# Technical Annex: Additional Technical Detail Concerning "Efficient Stochastic Routing in Path-Centric Uncertain Road Networks"

Chenjuan Guo[1,2*], Ronghui Xu[1], Bin Yang[1,2], Ye Yuan[2], Tung Kieu[2], Yan Zhao[2], Christian S. Jensen[2]
[1]East China Normal University, [2]Aalborg University
{cjguo,byang}@dase.ecnu.edu.cn,rhxu@stu.ecnu.edu.cn,yuanye@cs.aau.dk,{tungkvt,yanz,csj}@cs.aau.dk

This annex aims to report additional information that could not be included into the main paper due to the space limitation.

## 1 NAIVE STOCHASTIC ROUTING IN PACE

**Algorithm 1:** Naive Stochastic Routing in PACE

**Input:** Source $v_s$; Destination $v_d$; Budget $B$; PACE graph $\mathcal{G}^{\mathcal{P}}$;
**Output:** The path with the largest probability of getting $v_d$ within $B$;

```
Priority queue Q ← ∅;
Double maxProb ← 0; Path P* ← ∅;
Q.push(⟨v_s, v_s⟩, 0);
while Q ≠ ∅ do
    Path P̂ ← Q.peek();
    if P̂ reaches v_d and Prob(D(P̂) ≤ B) > maxProb then
        maxProb ← Prob(D(P̂) ≤ B); P* ← P̂;
    for each outgoing edge or T-path e do
        Extend P̂ by edge e;
        Q.insert(P̂, Expectation(D(P̂)));
return P*;
```

## 2 STATISTICS OF DATASETS

We report the statistics of Aalborg and Xi'an datasets in Table 1.

**Table 1: Statistics of datasets.**

| | Aalborg | Xi'an |
|---|---|---|
| Number of vertices | 32,226 | 117,415 |
| Number of edges | 78,348 | 236,733 |
| AVG vertex degree | 2.43 | 2.02 |
| AVG edge length (m) | 172.85 | 58.36 |
| AVG closeness centrality | $1.54\times10^{-2}$ | $1.64\times10^{-3}$ |
| Sampling rates of GPS records (Hz) | 1 | 0.2 |
| Number of traj. | 553,904 | 363,308 |
| AVG number of vertices of traj. | 5.41 | 27.05 |
| MAX number of vertices of traj. | 100 | 149 |

## 3 COMPLEXITY ANALYSIS

### 3.1 Algorithm 1

Given graph $\mathcal{G}^{\mathcal{P}}_{rev} = (\mathbb{V}, \mathbb{E}', \mathbb{P}', \mathbb{W}')$, the time complexity for building the binary heuristics is $O((|\mathbb{E}'| + |\mathbb{P}'|)lg|\mathbb{V}|)$, where $|\mathbb{E}'|$, $|\mathbb{P}'|$ and $|\mathbb{V}|$ represent the number of reversed edges, reversed T-paths, and vertices (i.e., road intersections or ends of roads), respectively.

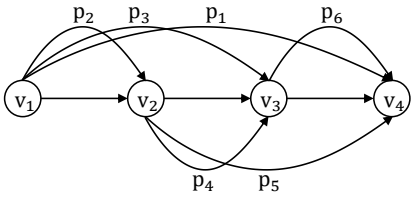



**Figure 1: Motivating Example**

The number of T-paths $|\mathbb{P}'|$ is polynomial in the length of the longest T-path. For instance, as illustrated in Figure 1, assume that the longest T-path is $p_1 = \langle v_1, v_2, v_3, v_4 \rangle$. From $v_1$, it generates three T-paths: $p_1 = \langle v_1, v_2, v_3, v_4 \rangle$, $p_2 = \langle v_1, v_2 \rangle$ , and $p_3 = \langle v_1, v_2, v_3 \rangle$. Likewise, from $v_2$, it generates two T-paths: $p_4 = \langle v_2, v_3 \rangle$, $p_5 = \langle v_2, v_3, v_4 \rangle$; and from $v_3$, it generates one T-path: $p_6 = \langle v_3, v_4 \rangle$. Hence, the total number of T-paths we can generate is $\frac{n(n-1)}{2}$, where $n$ is the length of the longest T-path.

Specifically, in the worst case, we visit all edges and T-paths, thus having $|\mathbb{E}'| + |\mathbb{P}'|$. For each visit, we need to update the priority queue, which at most has $|\mathbb{V}|$ elements, and thus the update takes at most $lg|\mathbb{V}|$.

The space complexity is $O(|\mathbb{V}|^2)$. For each destination, each vertex maintains a binary heuristic value.

### 3.2 Algorithms 2 and 3

Given graph $\mathcal{G}^{\mathcal{P}} = (\mathbb{V}, \mathbb{E}, \mathbb{P}, \mathbb{W})$, the time complexity for Algorithms 2 and 3 is $O(|\mathbb{V}| \cdot \eta^2 \cdot |out|)$, where $\eta$ is the number of columns in the heuristic table and $|out|$ is the largest outdegree of a vertex. We apply dynamic programming to build the heuristics table using Eq. (4). The heuristics table has in total $\mathbb{V} \cdot \eta$ elements. To fill in an element in the table, we need to visit at most $|out| \cdot \eta$ other elements because we may visit up to $|out|$ rows (see $Z \in ON(v_i)$ in Eq. (4)) and each row we may visit up to $\eta$ elements (see $\sum_{k=1}^{x}$ in Eq. (4)).

The space complexity is $O(|\mathbb{V}|^2 \cdot \eta)$. For each destination, we maintain a heuristic table taking $|\mathbb{V}| \cdot \eta$, therefore a total of $|\mathbb{V}| \cdot |\mathbb{V}| \cdot \eta$.

### 3.3 Building V-paths

Given graph $\mathcal{G}^{\mathcal{P}} = (\mathbb{V}, \mathbb{E}, \mathbb{P}, \mathbb{W})$, the time complexity for building V-paths is $O(|\mathbb{P}| \cdot |\mathbb{V}|)$.

Recall that V-paths are generated iteratively. Each iteration can at most generate $|\mathbb{P}|$ new V-paths and we can at most have $|\mathbb{V}|$ iterations. In the first iteration, overlapping T-paths are combined into V-paths. Thus, the number of generated V-paths must be smaller than the number of T-paths because the generated V-paths must have higher cardinality than the corresponding T-paths and they both represent the same path. In the next iteration, the same principle applies. Thus, each iteration can at most generate $O(|\mathbb{P}|)$ new V-paths.

Next, we explain why we can get most have $|\mathbb{V}|$ iterations. At each iteration, the cardinality of V-paths are at least one more than the V-paths in the previous iteration. In a graph with $|\mathbb{V}|$ vertices, the longest simple path (i.e., without loops) in the graph can at most have $|\mathbb{V}|$ vertices. Otherwise, there must be at least one vertex that appears twice in the path making a loop and thus the path is not a simple path anymore. Thus, we only need to perform at most $|\mathbb{V}|$ iterations as we need to ensure that V-paths are simple paths.

Space complexity is $O(|\mathbb{P}| \cdot |\mathbb{V}| \cdot |St|)$ where $|St|$ is the storage used for the longest V-path.

## 4 CROSS VALIDATION EXPERIMENT

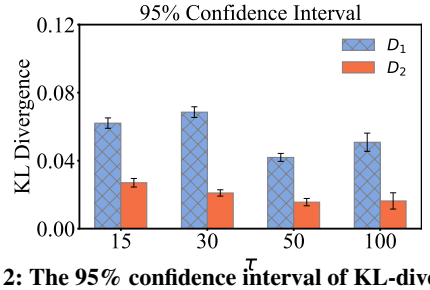



**Figure 2: The 95% confidence interval of KL-divergence**

We implement five-fold cross validation to assess the experiments. Then, We use KL-divergence to quantify the distances between the ground truth and the estimated distributions. This process is repeated five times, each time with a different training set. Subsequently, we computed the mean and variance of the results from these iterations. The results are used to depict 95% confidence interval for the KL-divergence, as depicted in Figure 2.

When the value of $\tau$ increases from 15 to 50 (T-path is instantiated by more trajectories), the value of KL-divergence gradually decreases, indicating that a larger $\tau$ value yields a more precise cost distribution for the T-path. Concurrently, the narrow range of the 95% confidence interval implies the minimal dataset variability's impact on the KL-divergence computation. Conversely, when $\tau = 100$, the KL-divergence starts to decrease. Moreover, the broadening of the 95% confidence interval at $\tau = 100$ indicates potential instability. These results may be due to the excessive exclusion of trajectories, which in turn affects the accuracy of cost distribution.

## 5 OFFLINE PRE-COMPUTATION COSTS

We report the overall offline pre-computation cost for building binary heuristics for $D_1$ and $D_2$ in Table 2, where run time and storage is reported by hours (h) and gigabytes (GB), respectively.

| Data Set $D_1$ | | | | | | |
|---|---|---|---|---|---|---|
| | **Off-Peak Hours** | | | **Peak Hours** | | |
| Methods | T-B-EU | T-B-E | T-B-P | T-B-EU | T-B-E | T-B-P |
| Run time (h) | 1.8 | 17.6 | 32.3 | 2 | 19.2 | 24.7 |
| Storage (GB) | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| Data Set $D_2$ | | | | | | |
| | **Off-Peak Hours** | | | **Peak Hours** | | |
| | T-B-EU | T-B-E | T-B-P | T-B-EU | T-B-E | T-B-P |
| Run time (h) | 8.5 | 92.8 | 224.5 | 12.6 | 108.1 | 239.2 |
| Storage (GB) | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 |

**Table 2: Binary Heuristics Pre-computation**

When building budget-specific heuristics, we did not use parallelization, which, however, is possible because building the heuristics for different vertices is independent. The offline pre-computation cost for $D_1$ and $D_2$ is reported in Table 3, where run time and storage is reported by hours (h) and gigabytes (GB), respectively.

| Data Set $D_1$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Off-Peak Hours** | | | | **Peak Hours** | | | |
| $\delta$ | 30 | 60 | 120 | 240 | 30 | 60 | 120 | 240 |
| Run time (h) | 43.2 | 33.4 | 20.4 | 16.8 | 56 | 28.5 | 17.4 | 13.2 |
| Storage (GB) | 1.87 | 1.37 | 1.02 | 0.92 | 1.34 | 1.07 | 0.97 | 0.92 |
| Data Set $D_2$ | | | | | | | | |
| | **Off-Peak Hours** | | | | **Peak Hours** | | | |
| $\delta$ | 30 | 60 | 120 | 240 | 30 | 60 | 120 | 240 |
| Run time (h) | 318.8 | 258.2 | 225.7 | 207.9 | 289.2 | 247.1 | 223.8 | 213.2 |
| Storage (GB) | 5.95 | 4.45 | 3.41 | 3.06 | 4.38 | 3.54 | 3.21 | 3.06 |

**Table 3: Budget-Specific Heuristics Pre-computation**

## 6 COMPARISON OF DIFFERENT METHODS

| Data Set $D_1$, Peak Hours | | | | | | |
|---|---|---|---|---|---|---|
| Methods | T-B-EU | T-B-E | T-B-P | V-B-P | T-BS-60 | V-BS-60 |
| Storage (GB) | 0.92 | 0.92 | 0.92 | 0.92 | 1.07 | 1.09 |
| Precomputation (h) | 2 | 19.2 | 24.7 | 32.3 | 29.1 | 37.0 |
| Routing (s) | 0.364 | 0.235 | 0.214 | 0.142 | 0.078 | 0.067 |
| Data Set $D_2$, Peak Hours | | | | | | |
| Methods | T-B-EU | T-B-E | T-B-P | V-B-P | T-BS-60 | V-BS-60 |
| Storage (GB) | 3.06 | 3.06 | 3.06 | 3.06 | 3.52 | 3.54 |
| Precomputation (h) | 12.6 | 108.1 | 239.2 | 284.5 | 248.5 | 302.1 |
| Routing (s) | 1.157 | 1.126 | 1.017 | 0.993 | 0.577 | 0.402 |

**Table 4: Comparison of different methods**

We provide statistics of the two heuristic methods in Table 4, where we set $\delta$ to 60 as recommended in the paper. As we can observe, the budget-specific heuristic method requires slightly more storage than the binary heuristics, and they mainly differ in offline processing run time and average online routing run time.

We do not recommend *T-B-EU* and *T-B-E*, as the naive PACE routing that uses simple binary heuristics requires high online running time, which is unbearable. In cases where sufficient one-time offline processing is possible, we always recommend *V-BS-*$\delta$, as it offers the fastest online routing services. In cases of limited offline process time, we recommend *T-B-P*, as it requires the least offline time compared to *T-B-P*, *T-BS-*$\delta$ and *V-BS-*$\delta$, though it requires more online routing time. The two designed heuristic methods well demonstrate the philosophy that if we trade off more offline pre-computation resources, we could obtain more efficient online routing service [1, 2].

## 7 CASE STUDY

To illustrate the benefits of the proposed methods, we analyze two representative queries from the datasets for peak hours. We compare V-BS-60 against Google Maps for Aalborg and Baidu Maps for Xi'an, as these platforms are widely utilized in the respective cities. Blue lines denote the routes provided by the platforms and red lines denote the routes provided by the proposed method. In instances of overlap, the blue lines are placed on top of the red lines. We assess the probability of arrival within the budget time for all paths based on the cost distributions.

As depicted in Figure 3 (a), we set a budget of 15 minutes, within which the V-BS-60 route has a 72.1% probability of arrival, outperforming the Google Maps route that has a 66.7% probability. Similarly, for the Xi'an dataset with a 35-minute budget (as shown in Figure 3 (b)), the probability of the V-BS-60 route exceeds that

of the Baidu Maps route. These cases illustrate how the proposed method is capable of enhancing the probability of arriving within a time budget.

# 8 EXPERIMENTAL RESULTS

We provide experimental results for stochastic routing with binary heuristics at off-peak and peak hours for data sets $D_1$ and $D_2$ in Figures 5 and 6, respectively.

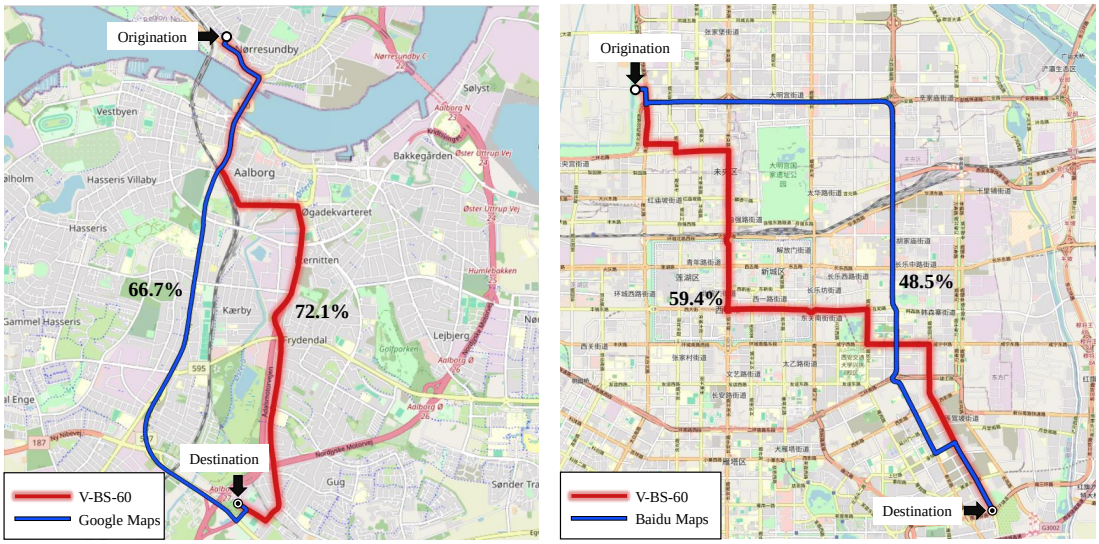
We provide experimental results for stochastic routing with budget-specific heuristics at off-peak and peak hours for data sets $D_1$ and $D_2$ in Figures 7 and 8, respectively.

We provide experimental results for V-path based stochastic routing at off-peak and peak hours for data sets $D_1$ and $D_2$ in Figures 9 and 10, respectively.

# REFERENCES

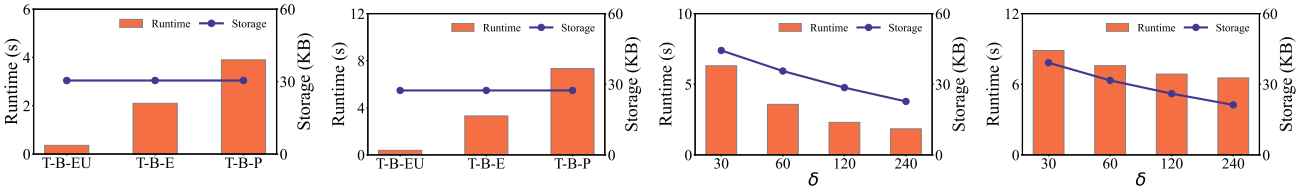
[1] Mehrdad Niknami and Samitha Samaranayake. 2016. Tractable Pathfinding for the Stochastic On-time Arrival Problem. In *International Symposium on Experimental Algorithms*. Springer, 231–245.

[2] Alberto Vera, Siddhartha Banerjee, and Samitha Samaranayake. 2022. Computing Constrained Shortest-Paths at Scale. *Oper. Res.* 70, 1 (2022), 160–178.



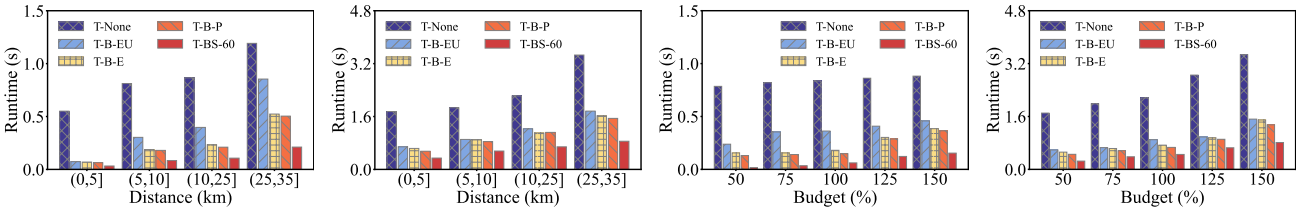

(a) Aalborg (budget for 15 minutes at peak hours) (b) Xi'an (budget for 35 minutes at peak hours)

**Figure 3: Case Study**



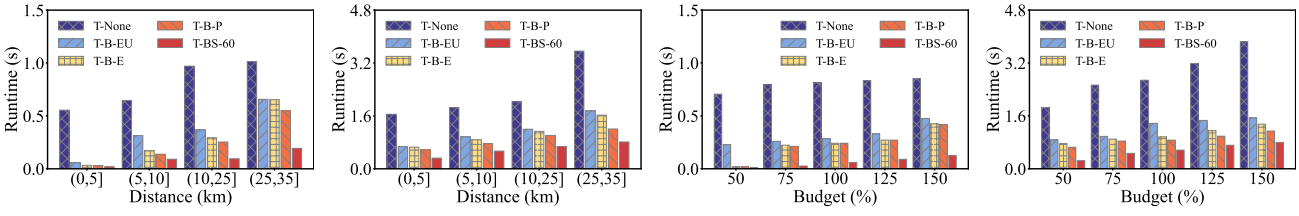

(a) Binary Heuristics, $D_1$ (b) Binary Heuristics, $D_2$ (c) Budget-Specific Heuristics, $D_1$ (d) Budget-Specific Heuristics, $D_2$

**Figure 4: Offline Building Binary Heuristics and Budget-Specific Heuristics**



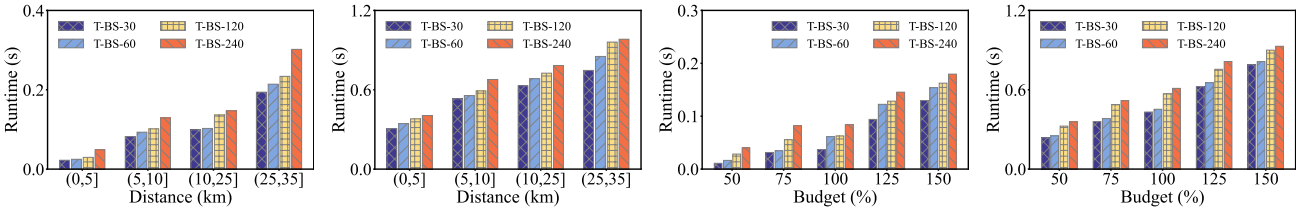

(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 5: Stochastic Routing with Binary Heuristics at Off-Peak Hours**



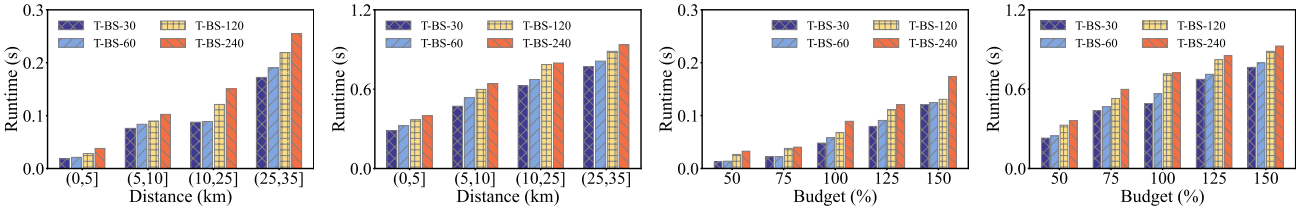

(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 6: Stochastic Routing with Binary Heuristics at Peak Hours**



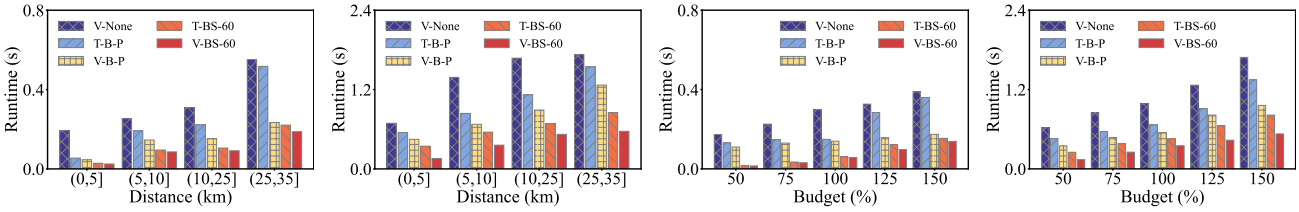

(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 7: Stochastic Routing with Budget-Specific Heuristics at Off-Peak Hours**



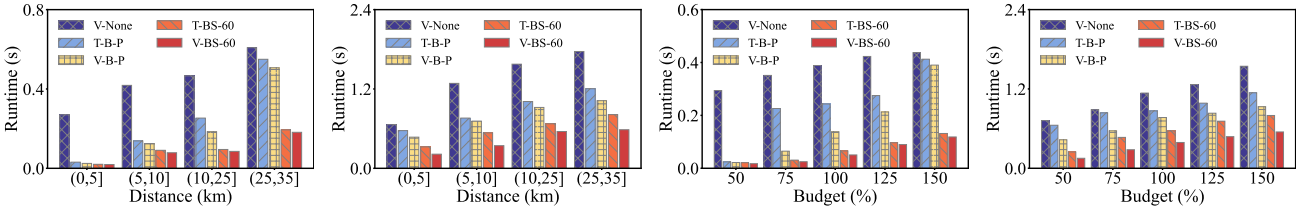

(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 8: Stochastic Routing with Budget-Specific Heuristics at Peak Hours**




(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 9: V-Path based Stochastic Routing at Off-Peak Hours**




(a) By distances, $D_1$ (b) By distances, $D_2$ (c) By budget values, $D_1$ (d) By budget values, $D_2$

**Figure 10: V-Path based Stochastic Routing at Peak Hours**