Many-to-Many Game-Theoretic Approach for the Measurement of Transportation Network Vulnerability

Nicholas E. Lownes, Qixing Wang, Saleh Ibrahim, Reda A. Ammar,
Sanguthevar Rajasekaran, and Dolly Sharma

The vulnerability of a transportation network is strongly correlated with the ability of the network to withstand shocks and disruptions. A robust network with strategic redundancy allows traffic to be redistributed or reassigned without unduly compromising system performance. High-volume edges with limited alternative paths represent system vulnerabilities—a feature of transportation networks that has been exploited to identify critical components. A mixed-strategy, stochastic game-theoretic approach is presented for the measurement of network vulnerability. This method is designed to incorporate all origins and destinations in a network in a computationally efficient manner. The presented method differs from previous efforts in that it provides a many-to-many measure of vulnerability and edge-based disruptions that may not reside on a common path. A game that considers all possible origin–destination pairs is constructed between a router, which seeks minimum cost paths for travelers, and a network tester, which maximizes travel cost by disabling edges within the network. The method of successive averages is used for routing probabilities, and a weighted entropy function is employed to compute edge-disruption probabilities. The method is demonstrated on a small example network and then applied to the Sioux Falls, South Dakota, network. Results indicate good correspondence with a previous method that used equilibrium assignment and rapid solution convergence.

Transportation network vulnerability is, unfortunately, a continued and growing concern across the world due to the increase in attacks on transportation infrastructure during recent decades. Jenkins and Butterworth show that between 1920 and 1970 there were 15 attacks against surface transportation infrastructure, whereas in 2007 alone there were nearly 120 such attacks that involved the use of explosives or incendiary devices (*1*). With constraint resources to protect, harden, and monitor, surface transportation infrastructure researchers have looked to build mathematical models that are capable of identifying which network vulnerabilities are most critical to the security and robustness of the network. These methods have approached the problem from one of two perspectives: a practical empirical perspective or a theoretical perspective. The first offers the benefits of manageable computational complexity and straightforward interpretation but suffers from a lack of rigor in the modeling of traveler behavior. The second method provides behavioral rigor at the expense of computational tractability and relative ease of interpretation.

This paper presents a straightforward approach that utilizes practical measures of edge activity within a game-theoretic framework. The basic method of Bell and Bell et al. was adopted, in which a game is played between a benevolent router and malevolent network tester (*2, 3*). The router assigns traffic to the shortest path between its origin and destination, with multiple shortest paths accommodated in a straightforward proportional fashion. The tester disrupts edges in such a way as to maximize the deterioration of the network's performance.

Computational issues quickly arise as networks approach a practical size, which has led to most game-theoretic approaches being applied to single origin–destination (O-D) pairs (*2, 3*). The issues are centered on the need to maintain and update all-pairs shortest paths for each iteration to accommodate the increased costs of edges chosen for disruption by the router. This issue is exacerbated if the analyst attempts to have the router make an equilibrium assignment at each iteration, though equilibrium would provide a more realistic assignment.

This paper presents a game-theoretic network vulnerability model that builds on the foundation laid by previous work that used games against malevolent opponents. Here, the consideration of all O-D pairs was allowed through the introduction of an all-pairs shortest-path routine, a path membership function for router assignments, and an entropy function for tester disruptions. The result is a many-to-many edge-based measure of network vulnerability. This paper begins with a review of relevant literature and previous relevant studies, followed by a description of the model and its components. A small example application is presented to clarify the basic concepts of the solution method, followed by an application to the well-known Sioux Falls, South Dakota, network. A summary of the findings and discussion of the results concludes the document.

LITERATURE REVIEW

Game-theoretic network vulnerability approaches that envision games between router and demon have been explored in prior studies that have developed mixed strategy games for vulnerability assessment (*2–4*). The structure common to this work is the game between a router and a tester or demon, in which the router seeks a shortest-path assignment and the tester seeks to maximally disrupt the network.

N. E. Lownes and Q. Wang, Center for Transportation and Livable Systems, Civil and Environmental Engineering, University of Connecticut, 261 Glenbrook Road, Unit 2037, Storrs, CT 06269-2037. S. Ibrahim, R. A. Ammar, S. Rajasekaran, and D. Sharma, Computer Science and Engineering, University of Connecticut, 371 Fairfield Road, Unit 2155, Storrs, CT 06269-2155. Corresponding author: Q. Wang, qiw09005@engr.uconn.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2263, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 1–8.
DOI: 10.3141/2263-01

Bar-Noy et al. have shown that the tester's problem is NP-hard, which has resulted in the use of different approaches to the tester problem (*5*). Among these approaches are the method of successive averages (MSA) and the introduction of an entropy function to the tester's objective, both of which allow for a closed-form solution to the tester probability (*2, 3*). Corley and Sha described a conflict situation as the relationship between a "defender" and an "interdictor" to interpret the most vital links and nodes in a network—those whose removal would result in the greatest increase in the shortest-path distance between two nodes (*6*). Ball et al. later defined the most vital arc problem in a weighted network and proved that it is an NP-hard problem (*7*).

Murray-Tuite and Mahmassani presented a non-zero-sum game between an evil entity and a traffic management agency to identify vulnerable edges in a transportation network (*8*). Reliability was defined as the existence of a feasible connection between an origin and a destination, with an acceptable edge-failure probability. Murray-Tuite described the incorporation of two types of substitution (method and target) into a methodology to determine a transportation system's risk profile (*9*).

Ukkusuri and Yushimito proposed a heuristic method that incorporated a user equilibrium-assignment procedure to assess the importance of highway transportation networks; this method used travel time as the primary measure of criticality (*10*). This work benefited from the user equilibrium-assignment procedure, which gave results that the researchers acknowledged were counterintuitive yet correct. The procedure ranked edges based on a criticality function that was derived from the difference in system travel time when a particular edge was removed. This approach allowed a comprehensive treatment of single edge removals; however, the problem size would grow exponentially if combinations of multiple edges were considered for removal. Latora and Marchiori proposed a method to identify critical edges based on the difference in shortest paths between baseline and disrupted networks, which was similar in structure to the method put forward by Ukkusuri and Yushimito but without an equilibrium-assignment component (*11*). Scott et al. developed a systemwide approach to identify critical links and evaluate network performance; the approach considered network flows, link capacity, and network topology, although the method was not optimization or game based (*12*).

Hollander and Prashkar reviewed game-theoretic approaches in transportation analysis (*13*). They described four categories of applications: games against a demon (tester), games between travelers, games between authorities and a single traveler, and games between all travelers and an authority. More recently, in an extension to robust rail network design, Laporte et al. proposed a game against a demon in which travelers could be served by the rail network or the complementary network (*14*).

The work presented in this paper seeks to add to the existing literature by proposing a straightforward methodology for assessing the vulnerability of transportation networks that incorporates all O-D pairs in such a manner as to facilitate rapid computational times for near real-time applications. This simple method will be used as a foundation for later works that incorporate more realistic travel dynamics.

GAME-THEORETIC APPROACH TO VULNERABILITY

In this paper, it is assumed that there are two opponents in a noncooperative zero-sum game with symmetric information: the router, a benevolent player who seeks the shortest paths for all travelers, and an evil tester, who tries to disable edges in the network to maximally disrupt network performance. Inelastic demand is assumed and congestion effects are not incorporated. A directed graph $G = (N, E)$, where N is the set of nodes and E is the set of edges, represents the transportation system, with the corresponding notation described in Table 1. It is assumed that failed edges increase travel time proportional to a constant β, which is large enough to present a severe cost to travelers while maintaining a connected network. In this manner, algorithmic problems of a disconnected network are avoided, as is the problem of estimating demand changes due to network disruptions—an important and relevant problem, but outside the scope of this work.

TABLE 1 Notation Index

$e \in E$	Edge *e* belongs to set of edges *E*.
$i \in N$	Node *i* belongs to set of nodes *N*.
$p \in P$	O-D pair *p* belongs to set of O-D pairs *P*.
$k_p \in K_p$	Shortest path k_p between O-D pairs *p* belongs to set of shortest paths K_p.
θ	Tester confidence parameter
β	Failed edge penalty weight
n	Iteration counter
ϵ	Sufficiently small convergence criterion
h_{kp}	Path choice probability between O-D pair *p*
$a_{e,kp}$	Parameter that takes value 1 if $e \in k_p$ and value 0 otherwise
d_p	Travel demand between O-D pair *p*
C_e^F	Cost of edge *e* in failure scenario *F*
C_e^-	Cost of edge *e* in a normal state
C_e^+	Cost of edge *e* in a failed state
S_e^n	Statistical expected (*s*-expected) cost of edge *e* at iteration *n*
x_e^n	Edge-use probability differential of edge *e* at iteration *n*
ρ_e^n	Probability the tester disables edge *e* at iteration *n*
γ_e^n	Probability the router chooses edge *e* at iteration *n*

The objective of the game between the router and the tester is represented by a minimax formulation, as shown in Equation 1. Minimax problems have been shown to be NP-hard, and, as previously discussed, the tester objective is itself an NP-hard problem. Bell has stated that the minimax problem decomposes into two constituent problems, provided certain assumptions hold true (*2*). This decomposition approach is used in this paper, and the constituents are referred to as the "router problem" and "tester problem."

$$\min_\gamma \max_\rho V(\gamma, \rho) = \sum_{e \in E} \gamma_e^n \rho_e^n C_e^{F,n} \tag{1}$$

subject to

$$\sum_{e \in E} \rho_e = 1 \qquad \rho \geq 0 \tag{2}$$

$$\sum_{k_p \in K_p} h_{k_p} = 1 \qquad \mathrm{h} > 0 \tag{3}$$

$$\sum_{k_p \in K_p} h_{k_p} a_{e,k_\rho} = \gamma_e \qquad \forall e \tag{4}$$

where

V = objective function value representing the measure of network vulnerability,
γ = edge-choice probability, and
ρ = tester edge-failure probability.

The router and tester problems present formidable challenges. Equation 1 is presented as an edge-based formulation, whereas most previous work has used a path-based model. A definitional constraint is provided that links paths and edges. The edge-based formulation simply makes this link explicit. However, a path-based router decision is still assumed and employed in the solution to the router problem, meaning that path enumeration is still an issue. As has been mentioned, the tester problem has been shown to be NP-hard (*5*).

SOLUTION APPROACH

The many-to-many network vulnerability model contains two basic phases centered on the two players. In the first phase, the router identifies the shortest paths based on the tester's previous failure strategy through the use of the statistical expected (*s*-expected) cost of each edge (S_e^n). In the initialization phase, the network failure probability is assumed to be equal across all edges. The MSA is applied to update the edge-choice probability (γ_e^n) (*15*). The MSA is a heuristic; therefore, the solution obtained cannot be guaranteed to be optimal. In the second phase, assuming perfect information, the tester identifies a strategy of edge disruption to induce the maximum cost to network travelers (ρ_e^n). The game continues until the change in the objective function [$V(\gamma, \rho)$] falls below a critical threshold. The critical steps in this process are shown below:

Step 0. Initialize the network.
Step 1. Update the edge costs ($C_e^{F,n}$) under failure scenario ρ_e^n.
Step 2. Calculate the *s*-expected edge costs (S_e^n).
Step 3. Identify the shortest path or paths ($k_p \in K_p$) for each O-D pair (p).
Step 4. Calculate the edge-use probability differential x_e^n for each edge (e).
Step 5. Update the edge-use probability γ_e^n using the MSA.
Step 6. Calculate the tester edge-failure probability (ρ_e^{n+1}).
Step 7. Update $V^n(\gamma, \rho)$.
Step 8. If $V^n(\gamma, \rho) - V^{n-1}(\gamma, \rho) < \epsilon$, stop. Otherwise, go to Step 1.

Initialize Network

The preliminary step of initializing the network sets the initial conditions for the proposed procedure. The initial tester probability is uniformly distributed across the network $\left(\rho_e^1 = 1/|E|, \forall e\right)$; the initial router probability is set to zero ($\gamma_e^0 = 0 \; \forall e$); and the failed-state penalty, tester confidence parameter, and convergence criterion (β, θ, and ϵ, respectively) are assigned. Lastly, the value of the objective function ($V^0 = 0$) is set.

Update Edge Costs

Edge costs are treated as binary parameters, with each edge having a normal and failed state. It is assumed that the failed-state cost is directly related to the free-flow or normal condition travel time through the failed-state penalty coefficient (β), as shown in Equation 5.

$$C_e^{F,n} = \begin{cases} C_e^- & \text{if } \rho_e^{n-1} = 0 \\ C_e^+ = \beta C_e^- & \text{if } \rho_e^{n-1} > 0 \end{cases} \tag{5}$$

β is defined as equal to 10, although other values can be explored, as suggested by Bell et al. (*3*).

Calculate *s*-Expected Edge Costs

The assumption of perfect information allows the router to update the expected costs of the network, based on knowledge of the tester's previous strategy. The *s*-expected cost is updated with Equation 6. If the tester will fail an edge (e) with probability $\rho_e^{F,n-1} = 1$, then the cost of the edge is its failed-state cost (C_e^+). If the tester will fail an edge (e) with probability $\rho_e^{F,n-1} = 0$, then the *s*-expected cost will be its normal-state cost (C_e^-).

$$S_e^n = \left(1 - \rho_e^{n-1}\right) C_e^- + \rho_e^{n-1} C_e^+ \tag{6}$$

Identify Shortest Paths

The shortest path or paths ($k_p \in K_p$) for all O-D pairs (p) in the network are identified through the use of the previously updated *s*-expected costs. Ahuja et al. have described a generic, pseudopolynomial all-pairs shortest-path problem and the Floyd-Warshall algorithm, a polynomial implementation of the generic algorithm (*16*). For this specific implementation, an all-pairs shortest-path implementation was adopted that was based on Djikstra's algorithm for path enumeration at each iteration.

Calculate Probability Differential

In this step, the edge-use probability differential (x_e^n) is introduced and calculated. This parameter allows the router to assign traffic proportional to the shortest paths in the network of which e is a member.

$$x_e^n = \begin{cases} \displaystyle\sum_{\{p: e \in K_p\}} \frac{d_p}{\sum_p d_p} \cdot \frac{1}{|K_p|} & \text{if } e \cap K_p \neq \{\varnothing\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

This method accommodates the existence of multiple shortest paths between an origin and a destination through the inclusion of $1/|K_p|$ in the expression, by which an O-D pair with multiple shortest paths will have an equivalent probability differential across each of the paths. In baseline networks this is rarely a problem; however, the proportional *s*-expected costs coupled with the MSA updating of router probability may make this a more significant concern in later iterations.

Update Edge-Use Probability

The MSA is applied to update the edge-use probability through the use of the probability differential (x_e).

$$\gamma_e^n = \left(\frac{1}{n}\right) x_e^n + \left(1 - \frac{1}{n}\right) \gamma_e^{n-1} \tag{8}$$

Calculate Edge-Failure Probability

To calculate the edge-failure probability, Bell et al. proposed the introduction of an entropy function to the tester's level of this problem (*3*). With a weighted entropy function, the tester's problem has an explicit solution that is unique in edge-failure probability. The modified objective function is shown in Equation 9, with the explicit solution shown in Equation 10.

$$\min_{\gamma} \max_{\rho} V^n(\gamma, \rho) = \sum_{e \in E} \gamma_e^n \rho_e^n C_j^{F,n} - \left(\frac{1}{\theta}\right) \sum_{e \in E} \rho_e^n \ln \rho_e^n \tag{9}$$

$$\rho_e^{n+1} = \frac{\exp\left(\theta \gamma_e^n C_e^{F,n}\right)}{\sum_{j \in E} \exp\left(\theta \gamma_j^n C_j^{F,n}\right)} \tag{10}$$

The parameter θ has been called the aggressiveness of the tester but could also be interpreted as the confidence of the tester in its strategy. As θ increases, the tester's strategy coalesces around a small number of links. If θ decreases, the strategy is less focused on a small set of edges. In fact, as θ goes to zero, the link-failure probability approaches the initial value $1/|E|$. As θ goes to infinity, the solution tends to that of Equation 1 (*3*). The introduction of the entropy function results in a heuristic for the tester problem in that the chosen strategy will no longer be the worst possible, but something approaching the worst possible, depending on the confidence or aggressiveness of the tester.

RESULTS

The method described will be demonstrated in three applications. The first is a small hypothetical network designed to provide basic intuition and understanding of the methodological components. The other two applications use the Sioux Falls network, with full O-D data, as well as O-D data for comparison with an equilibrium-based method (*10*). Figure 1 depicts the small example network; it contains four nodes and six edges, with an edge identifier and edge cost noted in parentheses next to each edge. Table 2 provides associated O-D data d_p.

Tables 2 through 4 provide information about the example problem as it proceeds through two iterations. Table 2 provides O-D data and a summary of the shortest-path cost through each of the two iterations. Table 3 provides edge cost, *s*-expected edge cost, edge-use probability, and edge-failure probability when the tester confidence (θ) equals 0.5. Table 4 presents the same information as Table 3, but with θ equal to 10.

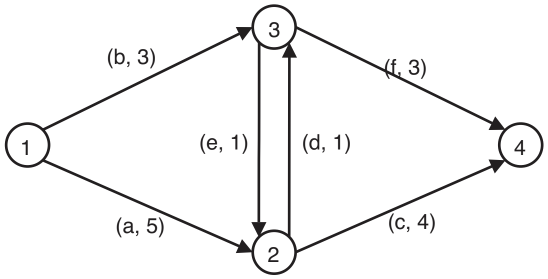

FIGURE 1 Example network (edge identifier, edge cost).

TABLE 2 Shortest Paths

		1st Iteration		2nd Iteration	
O-D Pair (*p*)	d_p	K_p	Cost of K_p	K_p	Cost of K_p
(1, 2)	1	b–e	4	a	5
(1, 3)	1	b	3	a–d	6
(1, 4)	1	b–f	6	a–c	9
(2, 3)	1	d	1	d	1
(2, 4)	2	d–f, c	4	c	4
(3, 4)	1	f	3	e–c	5

TABLE 3 Edge Probability and Failure Probability for $\Theta = 0.5$

		1st Iteration			2nd Iteration		
e	C_e^-	S_e^n	γ_e^n	ρ_e^{n+1}	S_e^n	γ_e^n	ρ_e^{n+1}
a	5	5	0	0.001	5.036	0.214	0.371
b	3	3	0.429	0.490	16.237	0.214	0.044
c	4	4	0.143	0.014	4.496	0.286	0.531
d	1	1	0.286	0.003	1.030	0.286	0.007
e	1	1	0.143	0.002	1.015	0.143	0.004
f	3	3	0.429	0.490	16.237	0.214	0.044

TABLE 4 Edge Probability and Failure Probability for $\Theta = 10$

		1st Iteration			2nd Iteration		
e	C_e^-	S_e^n	γ_e^n	$\rho_e^{F,n}$	S_e^n	γ_e^n	$\rho_e^{F,n}$
a	5	5	0	≈ 0	5	0.214	≈ 0
b	3	3	0.429	0.5	16.5	0.214	0.5
c	4	4	0.143	≈ 0	4	0.286	≈ 0
d	1	1	0.286	≈ 0	1	0.286	≈ 0
e	1	1	0.143	≈ 0	1	0.143	≈ 0
f	3	3	0.429	0.5	16.5	0.214	0.5

This method is presented as many-to-many, or for all O-D pairs, so at any given iteration the shortest path (or paths) for every O-D pair is being considered. The progression of the shortest-path cost can be seen in Table 2. In the first iteration, the tester's link-failure probability is equal for all edges, so the router simply chooses the free-flow shortest path for all O-D pairs. Table 3 shows that Edge a is the only edge that is not a member of a shortest path. Its use probability is therefore zero, along with its failure probability essentially being zero. At the end of the first iteration, the router has chosen an edge-use strategy, which, in turn, has led to the tester's edge-failure strategy. Note that $\Sigma_e \gamma_e^n > 1$, which is due to Edges b and f being members of two shortest paths. The simple modification $\gamma_a^n + \gamma_b^n/2 + \gamma_c^n + \gamma_d^n + \gamma_e^n + \gamma_f^n/2 = 1$ yields the expected result: unity.

During the second iteration, the edge costs and *s*-expected costs are updated according to the link-failure probabilities. This causes a shift in the shortest path, and Edge a no longer has a zero probability of selection by either the router or the tester.

The comparison between Tables 3 and 4 in this simple example demonstrates the effect of tester confidence or aggressiveness. In Table 3, in which θ is equal to 0.5, the tester's strategy is spread more widely among the six edges. The focus is Edge c, but Edges a, b, and f receive nontrivial attention after two iterations. When confidence is increased (i.e., θ = 10), the tester's strategy becomes much more rigid, focusing only on Edges b and f, regardless of a shift by the router. This rigidity is partially explained by the use of parameter β, which severely penalizes any edges with a nonzero probability of failure.

NETWORK APPLICATIONS

Figure 2 depicts the well-known and well-studied Sioux Falls network. Although this network bears little physical resemblance to Sioux Falls today, the network and its associated data have been used in a wide variety of network analysis studies.

The Sioux Falls network has 24 nodes and 76 edges. This paper will present results on the Sioux Falls network in two ways: (*a*) a comparison between the method presented in this paper and the findings of Ukkusuri and Yushimito (*10*), and (*b*) an application of the method presented in this paper that uses the traditional O-D data, which encompasses a larger number of nodes. The reduced O-D data for comparison with Ukkusuri and Yushimito are given in Table 5; the full application has 552 O-D demand pairs, equivalent to the number used by LeBlanc (*17*).

Table 5 illustrates the effect of tester confidence. As θ goes from one to 10, the tester (as displayed through edge-failure probabilities) focuses the strategy on a progressively smaller set of links. When θ equals 10, the tester's strategy comprises only six edges. Qualitatively, there is a good degree of correspondence between the strategies at the three levels of tester confidence, especially among the edges with the highest failure probability. There is also a high degree of correspondence between the method presented in this paper and previous work (*10*).

The method presented in this paper was run on the medium demand scenario from Ukkusuri and Yushimito, in which the authors measured the criticality of a link by reporting the percentage increase in system travel time with the removal of a particular link (*10*). The five most critical edges found by Ukkusuri and Yushimito are all near the top of the tester's strategies, and throughout the list there is a high degree of agreement between the two methods. Figure 2 depicts the origins and destinations of the trips used in constructing this comparison. Unsurprisingly, many of the critical edges cluster around the origins and destinations, as this is where the travel is concentrated. The critical edges found by Ukkusuri and Yushimito cluster around the origins and destinations somewhat more than the game-theoretic edges. This is illustrated by the boldfaced edge numbers in Table 5,

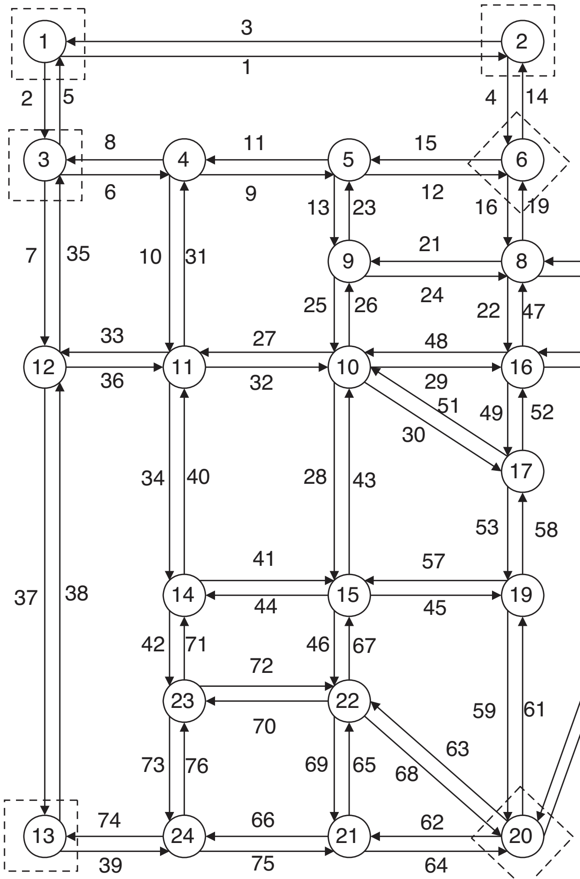

FIGURE 2 Sioux Falls network (dashed squares = origins for Table 5; dashed diamonds = destinations for Table 5).

TABLE 5 Tester Confidence or Aggressiveness and Comparison with Ukkusuri and Yushimito (*10*)

θ = 0.5, Iter. = 9114, RT= 63.5 s			θ = 1, Iter. = 3442, RT = 23.0 s			θ = 10, Iter. = 5049, RT= 40.3 s			Ukkusuri & Yushimito (*10*)	
Edge No.	Failure (%)	Use (%)	Edge No.	Failure (%)	Use (%)	Edge No.	Failure (%)	Use (%)	Edge No.	Critical (%)
4	31.51	27.15	4	34.08	24.66	4	34.78	23.20	39	80.34
6	13.97	29.87	39	16.62	29.03	39	20.55	28.87	4	71.78
12	13.96	29.87	6	15.69	28.88	6	17.66	28.83	**75**	67.87
39	10.99	28.67	12	15.65	28.88	12	16.87	28.82	64	50.02
20	4.93	32.88	36	5.23	17.43	36	8.32	19.10	20	46.23
7	4.74	24.46	20	2.89	32.87	64	1.81	18.84	16	42.96
2	2.98	22.14	2	2.87	24.64				1	30.43
64	2.89	14.66	7	2.68	24.47				2	26.19
75	2.62	28.67	64	1.95	15.78				60	20.17
36	2.60	14.31	32	0.91	17.42				50	19.09
32	1.27	14.31	75	0.91	29.03				7	17.61
68	1.18	14.01	29	0.16	17.43				54	16.08
16	0.95	32.88	68	0.11	13.25				37	12.65
9	0.70	29.87	16	0.11	32.87				6	11.02
29	0.62	14.31	9	0.05	28.88				**26**	9.02
37	0.47	17.24	37	0.03	17.25				12	7.84
60	0.35	11.42	60	0.02	11.78				**9**	7.79
38	0.33	14.83	38	0.01	14.47				3	7.71
50	0.20	11.55							68	4.63
56	0.18	8.20							**76**	4.61
18	0.18	16.08							**72**	4.48
35	0.17	7.73							**30**	4.06
65	0.14	14.01							**52**	3.99
1	0.14	4.51							**32**	3.84
47	0.07	2.77							**10**	3.30
All other failure < 0.07%			All other failure < 0.01%			All other failure < 0.01%				

NOTE: Iter. = iteration; RT = run time; No. = number.

which show the edges that were not incident to either an origin or destination.

The equilibrium method presented by Ukkusuri and Yushimito has a substantial number of nonincident edges in the critical set, though they tend to be clustered lower on the list than in the game-theoretic approach. This may be caused by the tendency of the game-theoretic approach with a low θ to spread a strategy over a larger number of edges, regardless of the appropriateness of the assignment. This loss in assignment realism is the price paid for the relatively quick solution time of the proposed algorithm, which has a convergence criterion of $\epsilon = 0.00001$.

Table 6 provides vulnerability results for the Sioux Falls network with the 552 nonzero demand pairs. Here, the tester's strategy is much broader, which makes sense given that every node is now an origin and a destination. When θ is higher, the tester appears to have more confidence in the higher-ranking edges and less confidence in the edges on the lower end of the list.

The link-use percentages are much smaller than those in Table 5. The link-use probability is updated through the probability differential (x_e^n), which is a measure of the proportion of O-D demand that might pass through a link. In the case of the full O-D matrix and its results in Table 6, the demand is spread much more evenly throughout the network, with the router concentrating less on particular links clustered around limited origins and destinations than in the example application in Ukkusuri and Yushimito (*10*).

Each application of this method will produce useful results relative to the particular application. That is, a failure probability of 12% in Table 6 cannot be compared to a failure probability of 25% in Table 5. Failure and usage probabilities are a measure of the probability that the tester or router, respectively, would include the edge in their strategy. These numbers can be used to give a relative degree of criticality within a particular example but offer no broad comparative measure across applications. Figure 3 shows the results from Table 6 in graphical form, with the boldest lines representing "high criticality" edges with failure probabilities greater than 5%, dashed lines representing edges with failure probabilities greater than 1% and less than 5%, and gray lines representing "lower criticality" edges with failure probabilities less than 1%. In this application, the highly critical edges are clustered in the middle, around Node 10, which is the highest activity node in the network in terms of O-D demand. Both directions of edges between node pairs are coded identically in Figure 3. This is probably because this method does not consider congestion effects, only the number of shortest paths an edge is a member of and the relative activity of those paths.

TABLE 6 Full O-D Matrix Vulnerability Results

θ = 1, Iter. = 84, RT = 0.704 s			θ = 5, Iter. = 335, RT= 2.95 s			θ = 10, Iter. = 72, RT = 0.63 s		
Link No.	Failure (%)	Link (%)	Link No.	Failure (%)	Link (%)	Link No.	Failure (%)	Link (%)
48	5.34	6.83	27	12.04	4.01	27	12.27	3.69
29	5.28	6.80	32	11.88	4.00	32	12.01	3.69
27	3.68	4.72	43	10.21	3.29	43	11.86	3.07
32	3.63	4.69	28	9.98	3.28	28	11.71	3.07
28	2.92	3.55	29	5.98	4.66	40	6.23	4.45
43	2.92	3.55	48	5.94	4.66	34	6.17	4.44
46	2.42	6.47	46	4.82	6.07	46	5.73	5.90
67	2.40	6.44	67	4.69	6.05	29	5.68	4.42
22	2.20	3.69	22	4.27	3.59	48	5.66	4.42
47	2.20	3.69	47	4.26	3.59	67	5.46	5.89
40	1.81	4.13	40	4.00	4.46	22	4.29	3.48
34	1.79	4.10	34	3.89	4.45	47	4.25	3.48
45	1.77	5.43	45	1.99	5.48	45	1.75	5.51
57	1.77	5.43	57	1.98	5.48	57	1.62	5.48
49	1.57	7.54	74	1.07	3.80	74	0.76	3.92
52	1.57	7.54	39	1.06	3.80	39	0.76	3.92
33	1.43	2.36	56	0.91	3.72	23	0.69	3.12
36	1.43	2.36	60	0.91	3.72	13	0.64	3.10
74	1.41	3.50	15	0.78	3.64	15	0.32	3.70
39	1.41	3.50	12	0.75	3.62	56	0.30	3.69
26	1.38	4.61	36	0.53	2.30	12	0.28	3.67
25	1.37	4.58	33	0.52	2.29	60	0.27	3.67
12	1.36	3.41	23	0.52	2.75	36	0.17	2.37
15	1.36	3.41	13	0.50	2.74	10	0.17	2.37
13	1.29	2.63	10	0.43	2.23	31	0.16	2.36

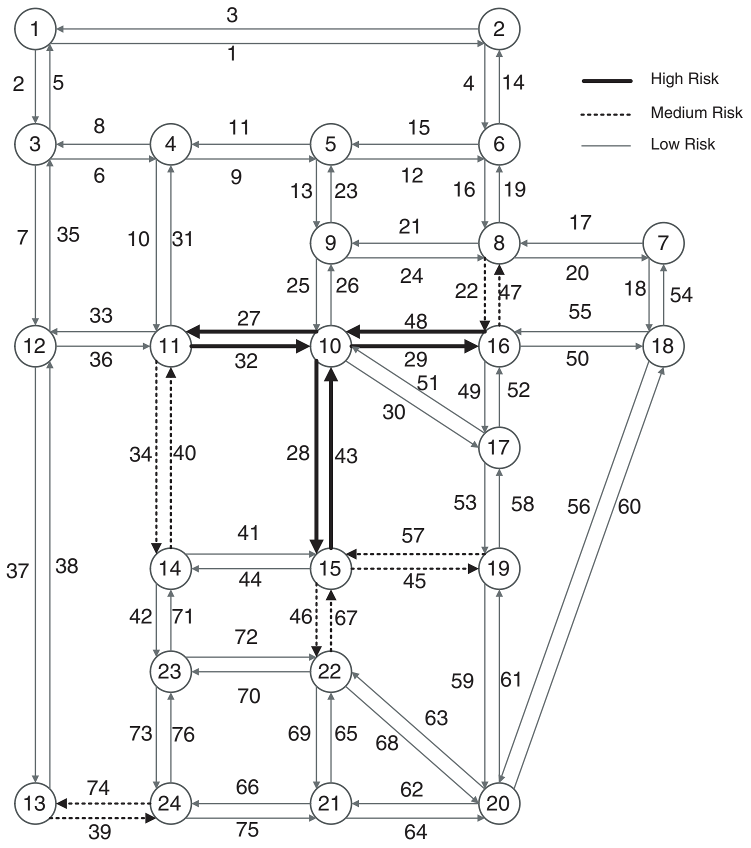

FIGURE 3 Full O-D results with tester failure probabilities.

CONCLUDING REMARKS

This paper presents a many-to-many method for the estimation of the vulnerability of transportation network components. The method adopts a game-theoretic framework and applies heuristics to solve both levels of the problem. The heuristics utilize practical measures of edge utilization and cost but do not provide an equilibrium assignment. The method solves the transportation network vulnerability problem rapidly on a small network, producing results that correspond well with an equilibrium-based methodology.

There are numerous challenges in the future for this, or any, method that provides a network vulnerability measure. Application to large networks presents substantial challenges, especially if on-demand applications (in emergency situations, for example) are to be pursued. The edge penalty function is relatively simple and can be improved, as can the probability differential metric, perhaps by incorporating the relative capacity of the paths. Ultimately, a method that combines the strengths of both the game-theoretic and equilibrium-assignment approaches should provide the level of reliability and realism that this important topic deserves. To this end, the work presented in this paper describes a straightforward method for the incorporation of all O-D pairs that could be applied on-the-fly in applications where the response to changes in the network needs to be coordinated in near real time. The authors are currently engaged in further development of the method, maintaining the computational efficiency while improving the realism by adding congestion effects.

ACKNOWLEDGMENTS

This material is based on work supported by the U.S. Department of Homeland Security. The authors thank Kelly Bertolaccini for her editorial assistance.

REFERENCES

1. Jenkins, B., and B. Butterworth. *Explosives and Incendiaries Used in Terrorist Attacks on Public Surface Transportation: A Preliminary Empirical Examination.* Report CA-MTI-10-2875. Department of Homeland Security; U.S. Department of Transportation, 2010.
2. Bell, M. G. H. The Use of Game Theory to Measure the Vulnerability of Stochastic Networks. *IEEE Transactions on Reliability,* Vol. 52, No. 1, 2003, pp. 63–68.
3. Bell, M. G. H., U. Kanturska, J. D. Schmöcker, and A. Fonzone. Attacker–Defender Models and Road Network Vulnerability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* Vol. 366, No. 1872, 2008, pp. 1893–1906.
4. Bell, M. G. H., and C. Cassir. Risk-Averse User Equilibrium Traffic Assignment: An Application of Game Theory. *Transportation Research Part B: Methodological,* Vol. 36, No. 8, 2002, pp. 671–681.
5. Bar-Noy, A., S. Kuller, and B. Schieber. *The Complexity of Finding the Most Vital Arc and Nodes.* Technical Report CS-TR-3539. Institute for Advanced Computer Studies, University of Maryland, College Park, 1995.
6. Corley, H. W., and D. Y. Sha. Most Vital Links and Nodes in Weighted Networks. *Operations Research Letters,* Vol. 1, No. 4, 1982, pp. 157–160.
7. Ball, M. O., B. L. Golden, and R. V. Vohra. Finding the Most Vital Arcs in a Network. *Operations Research Letters,* Vol. 8, 1989, pp. 73–76.
8. Murray-Tuite, P. M., and H. S. Mahmassani. Methodology for Determining Vulnerable Links in a Transportation Network. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1882,* Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 88–96.
9. Murray-Tuite, P. M. Transportation Network Risk Profile for an Origin–Destination Pair: Security Measures, Terrorism, and Target and Attack Method Substitution. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2041,* Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 19–28.
10. Ukkusuri, S. V., and W. F. Yushimito. A Methodology to Assess the Criticality of Highway Transportation Networks. *Journal of Transportation Security,* Vol. 2, No. 1–2, 2009, pp. 29–46.
11. Latora, V., and M. Marchiori. Vulnerability and Protection of Infrastructure Networks. *Physical Review E: Statistical, Nonlinear and Soft Matter Physics,* Vol. 71, No. 1, 2005, pp. 015103-1–015103-4.
12. Scott, D. M., D. C. Novak, L. Aultman-Hall, and F. Guo. Network Robustness Index: A New Method for Identifying Critical Edges and Evaluating the Performance of Transportation Networks. *Journal of Transport Geography,* Vol. 14, No. 3, 2006, pp. 215–227.
13. Hollander, Y. H., and J. Prashkar. The Applicability of Non-Cooperative Game Theory in Transport Analysis. *Transportation: Planning, Policy, Research, Practice,* Vol. 33, No. 5, 2006, pp. 481–496.
14. Laporte, G., A. M. Juan, and P. Federico. A Game Theoretic Framework for the Robust Railway Transit Network Design Problem. *Transportation Research Part B,* Vol. 44, No. 4, 2010, pp. 447–459.
15. Sheffi, Y. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods.* Prentice Hall, Englewood Cliffs, N.J., 1985.
16. Ahuja, R., T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, N.J., 1993.
17. LeBlanc, L. J. An Algorithm for the Discrete Network Design Problem. *Transportation Science,* Vol. 9, No. 3, 1975, pp. 183–199.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

The Transportation Network Modeling Committee peer-reviewed this paper.