Online Supplement to
Contextual Stochastic Vehicle Routing with Time Windows

Breno Serrano, Alexandre M. Florio, Stefan Minner, Maximilian Schiffer, and Thibaut Vidal

A. Feature projection function

The FPF is a function $\mathbf{f} : \mathbb{R}^p \times \Theta \times \mathcal{V} \setminus \{0\} \mapsto \mathbb{R}^{\bar{p}}$ that generates a $\bar{p}$-dimensional vector of projected features given a feature vector $\mathbf{x} \in \mathbb{R}^p$, a route $\theta \in \Theta$, a node $i \in \theta$, and travel time estimates $\hat{\mathbf{t}}$. The projected features are split into groups, as follows:

Table 3 Feature Projection Function: Predictors of the Penalty at Customer $i \in \theta$ Returned by $\mathbf{f}(\mathbf{x}, \theta, i; \hat{\mathbf{t}})$

Predictor	Group	Description
$x_1, \ldots, x_p$	(a)	Travel time covariates (original features)
e_i	(b)	Start of time window of customer $i \in \theta$
ℓ_i	(b)	End of time window of customer $i \in \theta$
$\ell_{\rho(i)}$	(b)	End of time window of customer $\rho(i) \in \theta$ that precedes $i \in \theta$
$c_{\rho(i),i}$	(b)	Transportation cost of the arc from $\rho(i)$ to i
$\hat{\sigma}^2_{\rho(i),i}$	(b)	Estimated variance in the travel time from $\rho(i)$ to i
k_i	(b)	Position of customer i along route θ
$a_\theta(i; \underline{\mathbf{t}})$	(c)	Lower bound on arrival time at customer $i \in \theta$ with free-flow travel times $\underline{\mathbf{t}}$
$s_\theta(i; \underline{\mathbf{t}})$	(c)	Lower bound on service start time at customer $i \in \theta$ with free-flow travel times $\underline{\mathbf{t}}$
$(a_\theta(i; \underline{\mathbf{t}}) - \ell_i)^+$	(c)	Lower bound on lateness
$\pi(a_\theta(i; \underline{\mathbf{t}}) - \ell_i)$	(c)	Lower bound on penalty
$\hat{t}_{\rho(i),i}$	(d)	Predicted travel time of the arc from $\rho(i)$ to i
$a_\theta(i; \hat{\mathbf{t}})$	(d)	Arrival times at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$s_\theta(i; \hat{\mathbf{t}})$	(d)	Service start time at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$(a_\theta(i; \hat{\mathbf{t}}) - \ell_i)^+$	(d)	Lateness at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$\pi(a_\theta(i; \hat{\mathbf{t}}) - \ell_i)$	(d)	Penalty at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$\xi_\theta(i; \hat{\mathbf{t}})$	(e)	Variability model of service start time at customer $i \in \theta$ (see Section A.1)

A.1. Variability of service start time

To derive penalty predictors that account for travel time variability and its impact on service start time, we introduce a measure of service start time variability, which we denote as *service start time risk*. Given a route θ and a customer $i \in \theta$, the distribution of $\max\{e_i, a_\theta(i; \tilde{\mathbf{t}})\}$ (the service start time at customer i) is, in general, truncated, because of possible early arrivals and waiting times at customer i and at other customers previously visited. Such a truncation decreases service start time variability; therefore, a covariate for service start time risk should consider both travel time variability and the likelihood of early arrivals along a route.

Let $\boldsymbol{\Sigma} = [\sigma_{ij,lm}]_{(i,j),(l,m)\in\mathcal{A}}$ be the travel times covariance, and let $\sigma_{ij}^2 = \sigma_{ij,ij}$. Given a route $\theta = (v_1, \ldots, v_L)$, we denote by $\xi_\theta(i)$ the service start time risk at customer $i \in \theta$, and consider the following risk propagation model:

$$\xi_\theta(v_k) = \begin{cases} (1 - \mathcal{P}_\theta(v_k))\sigma_{0v_k}^2, & \text{if } k = 1, \\ (1 - \mathcal{P}_\theta(v_k))(\xi_\theta(v_{k-1}) + \sigma_{v_{k-1}v_k}^2 + 2\sigma_{v_{k-2}v_{k-1},v_{k-1}v_k}), & \text{otherwise}, \end{cases} \tag{32}$$

where $\mathcal{P}_\theta(i) = \mathbb{P}(a_\theta(i;\tilde{\mathbf{t}}) < e_i \mid \mathbf{x}^{n+1})$ is the early arrival probability at customer $i \in \theta$ conditional on observed features $\mathbf{x}^{n+1}$, and $v_0 = 0$.

Model (32) propagates the variabilities of travel time and service start time along route θ as far as early arrival probabilities are low. When the early arrival probability $\mathcal{P}_\theta(i)$ is high, the service start time risk at customer i is low, since service occurs at instant e_i with high probability. In this case, the service start time risk at other customers along route θ following customer i also decreases. Note that the model accounts for travel time correlation between adjacent arcs in the network.

Service start time risk measures $\xi_\theta(i)$, $i \in \theta$, cannot be computed directly because the travel time distribution is unknown. Following our distribution-free approach, we estimate $\boldsymbol{\Sigma}$ and $\mathcal{P}_\theta(i)$ (and hence $\xi_\theta(i)$) from data. Let $\hat{\boldsymbol{\Sigma}} = [\hat{\sigma}_{ij,lm}]_{(i,j),(l,m)\in\mathcal{A}}$ be the estimated travel times covariance (as described in Section 3.4) and let $\hat{\sigma}_{ij}^2 = \hat{\sigma}_{ij,ij}$. In the remainder of this section, we discuss how to estimate $\mathcal{P}_\theta(i)$ for any route θ. To this end, we let $\mathbf{g} : \mathbb{R}^p \mapsto \mathbb{R}^{|\mathcal{A}|}$ be a travel time prediction model. Further, we assume that for each customer $i \in \mathcal{V} \setminus \{0\}$ a set of training routes Θ_i is available, where $i \in \theta$ for all $\theta \in \Theta_i$. This is an unrestrictive assumption as these training routes may be arbitrary routes, e.g., generated by solving other VRP models. Finally, let $\rho_\theta(v_k)$ be the node that precedes v_k in route $\theta = (v_1, \ldots, v_L)$, that is, $\rho_\theta(v_k) = 0$ if $k = 1$, and $\rho_\theta(v_k) = v_{k-1}$ if $k \geq 2$.

Given a vector $\mathbf{x}$ of travel time covariates, let $\mathbf{w}_{i,\theta}(\mathbf{x})$ be the vector of early arrival covariates, with components as described in Table 4. Clearly, if $e_i = 0$, then we have $\mathcal{P}_\theta(i) = 0$. Further, note that if $C_\theta > e_i$, then $\mathcal{P}_\theta(i) = 0$, since we assume that $\tilde{t}_{ij} \geq c_{ij}$. Finally, if $\max_{j\in\theta\setminus\{i\}}\{e_j\} > e_i$, then we have again $\mathcal{P}_\theta(i) = 0$.

Since our predicted quantity is a probability, a sensible learning model is a logistic regression. Let $S(z) = 1/(1+\exp(-z))$ be the logistic function, and let $\mathcal{L}_{\text{nll}}(a,b) = -(b\log a + (1-b)\log(1-a))$ be the negative log likelihood loss function. For each customer $i \in \mathcal{V} \setminus \{0\}$, we train the parameters $\hat{\phi}_i^0$ and $\hat{\boldsymbol{\phi}}_i \in \mathbb{R}^{p+2}$ of the logit model:

$$\hat{\phi}_i^0, \hat{\boldsymbol{\phi}}_i \in \underset{\phi_i^0\in\mathbb{R}, \boldsymbol{\phi}_i\in\mathbb{R}^{p+2}}{\arg\min} \quad \frac{1}{n|\Theta_i|}\sum_{k=1}^{n}\sum_{\theta\in\Theta_i} \mathcal{L}_{\text{nll}}\left(S(\phi_i^0 + \boldsymbol{\phi}_i^\top \mathbf{w}_{i,\theta}^k), \mathbb{I}(a_\theta(i;\mathbf{t}^k) \leq e_i)\right) + \lambda \left\|\boldsymbol{\phi}_i\right\|_1,$$

Table 4 Components of the early arrival covariates vector $\mathbf{w}_{i,\theta}(\mathbf{x})$ at customer $i \in \theta$ given feature vector $\mathbf{x}$

Predictor	Description
$x_1, \ldots, x_p$	Travel time covariates
e_i	Opening of time window
$a_\theta(i; \mathbf{g}(\mathbf{x}))$	Estimated arrival time at customer $i \in \theta$
$\hat{\sigma}^2_{\rho_\theta(i)i}$	Estimated variance in the travel time from $\rho_\theta(i)$ to i
$\max_{j \in \theta \setminus \{i\}}\{e_j\}$	Latest e_j along route θ before arriving at customer i
C_θ	Transportation cost of route θ up to customer i

where we use $\mathbf{w}^k_{i,\theta} \coloneqq \mathbf{w}_{i,\theta}(\mathbf{x}^k)$, λ is the regularization parameter and $\|\boldsymbol{\phi}_i\|_1$ is the ℓ_1 norm of $\boldsymbol{\phi}_i$. Regularization by the ℓ_1 norm leads to sparsity in the model parameters.

Hence, for any route θ we estimate the early arrival probability at customer $i \in \theta$ by

$$\hat{\mathcal{P}}_\theta(i) = S(\hat{\phi}^0_i + \hat{\boldsymbol{\phi}}_i^\top \mathbf{w}^{n+1}_{i,\theta}).$$

B. Proof of completion bounds

We provide proofs for the RCSP and knapsack bounds in the following.

Proof of Proposition 1. From the definition of the RCSP bound in Equation (20), we have:

$$\begin{aligned}
0 &\overset{\text{(a)}}{\leq} \overline{C}_\theta + \widehat{T}_{\text{RCSP}}(i, Q - q_\theta) \\
&\overset{\text{(b)}}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \pi(\delta_{\theta \oplus u_1} - \ell_{u_1}) - \gamma_{u_1} + c_{u_1 0} + \widehat{T}_{\text{RCSP}}(u_1, Q - q_{\theta \oplus u_1}) \\
&\overset{\text{(c)}}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \pi(\delta_{\theta \oplus u_1} - \ell_{u_1}) - \gamma_{u_1} + c_{u_1 0} \\
&\qquad - c_{u_1 0} + c_{u_1 u_2} + \pi(\delta_{\theta \oplus u_1 \oplus u_2} - \ell_{u_2}) - \gamma_{u_2} + c_{u_2 0} \\
&\qquad \ldots \\
&\qquad - c_{u_{L-1} 0} + c_{u_{L-1} u_L} + \pi(\delta_{\theta \oplus \mathcal{E}} - \ell_{u_L}) - \gamma_{u_L} + c_{u_L 0} + \widehat{T}_{\text{RCSP}}(u_L, Q - q_{\theta \oplus \mathcal{E}}) \\
&\overset{\text{(d)}}{=} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1} u_j} + c_{u_L 0} + \sum_{j=1}^{L} \Big(\pi(\delta_{\theta \oplus u_1 \oplus \ldots \oplus u_j} - \ell_{u_j}) - \gamma_{u_j} \Big) + \widehat{T}_{\text{RCSP}}(u_L, Q - q_{\theta'}) \\
&\overset{\text{(e)}}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1} u_j} + c_{u_L 0} + \sum_{j=1}^{L} \Big(\sum_{\omega \in \Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \Big) + \widehat{T}_{\text{RCSP}}(u_L, Q - q_{\theta'}) \\
&\overset{\text{(f)}}{=} \overline{C}_{\theta'} + \widehat{T}_{\text{RCSP}}(u_L, Q - q_{\theta'}) \overset{\text{(g)}}{\leq} \overline{C}_{\theta'}.
\end{aligned}$$

Inequality (b) follows from the fact that extending path θ to customer u_1 cannot lead to a smaller bound than the bound associated with the optimal path extension from the minimization operator in Equation (20). In Inequality (c), the same argument holds when extending path $\theta \oplus u_1$ to customers

$u_2, \ldots, u_j$. In Equation (d) we rearrange the terms, and Inequality (e) is due to the fact that, given a path θ ending at customer j:

$$\delta_\theta \leq \tau_\theta \leq a_\theta(j; \mathbf{t}^\omega), \quad \forall \omega \in \Omega \tag{33}$$

Finally, Equality (f) is due to the resource extension function for the reduced cost given by Equation (17), and Inequality (g) holds since going from u_L back to the depot incurs no additional cost and can not improve the completion bound. □

Proof of Proposition 2. From the resource extension function given by Equation (17), we have:

$$\begin{aligned}
\overline{C}_{\theta'} &\overset{\text{(a)}}{=} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1}u_j} + c_{u_L 0} + \sum_{j=1}^{L} \left(\sum_{\omega \in \Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \right) \\
&\overset{\text{(b)}}{\geq} \overline{C}_\theta + \sum_{j=1}^{L} \left(\sum_{\omega \in \Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \right) \\
&\overset{\text{(c)}}{\geq} \overline{C}_\theta + \sum_{j=1}^{L} \left(\pi(\tau_\theta + \min_{\omega \in \Omega} t^\omega_{iu_j} - \ell_{u_j}) - \gamma_{u_j} \right) \\
&\overset{\text{(d)}}{=} \overline{C}_\theta + \sum_{l \in \mathcal{V} \setminus \{0\}} -v_{il}(\theta)\, z^*_l \overset{\text{(e)}}{\geq} \overline{C}_\theta + \widehat{T}_{ks}(i, Q - q_\theta) \geq 0
\end{aligned}$$

where Inequality (b) is due to the triangle inequality, which implies that the cost of a route cannot decrease if we add customers to it. Inequality (c) is a consequence of Equation (22) and the fact that adding more customers to a route between i and u_j can only increase the arrival time at customer u_j. Equality (d) holds by our definition of the knapsack values. Inequality (e) is due to the optimality of the knapsack solution and the definition of the completion bound. □

C. Dynamic programming algorithm for RCSP

We present a DP algorithm for obtaining a RCSP bound. In Algorithm 1, variable $T_1[\delta, i, q]$ stores a lower bound on the reduced cost of extending a route that ends at customer i with remaining capacity q, departing from i at time δ (i.e., the arrival time at customer i is equal to δ). We relax elementarity by allowing routes with cycles but we still remove 2-cycles. Similarly, $T_2[\delta, i, q]$ stores the second best lower bound on the reduced cost. Finally, $N[\delta, i, q]$ stores the customer following i on the route associated with the best lower bound.

Algorithm 1: Dynamic programming algorithm for RCSP

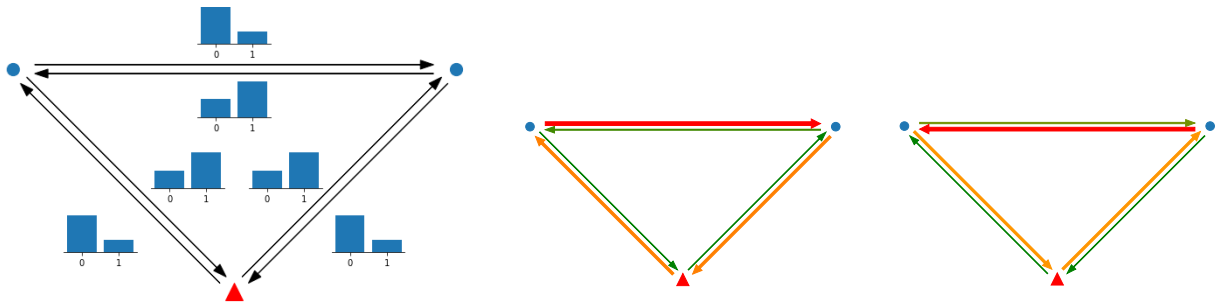
Result: matrix T_1 of lower bounds on the reduced costs of route extensions

$\ell_{\max} \leftarrow \max_{i \in \mathcal{V}^+}\{\ell_i\}$ // latest end of time window among all customers
$\Delta t \leftarrow \ell_{\max}/40$ // define a time step
for $\delta = 0, \Delta t, 2\Delta t, \ldots, \ell_{\max}$ **do**
  $T_1[\delta, i, q] \leftarrow \infty$, for $i \in \mathcal{V}^+$, $q = 1, \ldots, Q$ // initialize matrix T_1: lower bound on reduced costs
  $T_1[\delta, 0, q] \leftarrow 0$, for $q = 0, \ldots, Q$ // initialize matrix T_1
  $T_1[\delta, i, 0] \leftarrow c_{i0}$, for $i \in \mathcal{V}^+$ // initialize matrix B
  $T_2[\delta, i, q] \leftarrow \infty$, for $i \in \mathcal{V}$, $q = 0, \ldots, Q$ // initialize matrix T_2: second best cost
  $N[\delta, i, q] \leftarrow 0$, for $i \in \mathcal{V}$, $q = 0, \ldots, Q$ // initialize matrix N: next customer in the route
  for $q = 1, \ldots, Q$ **do**
    for $i \in \mathcal{V}^+$ **do**
      $T_1[\delta, i, q] \leftarrow T_1[\delta, i, q-1]$
      $T_2[\delta, i, q] \leftarrow T_2[\delta, i, q-1]$
      $N[\delta, i, q] \leftarrow N[\delta, i, q-1]$
      for $j \in \mathcal{V}^+$ **do**
        if $j = i$ *or* $q_j > q$ *or arc* (i, j) *is forbidden by branching* **then**
          continue // does not extend label L to customer j
        if $N[\delta, j, q - q_j] \neq i$ **then**
          $v \leftarrow c_{ij} - \gamma_j + T_1[\delta, j, q - q_j] + \pi(\delta - \ell_j)$
        else
          $v \leftarrow c_{ij} - \gamma_j + T_2[\delta, j, q - q_j] + \pi(\delta - \ell_j)$ // avoid 2-cycles
        if $v < T_1[i, q]$ **then**
          $T_2[\delta, i, q] \leftarrow T_1[\delta, i, q]$ // move best to second best
          $T_1[\delta, i, q] \leftarrow v$ // set new best
          $N[\delta, i, q] \leftarrow j$ // set next customer
        else if $v < T_2[\delta, i, q]$ **then**
          $T_2[\delta, i, q] \leftarrow v$ // just update second best
return T_1

D. Illustrative example

We consider a network with $N = 2$ customers and a training data set with $n = 2$ samples and $p = 1$ binary feature. Figure 5a shows the customer locations and arcs connecting each pair of customers. A small bar chart next to each arc shows the corresponding travel times on the y-axis as a function of the feature value on the x-axis. We omitted the values on the y-axis since this illustrative example is not concerned with specific travel time values but rather aims to show the relation between travel times and features. Figure 5b and 5c show the two possible scenarios, representing travel time realizations when the feature value equals $x = 0$ and $x = 1$, respectively. The color and line thickness of each arc indicate the level of congestion, with thicker lines representing greater congestion, and different colors indicating whether the arc is strongly congested (red), mildly congested (orange), or free from congestion (green). Under the scenario displayed in Figure 5b, a route that starts at the

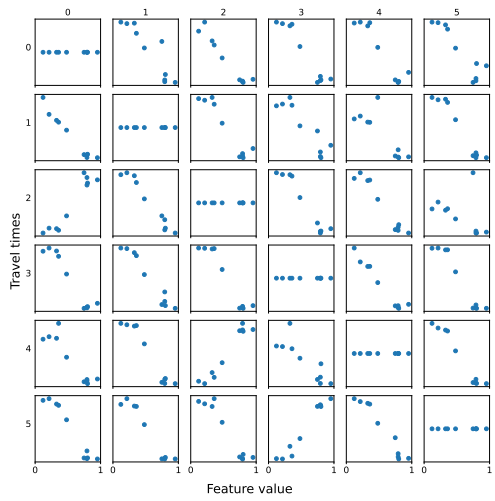
depot and visits customers in a clockwise direction will experience more congestion than a counter-clockwise route. Therefore, if the decision-maker finds herself in a scenario where $x = 0$, following the counter-clockwise route is optimal. Figure 5c shows the reverse pattern, i.e., when $x = 1$, the counter-clockwise route is more congested than the clockwise route. Existing methods based on SAA, which are widely adopted in the field, would provide a clockwise route that is optimal when $x = 0$ but not when $x = 1$. In contrast to featureless methods, feature-dependent solutions for the CS-VRPTW can provide the optimal route in both cases.



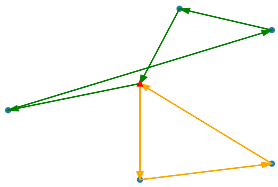
(a) Travel times as a function of feature value. (b) Travel times for $x = 0$. (c) Travel times for $x = 1$.

Figure 5 Joint distribution and different realizations of travel times and feature variable.

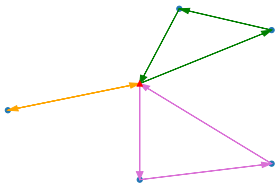
We now extend our previous example and consider a network with $N = 5$ customers and a training data set with $n = 10$ samples and $p = 1$ continuous feature with domain $x \in [0, 1]$. To capture different degrees of congestion, we assume sigmoidal travel times (see Section 5.1). In Figure 5a, each scatter plot in row $i \in \{0, \ldots, 5\}$ and column $j \in \{0, \ldots, 5\}$ shows the travel times and features in the training data set corresponding to arc $(i, j) \in \mathcal{A}$. For a more palatable exposition, we normalized the travel times of each arc based on its nominal free-flow travel time. We compare the optimal solutions of two featureless prescriptive methods, i.e., D-avg and SAA, against PSAA and full-information solutions. Figures 5c and 5b show the solution structures that emerge in our example. Note how the solution structures are fundamentally different from each other. In particular, Solution A requires three vehicles, whereas Solution B requires only two. In Figure 6d, we show the solution structures obtained by each method under different realizations of the feature variable. When $x = 0.28$, the featureless point-based approximation (D-avg) retrieves the full-information solution, but it fails to do so when $x = 0.83$. We see the opposite behavior for the featureless SAA method, i.e., it fails for $x = 0.28$ but can retrieve the full-information solution when $x = 0.83$. In contrast to the featureless approaches, PSAA provides feature-dependent solutions that match the full-information solution in both cases. This small example shows that ignoring information from feature data can lead to suboptimal solutions, underlying the benefit of the CS-VRPTW formulation.



(a) Relation between travel times and features in the training data set for each arc.



(b) Solution structure A.



(c) Solution structure B.

Feature realization	D-avg	SAA	PSAA	Full
$x = 0.28$	**A**	B	**A**	**A**
$x = 0.83$	A	**B**	**B**	**B**

(d) Optimal solution structures of each method.

Figure 6 Data set and solution structures of our illustrative example.

E. Generative models of travel times and features

Linear model. For each arc $(i,j) \in \mathcal{A}$, the deterministic cost c_{ij} defines a nominal travel time $\underline{t}_{ij}$ corresponding to the free-flow travel time of the arc. We define the stochastic travel times as the nominal travel times $\underline{\mathbf{t}} = [\underline{t}_{ij}]_{(i,j)\in\mathcal{A}}$ plus a random noise term which depends linearly on the features:

$$\tilde{\mathbf{t}}_{\text{linear}}(\tilde{\mathbf{x}}) = \underline{\mathbf{t}} + \mathbf{B}^\top \tilde{\mathbf{x}} + \tilde{\boldsymbol{\varepsilon}} \tag{34}$$

where $\mathbf{B}$ is a $p \times |\mathcal{A}|$ matrix whose columns are given by the vectors $\mathbf{b}_{ij} \in \mathbb{R}^p$ for $(i,j) \in \mathcal{A}$. We sample values in $\mathbf{b}_{ij}$ from a uniform distribution with support ranging from 1% to 20% of the corresponding nominal travel time $\underline{t}_{ij} = c_{ij}$. We assume that features $\tilde{\mathbf{x}}$ follow a uniform distribution between 0 and 1. The noise term $\tilde{\boldsymbol{\varepsilon}}$ follows a multivariate normal distribution with zero mean, and covariance matrix generated according to the method of Rostami et al. (2021), such that noise values at different arcs are correlated. Lastly, we assume that travel times on each arc must be greater than or equal to the nominal travel time, and we truncate travel times whenever necessary.

Exponential model. For each arc $(i,j) \in \mathcal{A}$, we consider that features are related to the travel times via an exponential function:

$$\tilde{t}_{ij} = \underline{t}_{ij} + 0.2\, \underline{t}_{ij} \exp\left(2\, \mathbf{b}_{ij}^\top \tilde{\mathbf{x}}\right) + \tilde{\varepsilon}_{ij} \tag{35}$$

where features $\tilde{\mathbf{x}}$ follow a uniform distribution between 0 and 1. We generate the parameter vectors $\mathbf{b}_{ij} \in \mathbb{R}^p$ by sampling each element from a uniform distribution between 0.1 and 0.3, and we

multiply each element by -1 with a probability of 0.2. Due to the exponential travel times, having normally distributed $\tilde{\varepsilon}_{ij}$ does not provide sufficient noise. Therefore, we assume that $\tilde{\varepsilon}_{ij}$ follows a log-normal distribution with zero mean and standard deviation $\sigma_\varepsilon = 1$.

Sigmoidal model. For each arc $(i,j) \in \mathcal{A}$, we generate travel times:

$$\tilde{t}_{ij} = \underline{t_{ij}} + \underline{t_{ij}}\,\sigma\Big(32(\tfrac{1}{2}\mathbf{b}_{ij}^\top \mathbb{1} - \mathbf{b}_{ij}^\top \tilde{\mathbf{x}})\Big) + \tilde{\varepsilon}_{ij} \tag{36}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. Features follow a uniform distribution between 0 and 1. The noise term $\tilde{\varepsilon}_{ij}$ follows a log-normal distribution with mean 0 and standard deviation $\sigma_\varepsilon = 1.2$. We generate the parameter vectors $\mathbf{b}_{ij}$ by sampling each element from a uniform distribution between 0.3 and 0.8, and we multiply each element by -1 with a probability of 0.2. Due to its characteristic shape, we can interpret the sigmoidal model as representing two feature-dependent *states of traffic*, e.g., a congested and a non-congested state.

F. Full-information benchmarks

We compare our models against benchmark solutions that rely on knowledge of travel times and feature distributions. The following benchmark solutions are impractical in a real-world setting, as the decision-maker does not know the underlying distributions. However, these benchmark solutions provide us with a relative measure of how well the proposed practical models perform.

Full-information solution. Based on the test data set, we can approximate the full-information solution of the CS-VRPTW, given a feature vector $\mathbf{x} \in \mathcal{X}$, as:

$$\hat{\mathbf{z}}^*_{\text{FULL}}(\mathbf{x}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} \frac{1}{n_\mathcal{T}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{x})} f_\Theta(\mathbf{z},\mathbf{t}), \tag{37}$$

with optimal objective value:

$$\hat{v}^*_{\text{FULL}}(\mathbf{x}) = \min_{\mathbf{z}\in\mathcal{Z}_\Theta} \frac{1}{n_\mathcal{T}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{x})} f_\Theta(\mathbf{z},\mathbf{t}). \tag{38}$$

Given the prescription $\hat{\mathbf{z}}^*_{\text{FULL}}$, the empirical test cost is given by:

$$\hat{R}_{\text{FULL}} = \hat{R}(\hat{\mathbf{z}}^*_{\text{FULL}}(\mathbf{x})) = \frac{1}{n_\mathcal{X}\cdot n_\mathcal{T}} \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{x})} f_\Theta\left(\hat{\mathbf{z}}^*_{\text{FULL}}(\mathbf{x}),\mathbf{t}\right) = \frac{1}{n_\mathcal{X}} \sum_{\mathbf{x}\in\mathcal{X}} \hat{v}^*_{\text{FULL}}(\mathbf{x}), \tag{39}$$

which provides a lower bound for the empirical test cost of any model. With this definition, we calculate the full-information percentage gap of a prescription $\hat{\mathbf{z}}$ as:

$$\rho(\hat{\mathbf{z}}) = \frac{\hat{R}(\hat{\mathbf{z}}) - \hat{R}_{\text{FULL}}}{\hat{R}_{\text{FULL}}}. \tag{40}$$

Predict with full information, then optimize. The performance of a prescription under the PTO framework depends on the choice of predictive model $\mathbf{g}(\cdot)$ providing travel time predictions $\hat{\mathbf{t}} = \mathbf{g}(\mathbf{x}; \hat{\boldsymbol{\varphi}})$. For the purpose of benchmarking, we define the PTO-F problem, which assumes that the predictive model perfectly predicts the expected travel times given observed features:

$$\mathbf{z}^*_{\text{PTO-F}}(\mathbf{x}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} f_\Theta\left(\mathbf{z}, \bar{\mathbf{t}}\right) \quad \text{with } \bar{\mathbf{t}} = \mathbb{E}\left[\tilde{\mathbf{t}} \,\middle|\, \tilde{\mathbf{x}} = \mathbf{x}\right]. \tag{41}$$

A solution for the PTO-F problem requires knowledge of the joint distribution of travel times and features. We can approximate the PTO-F solution using the test data set:

$$\hat{\mathbf{z}}^*_{\text{PTO-F}}(\mathbf{x}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} f_\Theta\left(\mathbf{z}, \hat{\mathbf{t}}\right) \quad \text{with } \hat{\mathbf{t}} = \frac{1}{n_\mathcal{T}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{x})} \mathbf{t}. \tag{42}$$

We note that although PTO-F often performs better than PTO with practical prediction models, PTO-F is not necessarily a lower bound for PTO regarding test cost since neither PTO nor PTO-F account for the structure of the downstream optimization problem. Specifically, the true conditional expected travel times do not necessarily correspond to the travel times leading to minimum cost.

G. Predictive Performance Analysis

This section discusses predictive metrics for the linear regression (PTO-OLS) and k-NN regression (PTO-kNN) for all three generative models. Table 6 reports R^2 and MSE values of the travel time predictions averaged across scenarios and instance types. For the linear generative model, both methods achieve high R^2 values with relatively low MSE, reflecting the strong alignment between the predictive models and the underlying data-generating process. Interestingly, OLS regression achieves even higher R^2 values under the exponential generative model than under the linear model, suggesting that the exponential relationship is well approximated by a linear model in our parameter regime. However, PTO-k-NN performance degrades substantially for the exponential model, suggesting that the method struggles with the increased complexity. The sigmoidal model presents the most challenging prediction setting: while R^2 values remain reasonably high, MSE values are 3 to 7 times larger than those of the linear and exponential models. R^2 measures the proportion of variance in the travel times explained by the features (remaining high when predictions capture mean relationships), whereas MSE reflects absolute prediction errors that directly impact route quality. The substantially higher MSE for the sigmoidal model confirms that stronger nonlinearities dominate prediction difficulty. Although linear regression achieves high R^2 by explaining overall

variance, PTO-based optimization fails to capture scenario-dependent travel time variability. In contrast, SAA-based methods perform better by explicitly modeling this variability through scenarios.

Table 6 Test-set prediction metrics (R^2 and MSE) for linear regression and k-NN regression methods across generative models and instance types.

Gen.	Inst.	OLS		kNN	
model	type	R^2	MSE	R^2	MSE
Lin.	C	0.976	6.45	0.970	8.01
	R	0.967	20.27	0.960	25.07
	RC	0.978	32.18	0.974	38.47
Exp.	C	0.982	9.45	0.948	27.98
	R	0.984	21.43	0.939	81.21
	RC	0.991	23.89	0.966	88.63
Sig.	C	0.869	46.58	0.814	66.21
	R	0.870	112.10	0.809	165.49
	RC	0.907	164.37	0.860	247.04

H. Sensitivity of PSAA to the Number of Scenarios

To study the impact of the number of scenarios on the PSAA method, we conducted a sensitivity analysis on a subset of instances generated under the linear generative model with $p = 10$ features and $n = 100$ samples. Figure 7 presents the distribution of test costs, reported as full-information gaps, for increasing numbers of scenarios. The left y-axis displays the test cost values, while the right y-axis shows the corresponding average run times. The results indicate that both the mean test cost and its variance decrease sharply as the number of scenarios increases from 1 to 5. Beyond approximately 10 scenarios, however, the improvements in test cost are no longer noticeable. Based on these results, we concluded that using 50 scenarios provides sufficiently low test costs while offering improved computational efficiency relative to using a larger number of scenarios.

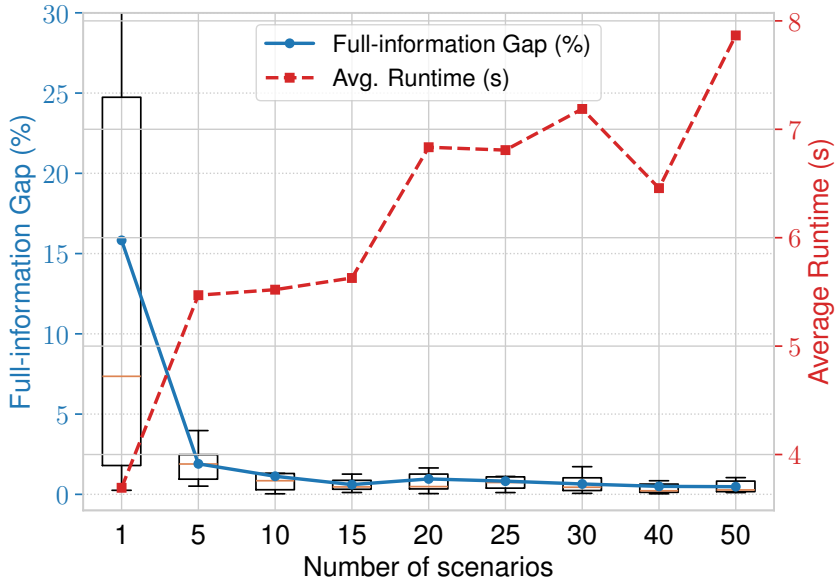


Figure 7 **Test costs for increasing number of scenarios for PSAA.**

I. Effects of Feature Dimension and Sample Size under Non-Linear Generative Models

Complementing the analysis presented in Section 6.2, we examine the effects of varying the feature dimension and sample size for non-linear generative models.

Figure 8 reports the performance of the prescriptive methods under an exponential generative model for travel times. The non-linear nature of the exponential model introduces additional estimation difficulty, which are reflected in higher overall test costs and increased variability compared to the linear setting. In the low-dimensional case ($p = 3$), SAA-based methods show the best performance and all methods represent limited sensitivity to the number of samples n.

For the setting with $p = 10$, the D-avg baseline method shows large gaps with high variance. Similarly, methods based on local information, such as PTO-kNN and SAA-kNN struggle to achieve low costs, especially in the low-data regime ($n = 50$). Nevertheless, the SAA-kNN method shows clear improvements with increasing sample size, as evidenced by the reduction in both average cost and variability at $n = 1000$. Overall, PSAA and RSAA consistently achieve relatively low costs and demonstrate robustness even with small sample sizes. In contrast, the classical featureless SAA baseline yields significantly higher costs and shows limited improvement with increasing sample size.

For the high-dimensional setting ($p = 100$), the adopted exponential generative model produces unrealistically large travel times, resulting in numerical instability. Therefore, we restricted this analysis to settings with $p = 3$ and $p = 10$.

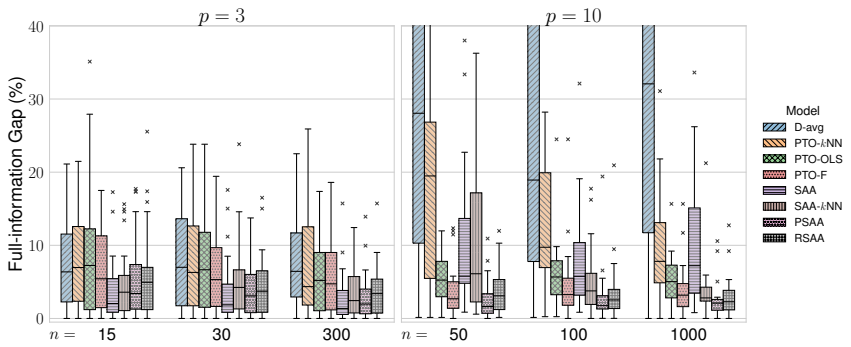


Figure 8 **Test costs for increasing feature dimension and sample size on instances with exponential travel times.**

Figure 9 reports results obtained under a sigmoidal generative model for travel times, comparing the performance of the different prescriptive methods. This non-linear generative model provides the most challenging settings, as evidenced by substantially higher test costs and increased variability across all methods.

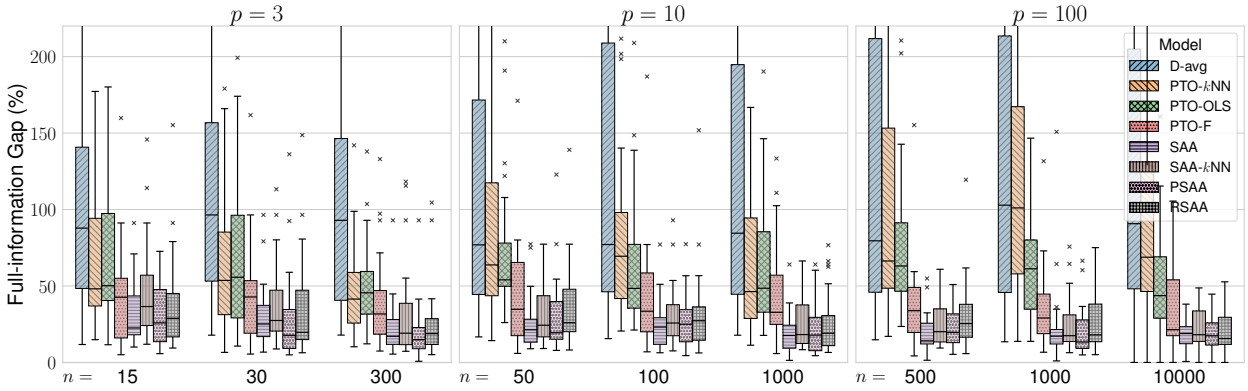


Figure 9 **Test costs for increasing feature dimension and sample size on instances with sigmoidal travel times.**

Consistent with the results obtained under the exponential generative model, the SAA-based methods provide the lowest costs overall. In particular, PSAA and RSAA consistently achieve relatively low costs and demonstrate robustness even with small sample sizes. Most methods show only marginal improvements in performance as the sample size increases. The classical featureless SAA method remains competitive with feature-dependent SAA variants, as it can already capture the variability in travel times. This behavior likely depends on the noise level applied to the generative model, which may affect the relative advantage of feature-based methods. In contrast to the classical SAA, the featureless D-avg baseline performs notably worse, showing large test costs and high variance across all configurations, regardless of the choice of feature dimension p and sample

size n. These results highlight the importance of accounting for travel time variability under the sigmoidal generative model.

J. Number of neighbors in kNN Regression

As explained in the experiments involving k-nearest neighbors (kNN) regression (see Section 5.3), the number of neighbors k was selected through a grid search using cross-validation. To provide further insight into the range of neighborhood sizes used by the kNN method, Figure 10 shows the distribution of the chosen k values across the three generative models considered, focusing on the base setting of instances with 25 customers, 100 historical observations, and 10 features.

As observed, the optimal values of k tend to concentrate between 6 and 12, with slight variations depending on the specific generative model. While in principle k could take values as large as the number of observations (i.e., up to 100 in this setting), the cross-validation procedure consistently favored moderately sized neighborhoods rather than very small or very large ones.

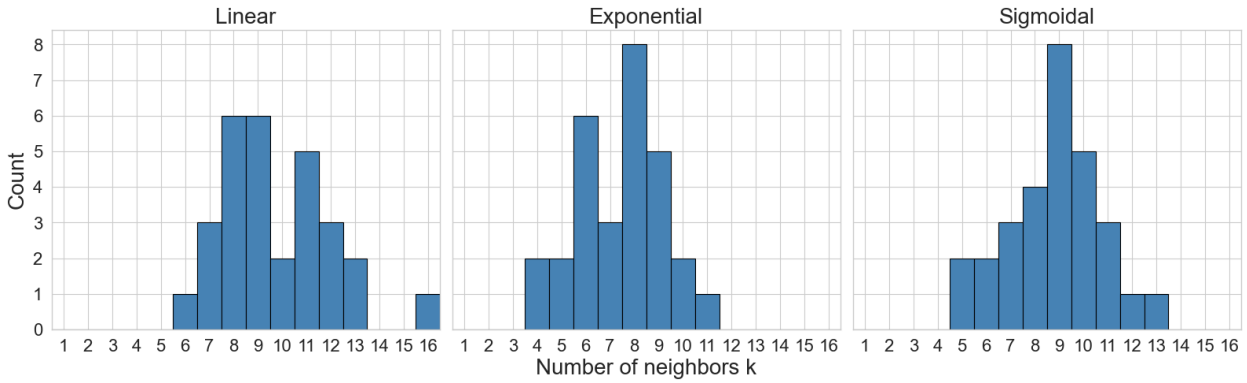


Figure 10 Distribution of the selected number of neighbors in kNN regression.