Supplementary Document for the Paper "Regulating Hazardous Materials Transportation by Dual Toll Pricing"

Appendix A

Recall that the first stage involves obtaining regular and hazmat flow vectors, $\breve{v} \in V$ and $\breve{x} \in X$ that solve the following non-convex problem we denoted as (MR):

$$\min_{v \in V, x \in X} z(v,x) = \sum_{s \in \mathcal{S}} \sum_{(i,j) \in \mathcal{A}} n^s c_{ij}(v_{ij}) \rho_{ij} x_{ij}^s. \tag{1}$$

In the next subsections, we explain the details of the approaches we underscored in Section 4.1 of the main paper.

I Modified Frank-Wolfe Algorithm

For the sake of notational simplicity, we assume the BPR function be of the form $c_{ij}(v_{ij}) = A_{ij} + B_{ij}(v_{ij})^4$, for some positive constants A_{ij} and B_{ij}. We describe the implementation of the Modified Frank-Wolfe algorithm to solve (MR) problem as follows.

The objective function (1) can be replaced by its first-order Taylor approximation at any arbitrary point $(\hat{v}, \hat{x})$:

$$z(v,x) \approx z(\hat{v},\hat{x}) + \frac{\partial z}{\partial v}(\hat{v},\hat{x})(v-\hat{v}) + \frac{\partial z}{\partial x}(\hat{v},\hat{x})(x-\hat{x}) + \ldots,$$

Ignoring the constant terms in the above equation and owing to the disjoint constraint sets V and X, the minimization problem in the Modified Frank-Wolfe algorithm can be written as two separate linear minimization problems for regular and hazmat. That is, in each iteration k with an incumbent (v^k, x^k), we consider linear approximations of $z(v,x)$ resulting to the following linear programs:

$$\min_{v \in V} \quad v^T \frac{\partial z(v^k, x^k)}{\partial v} = \sum_{s \in \mathcal{S}} \sum_{(i,j) \in \mathcal{A}} \left[4 n^s B_{ij} (v_{ij}^k)^3 \rho_{ij} x_{ij}^{sk} \right] v_{ij} \tag{2}$$

$$\min_{x \in X} \quad x^T \frac{\partial z(v^k, x^k)}{\partial x} = \sum_{s \in \mathcal{S}} \sum_{(i,j) \in \mathcal{A}} \left[n^s \left(A_{ij} + B_{ij} (v_{ij}^k)^4 \right) \rho_{ij} \right] x_{ij}^s \tag{3}$$

Let $v^{k+1/2}$ and $x^{k+1/2}$ denote the solutions of (2) and (3), respectively. Update the solutions as follows:

$$v^{k+1} = v^k + \alpha_k d^k \tag{4}$$

$$x^{k+1} = x^k + \alpha_k d^k \tag{5}$$

where α_k is the step size and d^k is the actual search direction evaluated using $v^{k+1/2}$ and $x^{k+1/2}$ as described in Fukushima (1984).

The next two following subsections are devoted to describing the two variations of the linearzation approach we established which are used to verify that the solutions calculated by Modified Frank-Wolfe are in fact valid approximations.

II Piecewise-Linear Model 1

Note that the first linearization approach involves two parts to transform the non-linear objective function (1). For illustration purposes and notational simplicity, we consider the BPR travel time function expressed as $c_a(v_a) = A_a + B_a(v_a)^4$, with v_a denoting the traffic volume in each arc $a \in \mathcal{A}$, as the only decision variable. In what follows, we represent this link travel time function as a piecewise-linear function.

Let the feasible domain of v_a (i.e., $[0, \bar{v_a}]$) be partitioned into N segments. For this purpose, for each link a, we choose a series of values of $K_{a,n}$ to partition the feasible domain of v_a into many small regions, where $0 < K_{a,n} < K_{a,n+1} < \bar{v_a}$ for $n = 1, \ldots, N-1$. Denote each region as $[n]$, where $K_{a,n} \leq v_a < K_{a,n+1}$. Now, we can consider the travel time function to be represented by a piecewise-linear function with N regions. That is we specify a linear function to approximate the non-linear travel time function within each region $[n]$, i.e.,

$$c_a(v_a) \approx \mu_n^a v_a + \nu_n^a \qquad \text{if} \quad K_{a,n} \leq v_a < K_{a,n+1}, \tag{6}$$

with μ_n^a and ν_n^a denoting the parameters of the linear function. To determine the parameters in each region $[n]$, we approximate the travel time function $c_a(v_a)$ with its first-order Taylor expansion. That is, when $K_{a,n} \leq v_a < K_{a,n+1}$, i.e., v_a falls in segment n, parameters μ_n^a and ν_n^a can be estimated by the partial derivatives of the travel time function with respect to v_a evaluated at $K_{a,n}$ as follows:

$$\mu_n^a = \left.\frac{\partial c_a}{\partial v_a}\right|_{K_{a,n}} = 4B_a(v_a)^3\big|_{K_{a,n}} = 4B_a(K_{a,n})^3, \quad \text{and} \tag{7}$$

$$\nu_n^a = c_a(K_{a,n}) - K_{a,n}\left.\frac{\partial c_a}{\partial v_a}\right|_{K_{a,n}} = \left(A_a + B_a(K_{a,n})^4\right) - K_{a,n}\left(4B_a(K_{a,n})^3\right). \tag{8}$$

One can observe that once we specify the region in which v_a falls, μ_n^a and ν_n^a become known parameters, making (6) a linear function in v_a. Replacing equation (6) in the objective function

(1), the objective function to be minimized is transformed to the following form:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} n^s(\mu_n^a v_a+\nu_n^a)\rho_a x_a^s = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} n^s\rho_a(\mu_n^a v_a x_a^s+\nu_n^a x_a^s), \quad \text{if} \quad K_{a,n}\le v_a < K_{a,n+1} \tag{9}$$

Moreover, let us introduce a new variable $c_a^s(v_a, x_a^s)$, for all arcs $a \in \mathcal{A}$ and shipments $s \in \mathcal{S}$, denoting the travel time of arc a incorporating the decision of both types of traffic. Specifically:

$$c_a^s(v_a, x_a^s) = \mu_n^a v_a x_a^s + \nu_n^a x_a^s, \quad \text{if} \quad K_{a,n}\le v_a < K_{a,n+1} \tag{10}$$

Note that equation (10) is the extended form of $c_a^s(v_a, x_a^s) = c_a(v_a)x_a^s$ with $c_a(v_a)$ replaced by its linear function approximation. When arc a is used by hazmat shipment s, i.e., $x_a^s = 1$, c_a^s equals the arc travel time, $\mu_n^a v_a + \nu_n^a$, otherwise with $x_a^s = 0$, and any regular traffic flow v_a, c_a^s equals 0.

To account for the piecewise-linear travel time function, the following equivalent mixed-integer linear constraints denoted by set (Π_1) must be added to (MR):

$$(\Pi_1): \begin{cases} L.\xi_{a,n} \le v_a - K_{a,n} \le U.(1-\xi_{a,n}) - \epsilon, \quad \forall a, \forall n, \\ \gamma_{a,n} = \xi_{a,n+1} - \xi_{a,n}, \quad \forall a, \forall n, \\ \xi_{a,n} \in \{0,1\}, \\ c_a^s \ge 0, \quad \forall s, \forall a, \\ L.(1-\gamma_{a,n}) \le c_a^s - (\mu_n^a v_a x_a^s + \nu_n^a x_a^s) \le U.(1-\gamma_{a,n}), \quad \forall s, \forall a, \forall n, \\ n = 1, \ldots, N, \\ a \in \mathcal{A}, \\ s \in \mathcal{S}. \end{cases}$$

In set (Π_1), L and U represent very large negative and very large positive constants, respectively, and ϵ is a very small positive constant. $\xi_{a,n}$ denotes a binary variable indicating whether v_a is greater than $K_{a,n}$ or not: $v_a \in [K_{a,n}, \infty)$ if $\xi_{a,n} = 0$; $v_a \in [0, K_{a,n})$ otherwise, which can be verified by the first constraint of set (Π_1). Another binary variable $\gamma_{a,n}$, defined as the difference of $\xi_{a,n}$ and $\xi_{a,n+1}$, determines whether v_a falls in the segment $[K_{a,n}, K_{a,n+1})$ or not: specifically, $\gamma_{a,n} = 1$ expresses the condition wherein $\xi_{a,n} = 0$ and $\xi_{a,n+1} = 1$, implying that $v_a \in [K_{a,n}, K_{a,n+1})$. For the other possible combinations of $\xi_{a,n}$ and $\xi_{a,n+1}$, we have $\gamma_{a,n} = 0$ which represents the situation $v_a \notin [K_{a,n}, K_{a,n+1})$.

When v_a lies in the region $[n]$, substituting the equivalent case, i.e., $\gamma_{a,n} = 1$, into the fifth constraint of set (Π_1), we have:

$$0 \le c_a^s - (\mu_n^a v_a x_a^s + \nu_n^a x_a^s) \le 0 \quad \Longleftrightarrow \quad c_a^s = (\mu_n^a v_a x_a^s + \nu_n^a x_a^s), \tag{11}$$

which produces the required simultaneous results in equations (9) and (10). Despite linearizing the travel time function $c_a(v_a)$, the transformation of the objective function in (9) still contains bilinear terms providing a mixed-integer bilinear program (BLP). Every bilinear term involves the product of a nonnegative continuous variable v_a and a binary variable x_a^s. Existing methods to deal

with general bilinear terms (i.e., the product of a nonnegative continuous and a nonnegative integer variable) are aimed at obtaining stronger relaxations, branching techniques, and heuristics within a spatial branch-and-bound framework. Recently, a different approach is proposed by Gupte et al. (2013) whose focus is on linearizing the bilinear terms and solving the BLP as a MILP. In doing so, they replace the integer variable with its binary expansion and then linearize the resulting product of continuous and binary variables. Although binary expansions are known to be inefficient, their method provides an exact MILP reformulation of BLP. From objective function (9), it is evident that having binary x_a^s obviates the need to use the binary expansion of the integer variables, and subsequently simplifies the transformation process. Hence, to linearize the bilinear terms we incorporate the Gupte et al. (2013) method which is described as follows:

Let $v_a \leq \bar{v_a}$ be the bounded continuous variable, for all arcs $a \in \mathcal{A}$. Also assume the bounded integer variable, for all arcs $a \in \mathcal{A}$ and shipments $s \in \mathcal{S}$, be $x_a^s \leq 1$. We define a new set of continuous variables w to replace with bilinear terms in (9). Specifically:

$$\{w_a^s : \quad w_a^s = v_a x_a^s, \quad v_a \leq \bar{v_a}, \quad x_a^s \leq 1, \quad \forall a \in \mathcal{A}, \quad \forall s \in \mathcal{S}\} \tag{12}$$

The above transformation is equivalent to the following linear constraints denoted by set (Π_2):

$$(\Pi_2): \begin{cases} w_a^s \leq \bar{v_a} x_a^s, \quad \forall s, \forall a, \\ w_a^s \leq v_a, \quad \forall s, \forall a, \\ w_a^s \geq v_a + \bar{v_a} x_a^s - \bar{v_a}, \quad \forall s, \forall a, \\ v_a \geq 0, \quad \forall a, \\ w_a^s \geq 0, \quad \forall s, \forall a, \\ x_a^s \in \{0,1\}, \quad \forall s, \forall a, \\ a \in \mathcal{A}, \\ s \in \mathcal{S}. \end{cases}$$

Now, replacing the bilinear term with its equivalent continuous variable w_a^s in the fifth constraint of set (Π_1) and in the objective function (9), the non-linearity of the problem is finally removed. That implies, combining all the results we can transform the non-linear (MR) problem in (1) to an equivalent MILP. For easy reference we put the whole formulation in one place as (MILP-1):

$$\text{(MILP-1)} \quad \min_{v,x,\xi,c,w} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} n^s \rho_a c_a^s$$

Subject to:

$$v \in V, \quad x \in X, \quad (\Pi_1), \quad \text{and} \quad (\Pi_2).$$

III Piecewise-Linear Model 2

Originally, Wang and Lo (2010) propose a region partitioning method for the nonlinear travel time function having two variables, namely arc traffic flow and arc capacity, and specify a linear

function within every region. Motivated by their idea, instead of separating, we suggest to consider the nonlinear term in problem (MR) as a single nonlinear function with two variables which can be linearized by piecewise-linear model 2.

For doing so, let us consider the nonlinear part in the objective function (1) denoted by $g_a^s(v_a, x_a^s)$. That is:

$$g_a^s(v_a, x_a^s) = c_a(v_a)x_a^s = \left(A_a + B_a(v_a)^4\right) x_a^s, \tag{13}$$

which is nonlinear with respect to decision variables v_a and x_a^s. Our goal is to represent this function (13) as a piecewise-linear function. We note that x_a^s is a binary variable, thus we adapt our method to consider for discrete levels of x_a^s as described in Wang and Lo (2010, Section 2.3.4).

In this transformation, the feasible domains of v_a (i.e., $[0, \bar{v_a}]$) and x_a^s (i.e., $\{0, 1\}$) are partitioned into N and M segments, respectively. Similar to model 1, for each link a, we choose a series of values of $K_{a,n}$ to partition the feasible domain of v_a into many small regions such that $0 < K_{a,n} < K_{a,n+1} < \bar{v_a}$, $n = 1, \ldots, N-1$. For the second variable, let m be the discrete number to be taken by x_a^s, i.e., 0 and 1, and M being the maximum value that m can take on. The feasible domain of x_a^s then becomes $\{m, \quad m = 0, \ldots, M\}$, and we denote $Q_{a,m}^s = m$. In partitioning the domain of x_a^s, for each link a and for each hazmat shipment s, a series of values of $L_{a,m}^s$ are chosen where $Q_{a,m}^s \leq L_{a,m}^s < Q_{a,m+1}^s$, for $m = 1, \ldots, M$, and each segment $[L_{a,m}^s, L_{a,m+1}^s)$ containing only one feasible value $Q_{a,m}^s$. Now, for each region $[n, m]$, where $K_{a,n} \leq v_a < K_{a,n+1}$, and $L_{a,m}^s \leq x_a^s < L_{a,m+1}^s$, we can specify a linear function approximating the non-linear function (13) within it. To account for such a piecewise-linear function with $N \times M$ regions, the following mix-integer linear constraints denoted by set(Π_3) are required:

$$(\Pi_3): \begin{cases} L\xi_{a,n} \leq v_a - K_{a,n} \leq U(1-\xi_{a,n}) - \epsilon, \quad \forall a, \forall n, \\ \gamma_{a,n} = \xi_{a,n+1} - \xi_{a,n}, \quad \forall a, \forall n, \\ L\psi_{a,m}^s \leq x_a^s - L_{a,m}^s \leq U(1-\psi_{a,m}^s) - \epsilon, \quad \forall s, \forall a, \forall m \\ \kappa_{a,m}^s = \psi_{a,m+1}^s - \psi_{a,m}^s, \quad \forall s, \forall a, \forall m, \\ L(1-\kappa_{a,m}^s) \leq x_a^s - Q_{a,m}^s \leq U(1-\kappa_{a,m}^s), \quad \forall s, \forall a, \forall m, \\ \theta_{n,m}^{s,a} = \gamma_{a,n} + \kappa_{a,m}^s, \quad \forall s, \forall a, \forall n, \forall m, \\ L(2-\theta_{n,m}^{s,a}) \leq g_a^s - (\mu_{n,m}^{s,a} v_a + \nu_{n,m}^{s,a}) \leq U(2-\theta_{n,m}^{s,a}), \quad \forall s, \forall a, \forall n, \forall m, \\ \xi_{a,n}, \psi_{a,m}^s \in \{0,1\}, \quad \forall s, \forall a, \forall n, \forall m, \\ g_a^s \geq 0, \quad \forall s, \forall a \\ a \in \mathcal{A}, \\ s \in \mathcal{S}, \\ n = 1, \ldots, N, \quad m = 1, \ldots, M. \end{cases}$$

where using the first order Taylor expansion we obtain:

$$\mu_{n,m}^{s,a} = \left.\frac{\partial g_a^s(v_a, x_a^s)}{\partial v_a}\right|_{K_{a,n}, Q_{a,m}^s} = 4B_a(v_a)^3 x_a^s\big|_{K_{a,n}, Q_{a,m}^s} = 4B_a(K_{a,n})^3 Q_{a,m}^s, \quad \text{and} \tag{14}$$

$$\nu_{n,m}^{s,a} = g_a^s(K_{a,n}, Q_{a,m}^s) - K_{a,n}\left.\frac{\partial g_a^s(v_a, x_a^s)}{\partial v_a}\right|_{K_{a,n}, Q_{a,m}^s}. \tag{15}$$

Following the logic as in set (Π_1), here for x_a^s, $\psi_{a,m}^s$ and $\kappa_{a,m}^s$ play the same roles as $\xi_{a,n}$ and $\gamma_{a,n}$ do for v_a, respectively. In simple words, $\kappa_{a,m}^s = 1$ indicates $L_{a,m}^s \leq x_a^s < L_{a,m+1}^s$, and 0 otherwise. Substituting the case of $\kappa_{a,m}^s = 1$ into the fifth constraint of set (Π_3), we have $0 \leq x_a^s - Q_{a,m}^s \leq 0 \Longleftrightarrow x_a^s = Q_{a,m}^s$.

The sixth and the seventh constraint of (Π_3) approximate the non-linear function (13) by piecewise-linear functions within every region $[n, m]$. Specifically, if both $\gamma_{a,n}$ and $\kappa_{a,m}^s$ equal 1, then $K_{a,n} \leq v_a < K_{a,n+1}$, and $L_{a,m}^s \leq x_a^s < L_{a,m+1}^s$, giving $\theta_{n,m}^{s,a} = \gamma_{a,n} + \kappa_{a,m}^s = 2$. Substituting the case of $\theta_{n,m}^{s,a} = 2$ into seventh constraint, we obtain:

$$0 \leq g_a^s - (\mu_{n,m}^{s,a} v_a + \nu_{n,m}^{s,a}) \leq 0 \Longleftrightarrow g_a^s = \mu_{n,m}^{s,a} v_a + \nu_{n,m}^{s,a}. \tag{16}$$

We note that in (16), the linear function is specified solely with respect to v_a as the value of x_a^s is set equal to $Q_{a,m}^s$ within region $[n, m]$.

Consequently, the second piecewise linearization approach transforms the nonlinear (MR) problem to another equivalent MILP as denoted by (MILP-2):

$$\text{(MILP-2)} \quad \min_{v,x,\xi,\psi,g} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} n^s \rho_a g_a^s$$

Subject to:

$$v \in V, \quad x \in X, \quad (\Pi_3).$$

IV Computational Experiments

In this subsection, we carry out experiments on the test problems characterized in Table 2 of the main paper. Although our main purpose is to evaluate the performance of the Modified Frank-Wolfe algorithm in solving problem (MR), we spend a considerable amount of effort to study the quality of the solutions provided by the two linearization approaches for comparison purposes.

We first present our numerical results for Modified Frank-Wolfe algorithm. For better exploration in the solution space, we solve (MR) for each problem instance using multiple starting feasible solutions. That is, for every initial feasible solution, the Modified Frank-Wolfe Algorithm iterates until convergence, i.e., the deviation from the optimal solution is below ϵ, each resulting in a Min-Risk flow with a corresponding Min-Risk objective value. Here, for all the experiments, we use similar termination criterion to that defined in Fukushima (1984) with ϵ equal to 0.1. Finally, the Min-Risk flow solution is the one whose MR objective value is the least among all other solutions.

Table A.1: Frank-Wolfe Algorithm-computational results

Test Problem	No. of Starting Feasible Solutions	MR Value	Total Run Time (sec)	Avg. Run Time	Max. Absolute % Deviation
8-node	5	81082.3561	2	**0.4**	0%
	10	81082.3561	4	0.4	
	50	81082.3561	13	0.26	
	100	81082.3561	24	0.24	
10-node	5	1.629975E8	4	**0.8**	0%
	10	1.629975E8	6	0.6	
	50	1.629975E8	27	0.54	
	100	1.629975E8	57	0.57	
15-node	5	500975.5646	98	19.6	1.8%
	10	492075.3572	227	**22.7**	
	50	492075.3572	883	17.66	
	100	492075.3572	1674	16.74	

We present our results in Table A.1.

In Table A.1, the third column represents the best Min-Risk value we found for (MR) problem among all solutions obtained from each initial feasible point. In contrast is the worst Min-Risk solution whose objective value is the highest. Although we are mainly interested in the best Min-Risk solution, to gain more insight over the performance of the heuristic and the variance of the provided solutions, we also keep track of the worst solution for all the experiments. Specifically, it is worthwhile to investigate the effect of network congestion on the efficiency of the algorithm according to its computational time and the difference between the best and worst solutions found by the heuristic. For doing so, a similar set of experiments as Table A.1 are conducted with a much smaller trip table for regular vehicles, i.e., a few OD pairs and less OD demand values, representing no-congestion case. Our observations reveal that without congestion, the (MR) problem becomes much simpler to solve, and the average required computation time per each problem instance is 0.06, 0.1, and 0.2 (sec) for 8-node, 10-node and 15-node problems, respectively. Moreover, the absolute percent deviation from the best solution to the worst one, for all problem instances, does not exceed above %2.8e-5 which implies that the heuristic algorithm provides quite stable solutions regardless of the starting point. Mathematically, one explains that when the volume of regular traffic, v_{ij}, is very low comparing to the arc capacity, l_{ij}, the BPR function ($c_{ij}(v_{ij}) = c^0_{ij}(1 + 0.15(v_{ij}/l_{ij})^4)$ lends itself to a linear function (the second term in the BPR function denoting the congestion can be ignored), improving the speed and the quality of the Modified Frank-Wolfe algorithm. Although such results are desirable, assuming no congestion does not provide a realistic representation of a network; hence does not express the performance of the heuristic effectively. For this reason, the results of the latter set of experiments are not reported.

For the highly congested network, i.e., Table A.1, the (MR) problem is still solved very quickly by the heuristic. One observes that the maximum computation time per problem instance, i.e.,

Table A.2: Solution performance of the Piecewise-Linear Model 1 under different discretization schemes

Test problem	Partition Scheme	MILP-1 Value	MR Value	%Gap between MILP-1 & MR	Optimality Gap(%)	Run Time(sec)
8-node	20	81076.0608	81077.1944	0.001	39.64	>3600
	50	81059.9759	81077.2050	0.022	0	110.96
	100	81049.2227	81134.4724	0.105	0	4.2
	150	80952.0046	81158.4546	0.254	0	1.69
10-node	10	1.629871E8	1.629871E8	0	0.05	> 3600
	50	1.618353E8	1.629871E8	0.706	0	19.02
	100	1.618119E8	1.630267E8	0.745	0	5.6
	150	1.570855E8	1.630267E8	3.644	0	2.73
15-node	100	485425.0481	486838.8639	0.290	80.73	>3600
	150	486262.5252	488137.6503	0.384	57.65	>3600
	200	484758.9999	489076.2424	0.882	42.49	>3600
	300	484758.9999	518306.6393	6.472	20.71	>3600

bold-type cases in the fifth column, are 0.4, 0.8, and 22.7 (sec) for 8-node, 10-node and 15-node problems, respectively. The reason behind different average computation times for each network is the running time to convergence is affected by the quality of the initial solution when applying Modified Frank-Wolfe to the (MR) problem. We also note that other than congestion which increases the time to convergence, the network size has a similar impact on the computation effort required by the algorithm. Another interesting observation is that generating more starting feasible solutions does not change the quality of the Min-Risk objective value considerably. This has been better shown in the last column by maximum absolute % deviation among the best solutions found by different numbers of starting points. For the 8-node and 10-node networks, the best Min-Risk value found among only 5 solutions is as good as the one found using 100 starting points. In the 15-node network, there is only 1.8% improvement in the solution quality as we double the number of starting points. From this, we can conclude that the algorithm is robust and the quality of the final solution does not depend on the initial point. Although we noticed that the % deviation between the worst and best case solutions in the congested network can be higher when compared with the no congestion case, we can still assure obtaining a high quality solution by generating sufficient starting points. So far we have shown that the Modified Frank-Wolfe algorithm is robust and computationally efficient. Nevertheless, further experiments need to be implemented in order to prove its convergence to optimality.

We now evaluate the performance of the two linearization approaches whose solutions are guaranteed to be globally optimal. In particular, we solve (MILP-1) and (MILP-2) problems for the test networks illustrated in Table 2 of the main paper.

Tables A.2 and A.3 depict a summary of the linearization results to our problem instances. Note that our pre-defined partitioning scheme has been shown in the second column with each

value denoting the length of each segment. Due to different nature of the problem instances, we applied different discretization resolutions to each in order to study the solution performance of the linearization approaches. For comparison purposes, in both tables, we show the final objective value obtained directly from the MILP-1/2 (i.e., the piecewise-linear approximation) in the third column, and the MR objective value determined from the corresponding MILP flow solutions, i.e., to remove the linearization error, in the forth column. Additionally, we include the gap between these two values indicating the accuracy level of the piecewise-linear approach in approximating the MR problem as well as the corresponding calculation time and the achieved optimality gap. For each network, we start from implementing a fine resolution and we gradually decrease it.

As one can predict, in Table A.2, for each problem instance, the finer scheme generally takes longer calculation times but yields better approximation results, leading to very small gaps (i.e., note the descending computation times versus the ascending gaps). Besides, the exact MR objective values are generally better in the finer schemes. In contrast, applying coarser resolutions produce coarser results and typically worse exact MR objective values than that of the finer schemes, ex., increasing it from 81077.1944 to 81158.4546 in Table A.2, 8-node network. The gap is also larger, denoting less accurate approximations. However, the calculation time reduces considerably.

The network size and the congestion level also affect the computation time of the MILP. Specifically, we notice that in Table A.2, despite the 8-node network is small, heavy demand makes it substantially congested; hence the computation time for MILP formulation is long, taking more than 10 hrs to reach to a small optimality gap when partition scheme is set to 20. Similarly, for the 15-node network, the computation of all applied partition schemes could not finish within the allotted 1 hr due to the problem size and high congestion while a finer resolution, i.e. scheme 10, with a lighter congestion in the 10-node network resulted in a sufficiently small optimality gap within the first 1 hr of calculation. Note that the cases with calculation time >3600 imply the instances where the optimality gap does not reach to zero within the first hr of run; thus the reported values are the ones corresponding to the instant of 1 hr. To ascertain solution's global optimality, we let the CPLEX run for a very long time. It appears that for all the cases, the MILP can produce quality solutions quite fast and then it spends time on improving the optimality gap. For example, from the conducted experiments for the 8-node network with partition scheme 20, we observe the reported solution in the table at the instant of 256 with optimality gap 50.86%. Afterwards, the solution does not change while the optimality gap reduction continues. Subsequently, by nearly 8.1 hrs of computation, the MILP still shows the same solution as reported in the table, however the optimality gap is reduced to 23.53%. Similarly, in the 15-node network with scheme 100, the reported solution in the table is observed at instant 3379.32 sec when the optimality gap is 81.66%. Since then, running the algorithm for 10 hrs could only reduce the optimality gap to 55.08% without any significant improvement in the solution. Therefore, our observations confirm that the solution provided by the linearization approach possesses the desirable property of global optimality for each test problem. Moreover, it is evident from Table A.2 that there is a significant trade-off between the partition scheme and the optimality gap. Lower optimality gaps can

Table A.3: Solution performance of the Piecewise-Linear Model 2 under different discretization schemes

Test problem	Partition Scheme	MILP-2 Value	MR Value	(%)Gap between MILP-2 & MR	Optimality Gap(%)	Run Time(sec)
8-node	20	81076.0602	81077.1944	0.001	0	994.10
	50	81059.9756	81077.1944	0.021	0	50.09
	100	81049.2225	81139.6341	0.111	0	6.08
	150	80952.0047	81157.3831	0.253	0	1.03
10-node	10	1.629871E8	1.629871E8	0	0	3237.55
	50	1.618353E8	1.629871E8	0.706	0	18.92
	100	1.618119E8	1.630264E8	0.744	0	4.46
	150	1.570856E8	1.630265E8	3.644	0	1.54
15-node	120	485752.2987	487423.1966	0.342	62.50	>3600
	200	484758.9999	488602.7552	0.786	48.17	>3600
	320	484759.0000	513489.1986	5.595	18.32	>3600
	350	484759.0000	521498.5218	7.044	0	1666.07

be achieved at the expense of lower quality solutions when incorporating larger partition schemes, e.g., note the increasing partition schemes and the descending optimality gaps achieved in 1 hr for the 15-node network.

For Piece-wise Linear Model 2, we assume the feasible domain of x_a^s is partitioned into $[0, 1, 2)$ for all instances. Here, the same conclusions as Model 1 can be drawn according to our observations in Table A.3. Nonetheless, it is valuable to understand how the two linearization approaches are different using a comparative analysis. For doing so, we define a stopping criteria for the MILP solver; either a time limit or an optimality gap. Then, the MILP-1/2 are solved and return the solution upon reaching to the specified stopping rule. Note that for every problem instance, the adopted partition scheme is the one leading to the best MR objective value found (if multiple of such values exist, we choose the one which results in a smaller approximation gap) from Tables A.2 and A.3 for MILP-1 and MILP-2, respectively. Table A.4 compares the two lineariztion methods on the basis of the MILP objective value, MR objective value, approximation gap, and the time required to reach to a specified optimality gap.

In Table A.4, the choice of optimality gap varies in different problem instances and is typically higher as the network size increases. This setting is only due to time saving purpose as the computational effort required by MILP increases substantially in larger and/or congested networks. Nevertheless, in the 10-node network, we observe that light demands cause both linearization methods to reach to medium level optimality gaps very quickly making the comparison inefficient; hence in this case, we choose to set optimality gaps to critical low values.

In the 8-node network, at different levels of optimality gaps, both MILP-1 and MILP-2 perform quite similarly in terms of linear approximation objective value, MR objective value, and approximation gap whereas the computation time is significantly reduced under Piecewise Linear Model

Table A.4: Solution performances of MILP-1 vs. MILP-2 under a specified optimality gap. Run times are in seconds unless stated otherwise.

		Piecewise Linear Model 1				Piecewise Linear Model 2			
Test Problem	Opt. Gap	MILP-1	MR	Gap [a] (%)	Run Time	MILP-2	MR	Gap[b] (%)	Run Time
8-node	40%	81076.06	81077.19	0.001	3477.61	81076.06	81077.19	0.001	329.44
	20%	81076.06	81077.19	0.001	8.5 ‡	81076.06	81077.19	0.001	834.53
	0%	81076.06	81077.19	0.001	12.8 ‡	81076.06	81077.19	0.001	994.10
10-node	10%	1.7705E8	1.7705E8	0	88.94	1.6671E8	1.6671E8	0	271.55
	5%	1.6871E8	1.6871E8	0	121.09	1.6671E8	1.6671E8	0	272.11
	0%	1.6298E8	1.6298E8	0	9.87 ‡	1.6298E8	1.6298E8	0	3237.55
15-node	80%	485425.05	486838.86	0.290	1.02 ‡	491836.98	492650.43	0.165	1207.12
	70%	485420.05	486763.65	0.276	2.65 ‡	485913.25	487330.62	0.290	1551.68
	50%	485420.05	486763.65	0.276	27.07 ‡	485750.01	487160.38	0.289	5.14 ‡

[a]Denotes the gap between the MR objective value and its approximation by MILP-1.
[b]Denotes the gap between the MR objective value and its approximation by MILP-2.
‡Represents the cases with run time measured in hours.

2. Also, in the 15-node network, one can see that MILP-2 formulation always takes shorter computational times but yields larger MR objective values when compared to the MILP-1 alternative formulation.

The two MILPs take a different approach in obtaining the optimal solution for the 10-node network. Specifically, we observe that MILP-1 reaches to low level of optimality gap, i.e., 10% quickly, but it can not make the same improvement in the solution quality. In contrast, our experiments under MILP-2 show that the time to attain optimality gap 10% is longer during which the algorithm can better improve the solution as well. However, achieving 5% of optimality gap reverses this trend in both MILPs. That is, for MILP-1 it takes about 32 sec (notice the corresponding computational times for optimality gaps 5% and 10% in Table A.4) to reduce the gap from 10% to 5% during which it is also able to markedly enhance the solution quality. In contrary, the same amount of progress in optimality gap can be achieved in less than a second by MILP-2, but leading to no reduction in the MR objective value. Finally, upon reaching to optimality gap 0%, the two MILPs produce exactly the same results except that making such a small reduction in optimality gap becomes computationally very demanding in lineariztion model 1, i.e., it takes roughly 9.8 hrs for MILP-1 to reach from 5% to 0%. Despite such different approach in the 10-node network, one can notice that upon converging to the optimal solution, the computational performances of the Piecewise Linear Model 1 and 2 display the similar trend as they do in the other two networks. That is, Piecewise Linear Model 1 always takes longer computational times to converge to the optimal solution when compared to Model 2.

In Table A.5, we present a comparison of the two lineariztion methods in terms of MILP objective value, MR objective value, approximation gap, and the optimality gap obtained when the allotted

Table A.5: Solution performances of MILP-1 vs. MILP-2 under a specified run time. Run times are in seconds.

		Piecewise Linear Model 1				Piecewise Linear Model 2			
Test problem	Run Time	MILP-1	MR	Gap[a] (%)	Opt. Gap(%)	MILP-2	MR	Gap[b] (%)	Opt. Gap(%)
8-node	500	81076.06	81077.19	0.001	50.47	81076.06	81077.19	0.001	30.81
	1500	81076.06	81077.19	0.001	46.10	81076.06	81077.19	0.001	0.0[c]
10-node	500	1.6298E8	1.6298E8	0	0.05	1.6302E8	1.6302E8	0	0.78
	1500	1.6298E8	1.6298E8	0	0.05	1.6298E8	1.6298E8	0	0.75
15-node	500	495531.99	495593.81	0.012	97.77	485752.30	487437.67	0.345	82.97
	1500	495531.99	495593.81	0.012	90.41	485752.30	487370.37	0.332	70.80
	7200	485420.05	486692.16	0.261	72.20	485752.30	487191.13	0.295	56.16

[a]Denotes the gap between the MR objective value and its approximation by MILP-1.
[b]Denotes the gap between the MR objective value and its approximation by MILP-2.
[c]The MILP reached to the 0% optimality gap earlier than the time limit and stopped before the termination criteria.

time for running the MILPs is limited to a certain value.

The observations from Table A.5 are consistent with our previous conclusions regarding the performance of the two linearization methods. Considering the smallest network, the MILP-2 performs similar to MILP-1 in terms of solution quality but it requires less computational effort (better optimality gaps are achieved in the allotted run time using MILP-2). However, as the network grows, one can notice that MILP-1 produces a better MR objective value than that of MILP-2 during a certain run time, with the exception of the 15-node network (when the time limit is 500 sec or 1500 sec). Our numerical results for the 15-node network show that MILP-1 can not significantly improve the solution quality in the first hour of run during which the other linearization method is able to find better solutions. However, when the calculation time increases to 2hrs, MILP-1 generates more attractive solutions as it does in general.

Appendix B

Recall that the true bilevel dual-toll pricing problem, i.e., (GTS), constitutes the administrator's problem of the following form:

$$\min_{v\in V, x\in X, \tau\geq 0, t\geq 0} z(v,x,\tau,t) = \sum_{s\in\mathcal{S}}\sum_{(i,j)\in\mathcal{A}} n^s c_{ij}(v_{ij})\rho_{ij}x_{ij}^s \tag{17}$$

$$+\varphi\Big(\sum_{(i,j)\in\mathcal{A}}\tau_{ij}v_{ij} + \sum_{s\in\mathcal{S}}\sum_{(i,j)\in\mathcal{A}} n^s t_{ij}x_{ij}^s\Big), \tag{18}$$

in the upper level, and problems (R) and (H) in the lower level. To evaluate the benefits of solving problem (GTS) by the 2-Step EDO heuristic versus the Modified EDO, we present results of our experiments on the test problems characterized in Table 2 of the main paper as well as Test network 1 in Suwansirikul et al. (1987, Figure 1). For benchmarking, we also report the results of the LINGO approach whose solution posses the desirable property of local or global optimality. LINGO, produced by LINDO Systems Inc., includes a set of built-in solvers to tackle a wide variety of linear, nonlinear, and integer problems. For nonlinear models, it applies various techniques to obtain locally or globally optimal solutions. In our experiments, we used Extended LINGO 11.0 which can handle unlimited number of constraints and variables. Table B.1 depicts a summary of our comparative analysis results. As a companion, Table B.2 describes the details of the network for each case.

For each test network, we consider multiple levels of regular and hazmat OD pair to capture different regular traffic congestion and hazmat travel demand. Increasing the number of OD pairs generates more constraints and consequently increases the computational effort needed to arrive at either a global or a local solution to the problem. We limit our attention to situations where the LINGO solver can find either a global or a local solution within 24 hours. For simplicity, all arcs are tollable without a specified upper limit. For this reason, in both heuristics the initial lower and upper bounds on the arc tolls are set to zero and a very large number, respectively.

We note that (GTS) is a bi-objective problem with two (possibly) conflicting terms, i.e., risk and revenue. There is typically no unique solution to a Multi-Objective optimization problem, and most often multiple optimal solutions can be obtained by improving one objective and deteriorating the other. A full scale Multi-Objective analysis is not considered. Instead we numerically investigate the efficiency of the heuristics building our comparisons based upon the total objective function values (18) assuming $\varphi = 1$. An important result not explicitly presented in the table is that for some cases there exists a slight trade-off between the obtained risk and revenue values. These differences are quite acceptable as long as their corresponding total objective function values are close.

As demonstrated by the objective gap, the dual-toll policy obtained from 2-Step EDO algorithm yields nearly identical total objective values for most of the cases when compared with LINGO. For the Modified EDO heuristic, the solution diverges from LINGO in almost all cases (except for the 10-node test problem), implying that the 2-Step EDO outperforms the Modified EDO in terms of the solution efficiency. Note the dramatically different and erroneous solution calculated by Modified EDO in cases 1 and 3 for the 4-node network as it is in all cases for the 8-node network as well as cases 23 and 26 for the 15-node network.

Our experiments demonstrate that for larger problems the solution status using LINGO remains either local or unknown after many hours. Nonetheless, to further test the proposed heuristics for more complicated networks, among such cases (using ‡) we report those where a local optimal solution is obtained by LINGO after 24 hours. It is evident from Table B.1 that unlike the Modified EDO, the 2-Step EDO algorithm yields, in a reduced computing time, either nearly identical local

optimal solutions, ex., cases 9 and 26 , or more attractive solutions, ex., cases 7, 8, 10 and 11, when compared to LINGO.

We conclude that the LINGO approach is computationally very demanding and infeasible for any but very small problems when compared to either of the heuristics, and its computation time exponentially grows with the network size, congestion and its complexity. For both heuristics, we observe that most of the computational effort is in solving the UE problem (R) using the Frank-Wolfe algorithm. It is also obvious that the Modified EDO algorithm, despite generating low quality solutions, consistently requires fewer iterations of Golden Section to satisfy the same stopping tolerance (Δ =0.2), indicating that, in general, it dominates the 2-Step EDO in terms of computational efficiency. Nonetheless, we notice that there exists cases, ex., 7, 9 and 16, wherein the trend is reversed. This occurs when less Frank-Wolfe iterations are undertaken by the 2-Step EDO to obtain the UE solution.

We note from Table B.1 that there are zero hazmat revenue terms for all the cases. Thus the toll revenue reflects toll collected from regular vehicles. We observe that, in majority of the cases, the most risky arcs are subject to toll, and hazmat carriers can always take a detour without paying tolls due to the size of the network. Here, hazmat tolls serve to discourage the carriers from using the risky arcs, misleading one to conclude that the dual-toll policy acts as the road ban policy which completely prohibits hazmat carriers from the use of road segments. We now develop an illustrative example to demonstrate that the dual-toll policy is in fact not restrictive, and can produce flexible solutions to the carriers.

For tractability, consider Test network 1 of Suwansirikul et al. (1987) with the data being as in Tables B.3 and B.4. Let arcs 1, 2 and 3 be subject to toll charges and assume that the upper bounds on regular tolls and hazmat tolls are 50 and 100, respectively. From Table B.4 we note that only one route exists for hazmat shipments 1 and 3, whereas the hazmat carriers of shipment 2 can be directed to either of the two existing routes (i.e., route I: arc 2 or route II: arcs 1 and 3) by the network administrator. For this test network, problem (GTS) is solved using the 2-Step EDO algorithm under two different cases. In case 1, we let the population of arc 3 be at its original value, i.e., $\rho_3 = 200$. As shown in Table B.5, in this case, the minimum value of the objective function (18) is obtained when there is no toll charged to hazmat carriers. In fact, due to parameter values of each arc and the congestion stemming from the regular vehicles in arc 2, the route made up of arcs 1 and 3 is not only less risky when compared to arc 2 but also is shorter in terms of travel time, and consequently is selected by shipment 2. This property obviates the need to charge any toll as the decision of hazmat carriers follows the administrator's desired solution.

Now let the arc 3 population increase to 600 for case 2. The UE solution remains nearly identical to case 1 maintaining the same relation between the travel times of routes I and II. That is, with no hazmat tolls, shipment 2 still tends to select route II. However, with the excessive population exposure in arc 3, route II now becomes very risky. To redirect the carriers of shipment 2 while considering their own preferences, a toll is charged on arc 1, indicating that now route I becomes shorter in terms of travel time, and is selected by carriers of shipment 2. Note the routes undertaken

Case	LINGO				Modified EDO				2-Step EDO			
	Solution Type	Risk	Toll Revenue[a]	Run Time	Risk	Toll Revenue[a]	Run Time	Objective Gap (%)[b]	Risk	Toll Revenue[a]	Run Time	Objective Gap (%)[b]
1	Global	2469.86	0	4 sec	2945.94	703.28	8 sec	47.75	2469.86	1.96	14 sec	0.08
2	Global	7150.91	1472.92	4 sec	8303.92	820.45	12 sec	5.80	7322.19	1401.18	32 sec	1.15
3	Global	3389.86	0	4 sec	3857.11	1366.07	9 sec	54.08	3389.86	1.96	14 sec	0.06
4	Global	8070.92	1472.92	15 sec	8766.06	1090.62	32 sec	3.28	8444.92	1278.42	89 sec	1.88
5	Global	10707.51	486.08	24 sec	10718.75	1187.49	64 sec	6.37	10394.49	1307.95	114 sec	4.55
6	Global	80913.27	0	15 min	173736.15	444.49	47 sec	115.27	80914.86	0	77 sec	0^+
7	Local	163788.1	2634.44	24 hrs	490756.52	2164.03	162 sec	196.19	81026.94	181.41	83 sec	‡
8	Local	124858.4	15223.15	24 hrs	536943.91	39936.78	110 sec	311.82	81102.14	16952.11	95 sec	‡
9	Local	113760.7	0	24 hrs	761750.14	92538.63	1069 sec	650.95	114085.94	127.87	971 sec	0.40
10	Local	118068.7	456.42	24 hrs	239783.27	3240.46	206 sec	105.04	116180	7.05	311 sec	‡
11	Local	273323.5	141034.44	24 hrs	568796.57	4804.51	60 sec	38.43	368686.4	0	51 sec	‡
12	Global	3.96974E7	5518.4293	1 min	3.97155E7	0	57 sec	0.031	3.96808E7	22219.54	45 sec	0^+
13	Global	1613.44E5	223.34	11 min	1616.23E5	666.07	26 sec	0.17	1616.21E5	0	46 sec	0.17
14	Global	1626.21E5	15548.94	13 min	1629.97E5	119246.93	22 sec	0.29	1629.97E5	119253.4	63 sec	0.29
15	Global	1632.15E5	408807	8 hrs	1634.4E5	205113.8	39 sec	0.01	1634.18E5	311824.2	886 sec	0.06
16	Global	3971.52E4	83.89	12 min	3973.20E4	429.13	62 sec	0.04	3969.61E4	19205.82	47 sec	0^+
17	Global	1613.82E5	164553.48	19 min	1616.45E5	196.32	22 sec	0.06	1616.35E5	229.82	52 sec	0.05
18	Global	1626.98E5	14582.40	14 min	1630.20E5	119246.93	23 sec	0.26	1631.07E5	56586.68	74 sec	0.28
19	Global	1630.24E5	0	16 min	1634.64E5	205315.56	37 sec	0.40	1634.37E5	240637.8	1433 sec	0.40
20	Global	5669.20E4	0	2 min	5672.79E4	0	71 sec	0.06	5670.43E4	3.30	44 sec	0.02
21	Global	2305.34E5	164553.48	16 min	2308.72E5	12055.26	24 sec	0.08	2309.83E5	14752.79	49 sec	0.13
22	Global	2319.45E5	316089.75	10 min	2324.62E5	119245.80	24 sec	0.14	2325.87E5	56586.76	70 sec	0.16
23	Global	100196.95	439.15	5 min	169497.65	390.38	67 sec	68.81	108656.77	593.14	320 sec	8.56
24	Local	143418.95	81168.55	24 hrs	174183.54	593.95	130 sec	‡	171064.71	1636.52	189 sec	‡
25	Local	190541.59	55318.41	24 hrs	180316.26	9616.55	132 sec	‡	169340.49	6420.99	452 sec	‡
26	Local	314151.08	14053.72	24 hrs	399844.79	58981.25	88 sec	39.80	328904.86	536.18	295 sec	0.38

Table B.1: Performance of the 2-Step EDO vs. Modified EDO and LINGO. Table B.2 provides additional information for each case.

[a] Denotes the total toll collected from regular and hazmat users. However, hazmat toll revenue is zero in all the cases.

[b] Indicates the gap between the total objective function value (18) of the corresponding heuristic and LINGO.

‡Indicates the cases where the corresponding heuristic provides better solutions than the local optima found by LINGO.

Table B.2: Details of the understudy network for each case of Table B.1. Additional information is found in Figure B.1 and Tables B.3 and B.4 (4-node); Tables C.1 and C.2 in the supplementary document (8-node); Tables C.3 and C.4 (10-node); and Tables C.5 and C.6 (15-node).

Test		No. of OD pairs		Test		No. of OD pairs	
Network	Case	Regular	Hazmat	Network	Case	Regular	Hazmat
4-node				10-node (*cont.*)			
	1	4	1		14	30	3
	2	6	1		15	39	3
	3	4	2		16	10	5
	4	6	2		17	20	5
	5	6	3		18	30	5
8-node					19	39	5
	6	5	2		20	10	8
	7	15	2		21	20	8
	8	20	2		22	30	8
	9	27	2	15-node			
	10	5	4				
	11	10	4		23	5	2
10-node					24	10	2
	12	10	3		25	15	2
	13	20	3		26	5	4

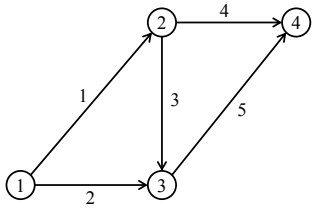

Figure B.1: An illustrative example: Test network 1 of Suwansirikul et al. (1987)

Table B.3: Arc attributes for the illustrative 4-node example in Figure B.1, with ρ_a: the population density along arc a and $c_a(v_a) = A_a + B_a(v_a/l_a)^4$

Arc a	A_a	B_a	l_a	ρ_a
1	4	0.6	40	200
2	4	0.3	50	150
3	6	0.9	40	200
4	5	0.75	40	400
5	3	0.45	40	250

Table B.4: Trip information for the illustrative example

	Regular vehicles						Hazmat trucks		
Trip information[a]	R1	R2	R3	R4	R5	R6	S1	S2	S3
Origin	1	2	2	1	1	3	1	1	2
Destination	4	3	4	3	2	4	2	3	3
Demand	50	60	40	200	45	70	4	5	4

[a]Ri and Sj represent regular trip i and hazmat shipment j, respectively.

Table B.5: Comparison of results for the illustrative example under the two cases

	Case 1. Arc 3 population = 200	Case 2. Arc 3 population = 600
Regular toll vector	(23.64,0,23.49,0,0)	(20.78,0,20.78,0,0)
Regular flow vector	(95,200,60,90,70)	(97,198,62,90,70)
Hazmat toll vector	(0,0,0,0,0)	(41.57,0,0,0,0)
Hazmat route choice	S1: (1,0,0,0,0) S2: (1,0,1,0,0) S3: (0,0,1,0,0)	S1: (1,0,0,0,0) S2: (0,1,0,0,0) S3: (0,0,1,0,0)
Risk	60576.83	105032
Revenue (Regular, Hazmat)	(3656, 0)	(3310,166)

by shipment 2 for cases 1 and 2 in Table B.5.

As mentioned by Marcotte et al. (2009), one disadvantage of the toll problem is that tolls are not necessarily set on risky arcs, and consequently may not be fair to all carriers. In this example, although arc 3 is considered as more risky, arc 1 is being charged and all the hazmat carriers cost is incurred by shipment 1 who does not go through the populated (risky) area. Indeed, they pay tolls just to prevent another carrier (shipment 2) from using a risky route. Nonetheless, from the standpoint of risk mitigation, dual-toll pricing is still interesting since it produces more acceptable solutions to drivers taking their own decision into account and can effectively direct them to less risky routes.

Appendix C

Appendix C provides additional information about the networks introduced in Table 2 of the main paper which are used in our numerical test. For each instance, the initial data includes start nodes, end nodes and arc free flow travel time. In the trip table, i.e., Origin-Destination (OD) pairs and OD demands, we consider all possible paired permutations of nodes for the regular traffic case while for the hazmat we randomly generate a few shipments due to the low volume of this type of traffic. In addition, we use Census site information to better estimate the arc population whereas the arc capacity and regular OD demand are sampled from the established test problems data available

on the Transportation Network Test problems website (Bar-Gera 2013) to effectively account for the network congestion. Table C.1 contains the network topology and arc attributes for the 8-node network, whereas the regular and hazmat OD demand information is described in Table C.2. Similarly, each pair of Tables C.3 and C.4, and C.5 and C.6 has the equivalent information for the 10-node network and the 15-node network, respectively.

Table C.1: Arc attributes for the 8-node network, with ρ_a: the population density along arc a and $c_a(v_a) = A_a(1 + 0.15(v_a/l_a)^4)$.

Arc a				
Start	End	A_a	l_a	ρ_a
1	2	6	900	701
1	3	4	400	193
2	3	6	700	701
2	4	5	200	1039
2	5	4	100	193
3	5	4	250	800
4	5	4	100	1295
4	6	4	200	800
5	6	2	300	1036
5	7	6	250	1030
6	7	2	150	1036
6	8	4	350	1423
7	8	5	100	1236

Table C.2: Travel demand matrix for the 8-node network. All single numbers represent regular demand except pairs which show regular and hazmat demand, respectively.

	1	2	3	4	5	6	7	8
1		70	50	(50,4)	90	(290, 3)	300	300
2			100	80	40	(100,2)	70	(300,7)
3					10	25	(150,2)	400
4					150	120	200	50
5						100	(90,1)	80
6							120	60
7								50

Table C.3: Arc attributes for the 10-node network, with ρ_a: the population density along arc a and $c_a(v_a) = A_a(1 + 0.15(v_a/l_a)^4)$.

Arc a				
Start	End	A_a	l_a	ρ_a
1	2	9	500	1249
1	4	4	450	582
1	5	8	150	1044
1	6	8	100	1321
1	7	6	300	637
2	3	8	300	608
2	4	10	200	561
3	2	6	600	1488
3	4	7	350	862
4	5	7	100	1292
5	6	5	170	531
7	6	6	200	947
8	1	6	100	554
8	2	5	250	1166
8	7	6	190	1177
9	2	8	300	1264
9	8	8	250	1199
9	10	10	370	751
10	2	6	480	1146
10	3	10	600	834

Table C.4: Travel demand matrix for the 10-node network. All single numbers represent regular demand except pairs which show regular and hazmat demand, respectively.

	1	2	3	4	5	6	7	8	9	10
1		40	100	(50,2)	120	30	140			
2			(40,2)	400	130	30				
3		200		100	(60,3)	300				
4					100	20				
5						(60,4)				
6										
7						90				
8	120	250	(70,1)	110	180	300	140			
9	30	50	45	70	(100,4)	(90,3)	80	110		130
10		40	60	45	200	(300,7)				

Table C.5: Arc attributes for the 15-node network, with ρ_a: the population density along arc a and $c_a(v_a) = A_a(1 + 0.15(v_a/l_a)^4)$.

Arc a				
Start	End	A_a	l_a	ρ_a
1	2	11	500	431
1	3	12	700	821
2	3	3	300	808
2	4	5	300	1171
3	2	5	200	828
3	4	4	400	778
3	5	1	400	1278
4	5	6	100	2196
4	6	4	500	1004
4	7	8	450	1255
5	6	1	50	4576
6	7	3	150	4576
6	8	5	350	1074
6	9	3	250	967
6	11	10	800	1026
7	8	1	600	4576
8	9	3	450	1163
9	10	5	200	492
9	11	7	350	978
9	12	12	150	947
10	12	5	250	205
10	13	2	100	890
10	15	8	50	1082
11	12	1	300	205
12	13	3	500	813
13	14	5	300	85
14	15	1	250	85
15	14	3	300	957

Table C.6: Travel demand matrix for the 15-node network. All single numbers represent regular demand except pairs which show regular and hazmat demand, respectively.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		100	40	20	10	140	110	90	80	170	50	(20,7)	120	(300,3)	300
2			60	50	100	90	40	150	120	60	20	100	(300,7)	250	240
3		10		50	50	60	80	400	100	200	45	35	50	30	(100,4)
4					85	40	10	60	(50,3)	100	30	10	300	60	100
5						25	45	50	30	200	150	100	65	80	90
6							10	40	25	80	45	70	45	60	50
7								100	25	30	50	100	120	20	35
8									90	150	(100,2)	50	70	120	140
9										30	40	50	10	20	200
10												300	20	300	25
11												50	40	80	60
12													50	40	100
13														45	50
14															60
15														120	

References

Bar-Gera, H. 2013. Transportation network test problems. http://www.bgu.ac.il/~bargera/tntp/. Accessed on May, 2013.

Fukushima, M. 1984. A modified frank-wolfe algorithm for solving the traffic assignment problem. *Transportation Research Part B: Methodological* **18**(2) 169–177.

Gupte, A., S. Ahmed, M. S. Cheon, S. Dey. 2013. Solving mixed integer bilinear problems using MILP formulations. *SIAM Journal on Optimization* **23**(2) 721–744.

Marcotte, P., A. Mercier, G. Savard, V. Verter. 2009. Toll policies for mitigating hazardous materials transport risk. *Transportation Science* **43**(2) 228–243.

Suwansirikul, C., T. L. Friesz, R. L. Tobin. 1987. Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transportation Science* **21**(4) 254–263.

Wang, D. Z., H. K. Lo. 2010. Global optimum of the linearized network design problem with equilibrium flows. *Transportation Research Part B: Methodological* **44**(4) 482–492.