Electric Vehicle Routing for Emergency Power Supply with Deep Reinforcement Learning

Extended Abstract

Daisuke Kikuta
NTT Corporation
Tokyo, Japan
daisuke.kikuta@ntt.com

Hiroki Ikeuchi
NTT Corporation
Tokyo, Japan
hiroki.ikeuchi@ntt.com

Kengo Tajiri
NTT Corporation
Tokyo, Japan
kengo.tajiri@ntt.com

Yuta Toyama
NTT DOCOMO
Tokyo, Japan
yuuta.toyama.zg@nttdocomo.com

Masaki Nakamura
NTT DOCOMO
Tokyo, Japan
masaki.nakamura.ua@nttdocomo.com

Yuusuke Nakano
NTT Corporation
Tokyo, Japan
yuusuke.nakano@ntt.com

ABSTRACT

To maintain telecom services even during power outages, maintaining the power of the base stations is essential. Here, we consider a solution where Electric Vehicles (EVs) go around to directly supply their power to the base stations whose power is continuously decreasing. The goal is to find EV routes that minimize both total travel distance and the number of downed base stations. In this paper, we formulate this routing as a new variant of the Electric Vehicle Routing Problem (EVRP) and propose a solver that combines a rule-based vehicle selector and a reinforcement learning-based node selector. We evaluate our solver on synthetic datasets and real datasets. The results show that our solver outperforms baselines in terms of the objective value and computational time. See https://ntt-dkiku.github.io/rl-evrpeps for details (full paper, code, visualization, etc).

KEYWORDS

Electric Vehicle Routing Problem; Emergency Power Supply; Deep Reinforcement Learning

ACM Reference Format:
Daisuke Kikuta, Hiroki Ikeuchi, Kengo Tajiri, Yuta Toyama, Masaki Nakamura, and Yuusuke Nakano. 2024. Electric Vehicle Routing for Emergency Power Supply with Deep Reinforcement Learning: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024,* IFAAMAS, 3 pages.

1 INTRODUCTION

With natural disasters increasing [4], maintaining infrastructures during disasters is becoming more critical. As a telecoms company, our company has a mission to maintain telecoms services even during power outages caused by disasters. One of the most fundamental challenges here is maintaining the battery of telecoms base stations. Although each base station has a backup battery, maintaining the base station's power over extended periods requires power supply from external sources.

This paper addresses this challenge by leveraging Electric Vehicles (EVs) as the external source: EVs go around to supply their power to the base stations directly. In contrast to existing approaches [3, 7–9], this approach works without Vehicle-to-Grid systems, whose installation cost is high, and is effective for the case where the number of EVs is less than that of base stations.

Here, we formulate the base station relief as a new variant of the Electric Vehicle Routing Problem (EVRP) [2], termed EVRP for Emergency Power Supply (EVRP-EPS). We also propose a solver that combines a rule-based vehicle selector and a reinforcement learning-based node selector. In contrast to existing EVRPs, EVRP-EPS additionally considers the battery discharge of EVs, as well as mandatory details such as preparation/cleanup time and EV discharge limit. With reinforcement learning, our solver enables deriving reasonable routes within a short time.

2 PROBLEM SETTING

Objective. Given a time horizon T (i.e., expected blackout duration) and sets of base stations, charge stations, and EVs, the objective is to maintain as many base station batteries as possible during the time horizon while minimizing the total travel distance of all EVs. Formally, the objective function below is minimized.

$$\mathcal{L} = \sum_{k} \sum_{a=1}^{A_k} \frac{\mathrm{d}\left(\boldsymbol{x}_{\pi_k(a)}, \boldsymbol{x}_{\pi_k(a+1)}\right)}{N_{\mathrm{ev}}} + \alpha \frac{1}{T} \int_{t=0}^{T} \frac{\sum_i \mathrm{I}\left(\mathrm{bsb}_i^t = 0\right)}{N_{\mathrm{bs}}} dt, \tag{1}$$

where A_k is the number of k-th EV's actions, $\mathrm{d}(\cdot,\cdot)$ is the distance between two points, $\pi_k(a)$ is the index of node visited by k-th EV at a-th action, α is the positive weighting factor, bsb_i^t is the i-th base station's battery at the time t, $N_{\mathrm{bs/ev}}$ is the number of base stations/EVs, and $\mathrm{I}(\cdot)$ is the Boolean indicator function.

Action Space and Sub-actions. EVs cycle through an action (*move*) and three subsequent sub-actions (*prepare*, *discharge*/*charge*, and *clean-up*). The action space here is to determine which node EVs *move* to from the current nodes. The sub-actions are deterministically and automatically conducted depending on the state when EVs arrive at base/charge stations.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Table 1: Our solver v.s. baselines. Metrics are averaged travel distance per EV (dist (km)), the time average of # downed base stations (down), the objective value with $\alpha = 100$ (obj), and total computational time (time). The smaller, the better for all values.

	Syn-ev6 (100 samples)								Syn-ev12 (100 samples)							
	T = 12h				T = 24h				T = 12h				T = 24h			
Model	dist	down	obj	time	dist	down	obj	time	dist	down	obj	time	dist	down	obj	time
w/o EVs	-	20.1	-	-	-	33.3	-	-	-	20.1	-	-	-	33.3	-	-
Greed	189	17.7	37.2	3s	312	29.8	62.8	3s	192	15.5	32.9	1s	317	26.6	56.4	2s
Rand (S=12800)	142	15.3	**32.0**	1m	263	26.5	**55.5**	2m	157	11.8	**25.1**	3m	283	21.1	**45.1**	6m
Ours (G)	81	12.7	26.1	1s	127	25.5	52.3	2s	80	6.80	14.4	1s	134	18.2	37.7	2s
Ours (S=1280)	79	12.4	25.7	48s	125	24.8	50.9	1m	77	6.54	13.9	1m	134	17.4	36.2	2m
Ours (S=12800)	78	12.4	**25.6**	8m	124	24.7	**50.7**	14m	76	6.50	**13.8**	12m	134	17.3	**35.9**	21m

	Real-ev6 (1 sample)								Real-ev12 (1 sample)							
	T = 12h				T = 24h				T = 12h				T = 24h			
Model	dist	down	obj	time	dist	down	obj	time	dist	down	obj	time	dist	down	obj	time
w/o EVs	-	8.25	-	-	-	17.5	-	-	-	11.3	-	-	-	25.9	-	-
Greed	30	5.87	19.4	1s	37	13.5	42.8	1s	89	9.34	21.7	1s	108	21.0	47.3	1s
Rand (S=12800)	32	3.74	**13.1**	2s	63	10.0	**33.8**	2s	128	4.21	**11.2**	5s	207	14.0	**33.7**	9s
Tsn ($\Delta t = 1.0$)	14	5.77	18.3	32s	52	12.2	40.1	30m	60	4.08	9.83	6s	113	13.9	32.0	9m
Tsn ($\Delta t = 0.5$)	18	3.36	**11.2**	3m	33	10.2	**32.9**	30m	55	2.88	**7.15**	35s	134	13.0	**30.3**	30m
Ours (G)	15	3.19	10.5	1s	25	8.73	27.8	1s	56	0.99	3.05	1s	92	11.7	27.0	1s
Ours (S=1280)	17	2.19	7.57	1s	28	8.12	26.1	2s	59	0.62	2.28	4s	110	10.9	25.6	7s
Ours (S=12800)	16	2.09	**7.20**	9s	30	8.03	**26.0**	15s	57	0.54	**2.09**	29s	101	11.0	**25.5**	48s

3 METHODOLOGY

Our solver generates EV routes by repeating the two phases: the vehicle selector selects an EV; the node selector determines the next destination node (base/charge station) of the selected EV. In the following, we describe the details of each selector.

Vehicle selector. In the action/sub-action cycle, EVs can start to move to the next destination only after finishing the *clean-up*. Therefore we employ a rule-based vehicle selector that always selects an EV finishing clean-up the soonest from the current time.

Node selector. After applying a linear projection to input features (the environment state of when an EV is selected) $\boldsymbol{h}^{(0)}$, the node selector produces the final embeddings of nodes and EVs by stacking L of two-tower Transformer encoders:

$$\boldsymbol{h}^{(l)}_{\mathrm{node}_n} = \mathrm{X{\small FMR}}^{(l)}_{\mathrm{node}_n}\left(\boldsymbol{h}^{(l-1)}_{\mathrm{node}_1}, \ldots, \boldsymbol{h}^{(l-1)}_{\mathrm{node}_N}\right), \tag{2}$$

$$\boldsymbol{h}^{(l)}_{\mathrm{ev}_k} = \mathrm{X{\small FMR}}^{(l)}_{\mathrm{ev}_k}\left(\boldsymbol{h}^{(l-1)}_{\mathrm{ev}_1}, \ldots, \boldsymbol{h}^{(l-1)}_{\mathrm{ev}_K}\right), \tag{3}$$

where $\mathrm{X{\small FMR}}^{(l)}_{\mathrm{node}_n}, \mathrm{X{\small FMR}}^{(l)}_{\mathrm{ev}_k}$ are the l-th Transformer encoders [5] for nodes and EVs, of which subscript indicate the output element. Note that no positional encoding is used here as nodes and EVs are permutation-invariant.

Finally, the visit probability for each node is computed from the scaled dot-product attention between the final embeddings of nodes and the selected EVs, as follows.

$$u(k,n) = \begin{cases} C \cdot \tanh\left(\dfrac{\boldsymbol{q}^{\top}_{\mathrm{ev}_k}\boldsymbol{k}_{\mathrm{node}_n}}{\sqrt{d_{\mathrm{k}}}}\right) & \text{if node}_n \text{ is feasible,} \\ -\infty & \text{otherwise,} \end{cases} \tag{4}$$

$$p_\theta\left(\Pi_{\bar{k}}(a) = n | \boldsymbol{h}^{(0)}, \bar{k}\right) = \frac{e^{u(\bar{k},n)}}{\sum_m e^{u(\bar{k},m)}}, \tag{5}$$

where $C(= 10)$ is the clipping width, the query $\boldsymbol{q}_{\mathrm{ev}_k} = W^Q \boldsymbol{h}^{(L)}_{\mathrm{ev}_k}$, the key $\boldsymbol{k}_{\mathrm{node}_n} = W^K \boldsymbol{h}^{(L)}_{\mathrm{node}_n}$, d_{k} is the dimension of the key, W^Q, W^K are trainable projection matrices, and $\bar{k}$ is the index of the selected EV. In terms of node feasibility, a base station is feasible if no other EVs visit it, and the selected EV can return to a charging station without running out of battery after visiting that base station. A charge station is feasible if the selected EVs can reach it from the current node without running out of battery.

The parameterized policy p_θ is trained by REINFORCE [6] with a greedy rollout baseline [1] so that Eq. (1) is minimized. For the decoding (i.e., route generation), we employ the sample decoding.

4 EXPERIMENTS

Setups. We evaluate our solver on synthetic datasets (Syn-ev6, Syn-ev12) and real datasets (Real-ev6, Real-ev12). The baselines in the evaluation are two naive approaches (Greed and Rand) and a constraint programming solver on a time-space network (Tsn). Greed and Rand replace the node selector of our solver with a greedy node selection that selects a base station with the lowest battery and a random node selection, respectively. Tsn solves subproblems divided by a heuristic clustering, meaning that solutions derived by Tsn are near-optimal. We set the time limit for solving EVRP-EPS to 30 minutes according to the actual requirements.

Results. Table 1 shows the evaluation results. We report the results of Tsn with two discrete time resolutions $\Delta t = 0.5, 1$h and those of our solver with different decoding: greedy (G) and sampling decoding (S=#samples). Overall, our solver consistently outperforms the baselines in terms of the objective value and computational time. In particular, S=12800 provides the minimum objective value in all cases. Regarding computational scalability, Tsn exhibits an exponential increase in computational time as the time horizon is doubled, whereas our solver restricts the escalation to a linear increment. Our solver also shows the scalability for the increase of the number of nodes and EVs (see the full paper), demonstrating the capability of handling larger-scale situations (i.e., a longer time horizon and more nodes/EVs) within a short time.

5 DISCUSSION

The experimental results reveal that our solver (reinforcement learning-based solver) is effective in solving the complicated EVRP within a limited time. On the other hand, some limitations remain, including the balance of travel distance among EVs and the estimation of travel time. We will address them by introducing a route-balancing strategy and considering actual travel distance and uncertainty of travel time due to traffic situations. In terms of unavailable roads due to a disaster, we may handle them by obtaining real-time road conditions from a provider and masking unavailable roads. We plan to validate the ideas in a demonstration experiment.

REFERENCES

[1] Wouter Kool, Herke van Hoof, and Max Welling. 2019. Attention, Learn to Solve Routing Problems!. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByxBFsRqYm

[2] Ilker Kucukoglu, Reginald Dewil, and Dirk Cattrysse. 2021. The electric vehicle routing problem and its variations: A literature review. *Computers Industrial Engineering* 161 (2021), 107650. https://doi.org/10.1016/j.cie.2021.107650

[3] Qixing Liu, Peng Xu, Yuhu Wu, and Tielong Shen. 2023. A two-stage algorithm for vehicle routing problem with charging relief in post-disaster. *IET Intelligent Transport Systems* 17, 8 (2023), 1525–1543. https://doi.org/10.1049/itr2.12344 arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/itr2.12344

[4] Vision of Humanity. 2022. Ecological threat report 2022: Analysing Ecological Threats, Resilience Peace. (2022).

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[6] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* 8, 3–4 (may 1992), 229–256. https://doi.org/10.1007/BF00992696

[7] Ning Zhou Xu, Ka Wing Chan, Chi Yung Chung, and Ming Niu. 2020. Enhancing Adequacy of Isolated Systems With Electric Vehicle-Based Emergency Strategy. *IEEE Transactions on Intelligent Transportation Systems* 21, 8 (2020), 3469–3475. https://doi.org/10.1109/TITS.2019.2929767

[8] N. Z. Xu and C. Y. Chung. 2016. Reliability Evaluation of Distribution Systems Including Vehicle-to-Home and Vehicle-to-Grid. *IEEE Transactions on Power Systems* 31, 1 (2016), 759–768. https://doi.org/10.1109/TPWRS.2015.2396524

[9] Zexin Yang, Tian Gao, Yu Liu, Xueliang Huang, Shan Gao, Qi Zhao, and Hongen Ding. 2023. A two-stage pricing strategy for electric vehicles participating in emergency power supply for important loads. *Electric Power Systems Research* 218 (2023), 109239. https://doi.org/10.1016/j.epsr.2023.109239