Reinforcement Learning for Solving Stochastic Vehicle Routing Problem

Zangir Iklassov   zangir.iklassov@mbzuai.ac.ae
Ikboljon Sobirov   ikboljon.sobirov@mbzuai.ac.ae
Ruben Solozabal   ruben.solozabal@mbzuai.ac.ae
Martin Takáč   martin.takac@mbzuai.ac.ae
MBZUAI, UAE, Abu-Dhabi

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

This study addresses a gap in the utilization of Reinforcement Learning (RL) and Machine Learning (ML) techniques in solving the Stochastic Vehicle Routing Problem (SVRP) that involves the challenging task of optimizing vehicle routes under uncertain conditions. We propose a novel end-to-end framework that comprehensively addresses the key sources of stochasticity in SVRP and utilizes an RL agent with a simple yet effective architecture and a tailored training method. Through comparative analysis, our proposed model demonstrates superior performance compared to a widely adopted state-of-the-art metaheuristic, achieving a significant 3.43% reduction in travel costs. Furthermore, the model exhibits robustness across diverse SVRP settings, highlighting its adaptability and ability to learn optimal routing strategies in varying environments. The publicly available implementation of our framework serves as a valuable resource for future research endeavors aimed at advancing RL-based solutions for SVRP.

Keywords: Stochastic Vehicle Routing Problem; Reinforcement Learning; Planning and Scheduling Optimization.

1. Introduction

Reinforcement Learning (RL) studies machine learning (ML) problems with a target to train algorithms to make decisions with a reward maximization function through interactions with their environment. With notable success of RL in numerous applications such as gaming Silver et al. (2017) and robotics Andrychowicz et al. (2020), their use cases in combinatorial optimization problems are gaining great attention. Combinatorial optimization problems are one of the most challenging optimization problems that is traditionally addressed with heuristics or solvers (e.g. Google OR tools). Heuristic approaches are generally fast but they are only locally optimal; solvers can provide globally optimal solutions but are significantly slow. RL approaches can resolve both of these issues with computationally efficient models that do not require human intervention to learn black-box heuristics. Such models, however, should be carefully formulated such that the environment and training settings are well-established for the problem at hand.

The vehicle routing problem (VRP) is a classic combinatorial optimization problem which aims to find an optimum route for a set of vehicles with limited capacity that deliver goods or services to a group of customers such that the overall distance traveled is minimized.

Nazari et al. (2018) proposed the first RL model that can tackle the VRP using a pointer network.

The **stochastic vehicle routing problem** (SVPR) is a subtype of VPR problems, and unlike deterministic VRP, it is even more challenging on account of uncertain parameters, which can only be determined after the vehicle completes several routes. While the SVRP poses certain difficulties, such a formulation of the VRP is more representative to the real-world applications that are commonly encountered in industry Cordeau et al. (2007); Gendreau et al. (1996). Several classical approaches to solving the SVRP have been studied in Laporte and Louveaux (1993); Pichpibul and Kawtummachai (2013); Goel et al. (2019); however, the problem has not been yet investigated using deep reinforcement learning.

A particular parameters of the SVRP, known as correlated stochastic demands and travel times, make the problem distinctly challenging due to the randomly variable customer demands that are non-independent/correlated based on certain underlying factors Rajabi-Bahaabadi et al. (2021). Solving the SVRP with correlated stochastic demands and travel times offers numerous benefits to companies that operate in transportation and logistics sectors, including reduction in carbon emissions and costs. In this work, our goal is to leverage RL to tackle the SVRP with correlated stochastic demands and travel times such that the formulation is closely aligned with the real-world settings.

In this research endeavor, we aim to enhance the applicability of our framework Iklassov et al. (2023) to real-world supply-chain scenarios by incorporating two important sources of stochasticity, namely stochastic demand and stochastic travel costs. We conduct experiments to assess the dependence of our proposed model on different settings of SVRP environment, which are frequently explored in classical research. It is important to note that the proposed architecture is designed to be flexible and capable of addressing SVRP instances with an arbitrary number of vehicles. Our specific **contributions** to this field are summarized as follows:

- In this study, we address the Stochastic Vehicle Routing Problem and propose a **an RL-based approach** to solve it. Our model incorporates stochastic customer demands and travel costs, which are crucial factors in real-world logistics scenarios. We compared our RL model to state-of-the-art approaches and demonstrated its superiority, achieving a **3.43% reduction in travel cost** compared to the best-performing classical model Goel et al. (2019).

- We conducted comprehensive experiments to evaluate the performance of our model across various VRP settings. We observed that incorporating weather information, which introduced correlations among customers, enabled our RL agent to achieve up to a **2% improvement in routing strategies**.

- Overall, our research showcases the effectiveness of RL in addressing the SVRP and its potential to outperform traditional approaches used in industry. The results highlight the **significant cost reductions and potential environmental benefits** that can be achieved by adopting RL-based routing solutions using **shared open-source framework**[1].

---

1. https://github.com/Zangir/SVRP

2. Related Work

Using RL methods for combinatorial optimization problems has been investigated in numerous studies. Bello et al. (2017) pioneered the use of RL to address the Traveling Talesman (TSP) and Knapsack problems, reporting a performance exceeding typical heuristics. In fact, their approach reached the results close to global optimum. Oroojlooyjadid et al. (2022) investigated the Beer Game problem using a Q-learning model, which can essentially be extended to solve decentralized multi-agent combinatorial problems. Tricks (2018) also dives into the application of RL techniques to search for optimal solutions for combinatorial problems such as Knapsack, Secretary, and Adwords. Such studies encourage the potential of RL applications in the field of combinatorial optimization problems.

Deterministic VRP is no exception to benefit from RL techniques in recent years. Nazari et al. (2018) reported the first work that leverage an RL method to solve VRP, authors proposed pointer network based architecture to solve deterministic VRP that gives superior performance compared to baseline heuristic and metaheuristic methods. Lu et al. (2020) developed a metaheuristic RL-based agent that supervises simple rules in the search for VRP solution. The approach sets the state-of-the-art (SOTA) with the lowest cost for VRP instances with 50-100 customers; however, it requires a domain knowledge from the researcher to generate a set of plausible heuristics. To alleviate all this problem, Wang (2021) proposed an RL framework that reaches a new SOTA performance for larger scale instances of VRP at 500, 1000, and 2000 customers.

While there are numerous research papers on the deterministic VRP that leverage RL methods, the use of such approaches in SVRP problems still remain a challenge. The current solutions for SVPR, as mentioned in Toth and Vigo (2014), rely heavily on the classical algorithms for SVPR, which can be summarized in the following categories:

- **Branch-and-Bound** algorithms - Branch-and-Cut Gauvin et al. (2014), Branch-and-Price Christiansen and Lysgaard (2007), Integer L-shaped Laporte and Louveaux (1993).

- **Heuristics** - LKH3 Helsgaun (2017), Clarke-Wright Pichpibul and Kawtummachai (2013).

- **Metaheuristics** - Tabu Search Li and Li (2020), Ant-Colony Optimization Goel et al. (2019).

Compared to the other branch-and-branch family members, the integrated L-shaped algorithm Laporte and Louveaux (1993) proved to be superior for the SVRP problem. The cost is substituted with a θ variable that acts as a lower bound evaluator through the application of branch-and-branch method. While this is desirable, its performance is computation-heavy with regards to the number of vehicles, making it undesirable in real-life applications. Among the many solutions for the VRP, LKH3 Helsgaun (2017) yields unmatchable performance for the deterministic VRP; however, its performance becomes unstable for SVRP. Clarke-Wright algorithm Pichpibul and Kawtummachai (2013) addresses that issue, performing similarly well for both deterministic and stochastic VRP, while being a simple heuristic-based approach. The Clarke-Wright algorithm has become the go-to reference/baseline model for comparison in the SVRP literature due to its consistency and

longevity. Metaheuristic methods slowly emerged to become prevalent in the literature as well, with the Ant-Colony optimization problem and such methods' SOTA performance on the problem Goel et al. (2019).

A separate subfield within the literature on SVRP involves instances with correlated stochastic parameters. In order to ensure the efficacy of classical methods for SVRP instances with non-i.i.d. parameters, it is crucial to consider the correlation in the design of the algorithm. For instance, the pheromone update formula in ACO should be adjusted to reflect the correlation between parameters Rajabi-Bahaabadi et al. (2021). Similarly, Branch-and-Bound algorithms must be modified to handle this correlation Bomboi et al. (2021). In general, classical algorithms have been shown to generally exhibit inferior performance on this type of problem due to the assumption of i.i.d. parameters in their design.

In recent years, a limited number of studies have applied ML algorithms to the SVRP. In Secomandi (2000), a policy-based model was proposed to address the SVRP with stochastic demands, where the authors trained a simple linear model and achieved improved performance compared to heuristic methods. In Niu et al. (2021), the authors utilized decision trees to solve the SVRP, while Niu et al. (2022) employed an evolutionary algorithm based on a radial basis network. Furthermore, Joe and Lau (2020) was the first study to use deep reinforcement learning to address the SVRP with stochastic customers. Mustakhov et al. (2023) extended this by considering stochastic customers and aiming to maximize the total goods served within a defined time frame. Notably, Iklassov et al. (2023) were the first to employ Reinforcement Learning (RL) for solving SVRP with stochastic correlated demands, achieving state-of-the-art performance in the considered problem domain. In our pursuit, we endeavor to augment this approach by integrating stochastic travel times, thereby subjecting the model to comprehensive evaluation.

3. Background

One of the most widely observed NP-hard combinatorial problems is the **Vehicle Routing Problem** (VRP). The VRP solves the optimization problem of delivering goods from a depot to customers with known locations and demands. The goal of the method is to minimize the overall travel cost required to meet the demands of all the customers. Several studies delved into the VRP optimization using different methods and heuristics Laporte (2009); but the initial VRP formulation was proposed by Dantzig and Ramser (1959).

The **Stochastic Vehicle Routing Problem** (SVRP) is a more challenging task due to the uncertainty in the parameters of the method, which can only be known after the completion of a route. Unlike the VRP, the SVRP is stochastic in such parameters, making it a harder task; however, because of its greater applicability in real-world industrial problems, it attracts more community in industry and research. It can be a better representative model of the real-life routing settings, and therefore, there are several works investigating the problem Cordeau et al. (2007); Gendreau et al. (1996). Toth and Vigo (2014) attributes three main sources as the stochasticity of the SVRP:

1. **stochastic demands** (VRPSD), i.e., uncertain customer demands,
2. **stochastic customers** (VRPSC), i.e., uncertain presence/absence of customers,
3. **stochastic travel times** (VRPSTT), i.e., uncertain travel costs for various routes.

The deterministic nature of VRP cannot suffice as an optimum solution for SVRP, claimed Louveaux (1998), and proposed a formulation that takes the stochasticity of the problem into account using a mathematical representation.

3.1. Classical Formulation

The objective of the problem is to

$$\text{minimize} \qquad \sum_{i,j \in C} c_{ij} x_{ij} + \mathscr{R}(x). \tag{1}$$

The problem at hand is characterized by a set of notations as follows: N - set of customers and depot, C - customers set, c_{ij} - stochastic travel cost between nodes i and j, c_{ijs} - s^{th} realization of $\{ij\}$ travel cost, ξ_i - stochastic demand of customer i, μ_{is} - s^{th} realization of the demand of customer i, K - set of vehicles, Q - maximum capacity of vehicle, x_{ij} - binary variable that shows whether (i, j) is used in the route.

In this formulation, the SVRP is modeled on a complete, undirected graph $G = (N, X)$. A depot node (i.e. node 0) and customer nodes are represented as the set of nodes $N = \{0, 1, \ldots, n\}$. The depot functions as the base for the set of K vehicles with an initial loading value of Q. A stochastic demand variable ξ_i, including its sth realization characterized as μ_{is}, exists per customer node $i \in C = N\backslash 0$. Arcs, given as $X = (i, j) : i, j \in N, i < j$, denote the associations between nodes; the travel cost of an arc $(i, j) \in X$ in a route is c_{ij}. The representation x_{ij} is an action of an agent with x_{ij}=1 when an arc (i, j) is traveled in the route and 0 otherwise. In case of situations when a vehicle fails to fulfill the demand of a customer due to a shortage of load, the recourse cost $\mathscr{R}(x)$ accounts for it. To be precise, the term is responsible for the cost incurred for traversing to the depot and back to refill in order to meet the customer demand. The detailed description of the classical problem, including all the constraints and additional information, can be found in the supplementary material.

3.2. Baselines

Several methods existent in the literature serve as baselines for the SVRP. The **Clarke-Wright** (CW) heuristic Pichpibul and Kawtummachai (2013) is such an approach that is based on the measure, so-called *savings*, that estimates the decrease in the total travel cost that occurs due to the merge of two customer nodes into a single route. The *saving* amount for two customer nodes i and j is computed as $saving_{ij} = \mathbb{E}[c_{0i}] + \mathbb{E}[c_{j0}] - \mathbb{E}[c_{ij}]$, where c_{0i} and c_{0j} are the traversing costs between depot and customer i and j, respectively, and c_{ij} is the traversing cost between them. These values are then collected for all the customer pairs in a savings list, sorted in decreasing order. The technique for merging the routes starts with picking the highest-value pair from the list. Provided that the expected demand given as $\mathbb{E}[\xi_i] + \mathbb{E}[\xi_j]$ is within the vehicle capacity Q, the two customer nodes are merged together in a single route. This process repeats until no viable merging is left in the savings list.

Another well-established optimization approach is **Tabu Search** (TS), which uses metaheuristics to solve the SVRP. With this approach, a random feasible solution is generated randomly, and its total cost is calculated. Using a set of predefined neighborhood heuristic procedures on the proposed solution, a new list of candidate solutions is generated over a certain number of iterations defined as $k_{Tabu,max}$. The algorithm then outputs the best candidate from the set of visited feasible solutions.

In Li and Li (2020), the authors demonstrate that choosing the right set of four heuristics (Specific neighborhood, 2-opt, Swap-operation, Reallocate-operation) can outperform other sets of heuristics studied in the older literature.

Ant Colony Optimization (ACO) is also a metaheuristics baseline that mimics how ants in nature behave foraging for food. Solution agents act as ants traveling the search space by investigating and exploiting the possible routes. When an agent finds a feasible solution, it leaves a pheromone on the arcs that build this route. The probability of an arc being a better candidate depends on the concentration of the pheromone laid by the agent, which can later be followed by subsequent agents. In other words, the higher the concentration of the pheromone, the better quality that arc has as a candidate. The pheromone fades away over time, again imitating the natural ant behavior. The search for new solutions goes on as ants explore more arcs to follow probabilistically.

4. Method

Objective. We propose a problem formulation where the objective is to
$$\text{minimize} \quad \tfrac{1}{|S|}\sum_{s\in S}(\sum_{i,j\in C}c_{ijs}x_{ij} + 2\sum_{i\in C}r_i c_{0is}),$$
where the agent minimizes the travel cost in the context of a set of scenarios, denoted as S. The agent aims to solve this problem by making decisions on action variables, specifically x_{ij} and r_i. Here, r_i represents a binary recourse action that determines whether the vehicle needs to return to the depot after visiting customer i in order to replenish its capacity and prevent potential issues with future customers along the route.

Stochastic Variables. The agent lacks prior knowledge of the distribution of stochastic variables and does not have access to the specific realization s of the demand μ_{is} or the travel cost c_{ijs}. The agent can only observe the value of μ_{is} after visiting customer i and the value of c_{ijs} after traversing the arc $\{ij\}$. However, the agent has the ability to estimate the values of stochastic variables by leveraging the knowledge of the observable variables. We define a set of random variables W that influence the realization of demand and travel costs in the following manner:
$$\theta_i = \bar{\theta}_i + \sum_m\sum_n \alpha_{imn}w_m w_n + \epsilon_i, \ \ \theta_i \in \{\xi_i, c_{i0}, ..., c_{iN}\},$$
where we define θ_i as a stochastic variable representing the demand of customer i or the travel cost from customer i to other nodes. The value of θ_i is determined by a fixed term $\bar{\theta}_i$, which is augmented by the weighted interaction between variables $w_m \in W$ and $w_n \in W$, as well as random noise ϵ_i. It is important to note that the agent does not possess knowledge of the distribution of the variables in W or their specific influence on θ_i. However, the agent does have access to the realizations of W variables, allowing him to implicitly learn the mapping from W variables to stochastic variables to enhance the routing strategy.

Weather Variables. This study focuses on a set of three W variables referred to as weather variables, namely temperature, pressure, and humidity. This selection is motivated by real-world scenarios where the agent lacks prior knowledge of demand and travel costs but can estimate them by observing the prevailing weather conditions, which exert a direct influence on these factors. For instance, the demand for ice-cream and the travel time between ice-cream shops may vary based on the temperature. By incorporating these weather variables, we aim to create a realistic environment where the agent utilizes observable

weather information to estimate the associated stochastic variables effectively. It is important to acknowledge that the presented formulation provides an abstract representation of the problem, and in practice, there may exist additional stochastic variables beyond demand and travel time that impact the agent's costs. Moreover, there is the possibility of incorporating additional W variables that the agent assumes to have an influence on the environment.

4.1. Routing Policy

State. The proposed SVRP formulation defines the environment as a state (I_s^t, h_k^t) associated with a particular realization s and vehicle k,

$$I_s^t \doteq \{(W, d_i^t, c_{i0s}, ..., c_{iNs}), i \in N\}, \ h_k^t \doteq (q_k^t, p_k^t), k \in K.$$

The state is dynamic and evolves over time $t \in \{1, ..., T\}$, where T represents the time step when all vehicles have returned to the depot and all customer demands have been fulfilled. I_s^t is characterized by a set of weather variables W, the current demand of customer i at time t, denoted as d_i^t, and the travel cost realizations c_{ijs} between all nodes $\{ij\}$. The weather variables and travel costs remain constant throughout the time horizon, while the demand d_i^t takes on the value of the demand realization μ_{is} at $t = 1$, and 0 at $t = T$ for all customers $i \in N$. h_k^t is characterized the current load q_k^t and the position p_k^t of each vehicle $k \in K$ at time t.

Action. At each time step t, the agent selects an action a_k^t for each vehicle, where the action represents the choice of the next node $i \in N$ to visit. The decision-making process is dynamic and governed by a policy π, such that $a_k^t \sim \pi(\cdot | I_s^t, h_k^t)$. In other words, the policy determines the probability distribution of selecting the next node for each vehicle based on the current state (I_s^t, h_k^t).

Transitions. Upon executing the action a_k^t, the positions of the vehicles are updated to p_k^{t+1} based on the nodes they move to. The representation of vehicle positions can be encoded using various techniques, such as coordinates on a 2D map or one-hot vectors. The demands and loads for vehicle k that visits the customer i are updated according to the following formulas:

$$d_i^{t+1} = \max\{0, d_i^t - q_k^t\}, \ d_j^{t+1} = d_j^t, \text{ for } j \neq i, \text{ and } q_k^{t+1} = \max\{0, q_k^t - d_i^t\}. \quad (2)$$

Objective. The agent aims to derive an optimal policy π with parameters Θ by employing the Reinforce algorithm, which seeks to minimize the cumulative expected cost

$$\mathcal{J}^\pi(\Theta) = \mathbb{E}[\sum_t^T C_t(I_s^t, a_k^t)].$$

The cost function $C_t(I_s^t, a_k^t)$ is defined as the sum of traversal costs for each vehicle k moving from node i_k to node j_k after executing action a_k^t. It is calculated as $\sum_{k=1}^K c_{i_k j_k s}$. In cases where vehicle k requires recourse action due to a failure, an additional recourse cost of $2c_{0j_k s}$ is incurred. The objective function is accumulated over a set of scenarios S, each consisting of T_s time steps, and the gradient of the objective function is computed as follows:

$$\nabla_\theta \hat{\mathcal{J}}^\pi(\Theta) \approx \frac{1}{S} \sum_{s=1}^S \sum_{t=1}^{T_s} ((C(I_s^t, h_k^t) - b_\phi(I_s^t, h_k^t)) \cdot \nabla_\Theta \log \pi_\Theta(a_k^t | I_s^t, h_k^t)),$$

where $b_\phi(I_s^t, h_k^t)$ denotes the baseline function with parameters ϕ that are trained to minimize $L(\phi) = \frac{1}{S} \sum_s^S \sum_{t=1}^{T_s} ||b_\phi(I_s^t, h_k^t) - C(I_s^t, h_k^t)||^2$.

4.2. Environment Settings

The model's robustness and performance improvement over time need to be evaluated across various environmental settings that closely resemble real-life scenarios. This experimentation is crucial to assess its ability to adapt and excel in diverse conditions. Therefore, we aim to examine key environmental factors that hold significance in classical SVRP research.

Inference. The performance evaluation of the model can be conducted by investigating its reliance on different inference strategies, encompassing widely employed methods like greedy sampling, random sampling, and beam search. In the case of greedy sampling, the action with the highest probability is chosen at each time step t to construct the routing solution. Random sampling generates multiple solutions by randomly selecting n_s actions in accordance with their respective probability distributions at each time step. In beam search, the top n_b cumulative most probable actions are selected. The utilization of these distinct inference strategies may yield disparate routing solutions, thereby leading to diverse performance outcomes.

Signal Ratio. The three components of θ_i, namely the constant term, weather-related factors, and random noise, are likely to exert varying influences on the distribution of stochastic variables. In order to quantify this influence, we introduce signal ratio variables A_i, B_i, Γ_i and investigate the impact of different signal ratio values on the model's performance.

$$A_i = \frac{\bar{\theta}_i^2}{T_i} \qquad\qquad B_i = \frac{\mathbb{E}\left[(\sum_m \sum_n \alpha_{imn} w_m w_n)^2\right]}{T_i}, \qquad\qquad \Gamma_i = \frac{\mathbb{E}\left[\epsilon_i^2\right]}{T_i},$$

$$T_i = \bar{\theta}_i^2 + \mathbb{E}[(\sum_m \sum_n \alpha_{imn} w_m w_n)^2] + \mathbb{E}\left[\epsilon_i^2\right].$$

Fill rate. The performance of the model may be influenced by the maximum capacity of each vehicle. To quantify its impact, we introduce a fill rate variable, denoted as $\Phi = \frac{Q}{\mathbb{E}[\xi_i]}$, which represents the relative value of the maximum capacity in relation to the expected value of the total demand.

A Priori vs Reoptimization. In SVRP, there are two distinct approaches that are both significant in industry and warrant consideration. The first approach, known as a priori, involves the agent constructing the entire route without prior knowledge of the realizations of stochastic variables. On the other hand, the reoptimization approach allows the agent to unveil the realizations of variables as it visits the corresponding customers and arcs. This enables the agent to dynamically update the routes based on the acquired information, thus addressing the problem in an online fashion.

Variable Estimates. Prior to constructing the route, the agent can utilize estimates of the stochastic variables as inputs, denoted as $\hat{\theta}_i$. In this study, we propose two types of estimates: the constant estimate and the k-NN estimate. The constant estimate simply represents the expected value of the stochastic variable, $\hat{\theta}_i = \bar{\theta}_i$. The k-NN estimate, on the other hand, involves the utilization of a historical weather dataset, which comprises tuples of $(w^g, \mu_{ig}, c_{i0g}, ..., c_{iNg})$. By employing the k-NN algorithm, the agent can leverage the current w vector to identify the expected value from the w_g closest tuples in the weather dataset,

$$\hat{d}_i = (\sum_{w_g \in N(w)}^{G} \frac{1}{||w_g - w||} \mu_{ig}) / (\sum_{w_g \in N(w)}^{G} \frac{1}{||w_g - w||}), \tag{3}$$

$$\hat{c}_{ij} = (\sum_{w_g \in N(w)}^{G} \frac{1}{||w_g - w||} c_{ijg}) / (\sum_{w_g \in N(w)}^{G} \frac{1}{||w_g - w||}). \tag{4}$$

Customer Positions. In this study, we introduce two distinct settings for customer positioning: fixed and flexible. In the fixed positioning setting, the positions of customers are predetermined and remain unchanged during the training and inference phases. As a result, the constant component of travel costs, denoted as $\bar{c}_{ij}$, remains constant throughout. This setting may yield improved performance due to the relatively less challenging nature of the problem. It is particularly relevant for businesses operating with fixed customers (e.g. B2B delivery). In contrast, the flexible customer positioning setting allows for varying customer positions in each problem instance. This results in different $\bar{c}_{ij}$ values for each instance (relevant to B2C delivery).

Delivery Types. There are two distinct delivery modes of goods: full delivery and partial delivery. In the full delivery mode, the vehicle is only capable of delivering goods to a customer in their entirety, meaning that the goods are indivisible. If a customer has a demand of μ_{is}, the vehicle must deliver the entire quantity at once. If the vehicle's capacity is insufficient, it must perform a recourse action. On the other hand, in the partial delivery mode, the goods can be divided, allowing the vehicle to deliver them in partial quantities.

5. Architecture Details

Customer Information. We present the architectural framework for the models illustrated in Figure 1. To obtain continuous representations of customers, referred to as state embeddings, we utilize a 1D convolutional layer with D filters. This layer plays a crucial role in extracting meaningful and continuous representations of the customer-related information $(w, d_i^t, c_{i0}^t, ..., c_{iN}^t)$.

Vehicle Information. The vehicle-related information, including the current position p_k^t and load q_k^t, is processed using an LSTM (Long Short-Term Memory) module with a state size of D. This LSTM module captures the temporal dependencies and generates the memory embedding h_k^t. The memory layer allows the model to retain and utilize information from previous time steps $[1, ..., t-1]$ regarding each vehicle k.

Next Node Probabilities. To combine the state and memory embeddings effectively, we employ an attention layer. This attention mechanism enables the model to assign importance weights to all node embeddings based on their relevance. The attention layer produces the output $P(p_k^{t+1}|\cdot)$, which represents the probability of each node i (including the depot) being the next node in the route of vehicle k.

Masking. To ensure the generation of feasible solutions, we employ a masking procedure. Specifically, we assign a value of $-\infty$ to the probability of infeasible position $P(p_k^{t+1}|\cdot)$ corresponding to customer nodes whose demand has already been fulfilled. By applying this masking technique, we effectively exclude these already serviced customer nodes from the set of potential next nodes in the route for each vehicle.

6. Experiments

Nodes. The model was evaluated on four distinct problem sizes involving 10, 20, 50, and 100 customers. To set up the problem instances, we randomly generated the positions of these customers on a 2D map defined by the interval $[0, 1] \times [0, 1]$. The depot was located at coordinates $(0.5, 0.5)$ as a fixed reference point.
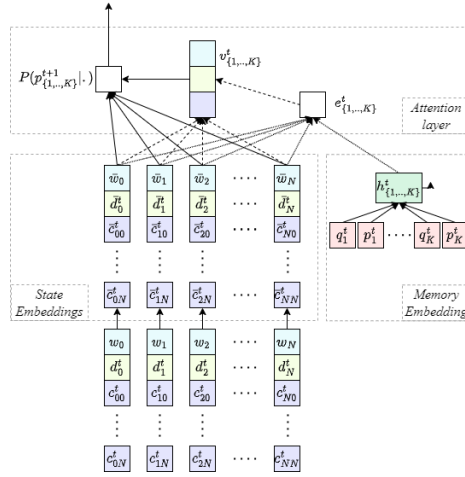
Figure 1: **Network architecture.** The lower section illustrates the model's input, which is composed of three elements: weather data w, the varying demand d_i^t, and travel costs c_{ij}^t. The model generates State Embeddings for each customer and encodes the vehicle's load (q_k^t) and current position (p_k^t) as the Memory Embedding (h_k^t). The embeddings are subsequently combined using an Attention Layer, resulting in probabilities assigned to the nodes. These probabilities represent the likelihood of each node being the next position for vehicle k.

Variables. The weather variables W consist of three variables, each following a random uniform distribution within the range of $[-1, 1]$. The noise term ϵ is generated from a multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where Σ is a non-zero covariance matrix. In the experiments, we vary the values of A_i, B_i, Γ_i, and Φ. To match these values, we generate the constant and noise components of the stochastic variables accordingly. By employing this generation process, we construct a weather dataset comprising 10,000 scenarios, represented by tuples $(w, \mu_{is}, c_{i0s}, ..., c_{iNs})$ for all $i \in N$. Additionally, we create a separate test dataset consisting of 1,000 scenarios for evaluation purposes.

6.1. Results

Baselines. To establish baseline models, we implemented stochastic approaches based on the works by Pichpibul and Kawtummachai (2013), Li and Li (2020), Goel et al. (2019). Since ACO is state-of-the-art algorithm for the SVRP, we conducted a hyperparameter search on validation dataset to identify its optimal configuration for our specific environment. Table 1 displays the results obtained from the search, which involved exploring the pheromone importance, heuristic importance, and the number of ants. Table 2 illustrates results on test data achieved by baselines and RL with beam search inference $(n_b = 3)$. The lower values of travel cost show better performance of the model. For fair comparison, the baselines and RL are tested in the same environment configuration (a priori, fixed customers, partial delivery, same distributions and fill rate). RL shows better performance on all sizes with 3.31%, 3.50%, 3.43% and 3.47% (3.43% on average) less travel cost compared to ACO results.

Table 1: The performance of the ACO algorithm on the SVRP validation dataset. The number of customers are represented in the columns. The results are presented as the average travel cost over the validation dataset, with lower values indicating better performance.

ACO parameters	10	20	50	100
3 / 4 / 20	3.73	7.61	15.88	34.86
6 / 8 / 24	3.70	7.56	15.77	34.63
4 / 9 / 15	3.67	7.49	15.62	34.31
3 / 10/ 12	**3.63**	**7.42**	**15.46**	**33.96**

Table 2: The results of the Baseline and RL models on the SVRP test dataset.

Baselines	10	20	50	100
Clarke-Wright	3.88	7.93	16.53	36.30
Tabu Search	3.76	7.67	16.00	35.13
Ant-Colony Opimization	3.63	7.42	15.46	33.96
RL-beam search	**3.51**	**7.16**	**14.93**	**32.78**

Environment Settings. To evaluate the sensitivity of the proposed model, we conducted experiments using the default configuration. This configuration included the use of beam search with a beam width of $n_b = 3$ for inference, k-NN variable estimates, fixed customer positions, partial delivery, signal ratios $A, B, \Gamma = 0.6, 0.2, 0.2$ respectively, a fill rate of $\Phi = 0.5$ and a single vehicle $K = 1$. The experimental results presented in Table 4 provide insights into the performance of different **inference strategies** with parameter settings of $n_s = 16$ and $n_b = 3$. The findings indicate that beam search exhibits superior performance compared to the other strategies, highlighting its effectiveness in optimizing the model's output. The comparison presented in Table 4 demonstrates that the k-NN estimation method surpasses the performance of constant demands.

Table 5 presents the results obtained from experiments conducted to evaluate the impact of different types of stochasticity. The first experiment involved a fully deterministic model, which naturally demonstrated the best performance. In the second experiment, stochastic demand was introduced, with a ratio of $A, B, \Gamma = 0.6, 0.2, 0.2$. The third experiment introduced stochastic travel cost using the same ratio. The results indicate that stochastic travel cost has a greater impact on performance compared to stochastic demand. This can be attributed to the fact that travel cost is influenced by a larger number of variables compared to demand. In the fourth experiment, both sources of stochasticity were combined, resulting in a more challenging problem setting.

Table 6 presents the impact of varying signal ratio settings on the performance of the model. The findings indicate that increasing the signal from the weather variable leads to a reduction in travel costs. This suggests that the RL model is capable of capturing and leveraging the implicit influence of weather on demands and travel times. It is worth noting that this impact may vary across different customers, yet the model can effectively utilize this information to enhance routing strategies. The performance of the model was evaluated through experiments examining the influence of **customer positions** on the outcomes, as depicted in Table 7. The findings suggest that fixed customer positions yield superior performance on the test data.

Table 3: The performance of inference strategies on the SVRP test dataset. The reported results pertain to both the a priori (Apr.) and reoptimization (Reopt.) settings.

	10		20		50		100	
INFERENCE STRATEGIES	APR.	REOPT.	APR.	REOPT.	APR.	REOPT.	APR.	REOPT.
RL-GREEDY	3.98	3.86	8.12	7.88	16.93	16.42	37.18	36.06
RL-SAMPLING	3.69	3.58	7.54	7.32	15.73	15.26	34.54	33.50
RL-BEAM-SEARCH	**3.51**	**3.40**	**7.16**	**6.95**	**14.93**	**14.48**	**32.78**	**31.80**

Table 4: The results obtained from two different types of variable estimates on the SVRP test dataset.

	10		20		50		100	
Variable Estimate	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
Constant variable estimate	3.76	3.64	7.67	7.44	16.0	15.52	35.13	34.07
kNN variable estimate	**3.51**	**3.40**	**7.16**	**6.95**	**14.93**	**14.48**	**32.78**	**31.80**

Table 5: The results of various stochasticity sources on the SVRP test dataset.

	10		20		50		100	
Stochasticity Sources	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
No Stochasticity	2.88	2.79	5.88	5.71	12.26	11.90	26.93	26.12
Stochastic Demand	3.13	3.04	6.39	6.20	13.33	12.93	29.27	28.39
Stochastic Travel Cost	3.38	3.28	6.90	6.70	14.40	13.96	31.61	30.67
Stochastic Demand / Travel Cost	3.51	3.40	7.16	6.95	14.93	14.48	32.78	31.80

We tested the performance variation between baseline models and RL models under correlated and uncorrelated (demand, travel cost) settings (Figure 1 of the Supplementary Material). It is evident from the results that the baseline models lack the ability to capture and utilize the correlations between customers, leading to suboptimal routing decisions. In contrast, the RL models demonstrate the capability to exploit these correlations, resulting in more cost-effective routes.

Table 8 presents the findings from experiments conducted with different values of the signal ratio. The results indicate that increasing the fill rate, which corresponds to a larger capacity of the vehicle, facilitates the routing task. This is reflected in the improved performance of the model, as it is able to identify more optimal routes under such conditions. The experiments were conducted with partial delivery, which allows the agent to deliver goods in a more convenient manner, leading to a comparatively easier problem setting. As reflected in the results, the performance of the model is higher under the partial delivery setting, indicating its ability to leverage this condition to identify more efficient routes (Table 1 of the Supplementary Material).

Multi-Vehicle Scenarios. Table 9 presents the results of experiments conducted to assess the impact of the number of vehicles on the performance of the model. Theoretically, increasing the number of vehicles should simplify the problem by enabling the division of a large route into multiple smaller routes, resulting in fewer recourse actions and reduced

Table 6: The results obtained from various signal ratios on the SVRP test dataset.

A/B/Γ	10		20		50		100	
	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
0.8 / 0.0 / 0.2	3.41	3.31	6.97	6.76	14.53	14.09	31.91	30.95
0.8 / 0.2 / 0.0	**3.35**	**3.25**	**6.84**	**6.64**	**14.26**	**13.84**	**31.32**	**30.38**
0.6 / 0.2 / 0.2	3.51	3.40	7.16	6.95	14.93	14.48	32.78	31.80
0.4 / 0.3 / 0.3	3.76	3.64	7.67	7.44	16.00	15.52	35.13	34.07

Table 7: The results obtained from two types of positions on the SVRP test dataset.

Customers Position	10		20		50		100	
	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
Flexible customer positions	3.88	3.77	7.93	7.69	16.53	16.03	36.30	35.21
Fixed customer positions	**3.51**	**3.40**	**7.16**	**6.95**	**14.93**	**14.48**	**32.78**	**31.80**

Table 8: The results obtained from various fill rates on the SVRP test dataset.

Fill Rate	10		20		50		100	
	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
0.1	3.88	3.77	7.93	7.69	16.53	16.03	36.30	35.21
0.5	3.51	3.40	7.16	6.95	14.93	14.48	32.78	31.80
0.9	**3.29**	**3.19**	**6.71**	**6.51**	**14.00**	**13.58**	**30.74**	**29.81**

Table 9: The results obtained from various number of vehicles on the SVRP test dataset.

Vehicle Number	10		20		50		100	
	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.	Apr.	Reopt.
1	3.51	3.40	7.16	6.95	14.93	14.48	32.78	31.80
2	3.43	3.33	7.01	6.80	14.61	14.17	32.08	31.12
3	3.41	3.31	6.96	6.75	14.52	14.08	31.88	30.92
5	**3.40**	**3.29**	**6.94**	**6.73**	**14.46**	**14.03**	**31.76**	**30.81**

Table 10: Inference time results in seconds on the SVRP test dataset.

Baseline	10	20	50	100
Clarke-Wright	0.005	0.016	0.058	0.185
Tabu Search	0.964	2.936	8.672	57.12
Ant-Colony Optimization	1.230	4.291	27.64	123.7
RL-greedy	0.061	0.121	0.191	0.394
RL-sampling	0.064	0.125	0.232	0.405
RL-beam search	0.072	0.178	0.293	0.416

total length. Consistent with this notion, our findings demonstrate that as the number of vehicles increases, the model's performance improves.

Time Complexity. Table 10 presents the inference times of RL models and baselines for a single problem instance. It is observed that RL and CW models exhibit inference times that are suitable for real-time solving, indicating their efficiency in finding solutions for industry (hundreds of customers). On the other hand, ACO and Tabu search models demonstrate

inference times that have an exponential dependence on the problem size, suggesting that their computational requirements increase significantly as the problem becomes larger.

7. Conclusion

In conclusion, the Stochastic Vehicle Routing Problem (SVRP) is a challenging yet significant task in the industry. It is characterized by two crucial sources of stochasticity: stochastic customer demands and stochastic travel costs. In this study, we have presented a formulation of the problem that incorporates both sources of uncertainty. Furthermore, we have proposed an RL agent architecture along with a training methodology to effectively address the formulated problem. In real-world industrial applications, companies often leverage available information that can potentially influence stochastic demand and travel costs. Weather information is one such example that can introduce correlation among customers. In our study, we integrated this type of information into the environment and architecture of our model. The results revealed that the RL agent, when provided with access to weather information, was able to achieve **routing strategies that outperformed those without such information by up to 2%**.

This finding highlights the significance of incorporating relevant external factors, such as weather, into the modeling and decision-making processes of stochastic routing problems. By utilizing this additional information, the RL agent was able to effectively capture and exploit the correlations among customers, resulting in improved routing strategies and ultimately leading to enhanced operational efficiency in real-world scenarios. We conducted a set of evaluations of the proposed model in various settings relevant to the VRP in industrial applications. The results demonstrate the robustness of the model in learning effective routing strategies across all tested scenarios. Furthermore, we compared the model to a state-of-the-art approach introduced by Goel et al. (2019), and our **RL agent exhibited a superior performance, achieving a significant 3.43% improvement** in travel cost.

The primary focus of this work was to introduce the first RL model capable of addressing the VRP with stochastic demands and travel costs and to showcase its superiority over classical models commonly employed in industrial settings. The results indicate that RL has the potential to replace traditional approaches and significantly reduce routing costs, consequently leading to reductions in logistics expenses and negative environmental impact. Future research in this field can delve deeper into the application of RL for the SVRP and explore the development of alternative architectures and training methods to further enhance the performance of the model. These **efforts can contribute to the advancement of efficient and sustainable routing solutions** in logistics and transportation domains.

References

Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Joshua Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39:20 – 3, 2020.

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *ArXiv*, abs/1611.09940, 2017.

Federica Bomboi, Christoph Buchheim, and Jonas Pruente. On the stochastic vehicle routing problem with time windows, correlated travel times, and time dependency. *4OR*, pages 1–23, 2021.

Christian H. Christiansen and Jens Lysgaard. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Oper. Res. Lett.*, 35:773–781, 2007.

Jean-François Cordeau, Gilbert Laporte, Martin WP Savelsbergh, and Daniele Vigo. Vehicle routing. *Handbooks in operations research and management science*, 14:367–428, 2007.

George B. Dantzig and John Hubert Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

Charles Gauvin, Guy Desaulniers, and Michel Gendreau. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Comput. Oper. Res.*, 50:141–153, 2014.

Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

Rajeev Kumar Goel, Raman Maini, and Sandhya Bansal. Vehicle routing problem with time windows having stochastic customers demands and stochastic service times: Modelling and solution. *J. Comput. Sci.*, 34:1–10, 2019.

Keld Helsgaun. An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems: Technical report. 2017.

Zangir Iklassov, Ikboljon Sobirov, Ruben Solozabal, and Martin Takác. Reinforcement learning approach to stochastic vehicle routing problem with correlated demands. *IEEE Access*, 11:87958–87969, 2023. doi: 10.1109/ACCESS.2023.3306076. URL https://doi.org/10.1109/ACCESS.2023.3306076.

Waldy Joe and Hoong Chuin Lau. Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In *ICAPS*, 2020.

Gilbert Laporte. Fifty years of vehicle routing. *Transp. Sci.*, 43:408–416, 2009.

Gilbert Laporte and François V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.*, 13:133–142, 1993.

Guoming Li and Junhua Li. An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows. *IEEE Access*, 8:158115–158124, 2020.

François V. Louveaux. An introduction to stochastic transportation models. 1998.

Hao Lu, Xingwen Zhang, and Shuang Yang. A learning-based iterative method for solving vehicle routing problems. In *ICLR*, 2020.

Taukekhan Mustakhov, Yernar Akhmetbek, and Aigerim Bogyrbayeva. Deep reinforcement learning for stochastic dynamic vehicle routing problem. *2023 17th International Conference on Electronics Computer and Computation (ICECCO)*, pages 1–5, 2023. URL https://api.semanticscholar.org/CorpusID:259158985.

Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem. In *Conference on Neural Information Processing Systems, NeurIPS 2018*, 2018.

Yunyun Niu, Detian Kong, Rong Wen, Zhiguang Cao, and Jian hua Xiao. An improved learnable evolution model for solving multi-objective vehicle routing problem with stochastic demand. *Knowl. Based Syst.*, 230:107378, 2021.

Yunyun Niu, Jie Shao, Jian hua Xiao, Wen Song, and Zhiguang Cao. Multi-objective evolutionary algorithm based on rbf network for solving the stochastic vehicle routing problem. *Inf. Sci.*, 609:387–410, 2022.

Afshin Oroojlooyjadid, M. Nazari, Lawrence V. Snyder, and Martin Takác. A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manuf. Serv. Oper. Manag.*, 24:285–304, 2022.

Tantikorn Pichpibul and Ruengsak Kawtummachai. A heuristic approach based on clarke-wright algorithm for open vehicle routing problem. *The Scientific World Journal*, 2013, 2013.

Mojtaba Rajabi-Bahaabadi, Afshin Shariat, Mohsen Babaei, and Daniele Vigo. Reliable vehicle routing problem in stochastic networks with correlated travel times. *Operational Research*, 03 2021. doi: 10.1007/s12351-019-00452-w.

Nicola Secomandi. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Comput. Oper. Res.*, 27:1201–1225, 2000.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Paolo Toth and Daniele Vigo. Vehicle routing: Problems, methods, and applications, second edition. 2014.

Learns Old Tricks. A new dog learns old tricks : Rl finds classic optimization algorithms. 2018.

Q. Wang. Alpha-t: Learning to traverse over graphs with an alphazero-inspired self-play framework. 2021.