Investigating Path-Based Solution Algorithms to the Stochastic User Equilibrium Problem

Shlomo Bekhor[a,*] and Tomer Toledo[b]

a. Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa, 32000, Israel

sbekhor@tx.technion.ac.il

b. Intelligent Transportation Systems Program, Massachusetts Institute of Technology, NE20-208 Cambridge MA 02139-4307 USA

toledo@mit.edu

Published in Transportation Research part B 39(3), pp. 279-295, 2005

* Corresponding author. Tel: (972)-4- 8292460, Fax: (972)-4-8225716

Abstract

This paper focuses on path-based solution algorithms to the SUE and investigates their convergence properties. Two general optimization methods are adapted to solve the logit SUE problem. First, a method that closely follows the GP algorithm developed for the deterministic problem is derived. While this method is very efficient for the deterministic user equilibrium problem, we use a simple example to illustrate why it is not suitable for the SUE problem. Next, a different variant of gradient projection, which exploits special characteristics of the SUE solution, is presented. In this method the projection is on the linear manifold of active constraints.

The algorithms are applied to solve simple networks. The examples are used to compare the convergence properties of the algorithms with a path-based variant of the Method of Successive Averages (MSA) and with the Disaggregate Simplicial Decomposition (DSD) algorithm.

1. Introduction

This paper investigates path-based algorithms to solve the stochastic user equilibrium (SUE) problem. Daganzo and Sheffi (1977) define the SUE as a state in which no driver can improve his/her perceived travel time by unilaterally changing routes. The SUE assignment problem is that of finding the link (and path) flows on a traffic network given travel demand between origins and destinations and the corresponding path choice sets (either explicit or implicit) and assuming a probabilistic route choice model.

The SUE problem has long been studied in the literature. Thorough reviews are presented in Sheffi (1985), Thomas (1991), Patriksson (1994), and Bell and Iida (1997). The SUE problem may be formulated and solved either in the space of link flows or in the space of path flows. Most solution algorithms proposed in the literature are link-based. An important advantage of link-based solutions is that they do not require explicit enumeration of the path choice set, and so, may be easily applied to large-scale networks. Instead of enumerating paths, link-based solutions assume an implicit choice set, such as the use of all efficient paths (Maher 1998, Dial 1999), or all cyclic and acyclic paths (Bell 1995, Akamatsu 1996). The correctness of the solution necessitates these implicit choice sets, which, however, may be unrealistic from a behavioral standpoint. It is clearly difficult to justify the assumption that cyclic paths should be used, but there may also be efficient paths that are unrealistic, such as paths involving repeatedly getting on and off a freeway. Path-based formulations allow a more flexible definition of the choice set that can accommodate these considerations.

New choice set generation methods that have been developed in recent years facilitate the use of path-based approaches. A well-known example is the labeling method proposed by

Ben-Akiva et al (1984). In this method, a large number of optimality criteria are defined on the basis of surveyed choice motivations. These criteria (labels) include the shortest route, the quickest route, the best signposted route, the route that maximizes motorway use, and so on. An optimal path is found for each criteria and the choice set consists of all the paths generated by the various criteria. The study found that six labels covered approximately 90% of all traveled routes, thus showing that choice sets need not be exhaustive. Cascetta et al (1996, 1997) incorporated the labeling method coupled with k-shortest path methods in solving the SUE problem.

At the same time, route choice behaviors have been further investigated, and new models were developed in an attempt to better capture the similarity among routes. While a unique one-to-one mapping between link-based and path-based solutions exists for SUE assuming a logit route choice model, it may not be so for other route choice models. Therefore, a path-based solution may be required in order to obtain path related information. Moreover, many of the link-based solution approaches, such as Dial (1999), are specific to the logit route choice model and may not be adapted to other route choice models. While the work presented in this paper focuses on the logit SUE assignment, it can be adapted to SUE formulations using other route choice models.

This paper focuses on path-based solution algorithms to the logit SUE and investigates their convergence properties. Two general optimization methods are adapted to solve the logit SUE formulation of Fisk (1980). First, a method that closely follows the GP algorithm developed for the deterministic problem by Berteskas and Gafni (1982) is derived. While this method is very efficient for the deterministic user equilibrium problem, we use a simple example to illustrate why it is not suitable for the SUE problem.

Second, a different variant of gradient projection, which exploits special characteristics of the SUE solution is presented. In this method the projection is on the linear manifold of active constraints. These methods are compared with the path-based variant of the Method of Successive Averages (MSA) and with the algorithm of Damberg et al (1996), which is an adaptation to the SUE problem of the Disaggregate Simplicial Decomposition (DSD) algorithm (Larsson and Patriksson 1992).

The rest of this paper is organized as follows: The next section reviews the logit SUE problem and algorithms developed to solve it. The adaptation of DSD algorithm for logit SUE is also described. Next, two new adaptations of gradient projection methods to the SUE problem are described followed by numerical results comparing the more useful of these algorithms with the MSA and DSD algorithms for simple networks. We conclude the presentation with a discussion of the results.

2. Logit SUE Assignment Problem

2.1 Notations and Formulations

Consider a directed graph $G=\left(N,A\right)$, where N and A are the sets of nodes and links, respectively. For each link, flow-dependent travel costs are defined, $c_a(x_a),\ \forall a\in A$, and we assume throughout this paper that the cost functions are continuous and differentiable. A set of origin-destination (OD) pairs, RS, with predefined demands for travel, $q^{rs}\ \ \forall rs\in RS$, from origin r to destination s, and sets of alternative paths, $K^{rs}\ \ \forall rs\in RS$, for each OD pair are also given. The logit SUE assignment problem concerns with finding the equilibrium flow patterns that will result if route fractions are defined by a logit model:

$$f_k^{rs} = \frac{\exp\left(-\theta C_k^{rs}\right)}{\sum_{j \in K^{rs}} \exp\left(-\theta C_j^{rs}\right)} q^{rs} \tag{1}$$

f_k^{rs} and C_k^{rs} are the flow and travel cost on path *k* connecting OD pair *rs*, respectively. θ is a positive dispersion parameter, which reflects an aggregate measure of drivers' perception of travel costs (Sheffi 1985). Higher values of θ indicate that drivers have a more accurate perception of travel times, and consequently tend to choose the least-cost path. As a result, route choices become less "dispersed" when θ increases.

Two equivalent formulations of the logit SUE problem have been proposed. Sheffi and Powell (1982) proposed a general formulation, given by:

$$min \;\; Z = \sum_{a \in A} x_a c_a(x_a) - \sum_{rs \in RS} q^{rs} S^{rs} - \sum_a \int_0^{x_a} c_a(w) dw \tag{2}$$

S^{rs} is a satisfaction function, defined as the expected perceived travel cost from *r* to *s*:

$$S^{rs} = E\left[\min_{k \in K^{rs}} \left\{ C_k^{rs} \right\} \right] \tag{3}$$

The above formulation may be applied to a variety of route choice models that meet certain conditions imposed on the satisfaction function. For the logit model, the satisfaction function is given by:

$$S^{rs} = -\frac{1}{\theta} \ln \sum_{k \in K^{rs}} \exp\left(-\theta C_k^{rs}\right) \tag{4}$$

This formulation includes path calculations, however, in the case of the logit model, coupled with the implicit choice sets described earlier, the objective function may be evaluated at the link level. This property is used to derive link-based solution algorithms to the logit SUE problem, as in Sheffi (1985) and Maher (1998).

Fisk (1980) developed a path-based formulation for the logit SUE problem, using an entropy term as follows:

$$min\ Z = \sum_{a \in A} \int_0^{x_a} c_a(w)dw + \frac{1}{\theta} \sum_{rs} \sum_{k \in K^{rs}} f_k^{rs} \ln f_k^{rs} \qquad (5)$$

Subject to:

$$\sum_{k \in K^{rs}} f_k^{rs} = q^{rs} \quad \forall rs \qquad (6)$$

$$f_k^{rs} \geq 0 \qquad \forall k, \forall rs \qquad (7)$$

For $f_k^{rs} = 0$, the term $f_k^{rs} \ln f_k^{rs}$ is defined by its limit and set equal to zero. Similar entropy-based formulations have been proposed for SUE assignment based on other route choice models, including cross-nested logit and paired combinatorial logit (Bekhor and Prashker 1999).

2.2 Link-Based Algorithms

The well-known method Method of Successive Averages (MSA) developed by Powell and Sheffi (1982) was the first algorithm applied to solve the SUE problem. This algorithm can be applied with any stochastic network loading method. The step size is predetermined by a descent sequence with respect to the iterations.

Maher (1998) developed link-based algorithms for the logit SUE problem, using Sheffi and Powell's formulation. Maher proposed an algorithm that uses the same search direction as the MSA algorithm, but calculates an approximately optimal step size in this direction, thus improving overall convergence. Two adaptations of the Davidon-Fletcher-Powell (DFP) method were also considered, but were found inferior to the above method.

Another example is the entropy-based algorithm developed by Dial (1999), which is specific for the logit route choice model. This algorithm exploits the fact that for the logit function, it is possible to map path flows from link flows and vice-versa.

Dial’s and Maher’s algorithms exploit mathematical properties of the logit function to develop efficient link-based algorithms. Recent research on path-based algorithms has demonstrated and established that it is a viable approach for deterministic traffic assignment problems with reasonably large network sizes: see, for example, Chen et al. (2002). Much of the attention has been focused on two algorithms: the disaggregate simplicial decomposition (DSD) algorithm and the gradient projection (GP) algorithm. Adaptations of these algorithms to the logit SUE are presented in subsequent sections.

2.3 Early Path-Based Algorithms

Several algorithms to solve the logit SUE problem can be found in the literature, exploiting Fisk’s formulation. Chen and Alfa (1991) proposed a modification of the Frank-Wolfe algorithm regarding the step size computation. This computation requires an inverse of a link-path incidence matrix, which makes the algorithm impractical for large networks. Huang (1995) proposed an improvement to this algorithm, which avoided the matrix inversion. The algorithm uses Dial's (1971) STOCH method to generate the path-set. However, the “efficient paths” obtained using this method may be different at each iteration of the algorithm, which may lead to inconsistent path flows.

Bell et al (1993) proposed an algorithm in which the step size computation is accomplished by iterative balancing, in a similar way to entropy-maximizing trip distribution models. The balancing is performed using the link-path incidence matrix, in which each entry in the matrix correspond to the entropy flow, and the marginals corresponds to the total link flows and total demand for each OD pair. The algorithm uses a column generation procedure to generate routes; therefore, the number of iterations limits the number of paths. This means that the dimensions of the problem (the link-path incidence matrix) increases with the number of the iterations. Leurent (1997) commented that this algorithm cannot guarantee a stable set of efficient paths when application data are slightly modified.

2.4 The Algorithm of Damberg-Lundgren-Patriksson

Damberg et al (1996) extended the DSD algorithm (Larsson and Patriksson 1992) to solve the logit SUE problem. This section briefly describes the method. Suppose that at iteration n a feasible path-flow solution is given. The first term in Fisk's formulation [equation (5)] is linearized, which amount to assuming that travel costs are fixed at their current values. The resulting sub-problem is given by:

$$min\ Z = \sum_{rs} \sum_{k \in K^{rs}} c_k^{rs(n)} f_k^{rs} + \frac{1}{\theta} \sum_{rs} \sum_{k \in K^{rs}} f_k^{rs} \ln f_k^{rs} \qquad (8)$$

$c_k^{rs(n)}$ is the travel cost on path k based on the vector of path-flows at iteration n.

The solution to the above sub-problem is given by:

$$h_k^{rs(n)} = q^{rs} \frac{\exp\left(-\theta c_k^{rs(n)}\right)}{\sum_{j \in K_{rs}} \exp\left(-\theta c_j^{rs(n)}\right)} \qquad \forall k, \forall rs \qquad (9)$$

If the vector $h^{(n)} - f^{(n)}$ is nonzero, it defines a descent direction with respect to the objective function (5). A line search in this direction is performed to find the optimal step size λ as follows:

$$\lambda^{(n)} = arg \min_{\lambda \in [0,1]} Z\left[f^{(n)} + \lambda \left(h^{(n)} - f^{(n)} \right) \right] \tag{10}$$

The new solution is given by:

$$f_k^{rs(n+1)} = f_k^{rs(n)} + \lambda^{(n)} \left(h_k^{rs(n)} - f_k^{rs(n)} \right) \quad \forall k, \forall rs \tag{11}$$

The flows calculated in equation (11) are then used to update link costs and path costs. The new sub-problem with the updated path costs is solved using equation (9), to produce a new descent direction. This iterative process continues until the convergence criterion is satisfied.

3. Alternative Algorithms

3.1 Algorithm GP – Gradient Projection using a reference path flow

In this section, we present an adaptation of the GP algorithm to solve the SUE problem that closely follows the algorithm developed for the deterministic problem by Bertsekas and Gafni (1982).

Consider a set of paths K^{rs} for a given origin-destination pair *rs*. Assume that $\bar{k}_{rs}$ is a reference path (the specific path will be defined later). The complement set of paths is defined as $\tilde{K}^{rs} = \left\{ k \middle| k \in K^{rs},\ k \neq \bar{k}^{rs} \right\}$. Demand constraints [equation (6)] can be expressed by:

$$f_{\bar{k}}^{rs} = q^{rs} - \sum_{\tilde{k} \in \tilde{K}^{rs}} f_{\tilde{k}}^{rs} \tag{12}$$

Fisk's SUE problem is redefined in the space of the complement-set path-flows:

$$\begin{aligned} min \;\; \tilde{Z} = & \sum_a \int_0^{x_a} c_a(w)dw + \frac{1}{\theta} \sum_{rs} \sum_{\tilde{k} \in \tilde{K}^{rs}} f_{\tilde{k}}^{rs} \ln f_{\tilde{k}}^{rs} \\ & + \frac{1}{\theta} \sum_{rs} \left(q^{rs} - \sum_{\tilde{k} \in \tilde{K}^{rs}} f_{\tilde{k}}^{rs} \right) \ln \left(q^{rs} - \sum_{\tilde{k} \in \tilde{K}^{rs}} f_{\tilde{k}}^{rs} \right) \end{aligned} \tag{13}$$

Subject to:

$$f_{\tilde{k}}^{rs} \geq 0 \;\; \forall \tilde{k}, \; \forall rs \tag{14}$$

The problem above has only the non-negativity constraints. The gradient for this problem is given by:

$$\frac{\partial \tilde{Z}}{\partial f_{\tilde{k}}^{rs}} = \sum_a C_a \left(\delta_{a\tilde{k}}^{rs} - \delta_{a\bar{k}}^{rs} \right) + \frac{1}{\theta} \left(\ln f_{\tilde{k}}^{rs} - \ln f_{\bar{k}}^{rs} \right) = c_{\tilde{k}}^{rs} - c_{\bar{k}}^{rs} + \frac{1}{\theta} \left(\ln f_{\tilde{k}}^{rs} - \ln f_{\bar{k}}^{rs} \right) \tag{15}$$

In the deterministic problem, first derivatives include only the path cost difference term. The reference path is then chosen to be the minimum cost path. Here, an equivalent reference path is the one that minimizes the generalized path cost (G_k^{rs}):

$$\bar{k}_{rs} = arg \min_{k \in K^{rs}} \; G_k^{rs}, \qquad G_k^{rs} = c_k^{rs} + \frac{1}{\theta} \ln f_k^{rs} \tag{16}$$

Second derivatives of the modified objective function (13) are given by:

$$\frac{\partial^2 \tilde{Z}}{\partial f_{\tilde{k}}^{rs} \partial f_{\tilde{l}}^{rs'}} = \sum_a \frac{\partial c_a}{\partial x_a} \left(\delta_{a\tilde{k}}^{rs} - \delta_{a\bar{k}}^{rs'} \right) \left(\delta_{a\tilde{l}}^{rs} - \delta_{a\bar{l}}^{rs'} \right) + \frac{1}{\theta} \left(\frac{1}{f_{\tilde{k}}} \delta_{\tilde{k}\tilde{l}} + \frac{1}{f_{\bar{k}}} \delta_{\bar{k}\bar{l}}^{rs} \right) \tag{17}$$

$\delta_{\tilde{k}\tilde{l}}$ is equal to 1 if $\tilde{k}=\tilde{l}$ and 0 otherwise. $\delta^{rs}_{\tilde{k}\tilde{l}}$ is equal to 1 if $\tilde{k}$ and $\tilde{l}$ connect the same OD pair *rs*, and 0 otherwise.

After calculating the derivatives, the gradient projection step can be evaluated. Given a feasible path flow solution at iteration *n*, link costs are calculated, and reference paths are found using equation (16) above. The updated flows on non-reference paths are obtained by:

$$\left(f^{rs}_{\tilde{k}}\right)^{n+1} = \left[\left(f^{rs}_{\tilde{k}}\right)^{n} - Q^{-1}\left(G^{rs}_{\tilde{k}} - G^{rs}_{\bar{k}}\right)\right]^{+} \tag{18}$$

Q^{-1} is the inverse of a positive definite scaling matrix. The "+" sign indicates projection on the positive orthant. This is needed to assure the non-negativity of the path flows. Flows on reference paths are obtained using the conservation constraints (12).

A key factor in the implementation of the algorithm is the choice of scaling matrix and calculation of its inverse. In the deterministic case, Bertsekas and Gafni (1982) and Jayakrishnan et al (1994) used a diagonal approximation of the Hessian matrix, enabling easy inversion. The diagonally scaled Newton method generates a "good" descent direction because the diagonal elements are dominant in the Hessian matrix. However, this may not be the case for the SUE problem, as illustrated by the following simple example.

Consider the 9-node grid network with 12 one-way streets and a single OD pair shown in Figure 1 below.

[Insert Figure 1 here]

Six different routes connect O and D. The following BPR link performance function was used:

$$c_a = c_a^0 \cdot \left(1 + 0.6\left(\frac{x_a}{s_a}\right)^4\right) \qquad (19)$$

c_a^0 is the free-flow travel cost on link a, x_a and s_a are the traffic flow and capacity of link a, respectively.

Both c_a^0 and s_a are input parameters to the assignment process. In this example, the capacity of all links is 1000 units. The free-flow travel costs are 1 unit each for links 6 and 8, and 2 units for the remaining links.

Initial solutions were obtained by applying the logit model with free-flow travel costs. Table 1 below summarizes the first derivative calculations for the first iteration.

[Insert Table 1 here]

In deterministic assignment, path 1, which is the least-cost path, would be used as a reference. In the stochastic case, path 6 has the minimum value of the generalized cost computed according to equation (16), and therefore is set as the reference path. Note that the first derivatives are always non-negative. However, the descent direction may be negative, depending on the Hessian matrix. Table 2 below shows the elements of the Hessian matrix.

[Insert Table 2 here]

Inspection of the Hessian matrix above shows that the diagonal elements are not significantly larger than the other elements and that the inverse of the Hessian matrix contains negative elements. Table 3 compares the updated flows computed using the full Hessian matrix with the ones generated by a diagonal approximation of the Hessian matrix. An optimal step size was used in this example.

[Insert Table 3 here]

With the diagonally scaled method, since both the gradient and the diagonal elements of the Hessian matrix are always positive, flows on non-shortest paths decrease, and the flow on the least-cost path increases. This behavior is desirable in the deterministic case, because of the equilibrium conditions. However, in the SUE assignment, equilibrium conditions also include the entropy term. The descent direction may lead to an increase in the flows on some of the non-shortest paths, as in paths 2, 3, 4 and 5. If the new step were computed according to the full Hessian, the new solution would be very close to the equilibrium solution (last column in Table 3).

Since inversion of the Hessian matrix is not practical for large networks, algorithm GP is not suitable to solve the SUE problem. We next propose a different variant of gradient projection in which the projection is on the linear manifold of active constraints.

3.2 Algorithm GP2 – Manifold optimization method

We first present the manifold optimization method. The discussion follows the presentation in Bertsekas (1999). Consider the general nonlinear minimization problem with equality constraints, in vector notation:

$$\begin{aligned} & min\ f(x) \\ & s.t.\ Ax = b \end{aligned} \tag{20}$$

Suppose that a feasible solution x^n at iteration n is given. A feasible descent direction, d^n, can be found by solving the following sub-problem:

$$\begin{aligned} & min\ \ \nabla f(x^n)'d + \tfrac{1}{2}d'Qd \\ & s.t.\ \ Ad = 0 \end{aligned} \tag{21}$$

Q is a positive definite matrix.

The solution, d^n, satisfies the equality constraints in (20), $A\left(x^n + d^n\right) = Ax^n + Ad^n = b$, and is therefore a feasible direction. It is also a descent direction since for $d^n \neq 0$: $\nabla f(x^n)'d^n + \tfrac{1}{2}d^{n\prime}Qd^n < 0$, and therefore $\nabla f(x^n)'d^n < -\tfrac{1}{2}d^{n\prime}Qd^n < 0$.

Sub-problem (21) can be easily solved in closed form by applying the Karush-Kuhn-Tucker (KKT) first order optimality conditions. The solution is given by:

$$d^n = -Q^{-1}\left(\nabla f(x^n) + A'\lambda\right) \tag{22}$$

$$\lambda = -\left(AQ^{-1}A'\right)^{-1} AQ^{-1}\nabla f(x^n) \tag{23}$$

The application of this algorithm to the SUE problem exploits the fact that the solution corresponds to a logit route choice model. The logit model assigns strictly positive choice probabilities to all paths in the choice set. This implies that non-negativity constraints will not be binding and can be ignored. Therefore, we can concentrate only on satisfying the demand equality constraints (6).

As with the GP algorithm, the selection of the scaling matrix Q must recognize the need to invert this matrix. Moreover, the structure of the binary A matrix is such that the direction finding problem (21) can be decomposed for each origin-destination pair if Q is diagonal (in fact, Q may be block-diagonal assuming it is arranged such that paths are ordered by OD pairs). Therefore, a natural selection is the diagonal of the Hessian matrix. The diagonal elements of this matrix are given by:

$$h_k^{rs} = \frac{\partial^2 Z}{\partial f_k^{rs2}} = \frac{\partial c_k^{rs}}{\partial f_k^{rs}} + \frac{1}{\theta}\frac{\partial \ln f_k^{rs}}{\partial f_k^{rs}} = \sum_a \frac{\partial c_a}{\partial x_a}\delta_{ak}^{rs} + \frac{1}{\theta f_k^{rs}} \tag{24}$$

The descent direction for path k is then obtained by (omitting the iteration index):

$$d_k^{rs} = -\frac{1}{h_k^{rs}}\left(\nabla f_k^{rs} - \frac{\sum_{l\in K_{rs}} \frac{\nabla f_l^{rs}}{h_l^{rs}}}{\sum_{l\in K_{rs}} \frac{1}{h_l^{rs}}}\right) = \frac{\sum_{l\in K_{rs}} \frac{\nabla f_l^{rs} - \nabla f_k^{rs}}{h_k^{rs} h_l^{rs}}}{\sum_{l\in K_{rs}} \frac{1}{h_l^{rs}}} \tag{25}$$

Another special case of the matrix Q is the identity matrix. In that case the search direction further simplifies to:

$$d_k^{rs} = \frac{\sum_{l\in K_{rs}} \nabla f_l^{rs}}{|K_{rs}|} - \nabla f_k^{rs} = \frac{\sum_{l\in K_{rs}} \left(c_l^{rs} + \frac{\ln f_l^{rs}}{\theta}\right)}{|K_{rs}|} - \left(c_k^{rs} + \frac{\ln f_k^{rs}}{\theta}\right) \tag{26}$$

$|K_{rs}|$ is the size of the path choice set for OD pair rs.

The above expression has an intuitive interpretation: the search direction is proportional to the differences between the average generalized cost and the generalized cost of the path in question, in an attempt to equalize these generalized costs.

In the examples above, the GP2 algorithm implements approximations of the Hessian matrix as the scaling matrix. Variable scaling methods, in which an estimate of the Hessian matrix (or its inverse) is updated at each iteration with first order information, may also be used. For example, an equality constrained version of the DFP method (e.g. Goldfarb 1969) may replace the inverse of the diagonalized Hessian. Maher (1998) proposed a similar approach for link-based algorithms using the unconstrained formulation of Sheffi and Powel (1982).

Table 4 shows the first iteration solution of the 9-node grid network starting with the free-flow travel cost solution.

[Insert Table 4 here]

Note that the new solution vector is quite similar to the solution obtained by the exact algorithm GP (full Hessian, see Table 3). However, in algorithm GP2, only the diagonal elements are needed to obtain an optimized descent direction. Therefore, this algorithm can be implemented to solve real size networks with affordable computer resources.

Although not required, it may be useful for the implementation of the algorithm to calculate an upper bound on the step size. A maximum step size that will ensure strict positive path flows is given by:

$$(\alpha)^n = \underset{k \in K_{rs}}{Min}\left(-\frac{f_k^{rs}}{d_k^{rs}}\right), \ \forall rs, \quad d_k^{rs} < 0 \tag{27}$$

In the above example, path 1 is the only one with a negative descent direction. The step size has to be strictly smaller than (466.9/50.7) = 9.2.

4. Results

This section presents some performance comparisons between the path-based MSA, DSD and GP2 algorithms. The GP algorithm is not compared, since it is clearly inferior, as explained in the previous section. A path-based implementation of the MSA algorithm using a step size (1/1+n) is presented to provide a benchmark for comparison. Exact step sizes were calculated for both the DSD and GP2 algorithms. All three algorithms were tested using an "all-at-once" implementation in which link and path costs are updated only after assigning all path flows in the network.

4.1 Grid Network

First, the network presented in Figure 1 (9-node grid network) is used. The demand is 1000 units. The progress of the various algorithms was evaluated using a measure of the deviation from the objective function at equilibrium (Leurent 1997):

$$\ln\left|\frac{Z^{(n)}}{Z^*} - 1\right| \tag{28}$$

$Z^{(n)}$ and Z^* are the values of the objective function evaluated at iteration n and at equilibrium, respectively. As Leurent (1997) pointed out, this measure is not representative of a practical application where Z^* would not be known a priori. However, it ensures that the testbed is fair to all competing algorithms.

Figure 2 compares the performance of three algorithms (MSA, DSD and GP2).

[Insert Figure 2 here]

The results presented in Figure 2 indicate that both DSD and GP2 outperform the MSA algorithm with a slight advantage of the DSD algorithm over the GP2 algorithm. However, the example is too simple to draw general conclusions. The next section presents comparison results for a more realistic network.

4.2 Sioux Falls Network

The next network compared is the well-known Sioux Falls network: see, for example, Leblanc (1973). This network is composed of 24 nodes, 76 links and 550 OD pairs. The paths were generated prior to the assignment, using a combination of the link elimination method and the k-shortest path method. In a dense network, the k-shortest path method generates routes with high degree of similarity. The link elimination method consists of successively removing links and finding shortest paths on the remaining links of the network. Only acyclic paths were considered in these methods. For more details on choice set generation methods, see Bekhor et al (2001). An average of 6.3 paths were generated for each OD pair, and the maximum number of routes generated was 12.

For the purposes of the present analysis, we assumed a value of 0.5 for the dispersion parameter θ based on empirical evidence (Ramming 2001), which shows that the coefficient for travel time ranges from –0.4 to –0.6, depending on the model structure and other parameters in the model. This value indicates that given a 5-minute difference between two paths, about 8% of the drivers will chose the route with the higher cost.

Figure 3 compares the performance of the algorithms tested (MSA, DSD and GP2).

[Insert Figure 3 here]

In this example, the convergence rates of both DSD and GP2 algorithms are similar. In contrary to the grid network example, for increasing precision levels, the GP2 converges slightly faster than DSD.

Note that the convergence rate for the GP2 algorithm is slower than for the other two algorithms in the first iterations. This may be the result of using an initial solution (common to all three algorithms) that was obtained with free-flow times. The GP2 descent direction yields some large negative flows, because of the big differences in the travel times. The step sizes are therefore relatively small, which compromises the speed of convergence. It may be possible to avoid this problem by performing a few iterations of a deterministic traffic assignment. These first iterations will produce travel times that are closer to the equilibrium solution. Once this "warm-up" assignment is finished, the GP2 algorithm can be applied.

4.2.1 Sensitivity Analysis

Many factors affect the performance of the algorithms, including the value of the dispersion parameter θ and the level of congestion in the network. There are well known results (see, for example, Sheffi 1985, and Thomas 1991) showing that as the dispersion parameter increases, the SUE solution becomes close to the deterministic UE solution. A similar effect is observed with respect to congestion level (Prashker and Bekhor 2000).

We conducted a test of the sensitivity of the performance of the various algorithms to the value of θ by varying its value and performing SUE assignments. Figure 4 presents the number of iterations required by the DSD and GP2 algorithms to achieve a solution within 0.01% of the equilibrium objective function value. In all cases, the MSA algorithm required more than 50 iterations to reach the same precision, and its results are therefore omitted.

[Insert Figure 4 here]

The performance of DSD and GP2 for the Sioux Falls network is quite similar for different values of θ, with GP2 algorithm performing slightly better than DSD.

A second sensitivity analysis was performed with respect to the total demand. We uniformly multiplied the Sioux Falls demand by a constant factor, and performed SUE assignment to observe the influence of the congestion level on algorithm performance. Since the Sioux Falls matrix is quite congested, we vary this factor from 0.1 to 1.5, in intervals of 0.1. Figure 5 presents the number of iterations required by the DSD and GP2 algorithms to achieve a solution within 0.01% of the equilibrium objective function value. As in the previous case, the MSA algorithm requires more than 50 iterations to reach the same precision level in all but the 4 lowest levels of demand considered, and is therefore omitted from the comparison.

[Insert Figure 5 here]

As in the previous example, the performance of DSD and GP2 for the Sioux Falls network is quite similar for different demand levels. For very high congestion level, the DSD algorithm slightly outperforms GP2. This may suggest that DSD is better when the SUE solution approaches the deterministic UE.

4.2.2 Influence of the Path Choice Set

It is also interesting to examine the impact of the size of the path choice set on the equilibrium solution reached. To test that, we varied the maximum numbers of paths allowed for each OD pair and preformed SUE assignments. Figure 6 below shows the equilibrium objective function values reached (same value for all algorithms).

[Insert Figure 6 here]

The results indicate that for the Sioux Falls network, relatively few paths are needed to achieve equilibrium. This is an encouraging result since it suggests that path-based SUE assignment may be performed without a need for complete path enumeration.

5. Conclusions

This paper investigated path-based algorithms for the SUE problem. Two new algorithms were presented, based on adaptations of gradient projection methods. An adaptation of the GP algorithm proposed for the deterministic UE problem was found to be unsuitable for the SUE problem. An different variant based on the projection of the gradient on the manifold of demand constraints (called GP2 in this paper) was found to suit better to solve the SUE problem.

Comparison results between MSA, DSD and GP2 algorithms were also presented using a small grid network and the well-known Sioux Falls network. The performance of DSD and GP2 is similar, and as expected, both significantly outperform the path-based MSA algorithm.

In this paper we applied a pre-defined path choice set to allow for a fair comparison between the algorithms. For larger networks, since it is not possible to enumerate all paths, different choice set generation methods should be tested, to verify both the convergence properties and the quality of the solution.

In this paper we limited the application to the logit SUE assignment. However, path-based solution algorithms (both DSD and GP2) may be applied to SUE assignment using more sophisticated route choice models within the generalized extreme value (GEV) family.

References

Akamatsu, T. (1996). Cyclic flows, Markov process and stochastic traffic assignment, *Transportation Research Part B*, **30**(5), pp. 369-386.

Akamatsu, T. (1997). Decomposition of Path Choice Entropy in General Transport Networks. *Transportation Science*, **31**(4), pp. 349-362.

Bekhor S., Ben-Akiva M., Ramming S. (2001). Route Choice: Choice Set Generation and Route Choice Models. Paper presented at the IV Tristan Conference, Azores, Portugal.

Bell, M.G.H. (1995). Alternatives to Dial's Logit Assignment Algorithm. *Transportation Research Part B*, **29(4)**, pp. 287-295.

Bell, M.G.H., and Iida Y. (1997). *Transportation Network Analysis.* John Wiley & Sons, Chichester, West Sussex.

Bell, M.G.H., Lam, W.H.K., Ploss, G. and Inaudi, D. (1993). Stochastic User Equilibrium Assignment and Iterative Balancing, *Proceedings of the 12th International Symposium on Transportation and Traffic Theory,* ed. C.F. Daganzo, Berkeley, CA, pp. 427-439.

Ben-Akiva, M., Bergman, M.J., Daly, A.J., and Ramaswamy, R. (1984). Modelling Inter Urban Route Choice Behaviour. In Proceedings of the 9th International Symposium on Transportation and Traffic Theory, eds. J. Volmuller and R. Hamerslag, VNU Press, Utrecht, pp. 299-330.

Bertsekas, D.P. (1976). On the Levitin-Goldstein-Poljak Gradient Projection Method. *IEEE Transactions on Automatic Control* **AC-21**, pp. 174-184.

Bertsekas, D.P. (1999). Nonlinear Programming. *Athena Scientific, Belmont, MA*.

Bertsekas, D.P. and Gafni, E. (1982). Projection Methods for Variational Inequalities with Application to the Traffic Assignment Problem. *Mathematical Programming Study* **17,** pp. 139-159.

Cascetta, E., Nuzzolo, A., Russo, F. and Vitetta, A. (1996). A Modified Logit Route Choice Model Overcoming Path Overlapping Problems: Specification and Some Calibration Results for Interurban Networks. In Proceedings of the International Symposium on Transportation and Traffic Theory, ed. J.B. Lesort, Lyon, pp. 697-711.

Cascetta, E., Russo, F., and Vitetta, A. (1997). Stochastic User Equilibrium Assignment with Explicit Path Enumeration: Comparison of Models and Algorithms, Proceedings of the 8th IFAC Symposium on Transportation Systems, ed. M.Papageorgiou and A.Pouliezos, pp. 1078-1084.

Chen, M., and Alfa, A.S. (1991). Algorithms for Solving Fisk's Stochastic Traffic Assignment Model. *Transportation Research Part B,* **25,** pp. 405-412.

Chen, A., Lee, D. and Jayakrishnan, R. (2002). Computational Study of State-of-the-art Path-based Traffic Assignment Algorithms. *Mathematics and Computers in Simulation,* **59,** pp. 509–518.

Damberg, O., Lundgren, J.T., and Patriksson, M. (1996). An Algorithm for the Stochastic User Equilibrium Problem. *Transportation Research Part B,* **30**, pp. 115-131.

Dial, R.B. (1971). A Probabilistic Multipath Traffic Assignment Algorithm Which Obviates Path Enumeration. *Transportation Research*, **5**, pp. 83-111.

Dial, R.B. (2001). Equilibrium logit traffic assignment: elementary theory and algorithms, *Transportation Research Part Bl,* ***In Press, Uncorrected Proof****, www.sciencedirect.com.*

Fisk, C. (1980). Some Developments in Equilibrium Traffic Assignment, *Transportation Research,* **14B**, pp. 243-255.

Goldfarb, D. (1969). Extension of Davidon's Variable Metric Method to Maximization under Linear Inequality and Equality Constraints. *SIAM J. Applied Math.*, **17**, pp.739-763.

Goldstein, A. A. (1964). Convex Programming in Hilbert Space, *Bulletin of the American Mathematical Society*, **70**, pp. 709-710.

Huang, H. (1995). A Combined Algorithm for Solving and Calibrating the Stochastic Traffic Assignment Model. *Journal of Operational Research Society*, **46**, pp. 977-987.

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

Larsson, T. and Patriksson, M. (1992). Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem. *Transportation Science*, **26**(1), pp. 4-17.

Leurent, F. (1997). Curbing the computational difficulty of the logit equilibrium assignment model. *Transportation Research Part B*, **31**(4), pp. 315-326.

Leventhal, T., Nemhauser, G. and Trotter Jr., L. (1973). A Column Generation Algorithm for Optimal Traffic Assignment. *Transportation Science*, **7**, pp. 168-176.

Levitin, E.S. and Polyak, B.T. (1966). Constrained Minimization Methods, *U.S.S.R. Computational Mathematics and Mathematical Physics*, **6,** pp. 1-50.

Maher, M. (1998). Algorithms for Logit-based Stochastic User Equilibrium Assignment. *Transportation Research Part B,* **32(8)**, pp. 539-549.

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

Prashker J., and Bekhor S. (2000). Some Observations on Stochastic User Equilibrium and System Optimum. *Transportation Research Part B*, **24(4)**, pp.277 - 291.

Ramming, M.S. (2001). Network Knowledge and Route Choice. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge.

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

Sheffi Y. and Powell, W.B. (1982). An Algorithm for the Equilibrium Assignment Problem with Random Link Times, *Networks*, **12**, pp. 191-207.

Thomas, R. (1991). *Traffic Assignment Techniques*. Avebury Technical, The Academic Publishing Group, Aldershot.

List of Tables and Figures (in the order of appearance in the text)

FIGURE 1 9-Node Grid Network

TABLE 1 Algorithm GP: First Derivative in iteration 1

TABLE 2 Algorithm GP: Hessian matrix in iteration 1

TABLE 3 Algorithm GP: Descent Direction

TABLE 4 Algorithm GP2: First Iteration

FIGURE 2 Convergence of path-based algorithms – grid network

FIGURE 3 Convergence of path-based algorithms – Sioux Falls network

FIGURE 4 Influence of the Dispersion Parameter on Algorithm Performance

FIGURE 5 Influence of Demand on Algorithm Performance

FIGURE 6 Influence of the Path Choice Set on the Equilibrium Solution

FIGURE 1 9-Node Grid Network

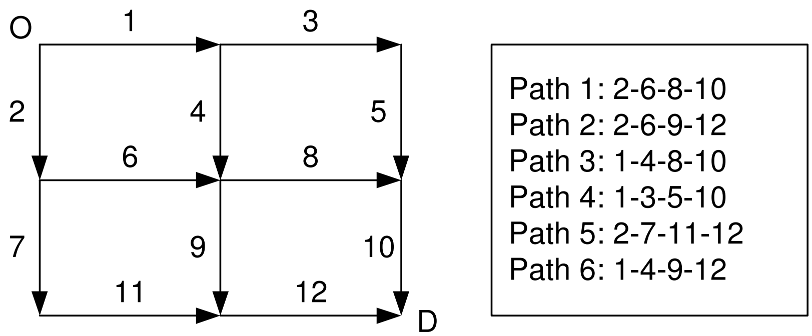

TABLE 1 Algorithm GP: First Derivative in iteration 1

Path	Flow	Cost	Generalized Cost	First Derivative
1	466.9	6.78	12.93	0.7558
2	171.8	7.40	12.55	0.3779
3	171.8	7.40	12.55	0.3779
4	63.2	8.30	12.45	0.2744
5	63.2	8.30	12.45	0.2744
6	63.2	8.03	12.17	

TABLE 2 Algorithm GP: Hessian matrix in iteration 1

	1	2	3	4	5
1	0.0229	0.0183	0.0183	0.0177	0.0177
2	0.0183	0.0241	0.0158	0.0159	0.0177
3	0.0183	0.0158	0.0241	0.0177	0.0159
4	0.0177	0.0159	0.0177	0.0336	0.0160
5	0.0177	0.0177	0.0159	0.0160	0.0336

TABLE 3 Algorithm GP: Descent Direction

		Diagonal Elements		Full Hessian		
Path	Current Solution	Direction	New Solution*	Direction	New Solution	Optimal Solution
1	466.9	33.0	401.9	74.5	392.4	391.3
2	171.8	15.7	140.9	-16.0	187.8	186.2
3	171.8	15.7	140.9	-16.0	187.8	186.2
4	63.2	8.2	47.1	-10.3	73.5	73.8
5	63.2	8.2	47.1	-10.3	73.5	73.8
6	63.2		222.1		85.1	88.7

* Optimized step size = 1.97

TABLE 4 Algorithm GP2: First Iteration

Path	Current Solution	Generalized Cost	Diagonal Hessian Elements	Direction	New Solution*
1	466.9	12.93	0.0067	-50.7	416.2
2	171.8	12.55	0.0083	4.5	176.3
3	171.8	12.55	0.0083	4.5	176.3
4	63.2	12.45	0.0176	8.0	71.2
5	63.2	12.45	0.0176	8.0	71.2
6	63.2	12.17	0.0162	25.6	88.8

* Optimized Step Size = 0.991

FIGURE 2 Convergence of path-based algorithms - Grid network

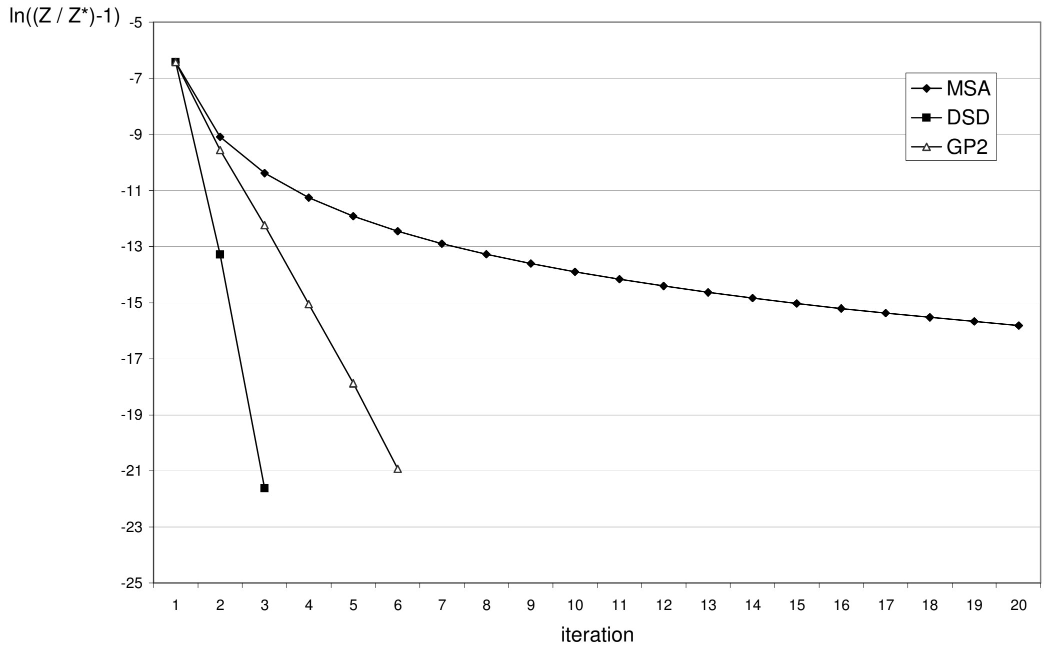

FIGURE 3 Convergence of path-based algorithms – Sioux Falls network

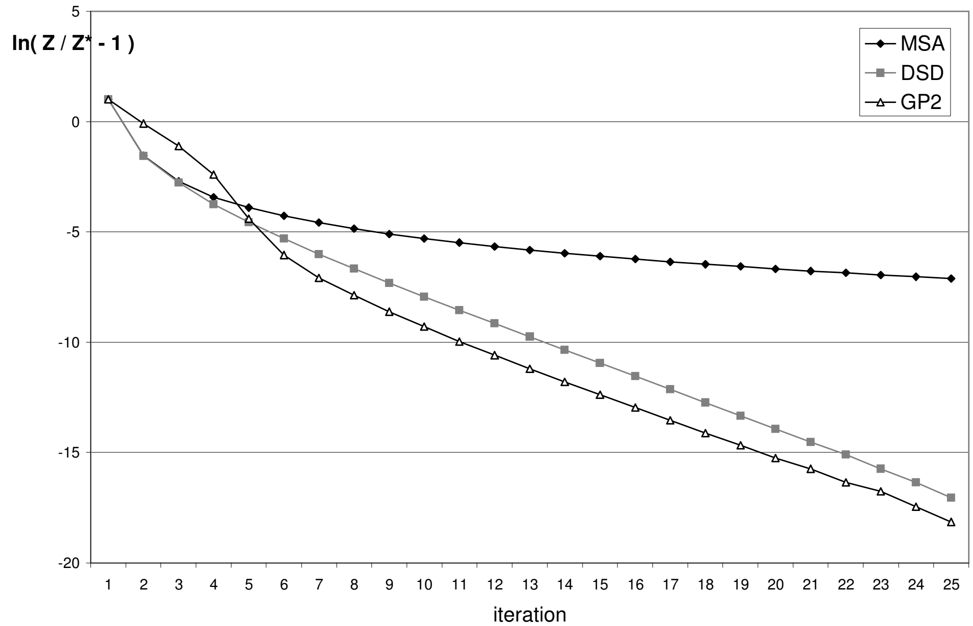

FIGURE 4 Sensitivity of the Algorithm Performance to the Dispersion Parameter

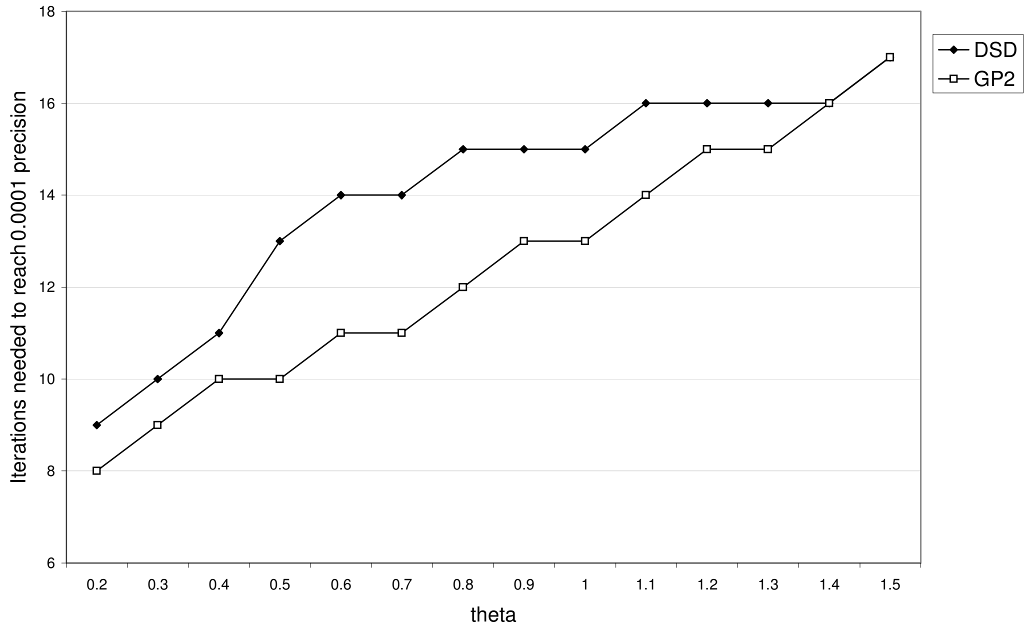

FIGURE 5 Sensitivity of the Algorithm Performance to the Level of Demand

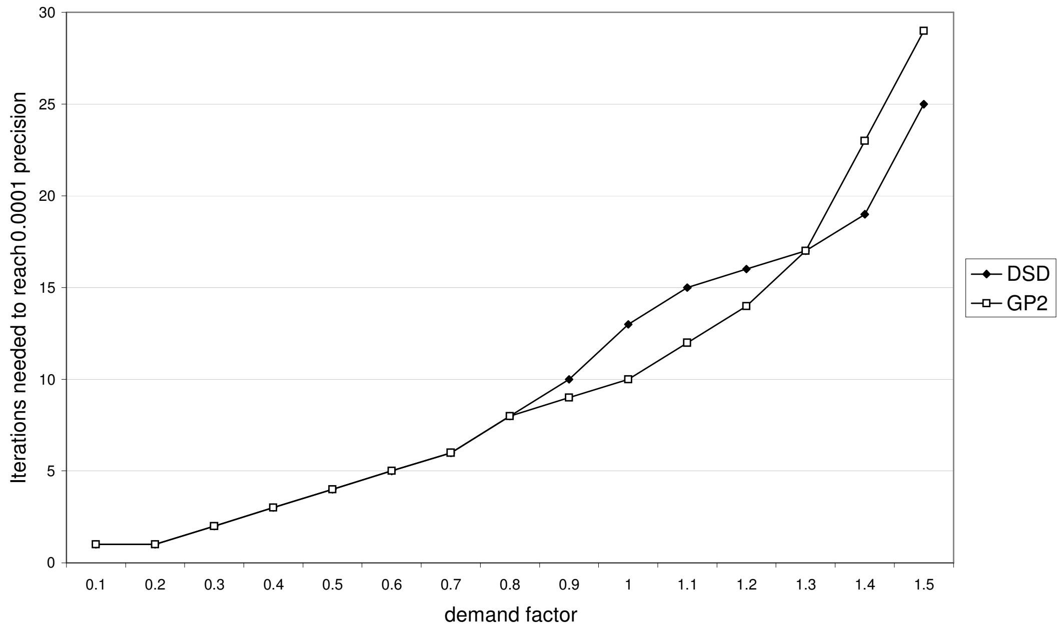

FIGURE 6 Influence of the Path Choice Set on the Equilibrium Solution