Heterogeneous mission planning of UAVs with attention-based deep reinforcement learning

Minjae Jung [1], **Hyondong Oh** [Corresp. 1]

[1] Mechanical Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea

Corresponding Author: Hyondong Oh
Email address: h.oh@unist.ac.kr

Large-scale and complex mission environments require UAVs to deal with various types of missions while considering their operational and dynamic constraints. This paper proposes a deep learning-based heterogeneous mission planning algorithm for a single UAV. We first formulate a heterogeneous mission planning problem as a vehicle routing problem (VRP). Then, we solve this by using an attention-based deep reinforcement learning approach. Attention-based neural networks are utilized as they have powerful computational efficiency in processing the sequence data for the VRP. For the input to the attention-based neural networks, the unified feature representation on heterogeneous missions is introduced, which encodes different types of missions into the same-sized vectors. Besides, a masking strategy is introduced to be able to consider the resource constraint (e.g., flight time) of the UAV. Simulation results show that the proposed approach has significantly faster computation time than that of other baseline algorithms while maintaining a relatively good performance.

Heterogeneous Mission Planning for a single UAV with Attention-Based Deep Reinforcement Learning

Minjae Jung and Hyondong Oh[1]

[1]**Ulsan National Institute of Science and Technology (UNIST), Ulsan 44919, Republic of Korea**

Corresponding author:

Hyondong Oh[1]

Email address: h.oh@unist.ac.kr

ABSTRACT

Large-scale and complex mission environments require UAVs to deal with various types of missions while considering their operational and dynamic constraints. This paper proposes a deep learning-based heterogeneous mission planning algorithm for a single UAV. We first formulate a heterogeneous mission planning problem as a vehicle routing problem (VRP). Then, we solve this by using an attention-based deep reinforcement learning approach. Attention-based neural networks are utilized as they have powerful computational efficiency in processing the sequence data for the VRP. For the input to the attention-based neural networks, the unified feature representation on heterogeneous missions is introduced, which encodes different types of missions into the same-sized vectors. Besides, a masking strategy is introduced to be able to consider the resource constraint (e.g., flight time) of the UAV. Simulation results show that the proposed approach has significantly faster computation time than that of other baseline algorithms while maintaining a relatively good performance.

INTRODUCTION

Recently, mission environments such as disaster management or logistics services become more complex and larger. The goal of these large-scale missions can be achieved safer and faster using unmanned aerial vehicles (UAVs) (Shakhatreh et al., 2019; Grzybowski et al., 2020; Kim et al., 2021). Since the complexity of task allocation by scheduling a large number of tasks in the mission to UAVs is high, it takes a long time for the human operator to plan these tasks manually without ensuring the optimal performance. The performance and computational time significantly impact the success rate of rescue in disaster management or the benefits of companies in logistics services (Atyabi et al., 2018). Therefore, autonomous mission planning algorithms need to be developed to solve these problems rapidly and efficiently.

Mission planning problems of the UAV can be represented as one of vehicle routing problems (VRPs). The VRP has various variations such as distance constraint (Karaoglan et al., 2020), multiple trip availability (Paradiso et al., 2020), and asymmetricity of costs (Ban and Nguyen, 2021) among many others. Thus, the VRP can represent some

39   of the mission planning problems for the UAV, which is the complex combinatorial
40   optimization problem. In addition, there are various studies to solve these VRP variations
41   in operation research or transportation research fields (Kumar and Panneerselvam, 2012).
42   Therefore, when representing and solving the mission planning problem of the UAV
43   as one of the VRPs while considering the characteristics of the UAV, it is possible to
44   develop more realistic mission planning algorithms with the help of VRP studies.

45   The VRP is a combinatorial optimization problem expressed in a graph form and
46   there are i) exact solvers, ii) heuristic algorithms, and iii) machine learning-based
47   approaches to solve the problem. The exact solver approach can obtain the optimal
48   solution with the branch and bound algorithm (Laporte and Nobert, 1983; Toth and
49   Vigo, 2002; Larrain et al., 2019) or dynamic programming (Secomandi, 1998; Mingozzi
50   et al., 2013). Although the exact solver approach can achieve the optimal cost, the
51   computation time grows exponentially with the scale of the problem. The heuristic
52   algorithm approach finds a feasible solution much faster than the exact approach. There
53   are various heuristic algorithms to solve the VRP such as variable neighborhood search
54   (Bräysy, 2003; Kytöjoki et al., 2007; Hemmelmayr et al., 2009), tabu search (Gendreau
55   et al., 1994; Fu et al., 2005; Qiu et al., 2018), and genetic algorithm (Baker and Ayechew,
56   2003; da Costa et al., 2018; Ruiz et al., 2019). Although the heuristic algorithms provide
57   reasonable performance, they need to be designed carefully for different problem setting,
58   which is often challenging and requires expert knowledge. The machine learning-based
59   approach utilizes data to train model parameters. If the model can approximate the
60   solver which maps input and output of the combinatorial optimization problem, it can
61   be trained flexibly with different setting of the problem given the sufficient amount
62   of data (Khalil et al., 2017). Besides, after training the model, fast computation time
63   can be achieved. Using the machine learning approach is a good option when data is
64   available thanks to its advantages of fast and flexible calculations; hence, this paper
65   adopts machine learning approach for UAV mission planning problems.

66   Among the machine learning approaches, supervised learning and reinforcement
67   learning can be considered to tackle the VRP. Vinyals et al. (2015) proposed a neural
68   network model, called the Pointer network, that approximates the solver of combinatorial
69   optimization. It uses the attention mechanism and the recurrent neural network (RNN)-
70   based encoder-decoder structure with supervised learning. However, supervised learning
71   needs a large amount of labeled data obtained from exact solvers that require significant
72   time for data generation. For this reason, the reinforcement learning approach that
73   generates data while interacting with the environment during the training process is often
74   preferred for the VRP (Bello et al., 2017; Kool et al., 2019; Mazyavkina et al., 2021).
75   Bello et al. (2017) proposed a reinforcement learning algorithm to solve combinatorial
76   optimization problems that uses the Pointer network structure from (Vinyals et al., 2015),
77   and then optimizes the model with the policy gradient algorithm. Kool et al. (2019)
78   utilized the Transformer (Vaswani et al., 2017) style-neural network model that modifies
79   the RNN structure of the Pointer network into multi head attention (MHA) network.
80   This model is proposed to solve various types of routing problems with its flexibility,
81   and the authors show that the model outperforms some heuristic algorithms and Pointer
82   network-based model in terms of the performance and the computation time; hence, we
83   build our approach based upon this MHA network.

84   The proposed approach in this study particularly considers the characteristics of the

UAV, which are the capability of handling heterogeneous missions and the flight time constraint. As UAV technology development continues, several tasks can be carried out by even a single UAV simultaneously or sequentially, such as delivering extra payloads, visiting specific areas to take images of landmarks, or flying over large areas to obtain information. Considering these heterogeneous tasks, the cost for completing each pair of tasks becomes different depending on the order of task completion. For instance, different path lengths for delivery or radius of the area for coverage make the cost matrix of the VRP asymmetric. Another characteristic of a UAV is that they have limited flight time due to fuel/battery capacity. This constraint makes a UAV refuels/recharges their fuel/battery at the depot and resume their work. Therefore, the mission planning problem of a UAV in this paper is represented as the multi-trip asymmetric distance constrained VRP (MTAD-VRP) which is one of the variations of the VRP. The heterogeneous mission planning considering these characteristics further increases the complexity of the VRP.

It is worthwhile noting that there are a few studies on heterogeneous mission planning problems for the UAV using heuristic optimization algorithms. Zhu et al. (2018) formulated the heterogeneous mission planning problem for UAV reconnaissance as multiple Dubins travelling salesman problem (MDTSP), which is one of VRPs and proposes the genetic algorithm-based approach to solve the problem. Chen et al. (2019) considered an additional constraint which is time window and formulates the heterogeneous mission planning problem as a multi-objective, multi-constraint nonlinear optimization problem. Then, they utilize the search-based algorithm for optimization. Gao et al. (2021) proposes ant colony-based algorithm for minimizing the weighted sum of the total UAV fuel consumption and the task execution time. The performance of the proposed algorithm is compared with other ant colony-based algorithms through numerical simulations. However, to our best knowledge, it is difficult to find the heterogeneous mission planning based on reinforcement learning approaches. As mentioned earlier, reinforcement learning-based approaches are expected to provide the superior performance compared with heuristic algorithms in terms of computation time and optimality. Besides, aforementioned works consider only single-trip problems which significantly limit the capability of the UAV.

To this end, this study proposes an attention-based reinforcement learning algorithm for the heterogeneous mission planning for a single UAV. We first formulate the heterogeneous mission planning problem as the MTAD-VRP expressed in a graph form to utilize the solvers for the VRP. Considering a realistic complex mission environment and characteristics of the UAV, we use the reinforcement learning approach with an attention-based neural network model to solve the problem with its fast computation time and flexibility. Although the existing learning-based algorithms can deal with various routing problems, most of them only consider homogeneous inputs (Vinyals et al., 2015; Bello et al., 2017; Kool et al., 2019). Thus, we introduce the unified mission representation for network inputs to contain the information of heterogeneous missions. And then, we design the masking strategy to deal with the flight time constraint to complete all tasks in a mission and help the training process of reinforcement learning. The proposed algorithm uses the MHA-based model architecture for better computational efficiency than that of the RNN-based model architecture while preventing the vanishing gradient effect when dealing with long data sequences. Furthermore, the MHA-based model has

131 a permutation invariant property that makes the model to be able to learn the robust
132 strategy regardless of input permutation. The REINFORCE algorithm (Sutton et al.,
133 2000) with a baseline updates the model to converge stably by reducing the variance of
134 the parameters' gradient. To validate the feasibility and the performance of the proposed
135 approach, we perform numerical simulations and compare the result with state-of-the-art
136 open-source heuristic algorithms.

PROBLEM DEFINITION

138 This study considers visiting, coverage, and delivery as heterogeneous missions. Here,
139 visiting is for capturing an image of the landmark building, coverage is for gathering
140 information of a large area with the spiral flight pattern, and delivery is for picking
141 and placing the package. Figure 1 illustrates heterogeneous missions. Note that, if
142 needed, more mission types could be readily incorporated into the problem thanks to the
143 flexibility of the learning approach.

144 Our purpose is to complete all the given heterogeneous mission while minimizing the
145 flight time of a single UAV dispatched from the depot. The flight time budget constraint
146 should be satisfied, and the UAV is allowed to return to the depot for recharging. Figure
147 2 shows a sample mission scenario in a 2-D view, where the black squares are the depot,
148 blue squares are visiting mission spots, circles are coverage mission areas, and a pair of
149 magenta diamonds with cyan arrows are the delivery mission with a specific direction.

To formulate the heterogeneous mission planning problem as the mathematical
formulation of the VRP, we abstract the problem into a graph instance. The mission graph
$G = (V, E)$ consists of k nodes $(v_1, v_2, \cdots, v_k) \in V$ and edges $(e_{12}, e_{21}, \cdots, e_{k1}, e_{1k}) \in E$.
Nodes represent the feature of each mission and the value of edges are constructed with
the travel time cost which depends on the type of mission. The solution of the problem
is the sequence of the index of nodes $\Omega = (n_1, \cdots, n_t)$, where $n_t \in \mathbf{N}$ and $1 \leq n_t \leq k$.
The total travel time cost L, which is the objective of the problem is the sum of every
value of edges between the selected nodes and the cost of returning to the depot as:

$$L = \sum_{t=1}^{k-1} e_{n_t n_{t+1}} + e_{n_k n_1} \quad n_t \in \Omega. \tag{1}$$

150 Assuming that the UAV flies with a constant velocity, the travel time cost between
151 missions is calculated by the total distance that the UAV need to fly. We ignore the time
152 of recharging and loading packages for simplicity. The type of mission affects the cost
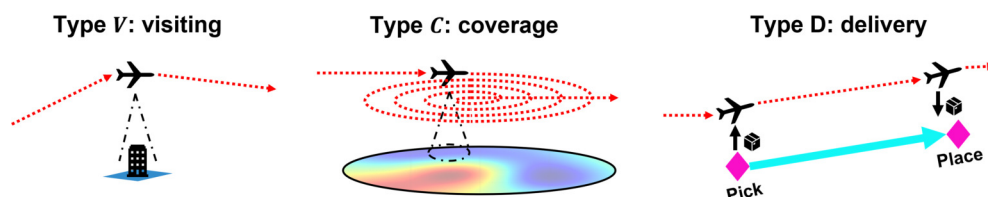
Figure 1. Illustration of heterogeneous missions.
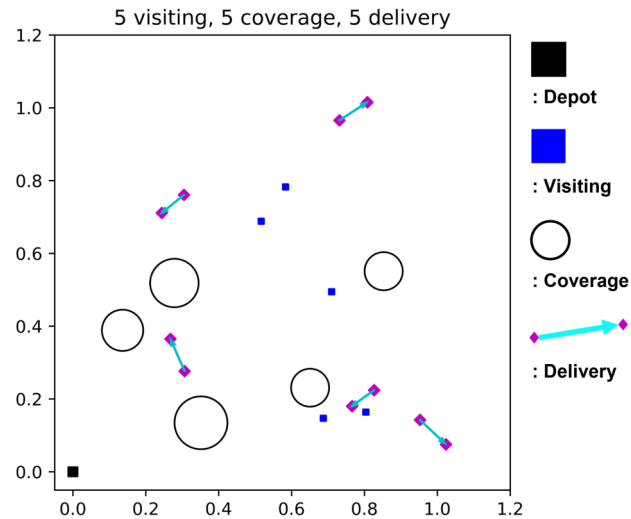
Figure 2. A sample mission scenario with 5 visiting, 5 coverage, and 5 delivery missions.
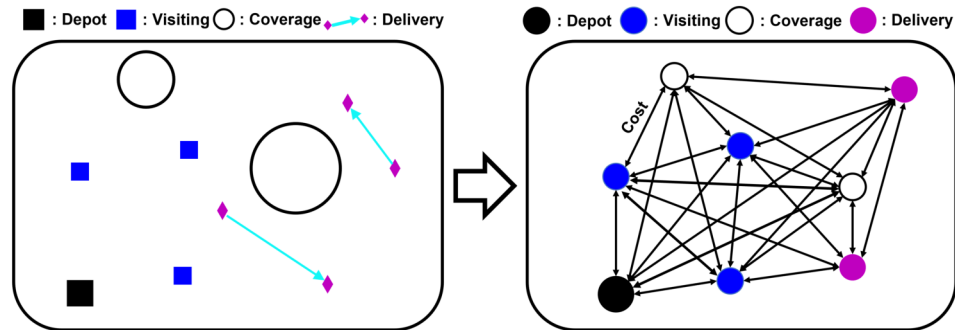
Figure 3. Visualization of abstracting a mission instance as a graph instance.

calculation as:

$$c_{xv} = d, \tag{2}$$
$$c_{xc} = d + S, \tag{3}$$
$$c_{xd} = d + l, \tag{4}$$

where $S = \pi r^2 / w$, c_{xv}, c_{xc}, and c_{xd} are the travel cost to the visiting, coverage, and delivery mission point, respectively, from the source mission point x. d is the distance between missions, S is the length of the spiral path to cover the area, w is the sensing range of the UAV, r is the radius of the coverage area, and l is the length of the delivery path. The cost for returning to the depot is the same as c_{xv} with the visiting mission point of the depot. Figure 3 provides the conversion of the mission instance to the graph representation.

Additionally, we consider the limited flight time constraint of the UAV for safe mission completion. Typically, the UAV can be recharged at the base station which
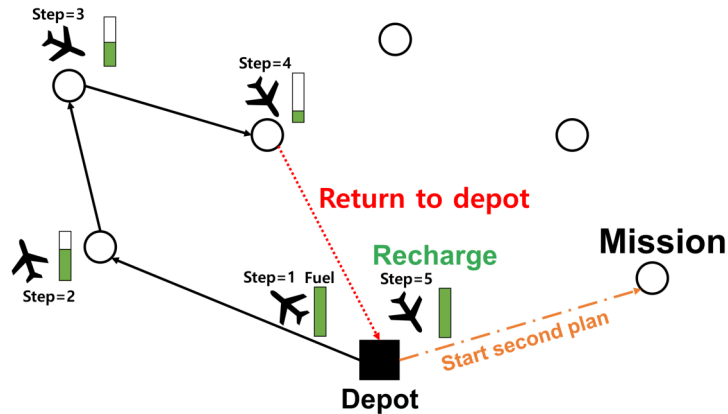
Figure 4. Illustration of recharging the UAV.

is considered as the depot in the VRP. Thus, we allow the UAV to be recharged by revisiting the depot. Figure 4 illustrates the recharging event graphically.

ATTENTION BASED DEEP REINFORCEMENT LEARNING

In this section, we first propose a unified feature representation to deal with heterogeneous missions. Then, we suggest a masking strategy to consider the flight time limitation constraint. We introduce the neural network model and reinforcement learning algorithm to solve the heterogeneous mission planning problem using these methods. The neural network model architecture consists of an encoder and decoder network with the attention mechanism for sequential data. The REINFORCE algorithm (Sutton et al., 2000), one of the reinforcement learning algorithms, is used to optimize the neural network.

Unified Feature Representation

We propose the unified feature representation $v = (x_1, y_1) || (x_2, y_2) || A || I_{Type}$ combining the spatial information of heterogeneous missions and indicator of each mission type, where (x_1, y_1) is the critical position of the mission, (x_2, y_2) is the end-position, A is the area information, and I_{Type} is the one-hot encoding indicator of each type. The operator $||$ means concatenation. The critical position represents the important position of each mission such as the position of the visiting mission, center position of the area of the coverage mission, and picking position of the delivery mission. The end position represents the position when the UAV completes the current mission. The delivery mission has a different end position while the others have the same end position as the critical position. The area information represents the radius of an area for the coverage mission, the length of the delivery mission, or zero for the visiting mission. The depot has the same representation as the visiting mission except I_{Type}. I_{Type} represents the type information of each mission, where $(0,0,0), (0,0,1), (0,1,0)$ and $(1,0,0)$ represent the depot, the visiting mission, the coverage mission, and the delivery mission, respectively.
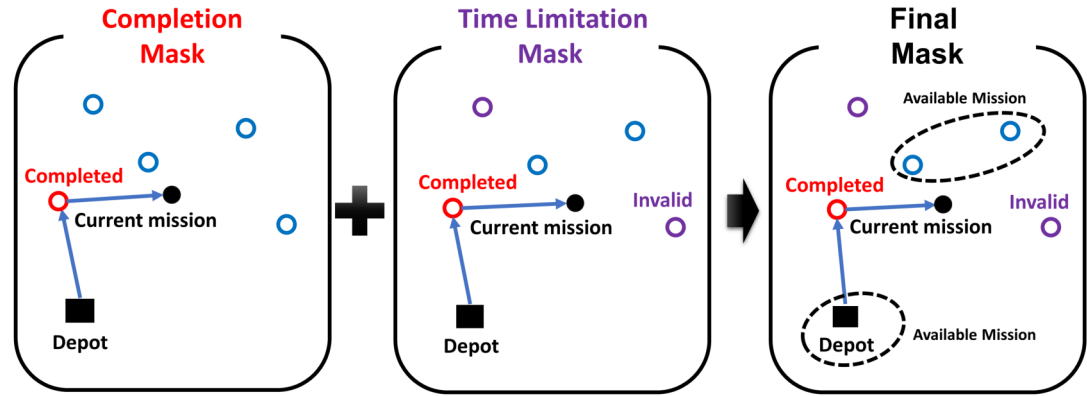
Figure 5. The example of the masking strategy.

Masking Strategy

The masking strategy generates the mask M to prevent selecting invalid actions in reinforcement learning. The mask consists of the completion mask $M_C = (m_{c_1}, m_{c_2}, \cdots, m_{c_k})$ for already completed missions and the time limitation mask $M_T = (m_{t_1}, m_{t_2}, \cdots, m_{t_k})$. The time limitation mask M_T masks the mission when the flight time of returning to the depot after completing the mission exceeds the remained flight time of the UAV, expressed as:

$$
m_{t_j} = \begin{cases} 1 & \text{if} \quad T_j + T_{Return,j} > T_{Remain} \\ 0 & \text{otherwise} \end{cases} \quad , \qquad (j = 1, \cdots, k) \tag{5}
$$

where T_j is the time to complete the mission j from the current mission, $T_{Return,j}$ is the flight time of returning to the depot time after completing the current mission, and T_{Remain} is the remained flight time of the UAV. The masking strategy generates the mask $M = M_C | M_T$, where the operator $|$ means the element-wise logical 'or' operation. Figure 5 shows an example of the masking strategy. Each circle in the figure represents an arbitrary mission. The completion mask M_C and the time limitation mask M_T are represented as red circles and purple circles, respectively. The agent in the example only can select unmasked missions to complete or the depot to return.

If every element of M_T is masked before finishing the whole mission, the UAV is forced to return to the depot for refueling/recharging itself. When the UAV arrived at the depot, the remained flight time of the UAV is initialized, and then every element of M_T is calculated by (5). After that, the UAV continues the subsequent tasks. Note that time for recharging is not explicitly considered in the cost; we assumed that the recharging can be done quickly by replacing the battery with a new one. However, if needed, we could easily include the recharging cost in the optimization problem formulation.

Model Architecture

The neural network model π_θ which approximates the VRP solver is parameterized with θ. The policy with the model can be represented as the probability $p_{\pi_\theta}(\Omega | s)$, where $s = (v_1, v_2, \cdots, v_k)$ is the given mission nodes as the input of the model and $\Omega = (n_1, n_2, \cdots, n_k)$ is the output of the model, which is the permutation of the index of

s. With the chain rule, the probability can be factorized as:

$$p_{\pi_\theta}(\Omega \mid s) = \prod_{t=1}^{k} p_{\pi_\theta}(\Omega_t \mid s, \Omega_{1:t-1}), \tag{6}$$

where Ω_t is the output value at $t \in \{1, \cdots, k\}$ and $\Omega_{1:t-1}$ is the partial sequence of Ω.

We utilize the Transformer style model architecture of (Kool et al., 2019) to approximate Eq. (6). The model takes the input which is a set of mission node data for the encoder and outputs the solution sequence with the decoder while satisfying the constraints. The encoder is implemented with multi-head attention (MHA) layers (Vaswani et al., 2017), and it generates the embeddings of each input element $(h_{e1}, h_{e2}, \cdots, h_{ek})$ where the embeddings represents the relationship among all of the other elements in the input mission nodes. To generates the embeddings with the encoder, the MHA layer utilizes query $Q_i = w_q v_i$, key $K_i = w_\kappa v_i$, and value $V_i = w_v v_i$ vectors, where w_q, w_κ, and w_v are linear layers for projecting mission node features. The attention mechanism inferences the relationship between query and key by calculating the attention score as:

$$u_{ij} = \frac{Q_i^T \cdot K_j}{\sqrt{d}}, \qquad (j = 1, \cdots, k) \tag{7}$$

where u_{ij} is the attention score and d is the embedding size. The attention score represents the similarity between Q_i and K_i. Using the attention score, the embedding h_{ei} of v_i and the context vector h_c are calculated as:

$$a_{ij} = softmax(u_{ij}), \tag{8}$$

$$h_{ei} = \sum_{j=1}^{k} a_{ij} V_j, \tag{9}$$

$$h_c = \frac{1}{k} \sum_{n=1}^{k} h_{en}. \tag{10}$$

Note that the calculation of the attention mechanism is parallelized by the heads which are part of the MHA layer. In this study, the encoder consists of 3 MHA layers with 8 heads and 128 embedding sizes. Figure 6 shows the embedding process of the encoder.

At each decoding step t, the decoder embeds the outputs from the encoder into $(h'_{e1}, h'_{e2}, \cdots, h'_{ek})$ with a MHA layer. Then, decoder selects the next node of the solution with the attention mechanism as described in (Vinyals et al., 2015). In this case, the query vector Q' consists of the context vector h_c, the partial solution information $\Omega_{1:t}$ which is abstracted as $\Omega' = (h_{en_1}, h_{en_t})$, where $n_t \in \Omega_{1:t}$, and the remained flight time budget T_{Remain} for considering the flight time budget constraint. Note that Ω' is initialized as $(0,0)$ before selecting the first mission to complete and then updated as (h_{en_1}, h_{en_t}), where h_{en_1} is the embedding of the first solution node and h_{en_t} is the embedding of the last solution node. This is because the agent of the VRP only needs to consider the uncompleted missions with respect to the last completed mission regardless of completed missions (Kool et al., 2019). Then, the probability of selecting each mission node is obtained by the attention score between the embeddings $(h'_{e1}, h'_{e2}, \cdots, h'_{ek})$, Q', and mask
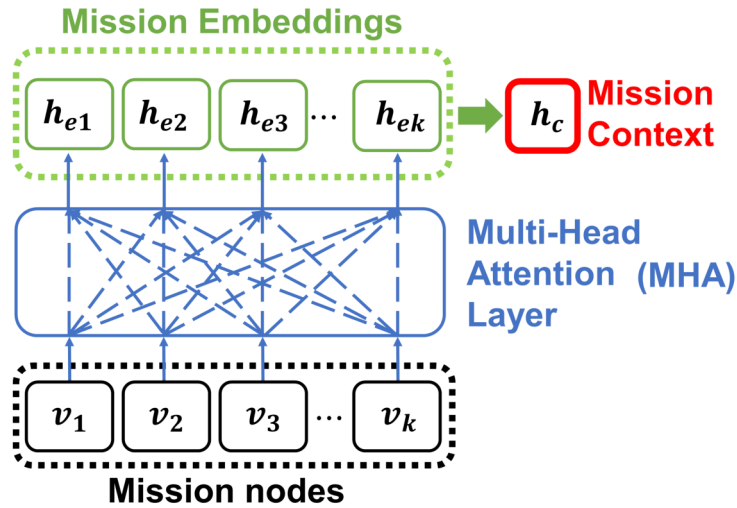
Figure 6. The encoder embeds the input mission nodes into embeddings with MHA.
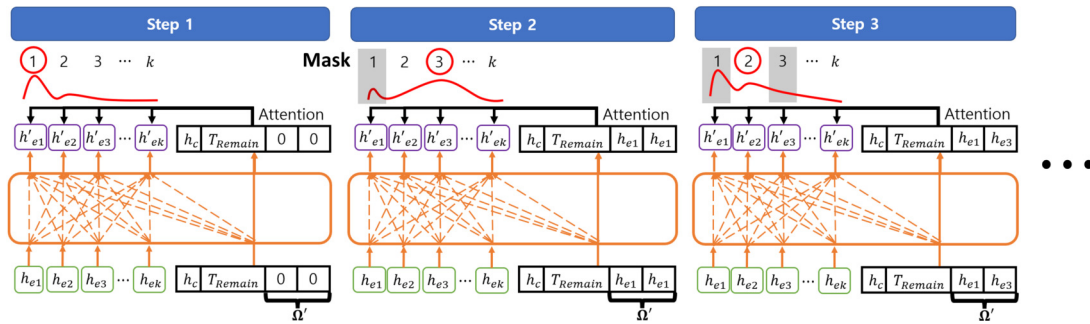
Figure 7. The example of decoding steps.

225   *M* from the masking strategy as:

$$u'_i = \begin{cases} -\infty & \text{if} \quad m_i = 1 \\ \dfrac{Q^T \cdot h'_{ei}}{\sqrt{d'}} & \text{otherwise} \end{cases}, \tag{11}$$

$$a' = softmax(u'), \tag{12}$$

226   where u'_i, d' and a' are the attention score, embedding size of the decoder, and the
227   probability of selecting each mission, respectively. The next solution n_t is selected by
228   sampling from the probability distribution a' and added to the last index of $\Omega_{1:t-1}$ to
229   construct $\Omega_{1:t}$. After selecting the next solution, T_{Remain} is reduced by the completion
230   time of the selected mission and Ω' is updated with $\Omega_{1:t}$. In this study, the decoder
231   consists of 1 MHA layer with 1 head and 128 embedding size. Figure 7 shows the
232   example of decoding steps.

REINFORCE with Baseline

234   To update the neural network model, we use the REINFORCE algorithm (Sutton et al.,
235   2000). In the Markov decision process (MDP) tuple $< s, a, r, \tau, \pi >$ for reinforcement

learning, the state s is the mission state, the action a is the selected mission from the agent policy $\pi_\theta(a|s)$ which is the neural network model parameterized with θ, the reward r is the cost in Eq. (1), and the transition probability $\tau = p(s'|s,a)$ is the next state after selecting a given s. Note that τ is deterministic in this work.

Since the REINFORCE algorithm produces the high variance gradient such that the algorithm might converge extremly slow during training, the baseline b is utilized to reduce the variance. Paremeters θ are updated with the policy gradient method as:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{k} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_{i,t} \mid \mathbf{s}_{i,t} \right) \left(\sum_{t'=t}^{k} r_{i,t'} - b_i \right), \tag{13}$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta), \tag{14}$$

where N is the number of batch size, k is the number of mission, α is the learning rate, and b is the baseline. Note that the baseline b of this study is moving average of the cost during training (Kool et al., 2019). The proposed algorithm is trained with $1,280,000$ mission instances with 512 batch size, 100 epochs, 1e-4 learning rate with the Adam optimizer (Kingma and Ba, 2015).

NUMERICAL SIMULATIONS

This section provides the comprehensive simulation results to show the performance of the proposed approach. Every simulation is run on NVIDIA GeForce RTX 2080 GPU, Intel(R) i9-9900KF CPU, and 64GB RAM.

The neural netowork is trained with the different number of missions in the range of (3, 30). The position of every mission is generated randomly in (0, 1) scaled two-dimensional (2-D) map with a uniform distribution. The coverage mission's radius is generated randomly in the range of (0.04, 0.08) with the uniform distribution. The place-position of the delivery mission is distant from the pick-position in the range of (-0.1, 0.1) with the uniform distribution. The position of the depot is the origin without loss of generality. The velocity of the UAV is 1 and the flight time budget is 6.

We compare our algorithm (termed as Transformer-RL) with the Google OR-Tools[1] which is the state-of-the-art solver for the combinatorial optimization problem. We modified the software into two types. The first baseline algorithm (OR-Type1) solves the given mission as a distance-limited VRP with a single-vehicle. The OR-Type1 generates a single route per iteration within the flight time budget. We give the penalty cost to the uncompleted missions to prevent generating an empty route. The penalty makes the algorithm try to generate a shorter route while satisfying the flight time limitation. Then, the OR-Type1 makes a plan iteratively until every mission is completed. The second baseline algorithm (OR-Type2) solves the mission instance as a distance-limited VRP setting, assuming that the number of available vehicles and the number of missions to be performed are the same. The assumption reduces the effort to solve the problem iteratively unlike the OR-Type1. Thus, the OR-Type2 generates multiple routes at once that complete every mission while deciding the desirable number of vehicles to utilize. We also compared ours with the simple greedy algorithm and the Pointer

---

[1] https://developers.google.com/optimization/routing/vrp

₂₇₀ network-based reinforcement learning algorithm (PointerNet-RL) (Bello et al., 2017).
₂₇₁ The simple greedy algorithm selects the next mission with the lowest cost from the
₂₇₂ current mission node while satisfying the flight time limitation, and the PointerNet-RL
₂₇₃ uses RNN (Vinyals et al., 2015) for the neural network structure instead of the attention
₂₇₄ mechanism used in this study. Figure 8 visualizes the sample solution and total cost of
₂₇₅ each algorithm. Note that the return path to the depot of each route is represented with
₂₇₆ the black dashed line. In Fig. 8, the OR-Type2 generates the best solution which has the
₂₇₇ lowest cost, and reinforcement learning-based algorithms show better performance than
₂₇₈ that of the OR-Type1 and the greedy algorithm. The OR-Type1 generates each route
₂₇₉ while completing the most missions possible and the greedy algorithm makes the most
₂₈₀ number of routes, which is inefficient due to its myopic strategy.
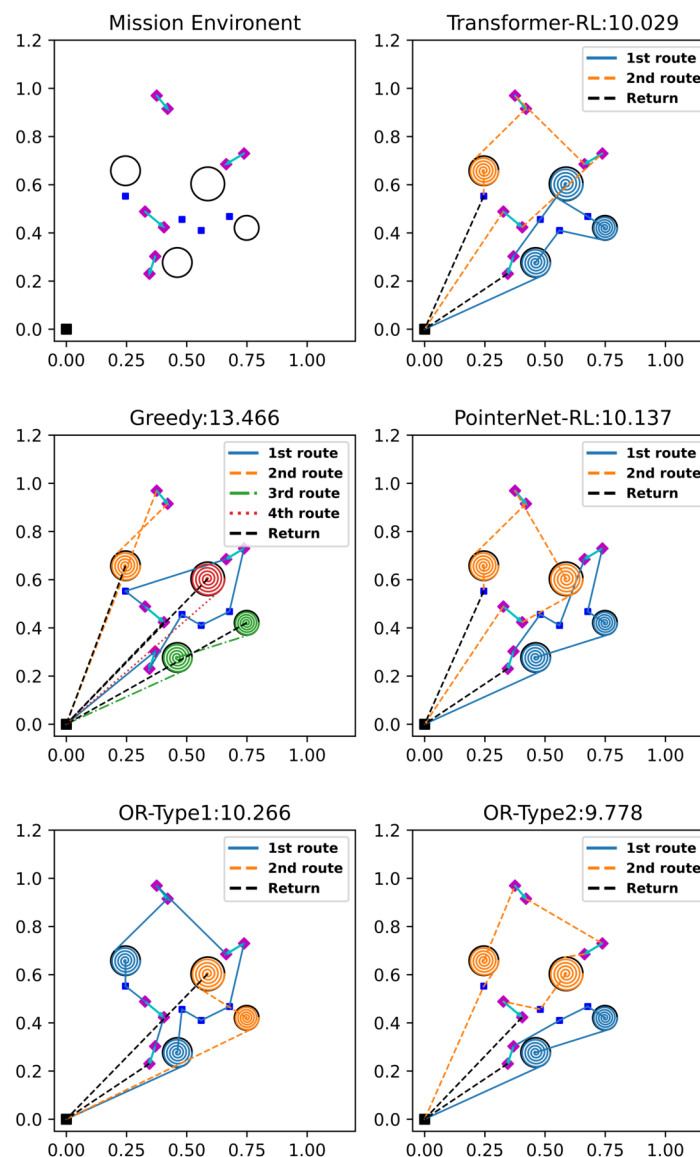
Figure 8. Sample solutions of the different algorithms on the mission environment for 12 missions. (4 visiting, 4 coverage and 4 delivery)

281 The total cost of the solution defined in Eq. (1) and the computation time is used to
282 measure the performance of the algorithms. We use 10,000 mission instance samples
283 to test the performance with the different number of missions in the range of (3, 30).
284 Figure 9 provides the result of the performance analysis. The OR-Type2 shows the best
285 performance in terms of the total cost, while Transformer-RL has a similar performance
286 with the OR-Type2. Figure 10 shows Transformer-RL is better than the PointerNet-RL
287 more clearly by the cost gap analysis with respect to the OR-Type2. Figure 11 provides
288 the computation time for each algorithm. The computation time of the OR-Type2, which
289 is the best algorithm for the cost, grows exponentially along with the scale of the mission.
290 On the other hand, the Transformer-RL and the PointerNet-RL show significantly faster
291 computation time than that of the other algorithms. The greedy algorithm also shows fast
292 computation time, but it has the worst cost performance. Table 1 summarizes statistical
293 results for a certain number of missions with the cost performance, the performance gap,
294 and the computation time.
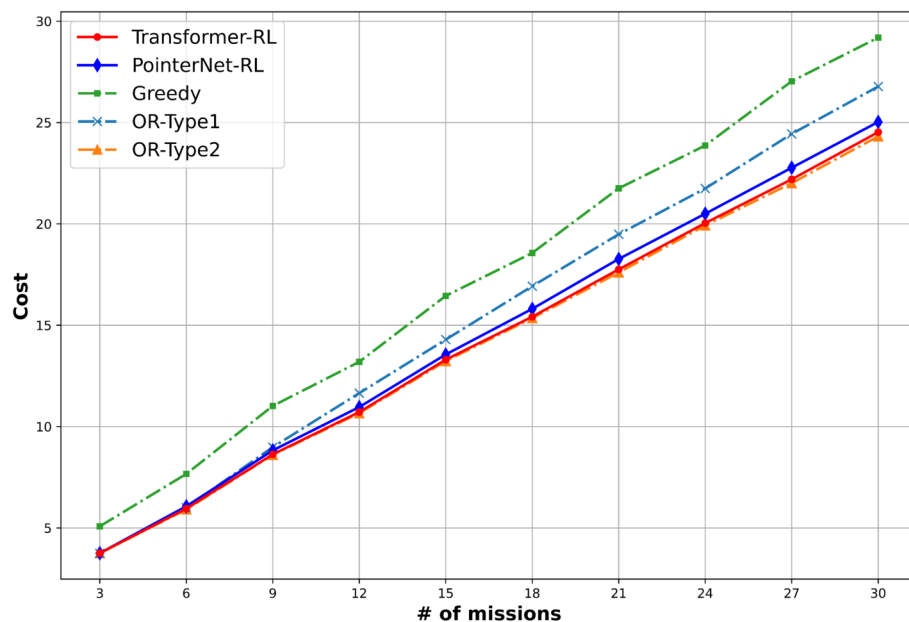
Figure 9. Mean cost performance of each algorithm.

Table 1. The performance of our proposed model compared with other algorithms. The cos gap [%] is with respect to the OR-Type2

Algorithm	$k=9$			$k=15$			$k=21$		
	Cost	Gap [%]	Time [s]	Cost	Gap [%]	Time [s]	Cost	Gap [%]	Time [s]
Transformer-RL	**8.76**	**1.846**	**1.22**	**13.55**	**2.461**	**2.05**	**18.14**	**3.08**	**2.78**
PointerNet-RL	8.82	2.525	2.26	13.55	2.453	3.64	18.26	3.78	5.18
Greedy	11.01	27.987	18.62	16.44	24.314	41.67	21.75	23.58	85.61
OR-Type1	8.98	4.381	85.23	14.29	8.012	173.73	19.48	10.71	315.71
OR-Type2	8.60	0.00	144.36	13.23	0.00	384.49	17.60	0.00	789.88
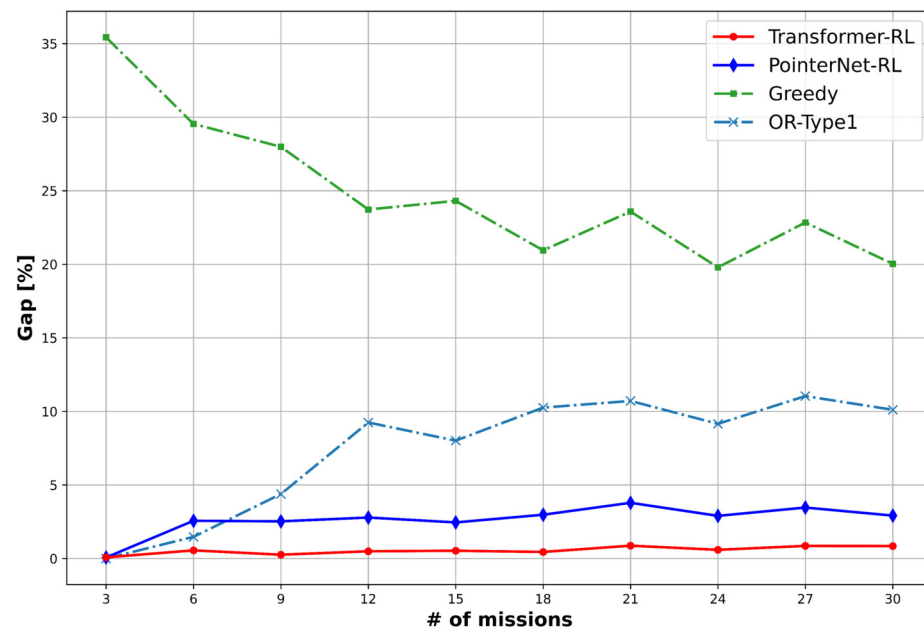
Figure 10. Cost gap [%] of the algorithms with respect to the OR-Type2.
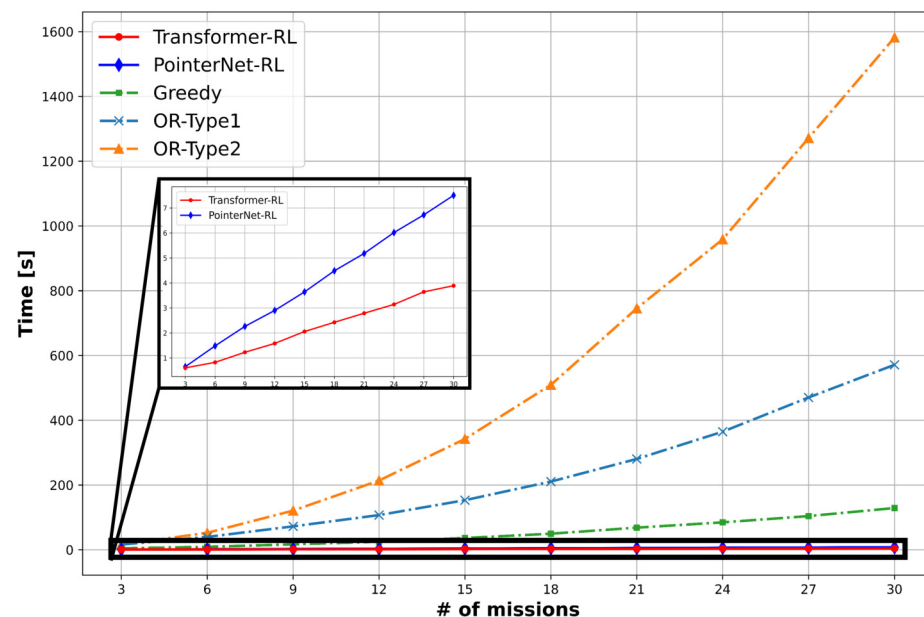
Figure 11. Computation time.

CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an algorithm for mission planning of heterogeneous missions for a single UAV. We formulate the mission planning problem into a vehicle routing problem that has various methods to solve. We used an attention-based deep reinforcement learning approach, expecting fast computation time and sufficiently good performance. The numerical experiments show that the proposed algorithm can be a good selection with the reasonable trade-off between performance and computation time. However, as

302 the proposed algorithm considers a deterministic mission environment and deals with a
303 single UAV, our future work will consider the uncertainty of the mission environment
304 such as the effect of the weather conditions and the operation of multiple UAVs with
305 multi-agent reinforcement learning approaches.

306 REFERENCES

307 Atyabi, A., MahmoudZadeh, S., and Nefti-Meziani, S. (2018). Current Advancements
308   on Autonomous Mission Planning and Management Systems: An AUV and UAV
309   Perspective. *Annual Reviews in Control*, 46:196–215.
310 Baker, B. M. and Ayechew, M. (2003). A Genetic Algorithm for The Vehicle Routing
311   Problem. *Computers & Operations Research*, 30(5):787–800.
312 Ban, H.-B. and Nguyen, P. K. (2021). A hybrid metaheuristic for solving asymmetric
313   distance-constrained vehicle routing problem. *Computational Social Networks*, 8(1):1–
314   19.
315 Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. (2017). Neural Combinatorial
316   Optimization with Reinforcement Learning. In *5th International Conference on
317   Learning Representations, Toulon, France, April 24-26*.
318 Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle-routing
319   problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368.
320 Chen, H.-X., Nan, Y., and Yang, Y. (2019). Multi-uav reconnaissance task assignment
321   for heterogeneous targets based on modified symbiotic organisms search algorithm.
322   *Sensors*, 19(3):734.
323 da Costa, P. R. d. O., Mauceri, S., Carroll, P., and Pallonetto, F. (2018). A genetic algo-
324   rithm for a green vehicle routing problem. *Electronic notes in discrete mathematics*,
325   64:65–74.
326 Fu, Z., Eglese, R., and Li, L. Y. (2005). A New Tabu Search Heuristic for The Open
327   Vehicle Routing Problem. *Journal of The Operational Research Society*, 56(3):267–
328   274.
329 Gao, S., Wu, J., and Ai, J. (2021). Multi-uav reconnaissance task allocation for hetero-
330   geneous targets using grouping ant colony optimization algorithm. *Soft Computing*,
331   25(10):7155–7167.
332 Gendreau, M., Hertz, A., and Laporte, G. (1994). A tabu search heuristic for the vehicle
333   routing problem. *Management science*, 40(10):1276–1290.
334 Grzybowski, J., Latos, K., and Czyba, R. (2020). Low-cost autonomous uav-based
335   solutions to package delivery logistics. In *Advanced, Contemporary Control*, pages
336   500–507. Springer.
337 Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood
338   search heuristic for periodic routing problems. *European Journal of Operational
339   Research*, 195(3):791–802.
340 Karaoglan, A. D., Atalay, I., and Kucukkoc, I. (2020). Distance-constrained vehicle
341   routing problems: A case study using artificial bee colony algorithm. In *Mathematical
342   Modelling and Optimization of Engineering Problems*, pages 157–173. Springer.
343 Khalil, E., Dai, H., Zhang, Y., Dilkina, B., and Song, L. (2017). Learning Combinatorial
344   Optimization Algorithms over Graphs. *Advances in Neural Information Processing
345   Systems*, 30.

346 Kim, S., Park, J., Han, D., Kim, E., and Lee, D. (2021). Development of a vision-based
347   recognition and position measurement system for cooperative missions of multiple
348   heterogeneous unmanned vehicles. *International Journal of Aeronautical and Space*
349   *Sciences*, 22(2):468–478.

350 Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd*
351   *International Conference on Learning Representations, San Diego, CA, USA, May*
352   *7-9*.

353 Kool, W., van Hoof, H., and Welling, M. (2019). Attention, Learn to Solve Routing
354   Problems! In *7th International Conference on Learning Representations, New*
355   *Orleans, LA, USA, May 6-9*.

356 Kumar, S. N. and Panneerselvam, R. (2012). A Survey on The Vehicle Routing Problem
357   and Its Variants. *Intelligent Information Management*, 4:66–74.

358 Kytöjoki, J., Nuortio, T., Bräysy, O., and Gendreau, M. (2007). An Efficient Variable
359   Neighborhood Search Heuristic for Very Large Scale Vehicle Routing Problems.
360   *Computers & Operations Research*, 34(9):2743–2757.

361 Laporte, G. and Nobert, Y. (1983). A branch and bound algorithm for the capacitated
362   vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85.

363 Larrain, H., Coelho, L. C., Archetti, C., and Speranza, M. G. (2019). Exact solution
364   methods for the multi-period vehicle routing problem with due dates. *Computers &*
365   *Operations Research*, 110:148–158.

366 Mazyavkina, N., Sviridov, S., Ivanov, S., and Burnaev, E. (2021). Reinforcement
367   Learning for Combinatorial Optimization: A Survey. *Computers & Operations*
368   *Research*, 134:105400.

369 Mingozzi, A., Roberti, R., and Toth, P. (2013). An Exact Algorithm for The Multitrip
370   Vehicle Routing Problem. *INFORMS Journal on Computing*, 25(2):193–207.

371 Paradiso, R., Roberti, R., Laganá, D., and Dullaert, W. (2020). An exact solution
372   framework for multitrip vehicle-routing problems with time windows. *Operations*
373   *Research*, 68(1):180–198.

374 Qiu, M., Fu, Z., Eglese, R., and Tang, Q. (2018). A tabu search algorithm for the vehicle
375   routing problem with discrete split deliveries and pickups. *Computers & Operations*
376   *Research*, 100:102–116.

377 Ruiz, E., Soto-Mendoza, V., Barbosa, A. E. R., and Reyes, R. (2019). Solving the open
378   vehicle routing problem with capacity and distance constraints with a biased random
379   key genetic algorithm. *Computers & Industrial Engineering*, 133:207–219.

380 Secomandi, N. (1998). *Exact and heuristic dynamic programming algorithms for the*
381   *vehicle routing problem with stochastic demands*. University of Houston.

382 Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I.,
383   Othman, N. S., Khreishah, A., and Guizani, M. (2019). Unmanned Aerial Vehicles
384   (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access*,
385   7:48572–48634.

386 Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy Gradient
387   Methods for Reinforcement Learning with Function Approximation. *Advances in*
388   *Neural Information Processing Systems*, pages 1057–1063.

389 Toth, P. and Vigo, D. (2002). Branch-and-bound Algorithms for The Capacitated VRP.
390   In *The Vehicle Routing Problem*, pages 29–51. SIAM.

391 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser,

392   L. u., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural*
393   *Information Processing Systems*, 30.
394 Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer Networks. *Advances in Neural*
395   *Information Processing Systems*, 28.
396 Zhu, W., Li, L., Teng, L., and Yonglu, W. (2018). Multi-uav reconnaissance task
397   allocation for heterogeneous targets using an opposition-based genetic algorithm with
398   double-chromosome encoding. *Chinese Journal of Aeronautics*, 31(2):339–350.