A Decomposition Scheme for Continuous Network Design Problem with Asymmetric User Equilibria

Xuegang Ban*
California Center for Innovative Transportation (CCIT)
Institute of Transportation Studies (ITS)
University of California, Berkeley
2105 Bancroft Way, Suite 300
Berkeley, CA 94720-3830
Tel: (510) 642-5112, Fax: (510) 642-0910
Email: xban@berkeley.edu
* **Corresponding Author**

Henry X. Liu
Department of Civil Engineering
University of Minnesota – Twin Cities
122 Civil Engineering Building
500 Pillsbury Drive S.E.
Minneapolis, MN 55455
Tel: (612) 625-6347, Fax: (612) 626-7750
Email: henryliu@umn.edu

Jiangang Lu
Department of Civil and Environmental Engineering
University of Wisconsin at Madison
1415 Engineering Drive
Madison, WI 53706
Tel: (608) 262-2524, Fax: (608) 262-5199
Email: jiangang@cae.wisc.edu

Michael C. Ferris
Computer Sciences Department
University of Wisconsin at Madison
1210 West Dayton Street
Madison, WI 53706
Tel: (608) 262-4281, Fax: (608) 262-9777
Email: ferris@cs.wisc.edu

Submittal to Transportation Research Record
March, 2006

Word Counts: 4,838 + 2,000 (3 tables and 5 figures) = 6,838 words

Abstract

In this paper, we formulate the continuous network design problem as a mathematical program with complementarity constraints (MPCC) and present a Gauss-Seidel decomposition scheme for the solution of the MPCC model. The model has an upper level as a nonlinear programming problem and the lower level a nonlinear complementarity problem. By applying the complementarity slackness condition of the lower level problem, the original bilevel formulation can be converted into a single level nonlinear programming problem. In order to solve the single level problem, a decomposition scheme is developed which can resolve the possible dimensionality problem (i.e., a very large number of defining variables). The decomposition scheme is tested and promising results are shown for well-known test problems.

1. INTRODUCTION AND MOTIVATION

The continuous network design problem (CNDP) aims to determine the optimal capacity enhancement for a set of selected links in a given network by minimizing both the total system cost and each driver's travel cost (*1*). CNDP has for long been formulated as a bilevel programming problem with the upper level a nonlinear programming (NLP) problem to minimize the system cost and the lower level a user equilibrium (UE) problem to account for driver's route choice behavior. It was first proposed by Morlok et al. (*2*) and subsequently studied by, Tan (*3*), Marcotte (*4*), Suwansirikul et al. (*5*), Friesz et al. (*6*), and Yang (*7*), to name but a few. Most of these works on CNDP were focusing on heuristic approaches for solving the bilevel model. For more detailed reviews on CNDP prior to year 2001, we refer to Yang and Bell (*8*).

It has been proven over the years that a bilevel formulation has broad applications not only in the transportation area but also in other engineering and science fields (*9*). Particularly, in the mathematical programming literature, the bilevel programming problem is also termed as mathematical programs with equilibrium constraints (MPEC) which has been extensively studied (*10*). However, solving such a problem is normally difficult due to the non-convex and non-smooth characteristics of MPEC. Therefore, how to reformulate rigorously and solve efficiently a general bilevel problem still remains an active research topic in both transportation area and the mathematical programming community.

By exploring the special structure of CNDP, Meng et al. (*11*) converted the bi-level problem to a single level yet smooth one via introducing a particular gap function for the lower level UE problem. Although still a non-convex model, the resulting single level problem can be solved using existing NLP solution algorithms. Nevertheless, Meng's model was based on the symmetry assumption on the lower-level problem, i.e., there is no interaction among flows on different links. A general UE problem can not be formulated as an NLP; instead, a nonlinear complementarity problem (NCP) or variational inequality (VI) formulation needs to be adopted. Marcotte (*12*) investigated such a general bi-level model, i.e., an NLP for the upper-level and a VI for the lower-level. By defining certain gap functions, he transferred the bi-level problem to a single level one and solved using the penalty method. More recently, Patriksson and Rockafellar (*13*) presented a new reformulation technique to convert an MPEC into a constrained and locally Lipschitz minimization problem which can be further solved using a decent algorithm proposed in the same paper. However, both Marcotte (*12*) and Patriksson and Rockafellar (13) didn't further test their models using well-known CNDP examples in transportation field.

By formulating the asymmetric user equilibrium (AUE) as a link-node based NCP, Ban et al. (*14*) modeled CNDP with AUE as a mathematical program with complementarity constraints (MPCC). As a special case of MPEC, MPCC has more plausible properties which make it easier to solve. In particular, a variety of methods can be applied to convert an MPCC to a single level NLP and then solve the NLP using existing solution techniques. Therefore, MPCC has been extensively studied recently (*15*-*21*). In particular, Ferris et al. (*22*-*23*) implemented a solver of nonlinear program with equilibrium constraints (NLPEC), as a sub-system of GAMS (general algebraic modeling systems, *24*). The MPCC model proposed in Ban et al. (*14*) was solved by

NLPEC directly. That is, the MPCC model was first converted to an equivalent single level NLP by applying the complementarity slackness condition. To solve the single level NLP, the strict complementarity condition is relaxed by a relaxation parameter. Then this parameter is progressively reduced, with the resulting relaxed NLP solved using existing NLP solvers. Ralph and Wright (*25*) proposed certain conditions under which the relaxation scheme can guarantee to solve the original MPCC successfully. Ban et al. (*14*) has demonstrated using well-known CNDP test problems that such a direct solution approach can generate promising results compared with existing solution techniques for CNDP.

Nevertheless, due to the fact that the lower level AUE has to be defined on the so-called disaggregated variables (i.e., link flows with respect to different destinations), the direct conversion (and solve) may bring the dimensionality problem (*11*) for large-scale CNDP. That is, the resulting single level NLP might have a large number of defining variables, especially for multiple origin and multiple destination (many-to-many) problems. In this paper, we observe that the lower-level AUE problem has a very special structure such that it can be easily separated according to individual destinations. Furthermore, the reformulation method we applied can still maintain this special feature. Therefore, by exploring such a special structure of the MPCC model for CNDP, we propose a decomposition scheme to resolve the dimensionality problem. Numerical examples in the paper show that the presented scheme can efficiently solve CNDP without losing too much of the quality of the solutions.

This paper is organized as follows. The MPCC based CNDP model is introduced in Section 2. In Section 3, the solution approach for the proposed MPCC model is discussed, especially the decomposition scheme for solving large-scale problems. Section 4 provides numerical examples demonstrating the effectiveness of the proposed decomposition scheme. Finally, concluding remarks and future research directions are given in section 5.

2. MPCC MODEL FOR CNDP

2.1 Link-Node NCP Formulation for Asymmetric User Equilibria

As shown by Ban et al. (*14*), AUE can be mathematically formulated as follows:

$$\begin{cases} 0 \le [\pi_j^s + t_{ij}(\sum_{s \in S} v_{ij}^s) - \pi_i^s] \perp v_{ij}^s \ge 0, \forall (i,j) \in A, s \in S \\ 0 \le [\sum_{(i,j) \in A} v_{ij}^s - \sum_{(j,i) \in A} v_{ji}^s - d_i^s] \perp \pi_i^s \ge 0, \forall i \in N, i \ne s, s \in S \end{cases}. \tag{1}$$

Here we denote a given transportation network as $G(N, A)$, where N is the set of nodes and A is the set of links. We use index i, j to denote nodes in N and (i,j) or ij to denote a link in A. Denote R as the origin node set which is a subset of N and generates origin-destination (OD) trips. Similarly, set S is defined as the destination set which is also a subset of N and absorbs OD trips. Further denotes π_i^s the minimum travel cost from node i to destination s, d_i^s the travel demand from i to s, v_{ij}^s the (disaggregated) flow for link (i, j) with respect to destination s, and t_{ij} the link travel cost for link (i, j). Symbol "$\perp$" is the "perp" operator such that $x \perp y \Leftrightarrow x^T y = 0$. Equation (1) can be rewritten in a matrix form as:

$$\begin{cases} 0 \le [-\Lambda_s^T \pi^s + t(\sum_{s \in S} v^s)] \perp v^s \ge 0 \\ 0 \le [\Lambda_s v^s - d^s] \perp \pi^s \ge 0 \end{cases}, \forall s \in S. \tag{2}$$

Here we define vectors $\pi^s = (\pi_i^s)_{i \in N, i \ne s}, v^s = (v_{ij}^s)_{(i,j) \in A}, d^s = (d_i^s)_{i \in N, i \ne s}$ for any given destination node $s \in S$, and $t = (t_{ij})_{(i,j) \in A}$. Also the standard node-link incidence matrix is represented as Λ and Λ_s denotes Λ with the row corresponding to destination *s* removed which guarantees that Λ_s is full row rank. Equation (2) is the link-node NCP formulation for AUE which will be utilized later for modeling the bilevel CNDP. We can easily observe that (2) has a very special structure such that it can be naturally decomposed according to individual destinations. The only place in which interactions exist for variables related to different destinations is the link travel cost vector t since *t* is defined on the aggregated link flows. This special structure has important impacts on how to design a solution algorithm for both the UE problem itself (*26*) and the CNDP problem constructed based upon (2), as will be discussed in more detail later.

2.2 MPCC Formulation for CNDP

Firstly, additional notation is listed as follows.

v_{ij} = the total (aggregated) link flow on link *(i,j)*, $v_{ij} = \sum_{s \in S} v_{ij}^s$

v = the vector of v_{ij}, $v \in R^{|A|}$

y_{ij} = the capacity enhancement for link $(i, j) \in A$

y = the vector of y_{ij}, $y \in R^{|A|}$

$t_{ij}(v, y_{ij})$ = the travel cost on link $(i, j) \in A$, defined as a function of the aggregated link flow v and the capacity enhancement of (i, j), i.e., y_{ij}

$g_{ij}(y_{ij})$ = the cost function of capacity enhancement for link $(i, j) \in A$

g = the vector of $g_{ij}(y_{ij})$, $g \in R^{|A|}$

θ = the relative weight of total capacity enhancement cost and total travel cost in the system design objective function.

l_{ij}, u_{ij} = the lower bound and upper bound for the capacity enhancement for link $(i, j) \in A$

l, u = the vector of l_{ij} and u_{ij}, respectively, $l, u \in R^{|A|}$

With this notation in place, as shown in Ban et al. (*14*), we can formulate CNDP with AUE as the following MPCC:

$$\min_{y,v^1,\cdots v^s,v^{|S|},\pi^1,\cdots\pi^s,\pi^{|S|}} \sum_{(i,j)\in A}[t_{ij}(\sum_{s\in S} v^s, y_{ij})v_{ij}]+\theta\sum_{(i,j)\in A} g_{ij}(y_{ij}), \tag{3a}$$

subject to

$$l_{ij} \le y_{ij} \le u_{ij},\ \forall (i,j)\in A, \tag{3b}$$

where $(v^s,\pi^s),\forall s\in S$ is the solution to the following NCP problem (AUE):

$$\begin{aligned} &0\le[\pi_j^s + t_{ij}(\sum_{s\in S} v^s, y_{ij})-\pi_i^s]\perp v_{ij}^s \ge 0, \forall (i,j)\in A, s\in S,\\ &0\le[\sum_{(i,j)\in A} v_{ij}^s - \sum_{(j,i)\in A} v_{ji}^s - d_i^s]\perp \pi_i^s \ge 0, \forall i\in N, i\ne s, s\in S. \end{aligned} \tag{3c}$$

Obviously, the MPCC based CNDP model (3) is defined on the upper level decision variable $y_{ij},\forall(i,j)\in A$ and the disaggregated link flow $(v^s,\pi^s),\forall s\in S$. Equation (3a) is the upper level objective of the MPCC model which tries to minimize a weighted summation of the total system travel cost and the enhancement cost; (3b) is the bound constraint for the upper level decision variable $y_{ij},\forall(i,j)\in A$; and (3c) is the lower level AUE formulation that $(v^s,\pi^s),\forall s\in S$ must satisfy. Using matrix notation, (3) can be rewritten as:

$$\min_{y,v^1,\cdots v^s,v^{|S|},\pi^1,\cdots\pi^s,\pi^{|S|}} [t(\sum_{s\in S} v^s, y)]^T v+\theta e^T g(y), \tag{4a}$$

subject to

$$l\le y\le u, \tag{4b}$$

where e is the vector of all 1's and $\{(v^s,\pi^s),\forall s\in S\}$ is the solution to the following NCP model:

$$\begin{cases} 0\le[-\Lambda_s^T\pi^s + t(\sum_{s\in S} v^s, y)]\perp v^s \ge 0 \\ 0\le[\Lambda_s v^s - d^s]\perp \pi^s \ge 0 \end{cases}, \forall s\in S\,. \tag{4c}$$

The MPCC model (4) can be tackled by being converted to a single level equivalence and then solved using a decomposition scheme which will be discussed in next section.

3. SOLUTION ALGORITHM

3.1 Single Level NLP Formulation for the MPCC Model

Because the NCP formulation (4c) can be readily replaced by its equivalent complementarity slackness condition and additional nonnegativity constraints, the MPCC model of CNDP (4) can be straightforwardly converted into a single level NLP as follows.

$$\min_{y,v^1,\cdots v^s,v^{|S|},\pi^1,\cdots\pi^s,\pi^{|S|}} \quad [t(\sum_{s\in S} v^s, y)]^T v + \theta e^T g(y), \tag{5a}$$

subject to

$$l \le y \le u\,, \tag{5b}$$

$$-\Lambda_s^T \pi^s + t(v,y) \ge 0, \forall s \in S\,, \tag{5c}$$

$$\Lambda_s v^s - d^s \ge 0, \forall s \in S\,, \tag{5d}$$

$$v^s \ge 0, \forall s \in S\,, \tag{5e}$$

$$\pi^s \ge 0, \forall s \in S \quad , \tag{5f}$$

$$(\Lambda_s v^s - d^s)_i \pi_i^s = 0, \forall s \in S, i \in N, i \ne s\,, \tag{5g}$$

$$[-\Lambda_s^T \pi^s + t(v,y)]_{ij} v_{ij}^s = 0, \forall s \in S, (i,j) \in A\,. \tag{5h}$$

Here the lower level NCP formulation in (4c) is replaced by its equivalent complementarity slackness condition in (5c) – (5h). Evidently, under the assumption that both the link travel cost function *t* and the function *g* are smooth, the single level NLP model (5) involves only smooth functions with respect to $(y, v, v^1, \cdots v^s, v^{|S|}, \pi^1, \cdots \pi^s, \pi^{|S|})$. Hence, it is a smooth and nonlinear optimization problem. However, this NLP formulation lacks sound mathematical properties because of the complementarity slackness constraints (5g) and (5h). Actually, because of these two constraints, the single level model is non-convex and most importantly, the Mangasarian Fromovitz Constraint Qualification (MFCQ) does not hold (*10*). Therefore, solving the single level NLP model directly is usually difficult and a progressive relaxation algorithm will be normally adopted instead.

Ban et al. (*14*) solved the single level NLP (5) directly by applying a relaxation scheme, particularly by the NLPEC solver developed by Ferris et al. (*22*). Promising results have been reported in their study for well-known CNDP test problems in the literature. However, note that the NLP equivalence (5) involves the disaggregated variables explicitly and hence has a large dimension for large scale problems. Such a dimensionality problem may very likely prohibit the application of the direct solve. Nevertheless, it is clear that all the constraints (5c) – (5h) are defined according to individual destinations, except for the interaction of disaggregated link flows on the link travel cost function *t*. This feature makes it possible to employ certain decomposition technique in order to solve the single NLP model more efficiently.

3.2 Decomposition Scheme for Solving Single Level NLP

For the single level NLP (5), constraint (5b) is defined on the upper level decision variable *y* only and those from (5c) – (5h) are defined according to each individual destination, except for the interaction of the disaggregated link flow variables $v^s, \forall s \in S$ and y in the link travel cost function t. Based on this observation, it is intuitive to apply certain decomposition scheme for solving the single level model (5). In the literature, decomposition schemes can be grouped into two categories: the Gauss-Seidel (or sequential) decomposition and the Jacobi (or parallel) decomposition (*27-28*). While the Jacobi decomposition method is amenable to parallel computing, the Gauss-Seidel (GS) approach has been proved to have better convergence performance since it can incorporate the newest available information (*29*). In this paper, we will concentrate on the GS method.

By applying GS, we can temporarily fix the interaction of $v^s, \forall s \in S$ and y in t. Then the (possible) large size single level NLP model can be converted into multiple, yet smaller dimensional, optimization problems. In our case, these smaller dimensional problems will be defined on y and individual $(v^s, \pi^s), \forall s \in S$, respectively. That is, the single level NLP (5) can be decomposed into the following $|S|+1$ smaller dimension NLPs.

$$\begin{aligned} &\min_{y} \ [t(\sum_{s\in S}\bar{v}^s, y)]^T(\sum_{s\in S}\bar{v}^s) + \theta e^T g(y), \\ &st. \, l \le y \le u, \end{aligned} \tag{7}$$

and

$$\left.\begin{aligned} &\min_{v^s,\pi^s} \ [t(\sum_{s'\in S, s'\neq s}\bar{v}^{s'} + v^s, \bar{y})]^T v^s \\ &st. \\ &-\Lambda_s^T\pi^s + t(\sum_{s'\in S, s'\neq s}\bar{v}^{s'} + v^s, \bar{y}) \ge 0, \\ &\Lambda_s v^s - d^s \ge 0, \\ &[-\Lambda_s^T\pi^s + t(\sum_{s'\in S, s'\neq s}\bar{v}^{s'} + v^s, \bar{y})]^T v^s = 0, \\ &(\Lambda_s v^s - d^s)^T \pi^s = 0, \\ &v^s \ge 0, \pi^s \ge 0, \end{aligned}\right. \quad \forall s \in S. \tag{8}$$

Essentially, (7) is defined on the upper level decision variable only with $v^s, \forall s \in S$ fixed as $\bar{v}^s$. While for each destination s, after temporarily fixing $v^{s'}, \forall s' \in S, s' \neq s$ as $\bar{v}^{s'}$ and y as $\bar{y}$, an NLP can be obtained, as shown in (8). We can immediately observe that the constraints in (8) actually define an NCP problem for each individual destination, i.e.,

$$\begin{cases} 0 \le [-\Lambda_s^T\pi^s + t(\sum_{s'\in S, s'\neq s}\bar{v}^{s'} + v^s, \bar{y})] \perp v^s \ge 0, \\ 0 \le [\Lambda_s v^s - d^s] \perp \pi^s \ge 0, \end{cases} \quad \forall s \in S, \tag{9}$$

Under certain monotonicity condition (*26*), (9) has a unique solution in terms of v^s. This means that solving the minimization problem (8) is equivalent to solve the NCP problem (9) for each destination $s \in S$. Because of the smaller dimension of both (7) and (9) compared with the original single level NLP (5), they can be solved much more efficiently. Note that solving these smaller dimension problems only tackles the decomposed version of the original single level NLP (5). The overall solution method is thus an iterative one with the decomposed problems solved at each iteration. Under the assumption that the obtained solution from the decomposed problems defines a descent direction to the original single level problem (although, verifying this is not trivial), the optimal step size for computing the next iterate can be obtained by a line search as follows.

$$\begin{aligned}&\min_{\alpha}[t(\sum_{s\in S}(\bar{v}^s+\alpha(\hat{v}^s-\bar{v}^s)),\bar{y}+\alpha(\hat{y}-\bar{y}))]^T[\sum_{s\in S}(\bar{v}^s+\alpha(\hat{v}^s-\bar{v}^s))]+\theta e^T g(\bar{y}+\alpha(\hat{y}-\bar{y})), \quad (10)\\ &\text{st. } 0\le\alpha\le 1\end{aligned}$$

where $\bar{v}^s$, $\bar{y}$ denotes current fixed variables, $\hat{v}^s$, $\hat{y}$ the solution obtained from (7) and (9), respectively, and α the step size.

To sum up, the iterative algorithm for solving (5) can be listed as follows.

Step 1 Initialization.

Assign initial values for the defining variables, v^{s0}, π^{s0}, y^{s0}, and set iteration count *n=0*.

Step 2 Gauss-Seidel Decomposition Scheme.

Step 2.1 Solve the decomposed problem (7) by setting $\bar{v}^s = v^{sn}, \forall s \in S$. Denote the obtained solution as $\hat{y}^n$.

Step 2.2 For each destination $s \in S$, solve the decomposed NCP (9) by setting $\bar{y} = y^{sn}$ and $\bar{v}^{s'} = v^{s'n}, \forall s' \in S, s' \neq s$. Denote the obtained solution as $\hat{v}^{sn}$, $\forall s \in S$.

Step 3 Line Search.

Solve the one-dimensional NLP (10) to obtain the optimal step size, denoted as α^*.

Step 4 Converge Test.

If certain convergence criterion is met, stop. Otherwise, set $v^{s(n+1)} = v^{sn} + \alpha^*(\hat{v}^{sn} - v^{sn})$, $y^{s(n+1)} = y^{sn} + \alpha^*(\hat{y}^{sn} - y^{sn})$, *n=n+1*, and go to Step 2.1.

Note that in Step 4 of the above algorithm, the stopping criterion has to consider both the upper level objective value and the lower level UE condition. For the upper level, we can check if the objective values remain stable over the past several iterations. While for the lower level UE condition, we apply a criterion similar to the relative gap in Boyce et al. (*30*):

$$\text{relGap} = \sum_{(i,j)\in A} t_{ij}(v_{ij}, y_{ij}) \cdot (v_{ij} - v_{ij}^{AON}) \Big/ \sum_{(i,j)\in A} t_{ij}(v_{ij}, y_{ij}) \cdot v_{ij}. \quad (11)$$

In (11), v_{ij}^{AON} denotes the all-or-nothing link flow provided the link travel time is fixed at link flow v_{ij} and capacity enhancement y_{ij}. Clearly, $\text{relGap} \ge 0$ and if $\text{relGap} = 0$, the lower level UE condition will hold exactly.

4. NUMERICAL EXAMPLES

In this section, we test the proposed decomposition scheme on a well-known and relatively large scale CNDP with both symmetric and asymmetric user equilibria. In particular, we will compare our solutions with those obtained using existing CNDP algorithms. Before introducing the actual network, however, we first provide the link travel time functions for both symmetric and asymmetric cases.

4.1 Link Travel Time Function

For the symmetric user equilibrium, the link travel time function is separable, implying the travel time on a particular link is only dependent on its own traffic flow. In the transportation field, the most popularly used function form for the symmetric case is the BPR (Bureau of Public Roads) function:

$$t_{ij}(v_{ij}, y_{ij}) = A_{ij} + B_{ij}(v_{ij}/(K_{ij} + y_{ij}))^4, \qquad (12)$$

where A_{ij} , B_{ij} , and K_{ij} are constants for any link *(i,j)* .

On the other hand, for the asymmetric case, no appropriate function form has been suggested in the literature. In this study, we simply adopt the following link travel time function:

$$t_{ij}(v_{ij}, y_{ij}) = A_{ij} + B_{ij}[(\sum_{(k,j)\in A} \rho_{ij,kj} v_{kj})/(K_{ij} + y_{ij})]^4, \qquad (13)$$

where $0 \le \rho_{ij,kj} \le 1$ denotes the "impact factor" of the flow on link (k, j) to the travel cost of link (i, j) . Apparently, we have $\rho_{ij,ij} = 1, \forall(i, j) \in A$ and further if $\rho_{ij,kj} = 0,$ $\forall j \in N, (i, j) \in A, (k, j) \in A, i \ne k$, equation (13) will reduce to the standard BPR function in (12). We should point out here that equation (13) is just an intuitive way to achieve (asymmetric) link interactions among adjacent links to demonstrate our proposed model and algorithm for CNDP with AUE. How to design a practically reasonable asymmetric link cost function for a given network is beyond the scope of this paper.

4.2 Test Networks

The Test network in this study is the Sioux-Falls network which was firstly constructed and studied by LeBlance (*31*). It contains 24 nodes and 76 links, as shown in FIGURE 1. All 24 nodes can be either origin or destination node, or both. The data of the network can be found in Suwansirikul et al. (*7*), including the parameters A_{ij} , B_{ij} , and K_{ij} in the link travel time function (12) or (13). In particular, only ten links are selected for capacity enhancement, namely, link 16, 17, 19, 20, 25, 26, 29, 39, 48 and 74 in FIGURE 1. The upper bound of the enhancement for each link is set to 25, i.e., $0 \le y_{ij} \le 25, \forall(i, j) \in A$. Furthermore, the cost function for capacity enhancement for each link is $g_{ij}(y_{ij}) = 0.001y_{ij}^2, \forall(i,j) \in A$. In other words, the upper level objection function of CNDP is

$$z(v,y) = \sum_{(i,j)\in A}[t_{ij}(\sum_{s\in S}v^s, y_{ij})v_{ij} + 0.001\theta_{ij}y_{ij}^2]\cdot \tag{14}$$

Finally, the impact factors in equation (13) for the asymmetric case are listed in TABLE 1. Note that TABLE 1 only shows the impact factors for $\rho_{ij,kj}, \forall j \in N, (i,j) \in A, (k,j) \in A, i \neq k$ and those $\rho_{ij,ij} = 1, \forall(i,j) \in A$ are not listed explicitly in the table.

4.3 Result Analysis

To ensure the convergence, we set the stopping criterion as $\text{relGap} \leq 1.0e-6$ and the fluctuation of the objective values for the last five iterations less than 1.0e-5. First, the solution for the symmetric case is shown in TABLE 2. This table also lists the solutions obtained by existing solution techniques. In particular, the simulated annealing (SA) method (*6*) obtained the best solution found so far. From this table, it is obvious that our decomposition scheme can generate a solution whose objective value is very close to that by SA. Furthermore, only 36 UE solves were performed in our scheme which is significantly less than most of other algorithms. This demonstrates that our proposed method can be much more time efficient than other approaches. Note that by applying the direct solution method in Ban et al. (*14*), we can also solve the symmetric problem with an even lower objective value (80.5157); however, the direct solution method tends to be more time consuming and can not be applied to large scale CNDPs due to the so-called dimensionality problem. So the direct solution method is not listed and compared in this paper.

FIGURE 2 further illustrates the convergence of the objective value. For each iteration of the algorithm, we mainly solve one run of UE. We can see from this figure the objective value converges very quickly. However, due to the non-convexity of the problem, the objective value may not decrease monotonically. Normally, the objective value decreases dramatically at the first several iterations. It then starts to fluctuate in a relatively small range. Finally, it becomes stable rapidly. This is more evident in FIGURE 4 for the asymmetric case. Meanwhile, the convergence of the lower level UE is depicted in FIGURE 3. Apparently, the UE condition also converges very fast. Combining FIGURE 2 and FIGURE3, we can conclude that if only an approximate solution is required, the proposed decomposition scheme can solve CNDP very efficiently by only several iterations. FIGURE 3 also shows that the relative gap is always nonnegative which has been discussed previously.

For the asymmetric case, since there is no solution reported in the literature (except the direct solution method which produces an objective value of 83.9095), TABLE 3 only shows the one we obtained using the proposed decomposition scheme. This table also demonstrates that solving the asymmetric case requires more iterations and the objective value is larger than that of the symmetric case. FIGURE 4 and FIGURE 5 further illustrate the convergence of the objective value and the lower level UE, respectively. We can easily observe from FIGURE 4 that the objective value becomes stable after only the first several iterations. This is identical to the symmetric case in FIGURE 2, implying that an approximate solution can be also efficiently obtained for the asymmetric case.

5. CONCLUSIONS

We proposed in this paper a decomposition scheme for solving the MPCC model for continuous network design problem with asymmetric user equilibrium. Instead of directly solving the equivalent single level NLP of the MPCC model, we applied the Guass-Seidel decomposition scheme to this single level problem by exploring its special structure. Then, the possible large size single level NLP can be converted into multiple yet smaller dimensional problems which can be tackled more easily. Numerical examples conducted in the paper showed that the presented decomposition scheme can generate promising results. Especially, it can solve efficiently a well-known CNDP test problem with the objective value very close to the best-known one in the literature.

For future studies, we will investigation conditions under which the proposed decomposition scheme can guarantee to generate an optimal solution for CNDP. Further, the decomposition scheme was only tested on Sioux-Falls network in this paper. Extensive testing of the proposed model and algorithm for solving larger scale CNDPs will be conducted in future research.

References

1. LeBlanc, L. and D.E. Boyce. A Bilevel Programming Algorithm for Exact Solution of the Network Design Problem with User-optimal Flows. *Transportation Research, Part B*, Vol. 20, pp. 259-265.
2. Morlok, E., J. Schofer, W. Pierskalla, and et al. *Development and Application of a Highway Network Design Model*. Publication FHWA Contract Number DOT-PH-11. FHWA, U.S. Department of Transportation, 1973.
3. Tan, H.-N., S.B. Gershwin, and M. Athana. *Hybrid optimization in urban traffic networks*. Report No. DOT-TSC-RSPA-79-7, Laboratory for Information and Decision System, MIT, Cambridge, MA, 1979.
4. Marcotte, P. Network Optimization with Continuous Control Parameters. *Transportation Science*, Vol. 17, 1983, pp. 181-197
5. Suwansirikul, C., T.L. Friesz, and R.L. Tobin. Equilibrium Decomposed Optimization: A Heuristic for the Continuous Equilibrium Network Design Problem. *Transportation Science,* Vol. 21, 1987, pp. 254-263.
6. Friesz, T.L., H. J. Cho, N.J. Mehta, and et al. A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints. *Transportation Science* Vol. 26, 1992, pp. 18-26.
7. Yang, H. Sensitivity Analysis for the Elastic Demand Network Equilibrium Problem with Applications. *Transportation Research, Part B,* Vol. 31, 1997, pp. 55-70.
8. Yang, H. and M.G.H. Bell. Transport Bilevel Programming Problems: Recent Mmethodological Advances. *Transportation Research, Part B,* Vol. 35, 2001, pp. 1-4.
9. Outrata, J., M. Kocvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Kluwer Academic Publisher, 1998.
10. Luo, Z.-Q., J.S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, UK, 1996.

11. Meng, Q., H. Yang, and M.G.H. Bell. An Equivalent Continuously Differentiable Model and a Locally Convergent Algorithm for the Continuous Network Design Problem. *Transportation Research, Part B*, Vol. 35, 2001, pp. 83-105.
12. Marcotte, P. and D.L. Zhu. Exact and Inexact Penalty Methods for the Generalized Bilevel Programming Problem. *Mathematical Programming*, Vol. 74, 1996, pp. 141-157.
13. Patriksson, M. and R.T. Rockafellar. A Mathematical Model and Descent Algorithm for Bilevel Traffic Management. *Transportation Science*, Vol. 36, 2002, pp. 271-291.
14. Ban, X.G, X.H. Liu, and M.C. Ferris. A General MPCC Model and Its Solution Algorithm for Continuous Network Design Problem. *Mathematical and Computer Modeling,* Vol. 43, No. 5-6, 2006, pp. 493-505.
15. Scheel, H. and S. Scholte. Mathematical Programs with Complementarity Constraints: Stationarity, Optimality, and Sensitivity. *Mathematics of Operations Research*, Vol. 22, No. 1, 2000, pp. 1-22.
16. Anitescu, M. On solving Mathematical Programs with Complementarity Constraints as Nonlinear Programs. Preprint ANL/MCS-P864-1200, MCS Division, Argonne National Laboratory, Argonne, IL, 2000.
17. Anitescu, M.. Global Convergence of An Elastic Mode Approach for A Class of Mathematical Programs with Complementarity Constraints. Preprint ANL/MCS-P1143-0404, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA, 2004.
18. Fletcher, R. and S. Leyffer. Solving Mathematical Program with Complementarity Constraints as Nonlinear Programs. *Optimization Methods and Software,* Vol. 19, No. 1, 2004, pp. 15-40.
19. Hu, X. and D. Ralph. A Note on Sensitivity of Value Functions of Mathematical Programs with Complementarity Constraints. *Mathematical Programming*, Vol. 93, No. 2, 2002, pp. 265-279.
20. Jiang, H. and D. Ralph. Smooth SQP Methods for Mathematical Programs with Nonlinear Complementarity Constraints. *SIAM Journal of Optimization*, Vol. 10, No. 3, 2000, pp. 779-808.
21. Jiang, H. and D. Ralph. Extension of Quasi-Newton Methods to Mathematical Programs with Complementarity Constraints. *Computational Optimization and Applications*, Vol. 25, No. 1-3, 2003, pp. 123-150.
22. Ferris, M.C., S.P. Dirkse, and A. Meeraus. *Mathematical programs with equilibrium constraints: automatic reformulation and solution via constrained optimization*. Numerical Analysis Group Research Report NA-02/11, Oxford University Computing Laboratory, Oxford University, USA, 2002.
23. Ferris, M.C., S.P. Dirkse, and A. Meeraus. Mathematical Programs with Equilibrium Constraints: Automatic Reformulation and Solution via Constrained Optimization. In T.~J. Kehoe, T.~N. Srinivasan, and J.~Whalley, editors, *Frontiers in Applied General Equilibrium Modeling*. Cambridge University Press, Cambridge, UK., 2004.
24. Brooke, A., D. Kendrick, A Meeraus, and et al. *GAMS, a user's guide*. GAMS Development Corporation, 1998.
25. Ralph, D, and S.J. Wright. Some Properties of Regularization and Penalization Schemes for MPECs. *Optimization Methods and Software*, Vol. 19, 2004, pp. 527-556.
26. Ban, X.G. *Quasi-Variational Inequality Formulations and Solution Approaches for Dynamic User Equilibria*, Ph.D. Thesis, University of Wisconsin-Madison, 2005.

27. Ortega, J.M. and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
28. Zadeh, N. A Note on the Cyclic Coordinate Ascent Method, *Management Science*, Vol. 16, 1969/1970, pp. 642-644.
29. Bertsekas, D.P. J.N. and Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, London, 1997.
30. Boyce, D.E., B. Ralevic-Dekic, and H. Bar-Gera. Convergence of Traffic Assignment: How Much Is Enough? *ASCE Journal of Transportation Engineering*, Vol. 130, No. 1, 2004, pp. 49-55.
31. LeBlanc, L. An Algorithm for the Discrete Network Design Problem, *Transportation Science*, Vol. 9, 1975, pp. 183-199.

List of Tables

List of Figures

TABLE 1 Impact factors for the tesing network

Link	6.8	7.8	8.6	8.7	10.9	10.16	13.24	16.10
4 (2.6)		.1	.05					
12 (5.6)			.05					
13 (5.9)					.06			
16 (6.8)		0.06						
17 (7.8)	0.08							
22 (8.16)					0.1			
24 (9.8)						0.1		
25 (9.10)	0.06	0.08						
25 (10.9)								0.08
54 (18.7)						0.05		
55(18.16)				0.06				
66 (21.24)						0.07		
73 (23.24)							0.05	

TABLE 2 Comparisons of results for the symmetric case

Variable	**MINOS** (Murtagh and Sanders 1981)	**HJ** Abdulaal and LeBlance (1979)	**EDO** Suwansirikul et al. 1987	**SA** Friesz et al. 1992	**AJ** Meng et al. 2001	**Decomposition Scheme** Proposed in this paper
Initial Value of y_{ij}	2	1	12.5	6.25	12.5	0
$y_{6.8}$ (Link 16)	4.8	3.8	4.59	5.38	5.5728	5.269770
$y_{7.8}$ (Link 17)	1.2	3.6	1.52	2.26	1.6443	1.378772
$y_{8.6}$ (Link 19)	4.8	3.8	5.45	5.5	5.6228	5.269853
$y_{8.7}$ (Link 20)	0.8	2.4	2.33	2.01	1.6443	1.378635
$y_{9.10}$ (Link 25)	2	2.8	1.27	2.64	3.1437	2.766501
$y_{10.9}$ (Link 26)	2.6	1.4	2.33	2.47	3.2837	2.766446
$y_{10.16}$ (Link 29)	4.8	3.2	0.41	4.54	7.6519	4.669070
$y_{13.24}$ (Link 39)	4.4	4	4.59	4.45	3.8035	4.350875
$y_{16.10}$ (Link 48)	4.8	4	2.71	4.21	7.382	4.668969
$y_{24.13}$ (Link 74)	4.4	4	2.71	4.67	3.6935	4.350856
Value of Objective Function	81.25	81.77	83.47	80.87	81.752	**81.102**
Number of Solved UE	58	108	12	3900	2700	36

TABLE 3 Solution for the asymmetric case

Initial y	y Link 16	y Link 17	y Link 19	y Link 20	y Link 25	y Link 26	y Link 29	y Link 39	y Link 48	y Link 74	Obj Value	# UE
12.5	4.96983	1.63806	5.27757	1.43692	3.12381	2.87238	5.25048	4.70207	4.92401	4.12252	**84.475**	56

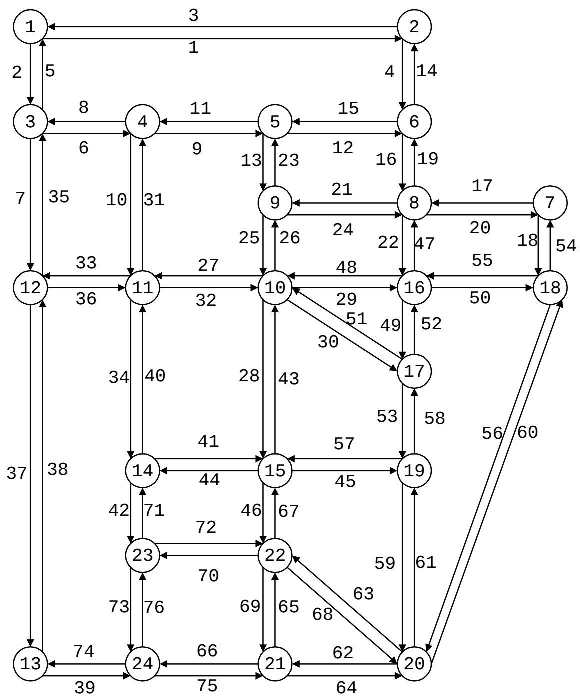


FIGURE 1 The test network (Sioux-Falls)

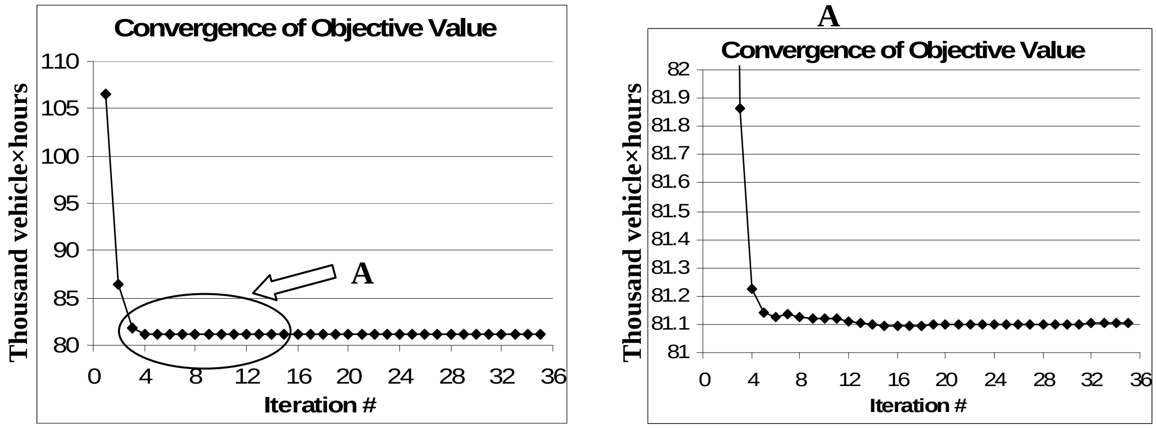


FIGURE 2 Convergence of the objective value for the symmetric case

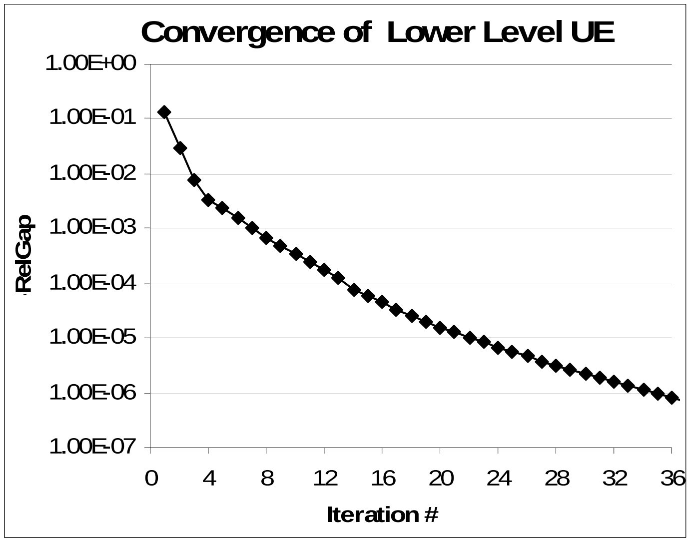
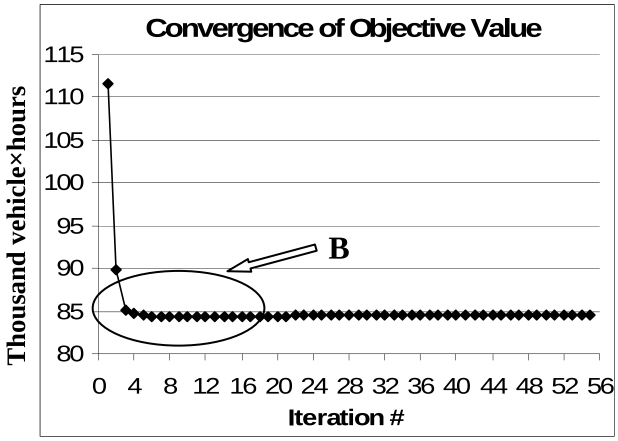


FIGURE 3 Convergence of the lower level UE for the symmetric case



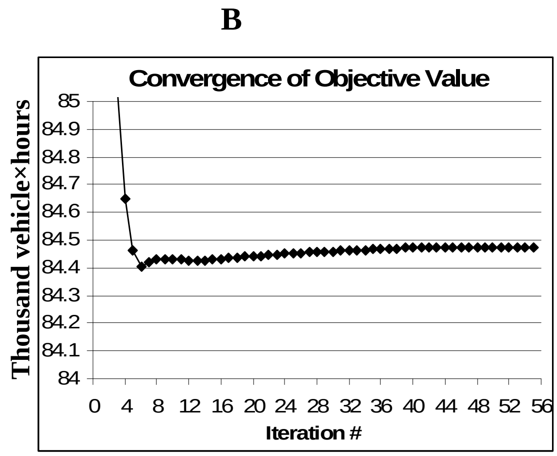


FIGURE 4 Convergence of the objective value for the asymmetric case

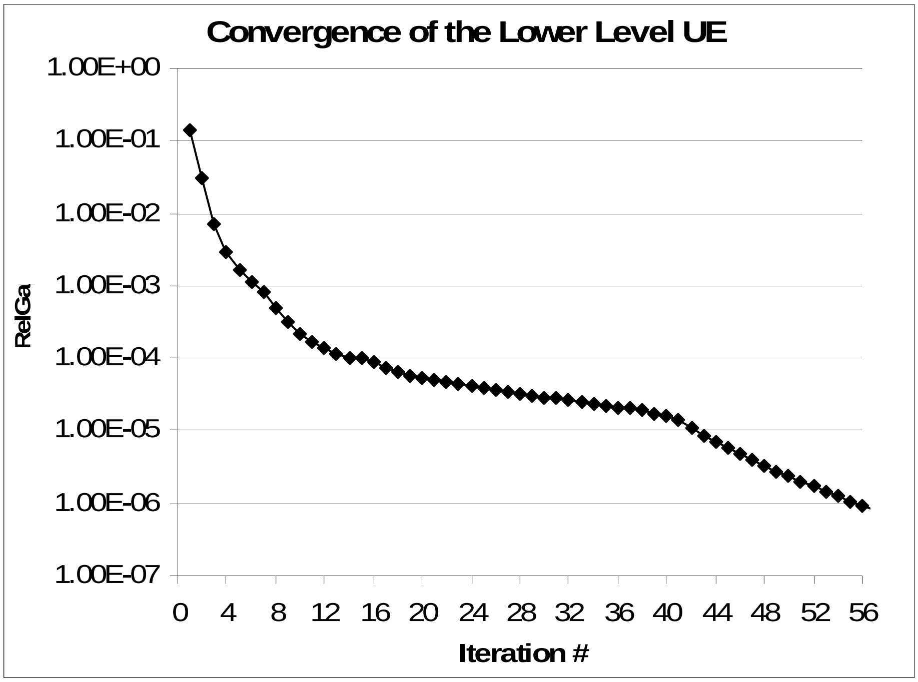


FIGURE 5 Convergence of the lower level UE for the asymmetric case