Appendix: Efficient Stochastic Routing in Path-Centric Uncertain Road Networks

Chenjuan Guo[1,2*], Ronghui Xu[1], Bing 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,xrhics@163.com,yeyuan.ds@gmail.com,{tungkvt,yanz,csj}@cs.aau.dk

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 COMPLEXITY ANALYSIS

2.1 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 $O((|\mathbb{E}'| + |\mathbb{P}'|) lg|\mathbb{V}|)$. 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.

2.2 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 η 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 η 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 $|\mathbb{V}| \cdot |\mathbb{V}| \cdot \eta$.

2.3 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.

3 OFFLINE PRE-COMPUTATION COSTS

We report the overall offline pre-computation cost for building binary heuristics for D_1 and D_2 in Table 1, where run time and storage is reported by hours (H) and megabytes (MB), respectively.

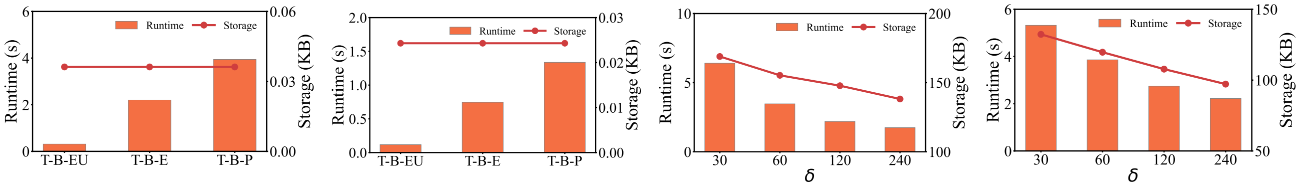
Data Set D_1	Off-Peak Hours			Peak Hours		
Methods	T-B-EU	T-B-E	T-B-P	T-B-EU	T-B-E	T-B-P
Run time (H)	1.8	17.6	32.3	2	19.2	34.7
Storage (MB)	1.14	1.14	1.14	1.14	1.14	1.14
Data Set D_2	**Off-Peak Hours**			**Peak Hours**		
	T-B-EU	T-B-E	T-B-P	T-B-EU	T-B-E	T-B-P
Run time (H)	0.56	5.1	8.9	0.67	5.5	10
Storage (MB)	0.66	0.66	0.66	0.66	0.66	0.66

Table 1: 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 2, where run time and storage is reported by hours (H) and gigabytes (GB), respectively.

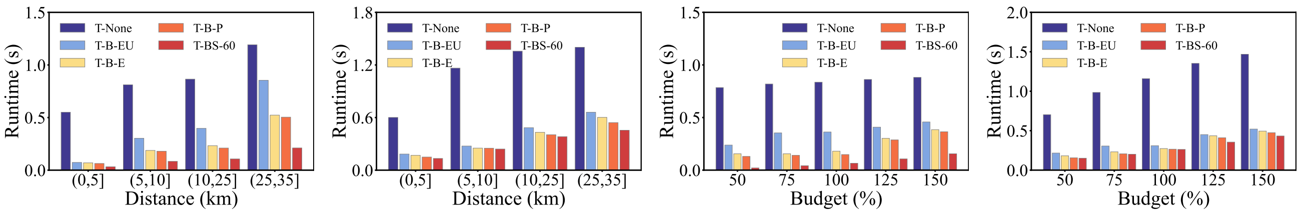
Data Set D_1	Off-Peak Hours				Peak Hours			
δ	30	60	120	240	30	60	120	240
Run time (H)	43.2	33.4	20.4	16.8	56	28.5	17.4	13.2
Storage (GB)	5.77	5.13	4.67	4.29	5.10	4.75	4.46	4.17
Data Set D_2	**Off-Peak Hours**				**Peak Hours**			
δ	30	60	120	240	30	60	120	240
Run time (H)	48.5	23.4	13.4	10.9	40.2	28.4	20.1	16.7
Storage (GB)	3.49	3.16	2.84	2.56	3.23	2.99	2.76	2.53

Table 2: Budget-Specific Heuristics Pre-computation



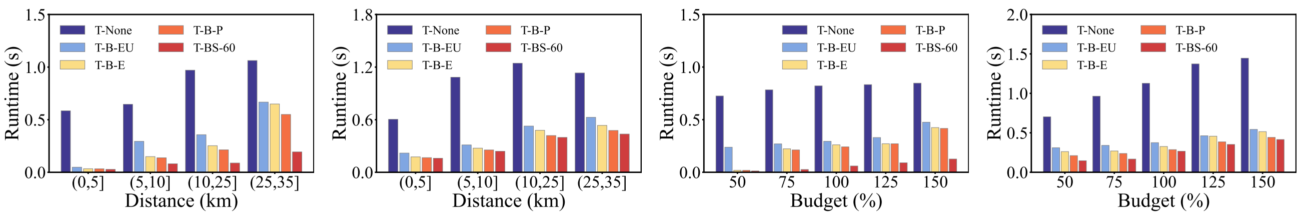
(a) Binary Heuristics, D_1 (b) Binary Heuristics, D_2 (c) Budget-Specific Heuristics, D_1 (d) Budget-Specific Heuristics, D_2

Figure 1: Offline Building Binary Heuristics and Budget-Specific Heuristics



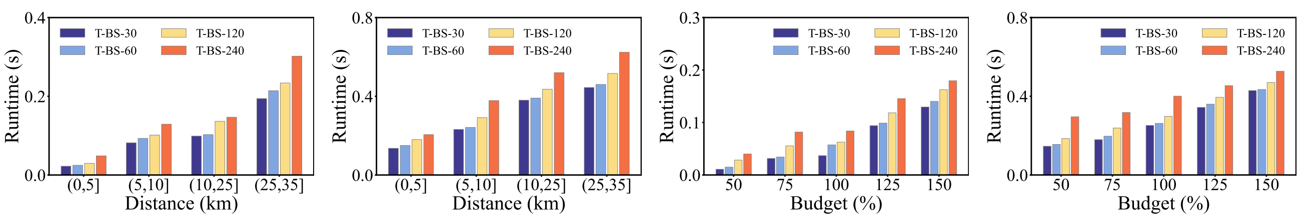
(a) By distances, D_1 (b) By distances, D_2 (c) By budget values, D_1 (d) By budget values, D_2

Figure 2: Stochastic Routing with Binary Heuristics at Off-Peak Hours



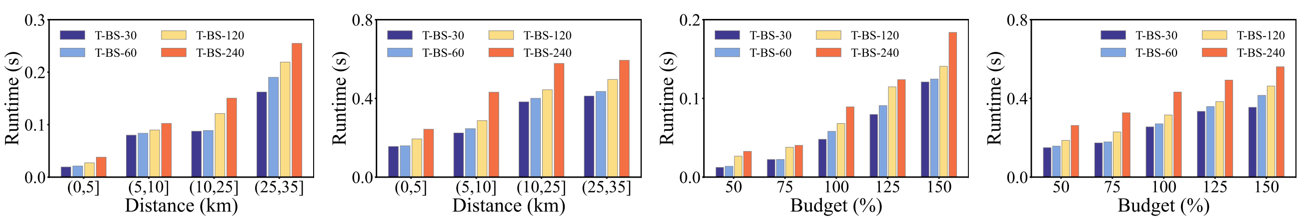
(a) By distances, D_1 (b) By distances, D_2 (c) By budget values, D_1 (d) By budget values, D_2

Figure 3: 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 4: 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

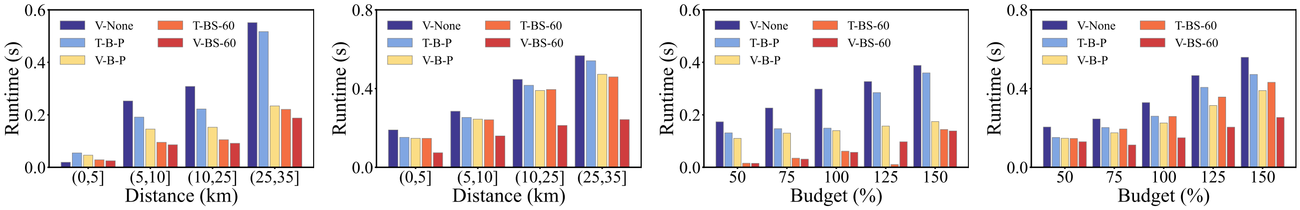
Figure 5: Stochastic Routing with Budget-Specific Heuristics at Peak Hours

4 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 Figure 2 and 3, respectively.

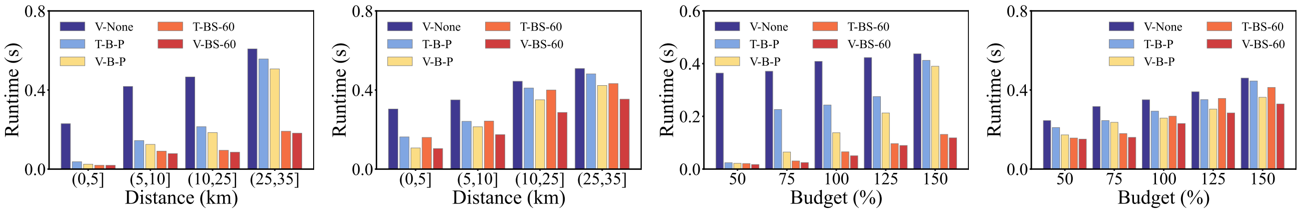
We provide experimental results for stochastic routing with budget-specific heuristics at off-peak and peak hours for data sets D_1 and D_2 in Figure 4 and 5, respectively.

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 Figure 6 and 7, respectively.



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

Figure 6: 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 7: V-Path based Stochastic Routing at Peak Hours