	CPU	49.7	44.7
Iterates	53	30	Iterates	62	48

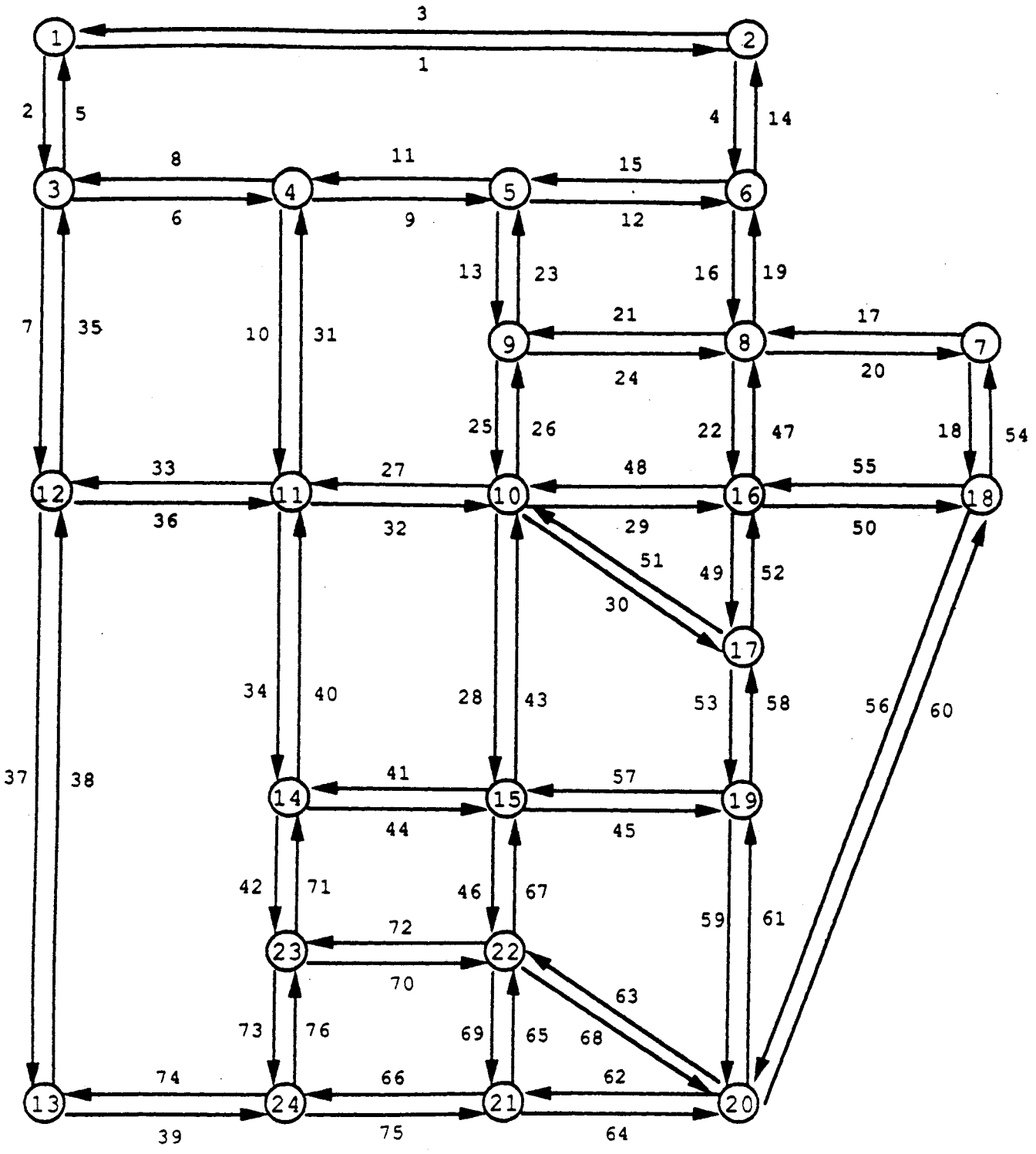
GRG2 were very similar, and the amounts of computer effort were roughly comparable, with GRG2 showing a small superiority for the larger Network 4.

To summarize, both GRG2 and SQP found similar solutions in all four network examples. The CPU time needed was about the same, but with GRG2 showing a slight superiority for the larger networks. Thus the hypothesized superiority of SQP over GRG2 was not observed here.

Experiment 3: A Larger Network.

In Experiment 1 the performance of an optimization algorithm, sequential quadratic programming, for solving the NDP with SUE assignment could not be shown to be definitely superior to DUE-based methods. In Experiment 2, SQP and an alternative algorithm, GRG2 were shown to find the same solutions in about the same amount of computer time on 4 different example problems. In this third experiment, the abilities of GRG2 and SQP to solve a larger problem was investigated. Figure 4 shows a network with 24 nodes and 76 directed links, originally constructed by Leblanc (1975) as a representation of the road system in Sioux Falls, South Dakota. Links 16, 17, 19, 20, 25, 29, 39, 48, and 74 are the candidates for expansion while all nodes serve as both origins and destinations. Suwansrikul et al. (1987) tested their algorithms on this problem, and details concerning the cost functions and the OD matrix can be found in their paper. Basically, they reported that the Hooke-Jeeves algorithm required 147 Frank-Wolfe iterations to find a reasonable solution to this problem while their EDO algortihm required only 89 iterations to find a somewhat similar (but not identical) solution.

Figure 4. The Sioux Falls Network



In attempting to solve this problem with GRG2, we first attempted to initialize the problem with an optimal solution from Suwansrikul et al. (1987), with the expectation that GRG2 would terminate with a similar solution in a few iterations. What in fact happened was that GRG2 would terminate with a message that it was unable to find a solution to the constraint equations (10) and hence unable to find a feasible trial solution. Since for the conditions used in this problem it is known that the solution for the constraint equations exists and is unique (Daganzo, 1982), this was at first puzzling, and a program for solving the constraint equations using the method of successive averages (MSA) (Sheffi, 1985) was written to investigate their solution properties. This program was again unable to find solutions to the constraint equations, and a close study of MSA's behavior indicated that the links which were included in the set of efficient paths as defined by Dial's algorithm tended to change from iteration to iteration in response to small changes in travel costs. That is, the set of paths as defined by Dial's algorithm was unstable, and this violated the sufficient conditions which guarantee the convergence of the MSA. When the Dial assignment algorithm was modified slightly so that it only considered a fixed set of routes (namely the Dial "efficient" routes for free-flow travel costs) the MSA converged without any problem.

When this modification of the Dial assignment was incorporated in the routine used by GRG2, and GRG2 was initialized with a "good" solution from Suwansrikul et al. (1987), the feasibility problem encountered earlier did not occur. However, GRG2 used 14 iterations and a time limit of one hour without substantially modifying the initial solution but also without being able to convice itself that it had a local optimum. This run cost about $700 in computer funds and at this

point the project's computation budget was exhausted. We point out however that the iteration count for GRG2 is not out of line with those reported by Suwansrikul et al. (1987), (for instance the Hooke-Jeeves algorithm required 71 Frank-Wolfe iterations to verify a near optimal solution) but a definite conclusion concerning the usefulness of GRG2 using SUE assignment could not be reached using the UM's mainframe systems. However, late in the project the Principal Investigator obtained a Sun Sparcstation 1+ workstation, with a FORTRAN compiler and the NAG Workstation subroutine library. Although computer code implementing GRG2 on the Sun was not available, the Workstation library did contain a routine implementing SQP, called E04VDF. The VAX SQP program was modified appropriately to run on the Sun, permitting a comparison of the performance of SQP to the DUE-based algorithms presented in Suwansrikul et al., (1987). The NDP with SUE was solved for a range of θ values, and the results of these runs are displayed in Table 7. At $\theta=10$, the total system cost is noticeably larger than that achieved by the DUE-based methods, due to the tendency of SUE procedure to assign some trips to routes with greater than minimum cost. As θ increases however, the SC for SQP decreases until it is approximately equal to that achieved by DUE-methods. (At $\theta=60$, the SQP algorithm began encountering underflow problems similar to those seen in Experiment 1). In all instances though, the computational effort as measured by the number of shortest path computations was substantially lower for the SQP approach.

TABLE 7

SOLUTION RESULTS FOR "SIOUX FALLS" NETWORK

Variable	SQP					Hooke-Jeeves	EDO
θ	10.	20.	30.	40.	50.		
y_{16}	3.96	4.11	4.22	4.31	4.33	4.80	4.59
y_{17}	0.86	0.79	0.80	0.81	0.83	1.20	1.52
y_{19}	4.23	4.21	4.28	4.34	4.36	4.80	5.45
y_{20}	1.21	1.15	1.14	1.14	1.15	0.80	2.33
y_{25}	1.86	2.00	2.06	2.06	2.05	2.00	1.27
y_{26}	2.29	2.46	2.49	2.48	2.46	2.60	2.33
y_{29}	1.84	1.60	1.45	1.36	1.31	4.80	0.41
y_{39}	4.28	4.12	4.16	4.23	4.19	4.40	4.59
y_{48}	1.12	1.05	1.04	1.05	1.06	4.80	2.71
Y_{74}	4.53	4.34	4.39	4.44	4.40	4.40	2.71
SC	101.8	90.4	86.5	84.8	84.0	80.8	83.1
Iterates	25	31	35	37	41	147	89

SUMMARY AND CONCLUSIONS

This report began by identifying the network design problem as an approach to making optimal decisions concerning roadway expansion. Although in its current formulation it is a somewhat idealized and simplified compared to the problems faced by highway decision makers, it is to be hoped that success in solving simplified versions will encourage progressive incorporation of realistic complexity, until useful practical tools result. Currently though no truly satisfying solution procedure exists for even these simple NDPs, primarily due to the numerical difficulties encountered when requiring that the link volumes be the outcome of a deterministic user equilibrium assignment. It was argued that employment of stochastic user equilibrium assignment avoided both the nondifferentiability problem encountered when using "implicit function" versions of the NDP with DUE assignment, and the problem with intractably large numbers of constraints encountered when using "constrained" versions, making the NDP with SUE assignment an attractive alternative to existing procedures. Computer codes for solving the SUE constrained NDP were written implementing two standard numerical optimization techniques, sequential quadratic programming and the generalized reduced gradient method. These were tested on several example problems, yielding the following results.

First, by using explicit expressions for the derivatives of logit-based link use probabilities and an algorithm which computes these derivatives without requiring path enumeration, the methods of differentiable optimization become computationally feasible candidates for solving continuous NDPs. Second, the results of the first and third numerical experiments indicate that, at least for the range of congestion likely to be encountered in real traffic networks, a sequential

quadratic programming algorithm for solving the SUE-constrained version of the NDP can be viewed as a computationally attractive alternative to DUE-based methods, even when the DUE is considered a more accurate representation of traveller behavior. Third, the results of the second experiment provide no clear-cut basis for choosing between sequential quadratic programming and the generalized reduced gradient algorithm for solving SUE-constrained NDPs. Particularly impressive is the performance of SQP in solving NDPs for the Sioux Falls network. Although it was not possible to get SQP to duplicate the solutions obtained by the DUE-constrained procedures, by increasing the value of the θ parameter it was possible to a solution which was roughly equivalent to the DUE solutions, with apparently less computational effort.

In conclusion, by showing how a logit-based route choice model can be used to solve one of the difficult problems in transportation research, this project has produced good evidence for the feasibility of using such models as tools in transportation planning. By developing the software necessary for implementing the logit-model, including routines for efficient computation of link-use and joint-link probabilities, this project has constructed a foundation for extending this model to more complicated transportation management problems. In fact such extensions are being actively pursued at this very moment. At this point two directions for further recommend themselves. First, efforts should be made to solve NDPs for larger, more realistic road networks. A particularly attractive line of work would be to exploit the natural parallel structure in traffic assignment to devise efficient solution procedures using parallel processors. Second, solution methods which allow for uncertainty, in the travel demand totals and in the traffic volumes, are needed to more accurately describe the situation actually faced by decision-makers, leading to a stochastic optimization approach to NDPs.

REFERENCES

Abadie, J. and Carpentier, J. (1969) Generalization of the Wolfe Reduced Gradient Method to the case on nonlinear constraints. in R. Fletcher (Ed.) *Optimization*.

Abdulaal, M. and LeBlanc, L. (1979) Continuous equilibrium network design models. *Transportation Research*, **13B**, 19-32.

Bazaraa, M. and Shetty, C. (1979) *Nonlinear Programming: Theory and Algorithms*, Wiley and Sons, New York.

Ben-Akiva, M. DePalma, A. and Kanaroglou, P. (1986) Effects of capacity constraints on peak-period traffic congestion. *Transportation Research Record*, **1085**, 16-26.

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

Chen, M. (1991) *A Study of Transportation Network Design Problems*, Doctoral Dissertation, Dept. of Mechanical and Industrial Engineering, University of Manitoba, Winnipeg, Manitoba.

Chen, M. and Alfa, A. (1991) A network design algorithm using a stochastic incremental traffic assignment approach. *Transportation Science*, **25**, 215-224.

Dafermos, S. (1980) Traffic equilibria and variational inequalities. *Transportation Science*, **14**, 42-54.

Daganzo, C. (1982) Unconstrained extremal formulation of some transportation equilibrium problems. *Transportation Science*, **16**, 332-361.

Dial, R. (1971) A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, **5**, 83-111.

Fisk, C. (1980) Some developments in equilibrium traffic assignment methodology. *Transportation Research*, **14B**, 243-256.

Friesz, T. (1985) Transportation network equilibrium, design and aggregation: Key developments and research opportunities. *Transportation Research*, **19A**, 413-427.

Gill, P., Murray, W., Saunders, M., and Wright, M., (1989) Constrained nonlinear programming, in Nemhauser, G. et al. (Eds.) *Optimization, Volume 1*, North-Holland, Amsterdam, 171-210.

Horowitz, J. (1984) The stability of stochastic equilibrium in a two-link transportation network, *Transportation Research*, **18B**, 13-28.

Lasdon, L. and Waren, A. (1982) *GRG2 User's Guide*, Dept. of General Business, University of Texas, Austin, TX.

LeBlanc, L. (1975) An algorithm for the discrete network design problem. *Transportation Science*, **9**, 183-199.

Marcotte, P. (1983) Network optimization with continuous control parameters. *Transportation Science*, **17**, 181-197.

Murtagh, M. and Saunders, M. (1982) A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Mathematical Programming Study*, **16**, 84-117.

Powell, W. and Sheffi, Y. (1982) The convergence of equilibrium algorithms with predetermined step sizes. *Transportation Science*, **16**, 45-55.

Sheffi, Y. (1985) *Urban Transportation Networks*, Prentice-Hall, Englewood Cliffs, NJ.

Sheffi, Y. and Powell, W. (1981) A comparison of stochastic and deterministic traffic assignment over congested networks. *Transportation Research*, **15B**, 53-64.

Smith, M. (1979) The existence, uniqueness and stability of traffic equilibria. *Transportation Research*, **13B**, 295-304.

Stephanedes, Y., Kwon, E. and Michalopolous, P. (1989) Demand diversion for vehicle guidance, simulation and control in freeway corridors. *Transportation Research Record*, **1220**, 12-20.

Suwansrikul, C., Friesz, T. and Tobin, R. (1987) Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. *Transportation Science*, **21**, 254-263.

Rinnooy Kan, A. and Timmer, G., (1989) Global optimization. in Nemhauser, G. et al. (Eds.) *Optimization, Volume 1*, North-Holland, Amsterdam, 631-662.

Tan, H., Gershwin, S. and Athans, M. (1979) *Hybrid Optimization in Urban Traffic Networks*, MIT Report Dot-TSC-RSPA-79-7.

Van Vliet, D. (1981) Selected node-pair analysis in Dial's assignment algorithm. *Transportation Research*, **15B**, 65-68.

Wardrop, J. (1952) Some theoretical aspects of road traffic research. *Proc. of the Inst. of Civil*

Engineers, Part II, 325-378.

Wolfe, P. (1963) Methods of nonlinear programming. R.L Graves and P. Wolfe (Ed.) Recent *Advances in Mathematical Programming*.

A STOCHASTIC USER EQUILIBRIUM APPROACH FOR SOLUTION OF THE CONTINUOUS NETWORK DESIGN PROBLEM

Final Report - Appendix

Prepared by

Gary A. Davis, Ph.D.
Department of Civil and Mineral Engineering
University of Minnesota

Submitted to

Research Administration and Development Section
Office of Materials and Research
Minnesota Department of Transportation

September, 1991

APPENDIX

We desire derivatives of the constraint functions with respect to both the link volumes and the capacity increases, i.e.

$$\frac{\partial}{\partial x_i}(x_k - \sum_{j=1}^{m} d_j q_{jk}(c)) \qquad \frac{\partial}{\partial y_i}(x_k - \sum_{j=1}^{m} d_j q_{jk}(c)) \tag{13}$$

The derivative of the constraint equation k with with respect to x_i is simply

$$\frac{\partial}{\partial x_i}(x_k - \sum_{j=1}^{m} d_j q_{jk}(c)) = \delta_{ki} - \sum_{j=1}^{m} d_j \frac{\partial q_{jk}(c)}{\partial x_i} \tag{14}$$

where δ_{ki} is the Kronecker delta. The problem becomes one of finding the derivatives of the link use probabilities $q_{jk}(c)$, and using the definition of $q_{jk}(c)$ given in equation (6), we have

$$\begin{aligned} \frac{\partial q_{jk}(c)}{\partial x_i} &= (\frac{\partial}{\partial x_i}) \sum_r \delta_{jrk} p_{jr}(c) \\ &= \sum_r \delta_{jrk} (\frac{\partial}{\partial x_i}) \frac{e^{-\theta c_{jr}}}{\sum_s e^{-\theta c_{js}}} \\ &= -\theta (\frac{\partial c_i}{\partial x_i}) (\sum_r \delta_{jrk} \delta_{jri} p_{jr} - \sum_r \delta_{jrk} p_{jr} \sum_s \delta_{jsi} p_{js}) \end{aligned} \tag{15}$$

where the last equality follows from the definition of the path costs given in equation (5). Defining

$$\tilde{q}_{jki} = \sum_r \delta_{jrk}\delta_{jri}p_{jr} \tag{16}$$

as the probability a trip between OD pair j uses both links k and i, and making the appropriate substitutions, we obtain

$$\begin{aligned}&(\frac{\partial}{\partial x_i})(x_k - \sum_{j=1}^{m} d_j q_{jk}(c)) \\ &= \delta_{ki} + \theta(\frac{\partial c_i}{\partial x_i})(\tilde{q}_{jki} - q_{jk}q_{ji})\end{aligned} \tag{17}$$

which is the formula given in equation (11). The derivatives with respect to the capacity expansion y_k can be obtained in a similar fashion.

Using the definitional equations (6) and (16) to compute the quantities $\tilde{q}_{jki}$ and q_{jk} would require first computing the route choice probabilities p_{jr}, a task which is prohibitively difficult for even moderate-sized networks. A similar problem arises in computing the assignment of traffic to links when route choices follow the logit relation, and in 1971 Robert Dial presented an algorithm which could compute such link volumes without first evaluating the route choice probabilities. Ten years later Van Vliet (1981) published a short paper which described how one might compute the link use probabilities q_{jk} by modifying Dial's original algorithm. To see how this is done, let O denote the origin node for OD pair j, D denote the destination node for OD

pair j, let K_1 denote the start node of link k and let K_2 denotes the end node of link k. A path which uses link k on the way from O to D must have the form

$$R_{O\text{-}K_1} \rightarrow R_{K_1\text{-}K_2} \rightarrow R_{K_2\text{-}D} \tag{18}$$

where $R_{K\text{-}I}$ denotes a path from node K to node I. Van Vliet showed that by modifying Dial's forward pass, it is possible to obtain a weighting term $W_{K\text{-}I}$ for the paths between nodes K and I and a nodal weighting term W_D, without enumerating the paths, such that such that the link use probability q_{jk} is given by

$$q_{jk} = \frac{W_{O\text{-}K_1} \times W_{K_1\text{-}K_2} \times W_{K_2\text{-}D}}{W_D} \tag{19}$$

Similarly, a route which travelled from O to D and traversed link k and later link i could be denoted by

$$R_{O\text{-}K_1} \rightarrow R_{K_1\text{-}K_2} \rightarrow R_{K_2\text{-}I_1} \rightarrow R_{I_1\text{-}I_2} \rightarrow R_{I_2\text{-}D} \tag{20}$$

so that the probability of using both links k and i would be

$$\tilde{q}_{jki} = \frac{W_{O\text{-}K_1} \times W_{K_1\text{-}K_2} \times W_{K_2\text{-}I_1} \times W_{I_1\text{-}I_2} \times W_{I_2\text{-}D}}{W_D} + \frac{W_{O\text{-}I_1} \times W_{I_1\text{-}I_2} \times W_{I_2\text{-}K_1} \times W_{K_1\text{-}K_2} \times W_{K_2\text{-}D}}{W_D} \tag{21}$$