The Multicommodity Maximal Covering Network Design Problem for Planning Critical Routes for Earthquake Response

Kannan Viswanath\(^1\) and Srinivas Peeta\(^2\)

\(^1\)Kannan Viswanath  
School of Civil Engineering  
Purdue University  
West Lafayette, IN 47907  
Phone: (765) 494-2206  
Fax: (765) 496-1105  
E-mail: viswanak@purdue.edu

\(^2\)Srinivas Peeta  
School of Civil Engineering  
Purdue University  
West Lafayette, IN 47907  
Phone: (765) 494-2209  
Fax: (765) 496-1105  
E-mail: peeta@ecn.purdue.edu
ABSTRACT

This paper formulates a multicommodity maximal covering network design problem (MCNDP) for identifying critical routes for earthquake response and seismically retrofitting bridges. The MCNDP seeks routes that minimize the total travel time over the selected routes and maximizes the total population covered, subject to a budget constraint on bridge retrofitting costs on the selected routes. The problem is formulated as a two-objective integer programming model and solved using the branch-and-cut module in the CPLEX optimizer. The model performance is analyzed using the transportation network of a seismically-prone region in southwest Indiana. A problem reduction strategy is introduced to reduce computational times by recognizing that the critical routes are not usually circuitous. Thereby, the search for the critical routes for an origin-destination (O-D) pair is confined to a limited geographical region around it. To further reduce computational costs, the formulation is extended to incorporate valid inequalities that exploit the problem structure. Computational experiments are conducted to investigate the effects of varying the budget, and the relative weights of the two objectives. Noninferior frontiers that illustrate the trade-offs between the conflicting objectives for different budgets are constructed to provide practical insights to decision-makers. In addition, a vulnerability analysis is performed for the various solution instances to infer on their ability to ensure connectivity between all O-D pairs following an earthquake.

Key Words: earthquake response, network design, discrete optimization, noninferior frontier
INTRODUCTION

Network-level disaster management planning is vital for effectively responding to natural calamities and security problems. For example, the availability of the transportation network is critical for efficient emergency response (1) under earthquakes. It entails the identification of critical routes in a planning context that remain functional following an earthquake, to enable the response operators to access as much population as possible in a minimum amount of time. This implies two objectives for the selection of the critical routes: (i) minimizing the total travel time on the routes between the origin-destination (O-D) pairs, and (ii) maximizing the total population that can be covered by these routes. The functionality or survivability of a route is governed significantly by its weakest elements, the bridges. The seismic retrofitting (2) of bridges can enhance the survivability of routes under earthquakes. However, due to the significant cost and effort involved in retrofitting, it is impractical to retrofit every bridge in an earthquake-prone region, especially with limited budgets. Hence, the budget serves as a constraint in the determination of the critical routes. Therefore, the critical routes problem can be viewed as a budget-constrained covering network design problem which seeks low cost routes that cover the maximum population while satisfying budget constraints introduced by the need to retrofit bridges on the critical routes.

Network design problems arise primarily in the capital investment phase of engineering problems such as transportation planning, distribution network design, computer networking and communication systems. The basic problem is to determine a network configuration that minimizes the sum of the fixed costs of the links chosen and the cost of routing the different commodities through the network defined by these links. Due to its usefulness in the aforementioned applications, the fixed cost network design problem has been well-studied. Magnanti and Wong (3) provide a comprehensive survey of integer programming-based approaches to address network design problems. They develop a generic discrete choice network design formulation that unifies the different types of network design models. Well-known problems in combinatorial optimization, including the shortest path, minimum spanning tree, Steiner tree and traveling salesman problems have been shown to be special cases of the generic network design model. However, these models have focused primarily on cost minimization and do not consider other important criteria like population coverage, environmental impacts and revenues.

Current et al. (4) introduce the notion of population coverage to the network design problem by formulating the maximum covering shortest path problem that minimizes the path cost for a single O-D pair and maximizes the total demand covered by that path. They associate some fraction of the population with each node, which is labeled as the demand of that node. They define a demand node as covered if the route includes the node or passes through another node that is within a prespecified distance from that node. They also formulate a variation of this problem (5) for hazmat routing applications, called the minimum covering shortest path problem, in which the demand covered is minimized. Hutson and Revelle (6) extend the concept of coverage to tree networks. A demand node is directly covered if there is a link in the tree that is incident upon it. Indirect coverage is assumed if the demand node is within a prescribed distance from a link in the tree. The maximal direct covering tree problem seeks to minimize the cost of the subtree and maximize the total demand covered by it. They suggest the use of these models for problems involving road network construction in sparsely populated areas under resource constraints.
Current and Schilling (7) adapt the coverage concept to the traveling salesman problem by formulating the covering salesman problem. The problem aims to identify a minimum cost subtour in a network such that every demand node is within a specified distance from a node on the subtour. They extend this model to develop the maximal covering tour problem (8), in which the objectives are to minimize the total tour length and maximize the demand covered by the tour, subject to a constraint on the number of nodes in the subtour. Suggested application domains include rural health care, overnight delivery systems, and bi-modal transportation systems. Gendreau et al. (9) solve a variant of the maximal covering tour problem in which a given set of nodes must always be present in a tour, and propose its application to the location of post boxes and the design of collection routes. They also propose classes of valid inequalities for use in a branch-and-cut algorithm to solve large problem instances in a reasonable amount of time.

The overview of the literature indicates that the coverage criterion has been successfully applied to identify a path, tree or tour, all of which are special network structures. This study extends the coverage criterion to a network problem with multiple O-D pairs, labeled the multicommodity maximal covering network design problem (MCNDP). Its objectives are to minimize the total routing costs over all O-D pairs and maximize the total demand covered, subject to a budget constraint based on the retrofit costs incurred on the chosen links. To our knowledge, there do not exist prior studies that consider budgetary limitations, routing costs and coverage criteria in a single model with multiple O-D pairs. Potential applications of the MCNDP other than the problem addressed here include the design of regional transit systems and electric power networks, and planning of truck shipment routes.

The context we consider here for the MCNDP is a geographical region consisting of population centers and an underlying road transportation network. The major population centers represent the origins and destinations. For efficient earthquake response, each O-D pair should be connected in the minimum time possible. However, using a shortest path algorithm to minimize travel times may not provide sufficient accessibility to the minor population centers. Hence, we consider the population that can be covered by a route while determining the critical routes. Coverage of a population center is viewed from a link standpoint, and defined as the ability of a link to provide access to it.

This paper is organized as follows. The next section formulates the MCNDP and shows that it is NP-hard. Then, some valid inequalities are specified to enhance the formulation. The model is applied to a case study and insights from its performance are discussed. Finally, some concluding comments are presented.

**MATHEMATICAL MODEL**

**Problem Description**

We formulate the MCNDP generically here, and apply it to the critical routes problem in a later section. We are given the locations of demand centers of a region and its associated undirected network. The network links have a fixed cost for their usage and a routing cost. There is a budget constraint on the total fixed cost incurred. The MCNDP seeks to allocate a limited budget to links such that the total routing costs for a set of O-D pairs is minimized and the total demand covered by the routes connecting them is maximized. The demand of a center is covered if a link in one of the selected routes provides access to it.
Notation and Problem Formulation
Let \( G(N, E) \) denote an undirected network with node set \( N \) and link set \( E \). The indices \( i \) and \( j \) denote a node in the network, \( i, j \in N \) and \( E \subseteq N \times N \), where \([i, j]\) denotes an undirected link between nodes \( i \) and \( j \) with a nonnegative fixed cost \( f_{ij} \). Let \( B \) denote the available budget. Each O-D pair in the network is represented as a unique commodity type. Let \( k \) represent the commodity type index, \( k \in K \), where \( K \) denotes the set of all commodities. One unit of flow of commodity \( k \) must be transported over the network from its origin \( O(k) \) to its destination \( D(k) \). To differentiate the direction of flow of a commodity, we consider two directed links \((i, j)\) and \((j, i)\) corresponding to each original undirected link \([i, j]\). Let \( A \) denote the set of the directed links; all links are uncapacitated. Let \( c_{ij}^k \) be the unit nonnegative routing cost for commodity \( k \) on link \((i, j)\), and \( m \) the demand center index, \( m \in M \), the set of demand centers in the region. Let \( r_m \) represent the demand associated with demand center \( m \), and \( E_m \) the set of links that are eligible to cover it. We treat the demand centers separately from the nodes in the network.

The formulation contains three types of variables: (i) the arc flow variables, which define the flow of different commodities in each of the selected links, (ii) the design variables, which define the links selected for the network design, and (iii) the coverage variables, which define whether or not a demand center is covered. They are defined as follows:

\[
x_{ij}^k = \begin{cases} 1, & \text{if there is a unit flow of commodity } k \text{ on link } (i, j) \\ 0, & \text{otherwise} \end{cases}
\]

\[
y_{ij}^k = \begin{cases} 1, & \text{if link } [i, j] \text{ is used in a flow path} \\ 0, & \text{otherwise} \end{cases}
\]

\[
z_m = \begin{cases} 1, & \text{if demand center } m \text{ is accessible from a link of a flow path} \\ 0, & \text{otherwise} \end{cases}
\]

Let \( x = \{x_{ij}^k\} \) be the vector of flow variables. Let \( S \) represent a restricted solution domain in which some flow variables are fixed \( a \ priori \). As discussed later, this is necessary to restrict the geographical region within which the search is performed to enable computational time savings. Presumably, the commodity paths to be determined would be confined to a subnetwork surrounding the O-D pair, and not circuitous.

The formulation has two objectives: \( Z_1 \), the total routing cost, and \( Z_2 \), the total demand covered where,

\[
Z_1 = \sum_{k \in K} \sum_{[i, j] \in E} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) \quad \text{and} \quad Z_2 = - \sum_{m \in M} r_m z_m.
\]

The integer programming formulation for the \textbf{MCNDP} is expressed as follows:

\[
\textbf{MCNDP:} \quad \begin{align*}
\text{Minimize } Z &= [Z_1, Z_2] \\
\text{subject to} \\
\sum_{(i, j) \in A} x_{ij}^k - \sum_{(j, i) \in A} x_{ji}^k &= \begin{cases} 1, & \text{if } i = O(k) \\ -1, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i, k
\end{align*}
\]
\[ z_m \leq \sum_{k \in K} \sum_{[i, j] \in E_m} (x_{ij}^k + x_{ji}^k) \quad \forall m \] 
\[ x_{ij}^k \leq y_{ij} \quad \forall k, [i, j] \in E \] 
\[ x_{ji}^k \leq y_{ij} \quad \forall k, [i, j] \in E \] 
\[ \sum_{[i, j] \in E} f_{ij} y_{ij} \leq B \] 
\[ \sum_{(i,j) \in A} x_{ij}^k \leq 1 \quad \forall i, k \] 
\[ \sum_{(i,j) \in A} x_{ij}^k \leq 1 \quad \forall k, j = D(k) \] 
\[ \sum_{(i,j) \in A} x_{ij}^k \leq |Q| - 1 \quad \forall k, Q \subseteq N, 2 \leq |Q| \leq |N| - 2 \] 
\[ x_{ij}^k, x_{ji}^k = 0 \text{ or } 1 \quad \forall [i, j] \in E \] 
\[ y_{ij} = 0 \text{ or } 1 \quad \forall [i, j] \in E \] 
\[ z_m = 0 \text{ or } 1 \quad \forall m \in M \] 
\[ \chi \in S \] 

In multiobjective programming, there may never exist a single solution that optimizes all the objectives. Thus, the notion of an optimal solution is not relevant here and is replaced by the concept of a noninferior solution set from which the decision-maker selects the most preferred solution. A solution to the above formulation is said to be noninferior if there exists no other feasible solution that improves one objective without degrading the other. There are two popular solution approaches to multiobjective programming: preference based techniques and generating techniques (10). The latter approach, which has been commonly employed due to its simplicity, generates the entire noninferior solution set or an approximation of it. We adopt this approach. An approximation of the set can be obtained by using a single objective function formed by different convex combinations of the objective functions. Our objective function becomes:

\[ Z(w_1, w_2) = w_1 Z_1 + w_2 Z_2 \] 

where \( w_1 + w_2 = 1 \) and \( w_1, w_2 \geq 0 \).

Due to the discrete nature of the MCNDP solution set, the above weighting method is incapable of identifying noninferior solutions that lie in the duality gap of the convex hull of the set. Solving a constrained version to enable such identification entails a substantial increase in the computational effort. Also, the number of noninferior solutions is large, though finite, for even small-size discrete optimization problems and in the worst case increases exponentially with problem size. Hence, it is not practical to generate the entire noninferior set. Instead, it is appropriate to focus on generating an approximation to the noninferior solution set.

Equations 2 denote the network flow conservation constraints that require \( x \) to describe a simple path from the origin to the destination for all commodities. Equations 3 represent the coverage constraints, which imply that a demand center is covered only when at least one of the links providing accessibility to the center is in a flow path. Constraint sets 4 and 5 are the forcing constraints; they state that no flow is allowed in either direction of link \([i, j]\) unless the fixed cost associated with it is incurred. This disaggregate version of the forcing constraints is preferred.
over the more compact aggregate form as it provides a tighter LP relaxation (3). Constraint 6 is the budget constraint; it states that the sum of the fixed usage costs of the links in any solution should not exceed $B$.

Due to constraint set 3, any solution can entail looping paths for one or more commodities as loops help to achieve extra coverage. We seek loopless paths and this is enabled through constraint sets 7 and 8. Constraints 7 state that the maximum flow of any commodity type exiting any node in the network should not exceed unity. Constraints 8 prevent the occurrence of loops at destination nodes, which is not precluded by 7. The coverage constraints can lead to isolated subtours that do not share any link with the corresponding commodity flow paths in the solution. These subtours are prevented by adding subtour elimination constraints 9, in which $Q$ denotes the nodes in a subtour. Since, $|Q|$ can potentially take several values between 2 and $|N|-2$, there are potentially an exponential number of such constraints. Constraint sets 10, 11 and 12 restrict the flow, design, and coverage variables, respectively, to 0-1 values. Constraint 13 restricts the flow variables to the set $S$. This aids in modeling the topological restrictions upon the commodity flows in the network.

**Complexity of MCNDP**

**Lemma:** The MCNDP is NP-hard.

**Proof:** We prove it by restriction. Consider the instance $\text{MCNDP}_R$ of problem $\text{MCNDP}$: $|M| = 0$, that is, there are no demand centers and the objective function is just the total routing cost of the various commodities. The coverage constraints 3 cannot be imposed as there are no demand centers. Also, the constraints sets 7, 8 and 9 are redundant in the new formulation and can be ignored. Solving this instance, which has only one objective, is equivalent to solving a budget design problem which is known to be NP-hard (11). Therefore, the $\text{MCNDP}$ generalizes the budget design problem. It follows that the $\text{MCNDP}$ is NP-hard.

**VALID INEQUALITIES**

Typically, integer programming formulations are solved using branch-and-bound type methods that solve linear programs at the nodes of the branch-and-bound tree. The computational time for such procedures is dependent on the number of tree nodes enumerated to obtain the solution, as well as the time required to solve the formulation at each enumerated node. The number of nodes to be enumerated is dependent on the quality of the bounds generated by associated linear programming (LP) relaxations. Valid inequalities, which are constraints based on the problem characteristics, are useful in this context. They are redundant in an integer programming formulation but can eliminate non-integer solutions that are optimal for the LP relaxations. Hence, they improve the lower bounds computed by the solution algorithms, thereby generating computational time savings. Valid inequalities can be appended to the $\text{MCNDP}$ to enhance the formulation.

**Proposition 1.** The constraint

$$x^k_{ij} + x^k_{ji} \leq y_{ij} \quad \forall k, [i, j] \in E$$

(15)

is a valid inequality for the $\text{MCNDP}$.

**Proof:** In any feasible design, a link $[i, j]$ is either chosen or not. Consider the case where link $[i, j]$ is selected, which implies $y_{ij} = 1$. Then, the corresponding set of inequalities for that link in
Viswanath and Peeta

Constraints 15 are its subtour elimination constraints for \(|Q| = 2\). If that link is not selected, it implies \(y_{ij} = 0\). Then, the forcing constraints imply that \(x_{ij}^k = 0, \forall k\), and the corresponding constraints in 15 are valid. This completes the proof.

This valid inequality is a virtual constraint vis-à-vis the problem as it is not a direct representation of a physical reality or a logical characteristic. However, it encompasses constraints 4 and 5, and part of constraints 9 (that is, for \(|Q| = 2\)). Hence, it can be used to generate some computational time savings by reducing the problem size.

**Proposition 2.** The following constraints are valid for the MCNDP.

\[
\begin{align*}
y_{ij} & \leq z_m & \forall m, [i, j] \in E_m \\
x_{ij}^k & \leq z_m & \forall m, k, [i, j] \in E_m \\
x_{ij}^k + x_{ji}^k & \leq z_m & \forall m, k, [i, j] \in E_m
\end{align*}
\]

**Proof:** If \(x_{ij}^k = 1\) for some \(k\) and \([i, j] \in E_m\) then \(y_{ij}\) assumes the value 1, and the demand center \(m\) is covered implying \(z_m = 1\). If \(x_{ij}^k = 0, \forall k, [i, j] \in E_m\) then no fixed cost is incurred for the \(\forall [i, j] \in E_m\) and the demand center \(m\) is not covered. This completes the proof of constraints 16. The proof of constraints 17 follows directly from the forcing constraints 4 and 5, and the result just proved. Constraint set 18 is redundant in the MCNDP formulation and directly follows from constraints 15 and 16.

These valid inequalities imply that if a demand center \(m\) is not chosen, then links in \(E_m\) do not appear in the solution. This simplifies the budget constraint, leading to potential computational efficiencies.

**Proposition 3.** Let \(m_1\) and \(m_2\) be two demand centers such that \(E_{m_1} \subseteq E_{m_2}\), then the following relation holds.

\[z_{m_1} \leq z_{m_2}\]

**Proof:** Suppose demand center \(m_1\) is covered by flow of some commodity \(k\) in arc \((i, j)\), this flow also covers demand center \(m_2\). If demand center \(m_1\) is not covered, constraint 19 reduces to the nonegativity constraint for the variable \(z_{m_2}\). The proof is complete.

This valid inequality can potentially reduce computations by exploiting the problem characteristic that if a demand center is covered by a link \((i, j)\), it is redundant to search other links to cover this demand center.

**APPLICATION OF THE MODEL**

**Case Study**

The MCNDP is used to determine the critical routes under earthquakes for a network representing southwest Indiana, as depicted in Figure 1. It consists of 184 nodes, 307 links, and 93 population centers. The demand of a center is set equal to its population as obtained from the US Census 2000 Data. 15 of them have a population greater than 5000 and are denoted as major population centers. The remaining 78 have a population between 3000-5000, and are labeled minor population centers. O-D node pairs, which represent the commodities, are chosen so as to
ensure connectivity between the major population centers. As shown in Figure 1, the 15 major population centers can be viewed as being located in five different layers from top to bottom: 3 centers each in the first three levels, 2 in the fourth level, and 4 in the fifth level. To enable connectivity across the region it is reasonable to choose O-D pairs that connect major population centers between adjacent layers and, also those within each layer. This approach generated 33 O-D pairs in the network. The fixed cost of a link is the cost to retrofit all bridges on that link. Due to the lack of bridge retrofitting cost data, we assume unit costs on all links. The link routing costs are their free-flow travel times, implying that link capacity is ignored. This is because the problem is being addressed for the planning stage of emergency response. The link travel times are assumed to be symmetric. The links that can potentially cover a population center are those that are within some pre-specified threshold distance of it.

Solving the MCNDP

Solution Procedure
The MCNDP is solved using the branch-and-cut algorithm in CPLEX. It is a search technique that uses the tree structure in which CPLEX dynamically adds cut constraints at the tree nodes to reduce the tree size. The solution procedure has two primary computational aspects. The first is the number of the tree nodes searched in the branch-and-cut method since each requires the solution of a linear program. Thereby, larger the number of tree nodes searched, the greater the computational time. This highlights the critical importance of the second computational aspect, the formulation size, which depends on the number of constraints. In this context, computational savings can be generated by including or excluding valid inequalities, and/or the subtour elimination constraints.

The solution procedure first excludes most subtour elimination constraints to avoid exponential computational times. The only ones included are those for $|Q|=2$ implying link subtours. This is because there are only a few constraints with $|Q|=2$, and preliminary analysis indicates that the associated subtours can occur frequently in the solution if they are not precluded. Next, the solution procedure identifies subtours in the solution obtained from CPLEX using this relaxed formulation. If subtours exist, it adds the necessary subtour elimination constraint(s) and reoptimizes the new formulation. This step is repeated until no more subtours are encountered in the solution.

The best known integer solution, which is the upper bound to the optimal solution, is the final solution obtained from CPLEX. The lower bound to the optimal solution is the best objective function value across all LP relaxation solutions of the branch-and-cut nodes. They are used to determine the percentage optimality gap, which is a measure of the quality of the solution. The percentage optimality gap is defined as [(upper bound – lower bound)*100]/lower bound, where the bounds are obtained from CPLEX. It indicates how close the best known integer solution is to the optimal solution.

Implementation Details
The computing environment consists of a Sun Ultra Enterprise server E6500 having 26 400Mhz UltraSparc II processors under the multi-user Solaris 7 operating environment with 23GB RAM, 131GB swap space and 8MB cache. A C++ program was implemented to solve the problem. It invokes CPLEX’s mixed integer programming (MIP) solver that provides the branch-and-cut procedure. Experiments are performed by excluding and including the valid inequalities in the
formulation. A preliminary analysis was performed to calibrate the CPLEX parameter settings for different experimental scenarios in terms of the budgets and the relative weights of the two objectives, and for different % optimality gaps.

**Problem Reduction Strategy**

Preliminary runs highlight the intractable nature of the MCNDP; an instance with budget $B = 150$ and weight pair $(w_1, w_2) = (0.99, 0.01)$ required over 5 days to solve to a 1% optimality gap. To generate significant computational time savings, a strategy to reduce the search domain was incorporated based on the notion that the solution for an O-D pair would lie within some restricted geographical area around it. Therefore, it is reasonable to expect that the path of a commodity would be confined to a subnetwork around that O-D pair. This enables us to set the commodity flow variables to zero for the rest of the network, which provides the restricted solution domain $S$ for the formulation.

**EXPERIMENTS AND STUDY INSIGHTS**

The objectives of the experiments are to: (i) identify an effective problem reduction strategy, (ii) use the selected strategy to analyze the computational performance and model sensitivity for different budgets and relative weights for the two objectives, and (iii) assess the effects of valid inequalities on computational performance. They also seek to develop noninferior frontiers that illustrate the trade-offs between the conflicting objectives for different budgets to provide practical insights to decision-makers. In addition, network vulnerability is briefly investigated.

**Identification of an Effective Problem Reduction Strategy**

To select an effective problem reduction strategy, three strategies $S_0$, $S_1$ and $S_2$ were considered. The corresponding percentages of the total number of flow variables set to zero a priori are 0, 72.27 and 83.46, respectively, based on some preliminary analysis. Hence, $S_2$ has the most restricted search domain among the three strategies. The minimum budgets for a feasible solution under the three strategies are 46, 51 and 64, respectively. The minimum required budget increases when more flow variables are fixed because the feasible solution space is more restricted.

Systematic variation of the budget from 70 to 150 under the three search strategies for the weight pair $(0.999, 0.001)$ yielded the optimal solutions, which were identical across all the budgets except for 70 (where $S_2$ performs slightly worse). This reinforces the proposed logic for the search strategy that the solutions typically exist within a restricted region around the O-D pair. The maximum CPU times for the instances solved under $S_0$, $S_1$ and $S_2$ were 5.13, 1.25 and 1.10 minutes, respectively, and for the budget of 80. This indicates that the problem reduction strategy is computationally effective.

Another set of experiments was conducted for the same budget range with the weight pair $(0.99, 0.01)$ under $S_1$ and $S_2$. This implies that greater importance is given to population coverage here. Figures 2 and 3 plot the CPU time (in minutes) and % optimality gap, respectively, for various budgets. The range of CPU times obtained under $S_1$ varied from 53.25 to 3873.43 minutes, and that for $S_2$ from 2.09 to 559.52 minutes. The solutions were identical for $S_1$ and $S_2$ for budgets 140 and 150. For 130, the solution using strategy $S_1$ is marginally better than that of $S_2$. For the range 90 to 120, the solution under $S_2$ is better. This is because the % optimality gap is smaller for $S_2$ compared to $S_1$ in this range, as illustrated in Figure 3. In summary, for the range
90 to 150, \( S_2 \) performs as well as or better than \( S_1 \) in terms of the objective function value. This is because under \( S_2 \) the search domain is smaller, leading to good solutions in a reasonable amount of time. For budgets 70 and 80, the objective function value for \( S_1 \) is better though the % optimality gap is smaller for \( S_2 \) compared to \( S_1 \). This is because smaller budgets restrict the number of non-zero design variables, leading to more circuitous routes. Since \( S_2 \) is more restrictive compared to \( S_1 \), it precludes more such routes due to which its solutions are relatively poorer under low budgets.

Figures 2 and 3 indicate that the CPU times and the % optimality gaps, respectively, are higher for (0.99,0.01) across all budgets when compared to the (0.999,0.001). The significantly higher computational times for larger \( w_2 \) values is because of the need to search more routes due to the increased importance of population coverage in the objective function. When \( w_2 \) is smaller, the problem approaches the budget design problem where the network structure can be favorably exploited.

The above results indicate that, for most budget values, \( S_2 \) performs better than \( S_1 \) in terms of the objective function value, % optimality gap and the computational time. Hence, further experiments are conducted using the \( S_2 \) strategy.

**Computational Performance and Model Sensitivity under \( S_2 \)**

The formulation was solved using \( S_2 \) for the following weight pairs: (0.999,0.001), (0.995,0.005), (0.99,0.01), (0.9825,0.0175), (0.975,0.025), (0.95,0.05) and (0.9,0.1) for the budget range 70 to 150. Figures 4 and 5 plot the CPU times and % optimality gaps, respectively, for various budgets. The various instances for the first two weight pairs were solved to optimality in few seconds, and are not displayed in the figure. The computational times increase with higher \( w_2 \) and lower budget values. However, the trend is sometimes skewed by the % optimality gap attained, and the need to reoptimize some instances caused by subtours in intermediate solutions. Also, in general, the CPU times and % optimality gaps for budget 70 are better than those for 80-110. This is because 70 is close to the minimum budget 64, and therefore has a relatively smaller number of feasible solutions implying an easier search.

As noted earlier, when the weight for the system travel time is relatively high, the problem closely approximates the budget design problem. Then, the network structure can be exploited for better computational results. When the budget is low, several commodities are likely to share links in the final solution rendering the problem more difficult to solve due to these interactions. This was also observed by Dionne and Florian (12). In our experiments, subtours occur in the intermediate solutions for most of the budgets with the weight pairs (0.95,0.05) and (0.9,0.1), and only for one instance in (0.975,0.025). This is because a larger weight for population coverage leads to physically meaningless isolated subtours in an effort to improve the objective function value. In general, for a specific \((w_1,w_2)\), the number of subtours and reoptimizations increase with higher budgets as more links are available to form subtours.

**Noninferior Frontiers: Travel Time-Population Coverage Trade-offs**

Figure 6 displays the noninferior solution set for the budget range 70 to 150 for the instances solved under \( S_2 \). In the figure, the solutions toward the left of the graph for any budget correspond to lower \( w_2 \) values; for example, the leftmost points correspond to \( w_2 = 0.001 \). In general, for a budget, higher values of \( w_2 \) lead to greater population coverage at the expense of increased system travel time. Some solutions, mostly towards the right end of the graph are inferior or dominated due to their bigger % optimality gaps. Hence, they are shown as isolated
points and do not form a part of the corresponding noninferior frontiers. For example, the instance with budget 100 and weight pair (0.9,0.1) was solved to an optimality gap of 4.36%. However, the system travel time is higher and the population covered is lower than that for the weight pair (0.95,0.05). This provides flexibility to the decision-maker by generating a range of solutions. Noninferior solution sets, like the ones shown here, can be very useful to the decision-maker as they highlight the tradeoffs between conflicting objectives. They also aid decision-makers to compare the additional benefits accrued in terms of system travel time savings and extra population coverage due to an additional investment. For example, the marginal benefits that could be realized from a budget of 80 instead of 70 would be higher than those under 90 instead of 80.

Effect of Valid Inequalities on Computational Performance
The high CPU times observed in the preceding experiments are due to the relatively low improvements to the lower bound of the branch-and-cut tree as enumeration progresses. To improve the lower bounds for better computational performance, valid inequalities are incorporated in the formulation. Preliminary analysis identifies the valid inequality combinations \{15, 16, 17, 18\} and \{15, 16, 17, 18, 19\} as effective candidates. Table 1 compares the computational performance of the solutions that include and exclude valid inequalities for different instances defined by the budgets and relative weights of the two objectives. All solutions are solved to the same % optimality gap for an instance so as to ensure consistent comparisons. The results suggest that the addition of the valid inequalities lead to tighter LP relaxations for a majority of the problems in the nodes of the branch-and-cut tree. This enables faster solution times and reduces the number of tree nodes enumerated. Compared to the experiments that exclude the valid inequalities, the computational time savings range from 96.65% to 6.01%. Similarly, the percentage reduction in terms of the number of nodes enumerated ranges from 97.71% to 1.59%. Between the two sets of solutions that involve the valid inequalities, the combination \{15, 16, 17, 18\} performs better for most cases. This is because, in general, a lesser number of nodes are enumerated under this combination. Also, the lesser number of constraints for this combination implies faster computation at each enumerated node. However, because the enumeration of nodes is specific to a problem instances, these trends cannot be generalized.

Network Vulnerability Analysis
In the MCNDP formulation for the critical routes problem, survivability is addressed indirectly by assuming that retrofitting bridges strengthens their ability to withstand earthquakes. While this formulation can be extended to explicitly incorporate the risk of link failures in critical routes using a stochastic setting, some measure of the vulnerability of the network can be computed even within the framework of this study. This is performed by determining the number of link-independent routes for each O-D pair in the solution (that is, the critical routes sub-network). Link-independent routes do not share common links, and hence provide redundancy. The sum of number of link-independent routes across all O-D pairs is viewed as a proxy for the network vulnerability.

Figure 7 shows the total number of link-independent routes for different budgets for the experiments that exclude valid inequalities. In general, the number of link-independent routes increases with the budget across all weight pairs. This measure of network vulnerability provides decision-makers value-added information.
CONCLUDING COMMENTS

In this paper, we introduced the MCNDP. It seeks routes that minimize the total travel time over the selected routes and maximizes the total population covered, subject to a budget constraint. It is useful in addressing network-level disaster management planning. Potential application domains include vehicular traffic networks, freight transportation, telecommunication networks, supply chain design and electric power networks. An integer programming formulation of the problem was presented, and it was shown to be NP-hard. The problem characteristics were exploited to develop some valid inequalities to improve computational performance. The MCNDP was applied to generate critical routes for earthquake response planning in southwest Indiana. A problem reduction strategy was proposed to reduce computational times. The branch-and-cut procedure of CPLEX was used to solve the problem.

The study results indicate that typically the computational performance improves with the budget and higher values of $w_j$. Noninferior frontiers are generated to provide practical insights for decision-makers. They assist the decision-maker in analyzing the tradeoffs among various objectives and making budget allocation decisions. A network vulnerability analysis was performed to provide the decision-maker a proxy measure for the ability of the critical routes sub-network to be functional under an earthquake. On-going research explicitly accounts for link failures in the identification of critical routes using a stochastic programming framework.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 9702612. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.
REFERENCES

LIST OF TABLES AND FIGURES

TABLE 1 Comparison of computational results for valid inequalities

FIGURE 1 Road network of southwest Indiana.

FIGURE 2 Computational time versus budget under weights (0.99,0.01)

FIGURE 3 Percentage optimality gap versus budget under weights (0.99,0.01).

FIGURE 4 Computational time versus budget for different weight pairs under $S_2$.

FIGURE 5 Percentage optimality gap versus budget for different weight pairs under $S_2$.

FIGURE 6 Noninferior solution set for different budget values under $S_2$.

FIGURE 7 Network vulnerability analysis.
<table>
<thead>
<tr>
<th>$(w_1, w_2)$</th>
<th>Budget</th>
<th>% Gap</th>
<th>Cplex MIP solver</th>
<th>Cplex MIP solver with the addition of valid inequalities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time (minutes)</td>
<td>Number of tree nodes</td>
</tr>
<tr>
<td>$(0.99, 0.01)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>0.01</td>
<td>84.17</td>
<td>22760</td>
<td>61.42</td>
</tr>
<tr>
<td>80</td>
<td>2.2</td>
<td>559.52</td>
<td>186771</td>
<td>183.43</td>
</tr>
<tr>
<td>90</td>
<td>2.2</td>
<td>322.17</td>
<td>142183</td>
<td>263.32</td>
</tr>
<tr>
<td>100</td>
<td>1.2</td>
<td>427.13</td>
<td>205580</td>
<td>1144.54</td>
</tr>
<tr>
<td>110</td>
<td>1.2</td>
<td>304.18</td>
<td>183392</td>
<td>107.38</td>
</tr>
<tr>
<td>120</td>
<td>1</td>
<td>65.25</td>
<td>47777</td>
<td>4.28</td>
</tr>
<tr>
<td>130</td>
<td>0.5</td>
<td>49.54</td>
<td>49355</td>
<td>11.40</td>
</tr>
<tr>
<td>140</td>
<td>0.36</td>
<td>21.42</td>
<td>22245</td>
<td>8.32</td>
</tr>
<tr>
<td>150</td>
<td>0.01</td>
<td>2.09</td>
<td>2198</td>
<td>2.30</td>
</tr>
<tr>
<td>$(0.9825, 0.0175)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>0.1</td>
<td>563.02</td>
<td>145629</td>
<td>406.10</td>
</tr>
<tr>
<td>80</td>
<td>2.91</td>
<td>3265.44</td>
<td>941418</td>
<td>599.52</td>
</tr>
<tr>
<td>90</td>
<td>2.36</td>
<td>537.25</td>
<td>197582</td>
<td>2824.14</td>
</tr>
<tr>
<td>100</td>
<td>2.06</td>
<td>1487.50</td>
<td>599062</td>
<td>115.03</td>
</tr>
<tr>
<td>110</td>
<td>1.7</td>
<td>3305.41</td>
<td>1820738</td>
<td>374.06</td>
</tr>
<tr>
<td>120</td>
<td>1.2</td>
<td>847.49</td>
<td>568568</td>
<td>269.29</td>
</tr>
<tr>
<td>130</td>
<td>0.5</td>
<td>518.39</td>
<td>384683</td>
<td>92.41</td>
</tr>
<tr>
<td>140</td>
<td>0.24</td>
<td>328.29</td>
<td>307484</td>
<td>37.20</td>
</tr>
<tr>
<td>150</td>
<td>0.01</td>
<td>21.13</td>
<td>32370</td>
<td>23.56</td>
</tr>
<tr>
<td>$(0.975, 0.025)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>0.5</td>
<td>1082.05</td>
<td>268307</td>
<td>627.48</td>
</tr>
<tr>
<td>80</td>
<td>3.0</td>
<td>1774.27</td>
<td>439795</td>
<td>821.32</td>
</tr>
<tr>
<td>90</td>
<td>3.8</td>
<td>1581.32</td>
<td>570695</td>
<td>537.03</td>
</tr>
<tr>
<td>100</td>
<td>3.12</td>
<td>8004.04</td>
<td>3822086</td>
<td>1999.20</td>
</tr>
<tr>
<td>110</td>
<td>2.49</td>
<td>3325.07</td>
<td>1343533</td>
<td>223.39</td>
</tr>
<tr>
<td>120</td>
<td>1.39</td>
<td>438.14</td>
<td>2025300</td>
<td>80.23</td>
</tr>
<tr>
<td>130</td>
<td>0.89</td>
<td>470.37</td>
<td>249426</td>
<td>51.05</td>
</tr>
<tr>
<td>140</td>
<td>0.35</td>
<td>96.24</td>
<td>57088</td>
<td>28.14</td>
</tr>
<tr>
<td>150</td>
<td>0.1</td>
<td>60.42</td>
<td>69767</td>
<td>38.01</td>
</tr>
</tbody>
</table>
FIGURE 1 Road network of southwest Indiana.
FIGURE 2 Computational time versus budget under weights (0.99,0.01).
FIGURE 3 Percentage optimality gap versus budget under weights (0.99,0.01).
FIGURE 4 Computational time versus budget for different weight pairs under $S_2$. 
FIGURE 5 Percentage optimality gap versus budget for different weight pairs under $S_2$. 
FIGURE 6 Noninferior solution set for different budget values under $S_2$. 
FIGURE 7 Network vulnerability analysis.