Sādhanā (2022)47:72
https://doi.org/10.1007/s12046-022-01833-2

Analysis of box and ellipsoidal robust optimization, and attention model based reinforcement learning for a robust vehicle routing problem

SWEETY HANSUWA[1,2], MOHAN RAJ VELAYUDHAN KUMAR[1,*] and RAJENDRAN CHANDRASEKHARAN[2]

[1]TCS Research, Tata Consultancy Services Limited, IITM Research Park, Chennai 600 113, India
[2]Department of Management Studies, Indian Institute of Technology Madras, Chennai 600 036, India
e-mail: v.raj@tcs.com

MS received 3 September 2021; revised 21 December 2021; accepted 18 January 2022

Abstract. In this work, we consider a class of vehicle routing problem that uses simultaneous pickup and delivery and is constrained by a hard service time window with an objective to minimize costs. In a realistic VRP environment, uncertainty or variability with constituent features and its values are the norm. We formulate and solve this class of vehicle routing problem as: (1) a mixed-integer linear programming (MILP) approach with box and ellipsoidal robust optimization mathematical model to handle uncertainty, (2) a MILP based exact box robust optimization mathematical model to handle uncertainty, and (3) a dynamic attention model based reinforcement learning approach to handle uncertainty. We have conducted computational experiments to analyse the impact of effectiveness on solution quality, problem scale, and solution performance in accounting for feature data uncertainty. Our study indicates that accounting for feature data variability using robust optimization approaches impacts solution cost. Simulation results using MILP based Robust optimization (MILP_RO) approaches and Attention Model (AM) based deep reinforcement (DRL) learning approaches show that we can cope with uncertainties to feature data without much of an impact to cost and performance for input customer graphs of smaller to medium node counts. Also, AM based DRL approaches give better quality results when compared with (MILP_RO) approaches for input customer graphs of higher node counts.

Keywords. Logistics and manufacturing; robust optimization; deep reinforcement learning.

1. Introduction

As one of the most studied combinatorial optimization problems for many years, vehicle routing problems (VRPs) are widely encountered in logistics such as package delivery & pickup services, manufacturing assembly lines, container terminal operations, transportation ridesharing, or dial-a-ride operations, etc.

The basic VRP requires efficiently arriving at a set of routes operated by a fleet of vehicles to service a given set of customers that start and end at a depot. Each customer should be visited once by only one vehicle at the lowest cost to serve all the customers. Thus, all the involved input data to the VRP problem is known in advance, such a problem referred to as a deterministic problem.

Practical logistic type VRP problems are one of the most challenging as these problems are dynamic and evolve with time. In the dynamic problem, a part or all of the input data is revealed dynamically (with/overtime) during the design or implementation of a statically planned/interim solution. The Dynamic Vehicle Routing Problem (DVRP) is one of the important variants of VRP, which necessitates building or designing an optimal set of routes using a fleet of vehicles to serve a given set of customers. In contrast, new customer orders arrive during the execution-run of the hitherto planned route allocation and schedule. Thus, routes must be re-configured dynamically for the next or future state(context) while still executing in the current context.

In this paper, we limit our work to VRP problems which exhibit uncertainty in feature data. This is also a prominent challenge in vehicle routing problems in real life. In this type of VRP problems, we usually encounter scenarios wherein exact input data is unknown initially but does exhibit variability – which we term as "uncertainty or variability or stochasticity" behaviour in data. To solve such problems, we must consider them within an *uncertainty* scope. There are many approaches for problems with uncertainty, such as sensitivity analysis, stochastic programming, and robust optimization to solve or account for data uncertainty.

*For correspondence
Published online: 05 April 2022

This paper focuses on the robust optimization (RO) approach, which addresses optimization problems with features that exhibit uncertainty in data using uncertainty set (characterized as uncertainty scope/degree), which often is agnostic to probability distribution patterns. Robust optimization aims to determine a feasible solution for any realizations of uncertain features and be conservatively optimal for all realizations of these uncertain features. In other words, robust optimization gives a decision that is ensured to be "good enough" for all possible realizations of uncertain features.

In VRP problems with uncertain feature data, one or more features exhibit uncertain or varying characteristics owing to partial or void information to start with. The features in VRP problems that typically exhibit uncertainty behaviour are listed as below:

- customers–there could be uncertainty with delivery and/or pickup quantity owing to factors such as customer's inventory needs or supplier's capacity to deliver or manage the demand, or priority-based supply or market driven factors etc.;
- vehicles – there could be uncertainty or variability in distance or time taken between source and destination locations due to weather or traffic conditions etc.;
- service– there could be uncertainty or variability with time expended to service (deliver and/or pickup items) at customer location(s) etc.

In today's competitive world, it is prudent that logistics companies should make strategic and operational decisions to optimize their VRP processes to be viable. Any viable VRP solution involves a need to manage two ends of the spectrum: a) build a potential VRP route and allocation plan or schedule to reduce cost (cost efficiency) at a strategic decision front, and b) account for operational decisions to meet customers' experience or service quality at operational decision front. Our work focuses on the VRP with simultaneous delivery and pickup and is constrained by service time windows, vehicle type, and capacity referred to as VRPSDPTW (figure 1).

We consider the problem where each customer might have some items or specific quantity to be delivered or some item to be picked up or have the need for some items to be simultaneously both be delivered and picked up at the same time. In this problem, every customer has to be visited at least one time, possibly by one or more vehicles, and the problem is referred to as the VRP with divisible or split delivery and pickup (VRPDDPTW). Such a split scenario is possible when the vehicle's capacity is lesser than the quantity to be picked or delivered or when items could be divided/split to enable optimized transportation routes. The objective of the overall vehicle route allocation and schedule is to minimize the total length of vehicles' route/tour paths and service all of the customers within their time window for an utmost customer satisfaction.

In our VRP approach, we have considered simultaneous delivery and pickup, whole or non-split demand/pickup serviceability, adhering to hard customer satisfaction involving heterogeneous vehicles.

The simultaneous delivery and pickup approach do have its benefit by allowing delivery and pickup operations in one go, thereby reducing fuel expenditure. Considering robustness into various prominent elements in our work helps the solution be rugged and helps arrive at a solution within the conservative bounds in accommodating uncertain data situations (neither overly conservative nor optimistic).

Some real-life examples that truly exhibit simultaneous pickup and delivery bi-directional flow requirements are:

- manufacturing and assembly lines–in-plant and out-plant material handling;
- bottled drinks industry–delivering full bottles and collecting empty ones simultaneously from the customers;
- mobile in-home patient care logistics–providing or delivering medicines or reports or care-materials or refills to patients at their home or care-facility and picking up soiled or test-samples for disposal and testing;
- post disaster relief and rescue operations–dispatching relief personnel and relief materials from transit-facility to affected areas (delivery) and moving affected or injured or displaced people from affected areas to transit facility (pickup);
- supply chain management–involving/moving (picking and delivering) products between echelon entities to right size inventories;
- dial-a-ride or ride sharing problem–wherein picking up and dropping off passengers from and to the requested locations; and
- last mile courier services–involving or delivering products from distribution-center's to customer's locations and collecting returns from the customer's back to the distribution-center's.

In this work, we have developed an exact Mixed Integer Linear Programming (MILP) model that is computationally more efficient (to existing MILP models) for solving the VRPSDPTW. Exact solutions are not possible all the time as VRPs are computationally considered to fall into NP-hard complexity problems. Due to this problem intractability, we have used meta-heuristic and reinforcement-learning (RL) solutions when the MILP exact approach fails to scale or converge to a solution within an estimated processing time limit. Meta-heuristic and RL approaches converge or produce solutions within an expected processing time but have a sub-optimal solution. This trade-off is acceptable when quick processing times are critical in real-life scenarios requiring timely actions or responses.

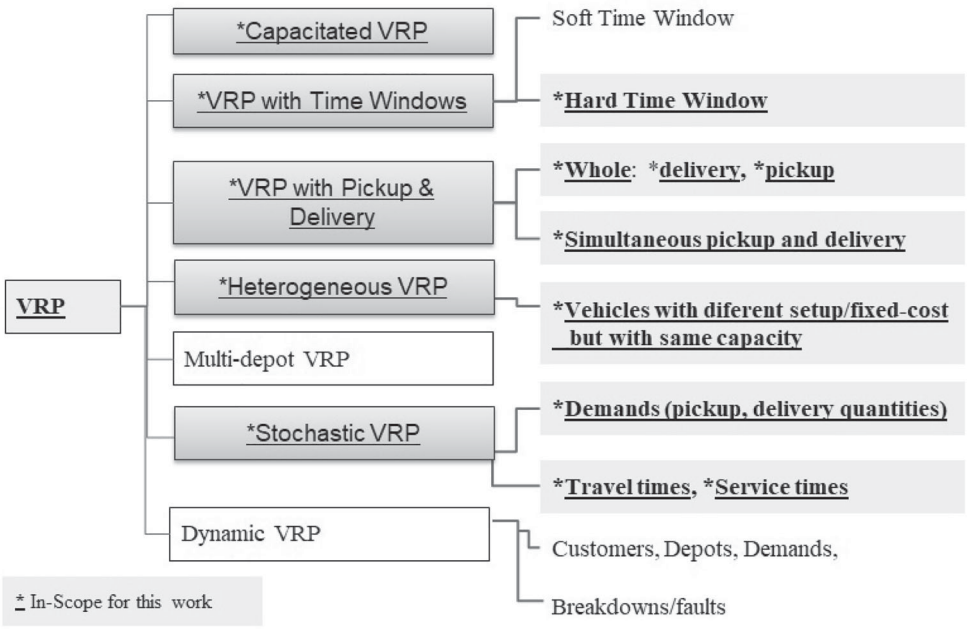

Figure 1. VRP (VRPSDPTW) taxonomy.

2. Contributions

In this work, we have studied, modelled, and solved a practical problem resulting from the fact that both delivery and pickup type scenarios exist in real-life logistics domain space with a good amount of feature data uncertainties.

More specifically, we have focused on a specific variant of VRP problem viz., Capacitated Simultaneous Delivery and Pickup with Time window (VRPSDPTW) with data level uncertainties w.r.t important features such as travel time, service time, delivery, and pickup quantities.

- We have formulated and proposed the following MILP based approaches specific to VRPSDPTW:
  - A general MILP approach for deterministic data scenario;
  - Two types of box uncertainty-based Robust Optimization (MILP_RO) approaches to handle different feature data uncertainties; and
  - A type of ellipsoidal uncertainty-based Robust Optimization (MILP_RO) approach to handle different feature data uncertainties;
- We have formulated and extended the following Attention-Model (AM) based DRL approaches specific to VRPSDPTW to train and handle different feature data uncertainties;
  - Static AM based vanilla *AM* [1] approach;
  - Dynamic AM based *DA_PSL* [2]; and
  - Dynamic AM based *DA_NL* (our approach);
- We have conducted extensive empirical tests to demonstrate the results and findings of the approaches. We have compared and analysed the usefulness and effectiveness of the robust optimization (MILP_RO) approach and Attention based Deep Reinforcement Learning approaches in handling uncertainties using realistic transportation domain public data sets.

3. Literature review

We briefly discuss the state-of-the-art works (table 1) with deterministic VRP and stochastic VRP.

3.1 *Deterministic VRP*

Jing Fan in [3] has addressed the vehicle routing problem with simultaneous pickup and delivery based on time window and proposed mathematical formulation to minimize the total length of vehicles and maximize the satisfaction of service quality. They have proposed the tabu search and reported the results of test instances. Hsiao and Ying-Yen in [4] have discussed simultaneous Delivery and Pickup Problem with Time Windows and proposed MIP model and genetic algorithm. Ran *et al* in [5] have addressed vehicle scheduling problems in home health care logistics and proposed Two mixed-integer programming models and a Genetic Algorithm (GA) and a Tabu Search (TS) method. The GA is based on a permutation chromosome, a split procedure, and a local search. The TS is based

Table 1. Review of papers.

Papers	Problem type	Objectives	Optimization approach	Uncertainty parameters	Solution methods
[3]	VRPSPD	Minimize transportation cost and maximize the customer satisfaction	Deterministic	–	MIP and tabu search
[4]	VRPSPDTW	Minimizing total dispatching cost and total travel cost	Deterministic	–	MIP and genetic algorithm
[5]	VRPSPDTW	Minimizing routing cost	Deterministic	–	MIP, genetic Algorithm and tabu search
[6]	VRPSPD	Minimizing total dispatching cost and total travel cost	Deterministic	–	Parallel simulated annealing algorithm
[7]	VRPSPDTW heterogeneous vehicles	Minimizing total dispatching cost and total routing cost	Deterministic	–	MILP and linear relaxation heuristic
[8]	VRPSPD	Minimize routing cost and handling cost	Deterministic	–	MIP, adaptive large neighbourhood search (ALNS) meta-heuristics
[9]	Green- VRPSPD	Minimize total fuel consumption cost	Deterministic	–	MIP and hyper-heuristic (HH-ILS) algorithm
[13]	CVRP	Minimize delivery cost	Robust	Demand	Robust formulation and an adaptive memory programming (AMP) meta-heuristic
[10]	VRPTW	Minimize the transportation cost	Robust	Demand and travel window	Robust formulation and branch-and-price-and-cut algorithm
[11]	VRPTW	Minimize the transportation cost	Robust	Demand and travel time	Robust optimization formulation and branch price and cut method
[12]	VRPTWMD (multiple delivery)	Minimize the operational cost	Robust	Demand	Robust formulation and heuristic
[14]	VRPTW	Minimize the travel time	Robust	Travel and service time	Robust formulation and adaptive large neighbourhood search (ANLS) algorithm

Table 1 continued

Papers	Problem type	Objectives	Optimization approach	Uncertainty parameters	Solution methods
[15]	Heterogeneous VRP	Minimum transportation cost and fleet composition	Robust	Demand	Meta-heuristics
[16]	VRPTWMD (multiple delivery)	Minimize the operational cost	Robust	Demand, travel and service times	Exact approach
[17]	humanitarian needs assessment planning	Maximizes the minimum excess coverage	Robust	Travel time	Robust formulation and tabu search
This paper★	VRPSPDTW	Minimizing total dispatching cost and total travel cost	Robust	Demand quantity, pickup quantity, travel time and service time	Robust formulation, heuristic to handle data uncertainty. (Please Note: Details on relevant DRL works in literature are covered in Section. 5.5)

on route assignment attributes of patients, an augmented cost function, route re-optimization, and attribute-based aspiration levels. Dong *et al* in [6] have discussed the vehicle routing problem with simultaneous pickup and delivery and proposed a parallel simulated annealing algorithm. Madankumar and Rajendran in [7] have discussed the vehicle routing problem with simultaneous delivery and pickup. They have proposed a MILP model and compared it with the existing MILP model. Richard *et al* in [8] have discussed the vehicle routing problem with simultaneous pickup and delivery and handling costs, formulated a MIP model, and showed how to adapt it to solve related problems. The proposed an adaptive large neighbourhood search (ALNS) heuristic to solve them. Olgun *et al* in [9] have addressed the green vehicle routing problem with simultaneous pickup and delivery and proposed a mathematical and hyper-heuristic (HH-ILS) algorithm based on iterative local search variables neighbourhood to solve large size problems.

3.2 *Stochastic VRP*

Da and Fatma in [10] have discussed the vehicle routing problem with time windows with demand uncertainty. They have proposed robust formulation and developed the branch-and-price-and-cut algorithm. Pedro *et al* in [11] have discussed the robust vehicle routing problem with time windows with customer demand and travel time uncertainties. They introduced a novel robust counterpart model on the budget uncertainty set and proposed a branch-price-and-cut method using a set partitioning formulation of the problem. Jonathan *et al* in [12] have addressed the vehicle routing problem with time windows and multiple deliverymen with demands uncertainty. They have proposed the robust mathematical formulation and constructive heuristic that extension of Solomon's heuristic.

Mehdi *et al* in [14] have discussed the vehicle routing problem with hard time windows with the uncertainty of demand, travel time, and service time. They have proposed an adaptive large neighbourhood search algorithm to derive the best robust solution with all uncertainties. Anirudh *et al.* in [15] have addressed heterogeneous vehicle routing problems and proposed a robust model and heuristic algorithm. Jonathan *et al* in [16] have proposed the exact approach for the vehicle routing problem with time windows and multiple deliverymen with demands, travel time, and service time uncertainty. The authors compared the solution with benchmark instances.

4. Problem description

Here, we consider a set of V vehicles; all of these vehicles are initially parked at a central depot 0 . Vehicles' have to travel to N customers before returning to the depot. Depot and

customers are located in different Geo-locations and are captured/tracked in as $x - y$ (*latitude*, *longitude*) planar position coordinates. Each travel by a vehicle between customers or depot and customer locations incur a cost owing to travel time. The vehicle can a) deliver items (in quantity); b) pickup items (in quantity); and c) both deliver and pickup items (in quantity) from either depot to customers (i.e., delivery) or customers to depot (i.e., pickup) incurring a service time at each customer site and possibility of wait time expend at some customer sites. VRP solution should adhere to constraints such as vehicle capacity, customer service time window duration, depot opening and closing times. Vehicles can be of heterogeneous configuration, such as capacity, fixed cost, operating cost, i.e., per unit usage based on time/distance, operational shift timings/duration, etc. A vehicle can choose to travel to a new location only when they have reached their previously decided target/location. A vehicle route starts from the depot, serves a set of customers, and returns to the depot. As soon as a vehicle returns to the depot after serving the customers, it must stay there in the depot. Once a customer is visited by a vehicle, it cannot be revisited by another vehicle. A vehicle route ends when there is no pending delivery and/or pickup customer orders, and all vehicles are stationed at the depot or have returned back to the depot. The goal is to find the routes taken by vehicles to serve the customers optimally to minimize the total costs (bi-objectivity: minimize vehicle dispatch and travel cost). Also, as part of the route formulation, there exists a possibility that the vehicle could wait at the current customer location post servicing or en-route, for a non-zero time period component (wait times or delayed start times) apart from the typical travel times, so that the vehicle can reach and meet the next customers' time-window criterion.

5. Solution approaches

Here, we describe in detail the following solution approaches to solve our VRP problem : a) MILP mathematical model; b) MILP Robust optimization model; c) Heuristic methods; and d) Reinforcement Learning and Deep Reinforcement Learning methods.

5.1 *Mathematical model*

5.1.1 *Indices and sets*

N total number of customers; depot is denoted by 0 and $N+1$.
V total number of vehicles

5.1.2 *Parameters*

α a constant for trade off between routing and dispatching cost
$c_{i,j}$ traveling cost between node i and node j
$t_{i,j}$ traveling time between node i and node j
s_i service time at node i
a^0 earliest start time of any vehicle at depot
a^i earliest start time for service at customer i
b^0 latest arrival time of any vehicle at depot
b^i latest arrival time for service at customer i
d_i demand for quantity to deliver at customer i
p_i demand for quantity to pickup at customer i
$fixed_v$ dispatching cost of vehicle v
C_v capacity limit of vehicle v
$FeatureSet_i$ Tuple of features that exhibit uncertainty for customer $i = (d_i, p_i, s_i, t_{i,j})$
M_1 a large positive value defined as: $= \left(\max_{0 \le i \le N, 0 \le j \le N} t_{i,j} \times (N+2) + \sum_{i=1}^{N} s_i \right)$
M_2 a large positive value defined as: $= 2 \times \left(\max_{1 \le v \le V} C_v \right)$

5.1.3 *Decision variables*

$x_{i,j,v}$ is equal to 1 if arc between customer i and customer j by vehicle v is selected, 0 otherwise
$del_{i,v}$ is equal to 1 if vehicle v is allocated to customer i, 0 otherwise
$start_v^0$ start time of vehicle v when it starts from depot
$st_{i,v}$ start time of service at customer i by vehicle v
$load_v^0$ total load on vehicle v at the time of starting from depot 0
$load_v^1$ total pickup on vehicle v at the time of starting from depot 0
$load_i$ total load on vehicle after completing the service at customer i

5.2 *Objective function*

The objective function of the problem is to find the optimal solution of routing plan to minimize the total cost that includes the routing cost and the fixed cost of vehicles as discussed in Eq. (1).

$$\begin{aligned} \text{minimize} \quad & \alpha \times \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N+1} \sum_{v=1}^{V} \left(c_{i,j} \times x_{i,j,v} \right) \\ & + (1-\alpha) \times \sum_{i=1}^{N} \sum_{v=1}^{V} \left(fixed_v \times x_{0,i,v} \right) \end{aligned} \tag{1}$$

subject to the following set of constraints:

The constraints (2)-(3) represents that a customer should be allocated to at least one vehicle.

$$\sum_{j=1}^{N+1}\sum_{v=1}^{V} x_{i,j,v} = 1, \quad i = 1,..,N \tag{2}$$

$$\sum_{j=0}^{N}\sum_{v=1}^{V} x_{j,i,v} = 1, \quad i = 1,..,N \tag{3}$$

Constraint (4) ensures that each vehicle should start from the depot before it starts serving a set of customers, and constraint (5) ensures that each vehicle should return back to the depot after serving the customers.

$$\sum_{j=1}^{N+1} x_{0,j,v} = 1, \quad v = 1,..,V \tag{4}$$

$$\sum_{i=0}^{N} x_{i,N+1,v} = 1, \quad v = 1,..,V \tag{5}$$

Constraint (6) represents the flow of constraint for each customer location such that if a vehicle visits customer i then the vehicle should leave the customer i after serving the customers' demand.

$$\sum_{i=0}^{N} x_{i,h,v} - \sum_{j=1}^{N+1} x_{h,j,v} = 0, \quad v = 1,..,V; h = 1,..,N \tag{6}$$

Constraint (7) is discussed the sub tours elimination in the route plan.

$$x_{i,i,v} = 0, \quad v = 1,..,V;\ i = 0,..,N+1 \tag{7}$$

Constraints (8)–(10) represents an arc that is selected for the route plan between customer i to j and the start time of service at j should be greater than the service time at customer i and travel time from customer i to customer j.

$$st_{i,v} + t_{i,j} + s_i - M_1 \times \left(1 - x_{i,j,v}\right) \le st_{j,v} \quad v = 1,..,V; i = 0,..,N; j = 1,..,N; i \ne j \tag{8}$$

$$start_v^0 + t_{0,i} - M_1 \times \left(1 - x_{0,i,v}\right) \le st_{i,v} \quad v = 1,..,V; i = 1,..,N \tag{9}$$

$$st_{i,v} + t_{i,N+1} + s_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \le st_{N+1,v} \quad v = 1,..,V; i = 1,..,N \tag{10}$$

Constraints (11)-(12) ensure the limits of total load and pickup at the vehicle when it starts from the depot.

$$load_v^0 = \sum_{i=0}^{N}\sum_{\substack{j=1 \\ i \ne j}}^{N} d_j \times x_{i,j,v} \quad v = 1,..,V \tag{11}$$

$$load_v^1 = \sum_{i=0}^{N}\sum_{\substack{j=1 \\ i \ne j}}^{N} p_j \times x_{i,j,v} \quad v = 1,..,V \tag{12}$$

Constraints (13)-(14) ensure the load at vehicle after visiting a customer.

$$load_i + p_j - d_j - M_2 \times (1 - x_{i,j,v}) \le load_j \quad v = 1,..,V; i,j = 1,..,N; i \ne j \tag{13}$$

$$load_v^0 + p_i - d_i - M_2 \times \left(1 - x_{0,i,v}\right) \le load_i \quad v = 1,..,V; i = 1,..,N \tag{14}$$

Constraints (15)-(19) represent the bound for decision variables:

$$a^0 \le start_v^0 \le b^0 \quad v = 1,..,V \tag{15}$$

$$a_i \le st_{i,v} \le b_i \quad v = 1,..,V; i = 1,..,N \tag{16}$$

$$0 \le load_v^0 \le C_v \quad v = 1,..,V \tag{17}$$

$$0 \le load_v^1 \le C_v \quad v = 1,..,V \tag{18}$$

$$0 \le load_i \le C_v \quad v = 1,..,V; i = 1,..,N \tag{19}$$

5.2.1 *Additional constraints* In the proposed MILP model, we define the constraints (20)–(22) to get the exact schedule given the service for each customer:

$$st_{j,v} \le st_{i,v} + t_{i,j} + s_i + M_1 \times \left(1 - x_{i,j,v}\right) \quad v = 1,..,V; i = 0,..,N; j = 1,..,N; i \ne j \tag{20}$$

$$load_j \le load_i + p_j - d_j + M_2 \times \left(1 - x_{i,j,v}\right) \quad v = 1,..,V; i,j = 1,..,N; i \ne j \tag{21}$$

$$load_i \le load_v^0 + p_i - d_i + M_2 \times \left(1 - x_{0,i,v}\right) \quad v = 1,..,V; i = 1,..,N \tag{22}$$

5.3 *Robust optimization approach: box uncertainty*

5.3.1 *Method 1* This section discusses the linear programming of the robust counterpart of MILP with box

uncertainty sets. The box uncertainty model was first proposed by the [18] and based which, we discuss the robust counterpart of a standard mixed-integer linear programming with box uncertainty sets as follows:

A standard mixed integer programming:

$$\text{minimize} \quad c_j x_j \tag{23}$$

$$s.t. \quad \sum_j a_{i,j} x_j \le b_i \tag{24}$$

$$x \ge 0 \tag{25}$$

Let us consider J_i the set of coefficients in row i that are subject to uncertainty, i.e. J_i is coefficient of $a_{i,j}$. The constraint (24) has an uncertain vector $a_{i,j}$ that is modeled as a symmetric with mean value that is equal to nominal value $\overline{a}_{i,j}$ and bounded by random variable $\widetilde{a}_{i,j}$, $j \in J_i$ and takes values in interval $\left[\overline{a}_{i,j} - \widetilde{a}_{i,j}, \overline{a}_{i,j} + \widetilde{a}_{i,j}\right]$. The equation is re-written as follows:

$$\sum_j \overline{a}_{i,j} x_j + max_{\left\{e_i \bigcup \{r_i\} | e_i \subseteq J_i, |e_i| = \lfloor \Gamma_i \rfloor, r_i \in J_i \setminus e_i\right\}} \left\{ \sum_{j \in e_i} \widetilde{a}_{i,j} |x_j| \right\} + (\Gamma_i - \lfloor \Gamma_i \rfloor) \widetilde{a}_{i,e_i} |x_{e_i}| \le b_i \tag{26}$$

J_i is an integer and the uncertainty in the constraint i and Γ_i is the uncertainty budget ($\Gamma_i = [0, |J_i|]$). To change the above equation into linear optimization, i^{th} constraint is protected by function $\beta(x, \Gamma_i)$ and defined as follow:

$$\beta(x, \Gamma_i) = max_{\left\{e_i \bigcup \{r_i\} | e_i \subseteq J_i, |e_i| = \lfloor \Gamma_i \rfloor, r_i \in J_i \setminus e_i\right\}} \left\{ \sum_{j \in e_i} \widetilde{a}_{i,j} |x_j| \right\} + (\Gamma_i - \lfloor \Gamma_i \rfloor) \widetilde{a}_{i,e_i} |x_i| \tag{27}$$

The protected function is equivalent to the following linear optimization problem:

$$\beta_i(x, \Gamma_i) = \text{maximize} \quad \sum_{j \in J_i} \widetilde{a}_{i,j} z_{i,j} |x_j| \tag{28}$$

$$s.t. \quad \sum_{j \in J_i} z_{i,j} \le \Gamma_i \tag{29}$$

$$0 \le z_{i,j} \le 1 \quad \forall j \in J_i \tag{30}$$

The duality of the above linear problem as follows:

$$\text{minimize} \quad \sum_{j \in J_i} \delta_{i,j} + \Gamma_i \lambda_i \tag{31}$$

$$s.t. \quad \lambda_i + \delta_{i,j} \ge \widetilde{a}_{i,j} |x_j| \quad \forall i, j \in J_i \tag{32}$$

$$\delta_{i,j} \ge 0 \quad \forall j \in J_i \tag{33}$$

$$\lambda_i \ge 0 \quad \forall i \tag{34}$$

$\delta_{i,j}$ and λ_i are the dual variables. The robust counterpart of linear programming can written as follows:

$$\text{minimize} \quad \sum_j c_j x_j \tag{35}$$

$$s.t. \quad \sum_j \overline{a}_{i,j} x_j + \lambda_i \Gamma_i + \sum_{j \in J_i} \delta_{i,j} \le b_i \tag{36}$$

$$\lambda_i + \delta_{i,j} \ge \widetilde{a}_{i,j} \eta_j \quad \forall i, j \in J_i \tag{37}$$

$$-\eta_j \le x_j \le \eta_j \quad \forall j \tag{38}$$

$$\delta_{i,j} \ge 0 \quad \forall i, j \in J_i \tag{39}$$

$$\eta_j \ge 0 \quad \forall j \tag{40}$$

$$\lambda_i \ge 0 \quad \forall i \tag{41}$$

In the above MILP model, we consider uncertain parameters such as travel time $t_{i,j}$ that is modeled as symmetric and bounded random variables that takes values $\in [\overline{t}_{i,j} - \widetilde{t}_{i,j}, \overline{t}_{i,j} + \widetilde{t}_{i,j}]$, service time s_i that takes values $\in [\overline{s}_i - \widetilde{s}_i, \overline{s}_i + \widetilde{s}_i]$, demand of delivery that takes values $\in \left[\overline{d}_i - \widetilde{d}_i, \overline{d}_i + \widetilde{d}_i\right]$ and pickup that takes values $\in [\overline{p}_i - \widetilde{p}_i, \overline{p}_i + \widetilde{p}_i]$. The robust counterpart of the MILP model problem with box uncertainty sets is discussed as follows:

Constraints (2)-(7) and (15)-(19) hold,

$$st_{i,v} + \overline{t}_{i,j} + \lambda^T_{i,j} \Gamma^T_{i,j} + \delta^T_{i,j} + \overline{s}_i + \lambda^S_i \Gamma^S_i + \delta^S_i - M_1 \times \left(1 - x_{i,j,v}\right) \le st_{j,v} \quad v = 1, .., V; i = 0, .., N; j = 1, .., N; i \ne j \tag{42}$$

$$\lambda^T_{i,j} + \delta^T_{i,j} \ge \widetilde{t}_{i,j} \times x_{i,j,v} v = 1, .., V; i = 0, .., N; \quad j = 1, .., N; i \ne j \tag{43}$$

$$\lambda^S_i + \delta^S_i \ge \widetilde{s}_i \times x_{i,j,v} v = 1, .., V; i = 0, .., N; \quad j = 1, .., N; i \ne j \tag{44}$$

$$start^0_v + \overline{t}_{0,i} + \lambda^T_{0,i} \Gamma^T_{0,i} + \delta^T_{0,i} - M_1 \times \left(1 - x_{0,i,v}\right) \le st_{i,v} \quad v = 1, .., V; i = 1, .., N \tag{45}$$

$$\lambda^T_{0,i} + \delta^T_{0,i} \ge \widetilde{t}_{0,i} \times x_{0,i,v} \quad v = 1, .., V; i = 1, .., N \tag{46}$$

$$st_{i,v} + \overline{t}_{i,N+1} + \lambda^{T}_{i,N+1}\Gamma^{T}_{i,N+1} + \delta^{T}_{i,N+1} + \overline{s}_i + \lambda^{S}_{i}\Gamma^{S}_{i} + \delta^{S}_{i} - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v} \quad v = 1,..,V; i = 1,..,N \tag{47}$$

$$\lambda^{T}_{i,N+1} + \delta^{T}_{i,N+1} \geq \widetilde{t}_{i,N+1} \times x_{i,N+1,v} \quad v = 1,..,V; i = 1,..,N \tag{48}$$

$$\lambda^{S}_{i} + \delta^{S}_{i} \geq \widetilde{s}_i \times x_{i,N+1,v} \quad v = 1,..,V; i = 1,..,N \tag{49}$$

$$load^{0}_{v} = \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{d}_j \times x_{i,j,v} + \lambda^{D}_{o}\Gamma^{D}_{o} + \sum_{j=1}^{N} \delta^{D}_{j} \quad v = 1,..,V \tag{50}$$

$$\lambda^{D}_{o} + \delta^{D}_{j} \geq \widetilde{d}_j \times x_{i,j,v} \quad v = 1,..,V; i = 0,..,N \quad j = 1,..,N; i \neq j \tag{51}$$

$$load^{1}_{v} = \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{p}_j \times x_{i,j,v} + \lambda^{PU}_{o}\Gamma^{PU}_{o} + \sum_{j=1}^{N} \delta^{PU}_{j} \quad v = 1,..,V \tag{52}$$

$$\lambda^{PU}_{o} + \delta^{PU}_{j} \geq \widetilde{p}_j \times x_{i,j,v} \quad v = 1,..,V; i = 0,..,N \quad j = 1,..,N; i \neq j \tag{53}$$

$$load_i + \overline{p}_j + \lambda^{PU}_{j}\Gamma^{PU}_{j} + \delta^{PU}_{j} - \overline{d}_j - \lambda^{D}_{j}\Gamma^{D}_{j} - \delta^{D}_{j} - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_j \quad v = 1,..,V; i,j = 1,..,N; i \neq j \tag{54}$$

$$\lambda^{PU}_{j} + \delta^{PU}_{j} \geq \widetilde{p}_j \times x_{i,j,v}\, v = 1,..,V; \quad i,j = 1,..,N; i \neq j \tag{55}$$

$$\lambda^{D}_{j} + \delta^{D}_{j} \geq \widetilde{d}_j \times x_{i,j,v} v = 1,..,V; \quad i,j = 1,..,N; i \neq j \tag{56}$$

$$load^{0}_{v} + \overline{p}_i + \lambda^{PU}_{i}\Gamma^{PU}_{i} + \delta^{PU}_{i} - \overline{d}_i - \lambda^{D}_{i}\Gamma^{D}_{i} - \delta^{D}_{i} - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_i \quad v = 1,..,V; i = 1,..,N \tag{57}$$

$$\lambda^{PU}_{i} + \delta^{PU}_{i} \geq \widetilde{p}_i \times x_{0,i,v} \quad v = 1,..,V; i = 1,..,N \tag{58}$$

$$\lambda^{D}_{i} + \delta^{D}_{i} \geq \widetilde{d}_i \times x_{0,i,v} \quad v = 1,..,V; i = 1,..,N \tag{59}$$

$$\delta^{T}_{i,j}, \delta^{S}_{i}, \delta^{PU}_{i}, \delta^{D}_{i}, \lambda^{T}_{i,j}, \lambda^{S}_{i}, \lambda^{PU}_{i}, \lambda^{D}_{i} \geq 0 \forall i, \quad \forall j \tag{60}$$

5.3.2 *Method 2* In this section, we discuss the exact solution approach using box uncertainty sets for VRPSDPTW. We extend the exact approach proposed by [16] for uncertainty in demand, pickup, travel, and service time parameters. This approach considers the recursive equations (for example, Eqs. (61)-(64) used to track uncertainty in travel and service times) to check the feasibility of the solution while considering the uncertain parameters. Each uncertainty set corresponds to a budgeted uncertainty set as discussed by [18]. In here, similar to Eq. (29), instead of using Γ_i, we use budgeted uncertainty set as $\Gamma^{(d,p)}$ and $\Gamma^{(t,s)}$ for uncertain demand, uncertain pickup, uncertain travel and service times. The exact method of MILP model with box uncertainty sets is discussed as follows:

Additionally, please note constraints (2)- (7) hold.

$$st_{i,v,\gamma} + \overline{t}_{i,j} + \overline{s}_i - M_1 \times \left(1 - x_{i,j,v}\right) \leq st_{j,v,\gamma} \quad v = 1,..,V \quad i = 0,..,N; j = 1,..,N; i \neq j; \gamma = 0,..,\Gamma^{(t,s)} \tag{61}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,j} + \widetilde{t}_{i,j} + \overline{s}_i - M_1 \times \left(1 - x_{i,j,v}\right) \leq st_{j,v,\gamma} \quad v = 1,..,V \quad i = 0,..,N; j = 1,..,N; i \neq j; \gamma = 1,..,\Gamma^{(t,s)} \tag{62}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,j} + \overline{s}_i + \widetilde{s}_i - M_1 \times \left(1 - x_{i,j,v}\right) \leq st_{j,v,\gamma} \quad v = 1,..,V \quad i = 0,..,N; j = 1,..,N; \quad i \neq j; \gamma = 1,..,\Gamma^{(t,s)} \tag{63}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,j} + \widetilde{t}_{i,j} + \overline{s}_i + \widetilde{s}_i - M_1 \times \left(1 - x_{i,j,v}\right) \leq st_{j,v,\gamma} \quad v = 1,..,V; i = 0,..,N; j = 1,..,N; \quad i \neq j; \gamma = 2,..,\Gamma^{(t,s)} \tag{64}$$

$$start^{0}_{v,\gamma} + \overline{t}_{0,i} - M_1 \times \left(1 - x_{0,i,v}\right) \leq st_{i,v,\gamma} \quad v = 1,..,V \quad i = 1,..,N; \gamma = 0,..,\Gamma^{(t,s)} \tag{65}$$

$$start^{0}_{v,\gamma-1} + \overline{t}_{0,i} + \widetilde{t}_{0,i} - M_1 \times \left(1 - x_{0,i,v}\right) \leq st_{i,v,\gamma} \quad v = 1,..,V \quad i = 1,..,N; \gamma = 1,..,\Gamma^{(t,s)} \tag{66}$$

$$st_{i,v,\gamma} + \overline{t}_{i,N+1} + \overline{s}_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v,\gamma} \quad v = 1,..,V; i = 1,..,N; \quad \gamma = 0,..,\Gamma^{(t,s)} \tag{67}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,N+1} + \widetilde{t}_{i,N+1} + \overline{s}_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 1,..,\Gamma^{(t,s)} \tag{68}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,N+1} + \overline{s}_i + \widetilde{s}_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 1,..,\Gamma^{(t,s)} \tag{69}$$

$$st_{i,v,\gamma-1} + \overline{t}_{i,N+1} + \widetilde{t}_{i,N+1} + \overline{s}_i + \widetilde{s}_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 2,..,\Gamma^{(t,s)} \tag{70}$$

$$load_v^0 \geq \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{d}_j \times x_{i,j,v} \quad v = 1,..,V \tag{71}$$

$$load_v^1 \geq \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{p}_j \times x_{i,j,v} \quad v = 1,..,V \tag{72}$$

$$load_{i,\gamma} + \overline{p}_j - \overline{d}_j - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_{j,\gamma} \quad v = 1,..,V; i,j = 1,..,N; i \neq j; \gamma = 0,..,\Gamma^{(d,p)} \tag{73}$$

$$load_{i,\gamma-1} + \overline{p}_j + \widetilde{p}_j - \overline{d}_j - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_{j,\gamma} \quad v = 1,..,V; i,j = 1,..,N; i \neq j; \gamma = 1,..,\Gamma^{(d,p)} \tag{74}$$

$$load_{i,\gamma-1} + \overline{p}_j - \overline{d}_j - \widetilde{d}_j - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_{j,\gamma} \quad v = 1,..,V; i,j = 1,..,N; i \neq j; \gamma = 1,..,\Gamma^{(d,p)} \tag{75}$$

$$load_{i,\gamma-2} + \overline{p}_j + \widetilde{p}_j - \overline{d}_j - \widetilde{d}_j - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_{j,\gamma} \quad v = 1,..,V; i,j = 1,..,N; i \neq j; \gamma = 2,..,\Gamma^{(d,p)} \tag{76}$$

$$load_{v,\gamma}^0 + \overline{p}_i - \overline{d}_i - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_{i,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 0,..,\Gamma^{(d,p)} \tag{77}$$

$$load_{v,\gamma-1}^0 + \overline{p}_i + \widetilde{p}_i - \overline{d}_i - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_{i,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 1,..,\Gamma^{(d,p)} \tag{78}$$

$$load_{v,\gamma-1}^0 + \overline{p}_i - \overline{d}_i - \widetilde{d}_i - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_{i,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 1,..,\Gamma^{(d,p)} \tag{79}$$

$$load_{v,\gamma-2}^0 + \overline{p}_i + \widetilde{p}_i - \overline{d}_i - \widetilde{d}_i - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_{i,\gamma} \quad v = 1,..,V; i = 1,..,N; \gamma = 2,..,\Gamma^{(d,p)} \tag{80}$$

$$a^0 \leq start_{v,\gamma}^0 \leq b^0 \quad v = 1,..,V; \gamma = 0,..,\Gamma^{(t,s)} \tag{81}$$

$$a_i \leq st_{i,v,\gamma} \leq b_i \quad v = 1,..,V; i = 1,..,N; \gamma = 0,..,\Gamma^{(t,s)} \tag{82}$$

$$0 \leq load_{v,\gamma}^0 \leq C_v \quad v = 1,..,V; \gamma = 0,..,\Gamma^{(d,p)} \tag{83}$$

$$0 \leq load_{i,\gamma} \leq C_v \quad v = 1,..,V; i = 1,..,N; \gamma = 0,..,\Gamma^{(d,p)} \tag{84}$$

5.4 *Robust optimization approach: ellipsoidal uncertainty*

We have adopted the ellipsoidal uncertainty robust counterpart model initially proposed by [19] to account for uncertainty. The details of this model are as follows:

$$\text{minimize} \quad c_j x_j \tag{85}$$

$$s.t. \quad \sum_j \overline{a}_{i,j} x_j + \Omega \sqrt{\sum_{j \in J_i} \widetilde{a}_{i,j}^2 x_j^2} \leq b_i \tag{86}$$

$$x \geq 0 \tag{87}$$

The robust counterpart of the MILP model problem with ellipsoidal uncertainty sets is discussed as follows:

Constraints (2)- (7) and (15)- (19) hold,

$$st_{i,v} + \overline{t}_{i,j} + \Omega \times \widetilde{t}_{i,j} + \overline{s}_i + \Omega \times \widetilde{s}_i - M_1 \times \left(1 - x_{i,j,v}\right) \leq st_{j,v} \quad v = 1,..,V; i = 0,..,N; j = 1,..,N; i \neq j \tag{88}$$

$$start_v^0 + \overline{t}_{0,i} + \Omega \times \widetilde{t}_{0,i} - M_1 \times \left(1 - x_{0,i,v}\right) \leq st_{i,v} \quad v = 1,..,V; i = 1,..,N \tag{89}$$

$$st_{i,v} + \overline{t}_{i,N+1} + \Omega \times \widetilde{t}_{i,N+1} + \overline{s}_i + \Omega \times \widetilde{s}_i - M_1 \times \left(1 - x_{i,N+1,v}\right) \leq st_{N+1,v} \quad v = 1,..,V; i = 1,..,N \tag{90}$$

$$load_v^0 = \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{d}_j \times x_{i,j,v} + \Omega \times \sqrt{\sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \widetilde{d}_j^2 \times x_{i,j,v}^2} \quad v = 1, .., V \tag{91}$$

$$load_v^1 = \sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \overline{p}_j \times x_{i,j,v} + \Omega \times \sqrt{\sum_{i=0}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \widetilde{p}_j^2 \times x_{i,j,v}^2} \quad v = 1, .., V \tag{92}$$

$$load_i + \overline{p}_j + \Omega \times \widetilde{p}_j - \overline{d}_j - \Omega \times \widetilde{d}_j - M_2 \times \left(1 - x_{i,j,v}\right) \leq load_j \quad v = 1, .., V; i, j = 1, .., N; i \neq j \tag{93}$$

$$load_v^0 + \overline{p}_i + \Omega \times \widetilde{p}_i - \overline{d}_i - \Omega \times \widetilde{d}_i - M_2 \times \left(1 - x_{0,i,v}\right) \leq load_i \quad v = 1, .., V; i = 1, .., N \tag{94}$$

5.5 *Robust deep reinforcement learning*

Deep learning using multi-level neural networks (DNN) enables Reinforcement Learning (RL) to scale to decision-making problems that were previously intractable or unfeasible [20]. DNNs though mainly used to make predictions, when used with Reinforcement Learning (RL), have enabled algorithms to learn to make decisions by interacting with the problem environment/context. Algorithms typically have heuristics expressed in rules, which can be interpreted as policies to make solutioning decisions. These policies can be parameterized using DNNs, and are further trained to obtain better models (or stronger algorithms), which are used to solve combinatorial optimization problems [1] using RL techniques are referred to as Deep Reinforcement Learning (DRL) approach.

Many of the exact or heuristics type solution approaches to VRP problems still lack the ability to extract knowledge from past experiences and blindly explore new solutions every time the customer configuration changes. This is a reason why approaches based on DNNs, and more specifically Deep RL, have been recently proposed to learn heuristics based on data automatically. DNN based RL can extract patterns and knowledge from past experiences and historical data, learning compact and informative representation to build a solution for a new instance of the same problem without any human expert intervention. RL approaches such as Pointer networks, Sequence to Sequence networks, Attention mechanisms, especially Multi-Head Attention layer, cyclic positional encoding (CPE), heuristics, Transformer encoder ([1, 2, 21–26]) etc., though have been used successfully to solve VRPs are single-agent solutioning approaches.

There are good amount of recent works that use DRL methods to solve stochastic VRP problems such as [21, 23–25, 27–29] etc., each of which solve a certain class of VRP problem using certain DRL approaches. In literature, we find very limited number of recent works that handle dynamic type VRP problems. One such work is by Waldy *et al* in [30] proposed a RL approach which uses Temporal Difference (*TD*) learning technique with experience replay to approximate the value function for the initial input data, which then is subsequently utilized and optimized by a Simulated Annealing (*SA*) algorithm which accounts for dynamic input data changes to generate the revised routes during run-time. The authors consider demand uncertainty but have not considered vehicle capacity constraint, and travel time or service time uncertainties. Balaji *et al* in [27] considered stochastic and dynamic capacitated VRP problems with pickup and delivery, time windows, and service guarantee, The author's showed that the DRL approach can directly be trained to solve very small-sized problems (synthetically generated and oversimplified) and produce competitive or superior results compared with classical baseline methods. One drawback with this work is that the authors have not looked at a realistic-sized problem space to ascertain the solution viability and quality competitiveness.

Though we understand that the applicability of DRL best suits to problem setting with high dimensional state and action spaces [20], we are of the belief that DRL would work right for complex VRP problems of high dimensional state-action spaces and more importantly its workability in practical scenarios encompassing data quality uncertainties [31]. This is an interesting open problem that we have tried understanding this problem scalability and data quality uncertainty perspective in this work of ours.

In this work, our focus is on Attention-Model (AM) method based VRP solutioning. One of the early works using attention model mechanism to efficiently solve VRPs and TSPs was proposed by [1]. In this paper, the authors have introduced a Monte-Carlo baseline to reduce the variance of the gradient estimator during training. The Attention Model (AM) learns to represent the graph of customers through a Transformer encoder. A context vector is created from the encodings of the involved parameters, which get updated after each decision. This context vector drives an attention mechanism that computes a score for each node, which is used as the probability to select it as the next target (node) in the route.

There are few works such as [2, 32, 33] which have adopted AM method propose to address variants of VRPs. Author's Li *et al* in [32], have proposed a DRL with

attention mechanism-based approach to solve precedence criteria with pickup-delivery pairings (i.e., pickup node must precede the paring delivery node) heterogeneous attention mechanism specifically prescribes attentions for each role of the nodes while accounting for precedence constraint, i.e., the pickup node must precede the pairing delivery node. This notion of pickup-delivery connotation is different to our work, in which we have considered accounting for customer node which has both delivery and pickup needs, and we do serve both delivery and pickup needs or each customer simultaneously. Also, the authors have not mentioned the features that they have considered and have not tested their approach at different levels of uncertainty. We believe that the focus is on delivery and pickup quantities and have not considered travel time or service time uncertainties.

Author's Li *et al* in [33] have proposed a DRL with attention mechanism approach to solve VRP problem comprising of vehicles of different capacities. They use two attentions in each step to track vehicle selection and node selection. They have used a small vehicle fleet sizes (3 and 5) to generate vehicle routes aimed at minimizing the longest or total travel time of the vehicle(s). Also, the authors have not mentioned the features that they have considered and have not tested their approach at different levels of uncertainty. We believe that the focus is on delivery quantities and have not considered travel time or service time uncertainties. Author's Peng et al. in [2] have used dynamic attention model with dynamic encoder-decoder architecture. In this work, the authors have invoked the encoder embedding each time a partial solution gets created.

We refer to the AM based method by [1] as vanilla approach in the rest of this paper. In the vanilla *AM* approach, the initial input is passed into the encoder to generate the embeddings and the mean context vector. The decoder iteratively uses this initial embedding and context vector in arriving at the complete solution.

In our work, we have considered a capacitated VRP problem to handle simultaneous delivery and pickup at each customer node and adhere to a non-flexible customer service time window constraint (i.e., hard service deadlines to be met and does not involve penalty factor considerations) with an objective to minimize the complete journey route's cost. We have considered heterogeneous vehicle setup, in which set of vehicles have varied fixed/capital/setup cost whilst with same fixed capacity. Also, we have assumed that the use of the vehicles in generating a vehicle's route schedule follows a pre-defined deployment sequence/order which is provided as an input configuration [vehicle's fixed-costs for the given input data-set is captured in table 2].

In the below subsection's, we provide further details on our *DA_NL* approach which uses node-level attention model embedder invocation approach to arrive at the complete solution.

5.5.1 *VRP RL problem formulation and preliminaries* Specifically, for VRP instance, the input $X = x_0, ..., x_N$ is a set of nodes and x_0 is the depot. Each node consists of six features or elements $x_i = (loc_i, d_i, p_i, a^i, b^i, s_i)$, where loc_i is a 2-dimensional coordinate of customer or node i in Euclidean space (please refer to 5.1 for other notations used in here). The solution π is a sequence defined as: $\pi = (\pi_1, ..., \pi_T)$, $\pi_t \in \{x_0, ..., x_N\}$,

Table 2. VRP dataset details.

Dataset instance	# of Nodes	Vehicle fixed cost/unit	Vehicle capacity	Problem size	Ref.
2_1	10	[110, 122]	200	Small	[7]
2_2	10	[94, 104]	200	Small	
2_3	10	[94,104]	1000	Small	
2_4	20	[51,59,66,74]	200	Small	
2_5	20	[51, 59, 66, 74]	200	Small	
2_6	20	[95, 111,124,139]	1000	Small	
2_7	20	[71,83,93,104]	700	Small	
2_8	20	[95,111,124,139]	200	Small	
2_9	20	[106, 124, 138, 154]	200	Small	
2_10	30	[55,55, 64, 71, 80, 95]	200	Medium	
2_11	30	[55,55,64,71,80,95]	200	Medium	
2_12	30	[96,96,112,124,139,166]	200	Medium	
2_100	100	[168]	200	Large	***
15_15	100	[110]	100	Large	[36]

Note: Vehicle operational cost per unit = 1

***Synthetically generated sample

where each customer node is visited exactly once and the depot can be visited multiple times. T is the length of sequence π that may vary across possible solutions.

$$p_\theta(\pi|X) = \prod_{t=1}^{T} p_\theta(\pi_t|X, \pi_{1:t-1}) \tag{95}$$

5.5.2 *Encoder* Encoder uses a graph attention network to encode node features into encoder embedding and context vectors similar to encoder in transformer architecture in [34]. Firstly, for each dED_x-dimensional (for VRP, dED_x = 7, the coordinate, delivery, pickup, earliest start time, latest arrival time, service time) input node x_i, the dED_h-dimensional ($dED_h = 128$) initial node embedding $h_i^{(0)}$ for for depot (where $i = 0$) and non-depot nodes ($\forall i = 1, ..N$) is computed through a linear transformation with learnable parameters of weights $W \in \mathbb{R}^{dED_h \times dED_x}$ and bias $b \in \mathbb{R}^{dED_h}$ for non-depot locations, and separate weight parameter W_0 and bias parameter b_0 for depot. The initial node embeddings $h_i^{(0)}$ is fed into the first layer of graph attention network and updated with the number of attention layers L being used. In our work, we have considered $L = 3$ attention layers. Each attention layer consists of a multi-head (MHA) sub-layer and a fully connected feed-forward (FF) sub-layer.

Multi-Head Attention sub-layer: Multi-head attention is used to extract different types of information as detailed in [34]. In this layer $\mathcal{L} \in 1..L, h_i^{(\mathcal{L})}$ is denoted as the node embedding of each node i, and the output $h_0^{\mathcal{L}-1}, ..., h_N^{\mathcal{L}-1}$ of the layer $(\mathcal{L} - 1)$ is the input to the layer $\mathcal{L}$. The multi-head attention (number of attention heads M= 8) vector is is computed as per [34] and is defined as follows:

$$MHA_i^{\mathcal{L}}(h_0^{\mathcal{L}-1}, ..., h_N^{\mathcal{L}-1}); \forall node x_i \tag{96}$$

Feed-Forward sub-layer: In this sub-layer, for each node i, based on multi-head attention vector, $h_i^{\mathcal{L}}$ is computed by skip-connection and fully connected feed-forward (FF) network. FF sublayer uses ReLu activation.

$$\hat{h_i^{\mathcal{L}}} = tanh(h_i^{\mathcal{L}-1} + MHA_i^{\mathcal{L}}(h_0^{\mathcal{L}-1}, ..., h_N^{\mathcal{L}-1}); \forall node x_i \tag{97}$$

$$h_i^{\mathcal{L}} = tanh(\hat{h_i^{\mathcal{L}}} + FF(\hat{h_i^{\mathcal{L}}})); \forall node x_i \tag{98}$$

$$h_i^{L} = ENCODE_i^{L}(h_0^0, ..., h_N^0); \forall node x_i \tag{99}$$

$ENCODE_i^L(h_0^0, ..., h_N^0); \forall node x_i$ is computed using Eqs. (96)-(98). The encoder computes an aggregated embedding $\overline{h}_i^{\mathcal{L}}$ of the input graph as the mean of the final node embeddings $h_i^{\mathcal{L}}$;

$$\overline{h}_i^{L} = \frac{1}{N} \sum_{i=1}^{N} (h_i^L) \tag{100}$$

Both the node embeddings h_i^L and context graph embeddings $\overline{h}_i^L$ are used as input to the decoder.

5.5.3 *Decoder* Decoding happens sequentially, and at timestep $t \in 1, ...T$, the decoder selects one node to visit based on partial solutions $\pi_1, .., \pi_{t-1}$ and the embedding of nodes (visited nodes till step $t - 1$ are treated as masked), and the mean embeddings of the unvisited nodes (step t) as the mean context vectors. Context vector h_c is computed by M-head attention mechanism following the approach proposed by [1]. in order to construct a feasible solution, the node that violates the constraints will be masked (refer to section 5.5.5). Finally, the probability $p_\theta(\pi_t|X, \pi_{1:t-1})$ is computed with a single-head attention mechanism [1].

5.5.4 *Dynamic attention model* After the vehicle updates the solution sequence with the next selected node, the remaining or unvisited nodes could be considered as a new (smaller) instance (graph) to be solved. Here the idea is to update embedding of the remaining or unvisited nodes using encoder as and when the vehicle updates the solution sequence with the next selected node [Eq. (101)].

$$h_i^{L} = ENCODE_i^{L}(h_0^0, ..., h_N^0) \tag{101}$$

The modified encoder-decoder architecture with dynamic attention model setup is shown in figure 2. Firstly, the structural features of the input instance are extracted by the encoder to generate mean context vectors and embedding, which is used by the decoder to construct the solution incrementally. In the decoder's construction step, next node is selected based on probability distribution (Eq. 95) over nodes and appended to the π_t. In our modified approach, the encoder embedding, and mean context vector will be recomputed each time as and when the next node is selected accounting for granular *state* and context level changes. This $Decoder - Encoder'$ processing cycle continues till all nodes have been visited, and finally produces the complete solution π.

5.5.5 *VRP RL problem context*

- *State*: *State* of the system connotates to various aspects such as a set of visited customers, set of pending or non-visited customers, the last customer served by each vehicle, the time elapsed since each vehicle left the depot, customers being served at this time (epoch), will the vehicle reach the customer's location adhering

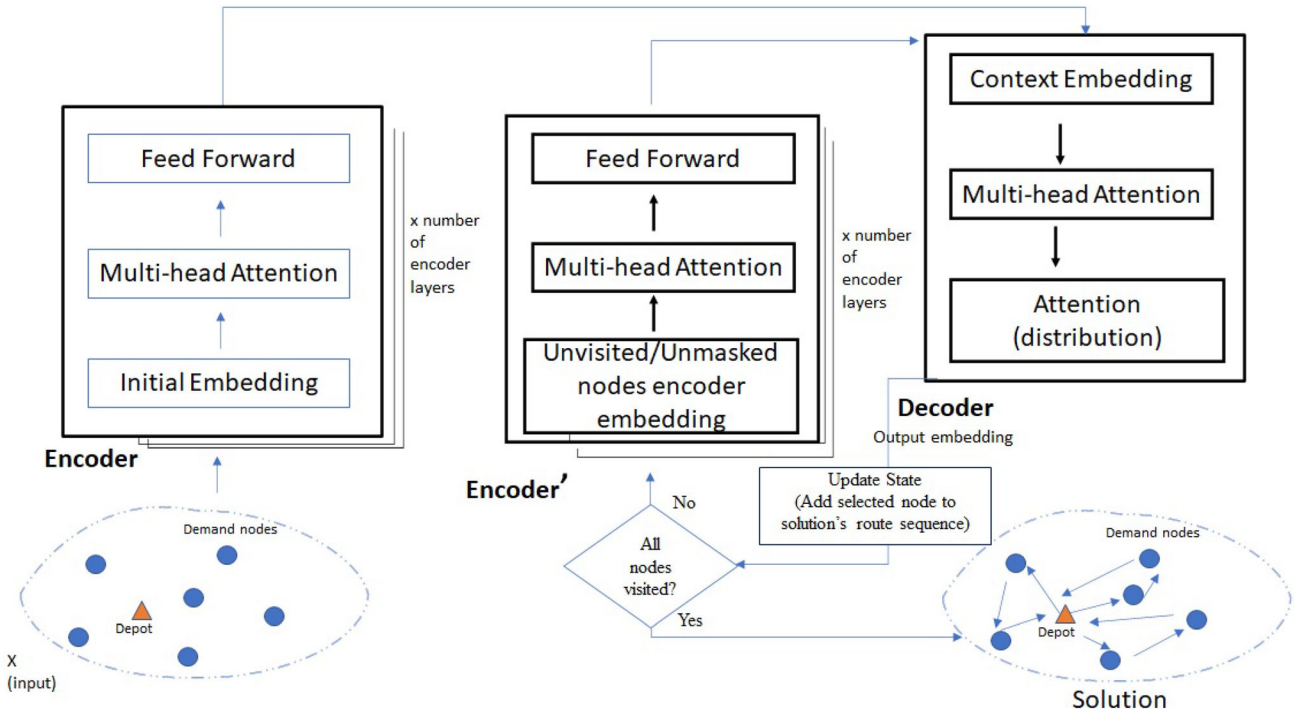

Figure 2. Dynamic encoder-decoder architecture.

to customer's time window, whether the vehicle has reached the goal state, i.e., depot, distance travelled and/or travelled time until now, current vehicle location, vehicles' load or used capacity, vehicles' remaining capacity, cumulative pickup quantity from the customers serviced till now, yet to be serviced customer's etc.

- *Mask*: *Mask* represents the different constraints required in formulating a valid vehicle-customer allocation and route schedule. Following are the constraints that have been considered:
  - capacity/load constraints– which accounts for customer delivery and/or pickup quantities to be $\leq$ vehicles' capacity at each and every node(includes depot) that the vehicle visits. This constraint is defined in Eqs. (17–18)
  - service window constraints– vehicle to reach (arrive at) customer's(node) location within the defined service time window constraints– vehicle to return back to the depot within the depot's round trip close time window. $st_{j,v} = \max((st_{j,v} + t_{i,j} + s_i), a_j) \times x_{i,j,v}; v = 1, .., V; i = 0, .., N; j = 1, .., N; i \neq j$ constraints that need to be satisfied are as follows:
    $st_{j,v} \geq st_{i,v}; v = 1, .., V; i = 0, .., N; j = 1, .., N; i \neq j$
    $st_{j,v} \geq a_j; v = 1, .., V; j = 0, .., N$
    $st_{j,v} \leq b_j; v = 1, .., V; j = 0, .., N$
  - visited node constraints– already visited nodes should not be considered
- *Reward*: We have used (negative) objective function cost as defined in Eq. (1) as the *Reward*. Here the objective is to arrive at an acceptable trade-off between fixed cost driven primarily by the vehicle-type and operational cost driven by the vehicle's travel distance or travel time. In our work, we have considered travel distance and travel time to be the same (same denomination or unit)
- *Action*: is the decision as to which node or customer to be handled next by the vehicle in its route or the decision on which vehicle should handle the next unvisited node from the state contexts.

5.5.6 *Model training* The attention model (AM) is trained using REINFORCE (Algorithm 1) [35] with a simple but effective greedy rollout baseline. Given an instance X, our training objective is the tour cost (Eq.(1)) of solution π.

In here, $\bigtriangledown L$ is the gradients of the parameter θ approximated by Monte Carlo sampling, π^s and π^g are the solutions of instance X_i constructed by sample rollout (to reduce the variance of gradients and accelerate convergence) and greedy rollout, and X_i is drawn from distribution S.

Algorithm 1 REINFORCE steps

Input: number of epochs E, steps per epoch F, number of batches B
Initialize pameters θ, $\theta^{BL} \leftarrow \theta$
for $epoch = 1, ..E$ **do**
  for $step = 1, ..F$ **do**
    X_i = RandomInstance() $\forall i \in 1, ..B$
    π_i^s = SampleRollout(X_i, p_θ) $\forall i \in 1, ..B$
    π_i^g = GreedyRollout(X_i, p_θ^{BL}) $\forall i \in 1, ..B$
    $\nabla L = \frac{1}{B} \sum_{i=1}^{B}$ (L(π_i^s) - L(π_i^g)) $\nabla_\theta \log p_\theta$ (π_i^s | X_i)
    θ = Adam(θ, ∇L)
  end for
  if $TTest(p_\theta, p_\theta^{BL})$ **then**
    $\theta^{BL} \leftarrow \theta$
  end if
end for

6. Experimentation, results and discussion

6.1 *Dataset and system configuration*

We briefly describe the problem dataset used and present the results of computational experiments. Test data-set instances include "Set-2" data-set from [7] for problems with smaller customer counts (of range 10 to 30), for higher-order customer counts of 100, we have used data-set instances from public data-set [36] and also a synthetically generated data-set instance. Further details of the data-set specifications are captured in table 2.

The MILP model, robust optimization approach for box uncertainty method 1, method 2, and ellipsoidal uncertainty were tested on a system with Intel Xeon 2.4 GHz (dual-core) and 64 GB RAM. MILP specific methods were implemented in OPL CPLEX 12.8 and with a threshold maximum time limit of two hours to execute the code. DRL experiments were tested on a 8 vCPUs, 30 GB RAM, 10 GB GPU Memory, A100 GPU system. DRL code were implemented using Python 3.8 and Pytorch 1.9 framework. GA experiments were tested on Intel i5 laptop with 32 GB RAM, and no dedicated GPU. We have used the DEAP python framework[1] in building the VRP solution based on GA using the hyper-parameters as listed in Appendix III.

6.2 *Test scenarios and solution methods*

In our work, to annotate sensitivity analyses for MILP robust models, we have considered the value of probabilities of constraint violations (PCV) to be equal to 1 (i.e., $\Gamma = 1$ [16]) and three levels (10%, 20%, and 30%) of uncertainty data of deviation from the nominal value (DNV). We have considered $\alpha = 0.5$ to allow equal weightage with objective function constituents. Also, in our work with both MILP and DRL methods, we have explicitly formulated, tested and compared the results from our approaches against few of the state-of-art Attention-Model (AM) based baseline approaches on VRPSDPTW using data samples generated with uncertainties to features such as: delivery quantity, pickup quantity, travel time and service time (i.e., $FeatureSet_i$ of customer i). We have used bootstrapped perturbation to generate data samples with a controlled level of variability or uncertainty in features represented by $FeatureSet_i$, so as to be in-line to uncertainty levels considered in our Robust-Optimization (MILP_RO) approaches. In here, bootstrapped perturbation uses mean value on each of the above-mentioned features as the basis to arrive at the feature specific perturbed value. We call this controlled perturbation-based approach as *BOOT* approach. In literature, common practice to induce data perturbation is by using a randomized distribution with lower and upper bounds to generate samples. We call these standard perturbation-based approach as *GEN* approach. In our empirical tests, based on how we generate the samples, we have considered three bootstrapped based Attention-Model (AM) *BOOT* approaches viz., vanilla *AM* (Ref. [1]), partial-tour level *DA_PSL* (Ref. [2]), and node level *DA_NL* (our approach) and the corresponding three *GEN* approaches viz., vanilla *AM*, partial-tour level *DA_PSL*, and our node level *DA_NL*. In our empirical study, we have analysed and compared MILP based approaches with Attention-Model approaches with both *BOOT* and *GEN* sample data generation techniques.

Summarized details of the methods with uncertainty levels are listed in table 3.

[1] https://pypi.org/project/deap/

Table 3. Experimentation Methods to solve VRPSPDTW variant.

Method	Description
MILP	Deterministic MILP without considering uncertainty in data
DNV10	MILP Robust optimization model with probabilities of constraints (PCV) is considered as unit value [i.e., $\Gamma = 1$] with 10% uncertainty data of deviation on features in $FeatureSet_i$ from their nominal value (DNV)
DNV20	MILP Robust optimization model with probabilities of constraints (PCV) is considered as unit value [i.e., $\Gamma = 1$] with 20% uncertainty data of deviation on features in $FeatureSet_i$ from their nominal value (DNV)
DNV30	MILP Robust optimization model with probabilities of constraints (PCV) is considered as unit value [i.e., $\Gamma = 1$] with 30% uncertainty data of deviation on features in $FeatureSet_i$ from their nominal value (DNV)
AM	Static Attention Model [1] based method with no uncertainty in training data w.r.t features in $FeatureSet_i$
AM10	Static Attention Model [1] based method with 10% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{10}{100})\}$
AM20	Static Attention Model [1] based method with 20% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{20}{100})\}$
AM30	Static Attention Model [1] based method with 30% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{30}{100})\}$
AMMax	Static Attention Model [1] based method with standard (min-max bound) uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{minimum\_value, ..maximum\_value\}$
DA_PSL	Dynamic Attention Model [2] based method with no uncertainty in training data w.r.t features in $FeatureSet_i$
DA_PSL10	Dynamic Attention Model [2] based method with 10% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{10}{100})\}$
DA_PSL20	Dynamic Attention Model [2] based method with 20% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{20}{100})\}$
DA_PSL30	Dynamic Attention Model [2] based method with 30% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{30}{100})\}$
DA_PSLMax	Dynamic Attention Model [2] based method with standard (min-max bound) uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{minimum\_value, ..maximum\_value\}$
DA_NL	Dynamic Attention Model (our approach) based method with no uncertainty in training data w.r.t features in $FeatureSet_i$
DA_NL10	Dynamic Attention Model (our approach) based method with 10% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{10}{100})\}$
DA_NL20	Dynamic Attention Model (our approach) based method with 20% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{20}{100})\}$
DA_NL30	Dynamic Attention Model (our approach) based method with 30% uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{nominal\_value, ..nominal\_value \times (1 + \frac{30}{100})\}$

Table 3 continued

Method	Description
DA_NLMax	Dynamic Attention Model (our approach) based method with standard (min-max bound) uncertainty in training data w.r.t features in $FeatureSet_i$ where $feature\_value \in \{minimum\_value, ..maximum\_value\}$
GA	Genetic Algorithmic approach without considering uncertainty in data

Table 4. MILP and box uncertainty method 1 for PCV=1.

Dtataset	MILP			DNV 10 %			DNV 20 %			DNV 30 %		
Instance	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%
2_1	2	236.69	0	2	238.78	0	3	294.87	0	3	298.61	0
2_2	2	198.09	0	2	223.42	0	2	224.19	0	3	294.09	0
2_3	1	182.01	0	2	198.09	0	2	198.09	0	2	198.09	0
2_4	2	157.91	3.82	2	146.80	0	3	191.16	19.13	3	195.80	0
2_5	3	186.97	0	3	174.91	0	3	186.45	0	3	187.43	0
2_6	1	230.46	0	1	225.52	0	1	231.66	0	1	242.52	0
2_7	2	198.45	0	1	150.55	0	1	150.55	0	1	152.47	0
2_8	4	475.73	0	5	510.94	0	5	526.94	0	6	620.03	0
2_9	3	333.93	65.35	3	346.66	64.46	4	457.09	69.02	4	447.14	52.46
2_10	3	206.50	26.79	4	251.60	35.8	4	279.66	37.13	5	304.38	28.11
2_11	3	205.27	0	4	233.19	18.18	4	246.88	16.74	4	253.64	18.1
2_12	3	684.80	18.46	7	727.73	14.61	8	814.93	7.53	9	888.79	0
2_100	31	3801.30	93.25	–	–	–	–	–	–	37	4461.49	94.23
15_15	33	3175.59	88.39	–	–	–	–	–	–	40	3977.23	88.23

– this scenario was not tested

$\#V$ Vehicle count

Table 5. Box uncertainty method 2 for PCV = 1.

Dataset	DNV 10 %			DNV 20 %			DNV 30 %		
Instance	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%
2_1	–	–	–	–	–	–	–	–	–
2_2	2	223.42	0	2	223.42	0	2	223.42	0
2_3	2	198.09	0	2	198.09	0	2	198.09	0
2_4	3	189.15	19.91	3	191.45	20.4	3	196.71	0
2_5	3	189.15	0	3	190.61	0	3	192.31	0
2_6	–	–	–	–	–	–	–	–	–
2_7	1	139.79	0	1	139.79	0	1	150.55	0
2_8	–	–	–	–	–	–	–	–	–
2_9	4	456.53	72.85	4	456.79	73.02	4	456.53	70.09
2_10	4	244.63	34.87	4	250.93	36.89	5	299.45	31.93
2_11	8	244.85	15.97	4	249.64	10.04	4	255.26	11.33
2_12	–	–	–	–	–	–	–	–	–
2_100	–	–	–	–	–	–	–	–	–

– this scenario was not tested

$\#V$ Vehicle count

Table 6. Ellipsoidal uncertainty for PCV = 1.

Dataset Instance	DNV 10 %			DNV 20 %			DNV 30 %		
	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%
2_1	2	238.78	0	3	294.87	0	3	298.61	0
2_2	2	223.42	0	2	223.53	0	2	224.19	0
2_3	2	198.09	0	2	198.09	0	2	198.09	0
2_4	3	190.90	20.21	3	193.81	11.12	3	197.07	0
2_5	3	189.15	0	3	190.40	0	3	194.03	0
2_6	1	221.56	0	1	231.66	0	1	242.52	0
2_7	1	150.55	0	1	150.55	0	1	152.47	0
2_8	5	510.94	0	5	526.94	0	6	620.03	0
2_9	4	457.84	72.7	4	456.91	55.52	4	461.08	71.94
2_10	4	249.56	34.93	5	305.30	32.90	5	304.88	28.44
2_11	4	247.60	17.71	4	250.28	0	4	260.54	0
2_12	7	727.73	8.39	8	814.34	4.82	9	888.79	0
2_100	–	–	–	–	–	–	–	–	–

– this scenario was not tested

$\#V$ Vehicle count

6.3 *Results on MILP and MILP_RO tests*

Our aim with this work is to understand handling data uncertainties and arrive at a robust solution. The use of exact solutioning approaches such as CPLEX MILP branch-and-bound or branch-and-control type methods does not scale well when the number of customers or nodes in the problem graph goes beyond a specific number. From our tests, we observe that with 100 customers, the exact approach produces a solution but with higher optimality gap.

The solution results of MILP, box uncertainty using method 1, box uncertainty using method 2 and ellipsoidal uncertainty models is summarized in table 4, table 5 and table 6 respectively. In this work, we compare the proposed MILP model solutions in table 4 with the state-of-art/existing model of MILP by [7]. We observe that the objective value results of our proposed MILP and the existing MILP model by [7] are matching for many of the test instances and slightly varying for a few instances, possibly due to the CPLEX solver's execution time threshold criteria/value used in our experimentation runs. We consider our proposed MILP model to be the baseline model against which we extend and compare the results from MILP robust optimization solutions using box uncertainty and ellipsoidal uncertainty methods. Table 4 and table 5 captures the result of our robust method run based on the box uncertainty set using method 1 and method 2. Based on the results, we observe that with the increase of uncertainty level (DNV10, DNV20, DNV30) for the box uncertainty, the objective function values (using both methods) in comparison with the baselined proposed MILP solution. However, the box uncertainty set for method 2 results is almost similar in terms of the objective function and a few instances better than the number of vehicles selected compared with the method 1 results. However, some dataset instances did not successfully complete within the given time limits due to the underlying CPLEX solver execution time criteria. Based on the results, we observe that with each increase in uncertainty level for the box uncertainty, the objective function values (using both method 1 and method 2) show an increasing trend compared to the base-lined proposed MILP solution. However, Robust optimization using box uncertainty method 2 yields almost similar results in terms of the objective function value. On some instances and some instances, Robust optimization using box uncertainty method 2 yields results that show improvement w.r.t the number of vehicles selected compared with the method 1 results. On a few dataset instances, method two did not yield results, possibly due to the CPLEX solver's execution time threshold/criteria used in the experimentation runs. Table 6 captures the result of the robust approach using an ellipsoidal uncertainty set. We are based on which we can infer that a robust optimization approach using box uncertainty performs better than ellipsoidal uncertainty. Additionally, we have studied and compared the results of our proposed MILP approach with and without additional constraints (section 5.2.1). The results of this study are captured in Appendix I. One drawback or limitation with MILP without additional constraints approach is that it does not reflect the exact in-transit schedule. Hence, we have focused on MILP with additional constraints as the baseline method in our work to compare results against Robust optimization and DRL approaches.

Table 7. Results: Solution Cost and number of vehicles required- Stochastic VRPSPDTW for various MILP and DRL approaches.

Uncertainty scenario	Approach	10 Nodes			20 Nodes			30 Nodes			100 Nodes		
		Cost	MAE	V#	Cost	MAE	V#	Cost	MAE	V#	Cost	MAE	V#
0%	MILP	205.60	0.00	1.7	263.91	0.00	2.5	365.53	0.00	3.0	3488.45	0.00	32.0
	GA	230.66	0.11	2.0	472.10	0.44	4.3	779.76	0.53	9.0	4841.71	0.28	40.0
	AM	238.09	0.14	2.3	280.57	0.06	2.7	428.50	0.15	4.7	2768.17	− 0.26	25.0
	DA_PSL	237.14	0.13	2.3	278.57	0.05	2.7	391.00	0.07	4.0	2610.20	− 0.34	23.5
	DA_NL	234.55	0.12	2.3	291.15	0.09	2.8	418.95	0.13	4.7	2586.95	− 0.35	23.0
10%	DNV10	220.10	0.00	2.0	259.23	0.00	2.5	404.18	0.00	5.0	–		
	AM10	238.17	0.08	2.3	312.92	0.17	3.2	441.63	0.08	4.7	2689.38		24.0
	DA_PSL10	239.40	0.08	2.3	298.69	0.13	3.0	441.78	0.09	5.0	2680.04		24.0
	DA_NL10	238.96	0.08	2.3	298.90	0.13	3.0	431.11	0.06	5.0	2386.58		21.5
20%	DNV20	239.05	0.00	2.3	290.65	0.00	2.8	447.16	0.00	5.3	–		
	AM20	234.51	− 0.02	2.3	309.37	0.06	3.2	426.08	− 0.05	4.7	2603.64		23.5
	DA_PSL20	238.42	0.00	2.3	303.35	0.04	3.2	465.38	0.04	5.3	2725.50		24.5
	DA_NL20	236.84	− 0.01	2.3	307.94	0.06	3.2	442.61	− 0.01	5.0	2622.83		24.0
30%	DNV30	263.60	0.00	2.7	307.57	0.00	2.8	482.28	0.00	5.3	4219.36	0.00	38.5
	AM30	243.90	− 0.08	2.3	339.55	0.09	3.5	487.80	0.01	5.3	2639.19	− 0.60	23.5
	DA_PSL30	244.83	− 0.08	2.3	325.36	0.05	3.3	441.13	− 0.09	5.0	2645.95	− 0.59	23.5
	DA_NL30	245.08	− 0.08	2.3	327.43	0.06	3.3	466.67	− 0.03	5.0	2587.53	− 0.63	23.0
Min-Max	AMMax	300.69		2.7	386.69		3.7	653.55		7.7	2471.81		22.0
	DA_PSLMax	307.51		2.7	372.06		3.7	560.64		6.0	2727.42		24.5
	DA_NLMax	260.24		2.3	385.76		3.7	682.98		7.3	2697.68		24.0

– this scenario was not tested

V# Vehicle counts

Table 8. Results: Experimentation run-timings - Stochastic VRPSPDTW for various MILP and DRL approaches.

Uncertainty scenario	Approach	10 Nodes		20 Nodes		30 Nodes		100 Nodes	
		Time		Time		Time		Time	
		Training (secs)	Testing (secs)	Training (secs)	Testing (secs)	Training (secs)	Testing (secs)	Training (secs)	Testing (secs)
0%	MILP	0	7.00	0.00	2576.67	0.00	4889.33	0.00	54000
	GA	0	187.67	0.00	204.50	0.00	223.00	0.00	308.50
	AM	715.35	0.10	1382.28	0.45	3235.85	0.08	4661.52	0.21
	DA_PSL	821.90	0.41	1577.35	0.11	3753.28	0.07	7223.84	0.55
	DA_NL	946.85	0.07	2009.56	0.18	4770.41	0.14	7396.42	0.53
10%	DNV10	0.00	5.00	0.00	2519.67	0.00	7200.00	–	
	AM10	721.06	0.04	1400.37	0.05	3327.06	0.07	4697.93	0.20
	DA_PSL10	803.85	0.04	1665.30	0.11	3858.73	0.08	7381.84	0.31
	DA_NL10	941.62	0.15	2072.51	0.09	4865.58	0.14	7248.50	0.56
20%	DNV20	0.00	12.67	0.00	2792.17	0.00	7200.00	–	
	AM20	725.19	0.04	1415.09	0.06	3207.18	0.07	4823.77	0.21
	DA_PSL20	818.22	0.15	1711.48	0.06	3898.66	0.08	7068.73	0.25
	DA_NL20	940.52	0.06	2109.06	0.19	4915.40	0.13	7006.62	0.53
30%	DNV30	0.00	27.00	0.00	3061.33	0.00	7200.00	0.00	54000
	AM30	717.73	0.22	1451.87	0.12	3243.69	0.07	4864.98	0.20
	DA_PSL30	810.88	0.05	1773.30	0.06	3903.96	0.08	6854.69	0.28
	DA_NL30	941.80	0.05	2125.41	0.12	4875.90	0.26	7033.82	0.54
Min-Max	AMMax	588.01	0.11	1332.47	0.05	2840.69	0.07	4106.09	0.20
	DA_PSLMax	694.51	0.04	1772.00	0.17	4007.99	0.08	6445.86	0.25
	DA_NLMax	804.50	0.07	2009.56	0.18	4496.21	0.15	6575.81	0.53

– this scenario was not tested

6.4 *Solution comparisor: error or gap metric*

We use Mean Absolute Error (MAE) to evaluate results from specific approaches against baselined approach, which is defined as follows:

$$MAE = \frac{X_i - \hat{X_i}}{X_i} \tag{102}$$

Where, $\hat{X_i}$ is the value from the baselined approach, X_i is the observed value using a specific approach run on i^{th} instance dataset from amongst n total dataset instances.

6.5 *Results on attention-model tests*

DRL Attention-Model (AM) based approaches consists of training phase and testing phase. For each problem dataset [table 2], in training phase, model is trained with instances that are generated on the fly. Each of our training epoch consists of 128000 instances/samples grouped into 1000 batches with batch size of 128. Please note that we have used 128000 samples with 5 training epochs for problems with small and medium node counts (10, 20, 30), and 128000 samples with 1 training epoch for problems with 100 nodes. One pertinent observation with DRL is that training the baseline model is a time-consuming process, but post which using the created trained model to solve the new real-time data input is extremely quick [27].

For each problem instance, the 2-dimension customer locations of nodes are first linearly projected to a 128-dimension vector, then processed by a 3-layer encoder attention mechanism before being fed into the decoder that shares the same hidden layer dimension with the encoder. Moreover, we have used Adam Optimizer to train the policy network with constant learning rate 10^{-4}. We have listed the hyper-parameters used in our experimentation runs in Appendix II. We find that the results obtained using 3-layer encoder setup is better than 2-layer encoder setup, in terms of cost and performance trade-off. Hence, we have used 3-layer encoder setup.

We have considered set of input datasets for each of the problems (with varied node counts i.e., 10, 20, 30 and 100 nodes) and uncertainty levels. The results capture the average solution cost, vehicle counts, and computation time grouped by problem size (node counts). Table 7 captures the results of different MILP and DRL approach runs w.r.t cost and number of vehicles required. Table 8 captures the results of different MILP and DRL approach runs w.r.t the performance (training and testing time taken).

Results on scenarios without feature data uncertainty

The key takeaways are that AM based RL approach performs close to the MILP solution w.r.t number of vehicles, and solution cost. In terms of solution cost, AM based RL approach gives a solution cost, which is 5% to 15% above MILP solution for small and medium graph node problems. But for large graph node problems, AM based RL approach outperforms MILP solution by 26% to 35%. However, RL has a significant advantage over GA in terms of solution cost. Both RL and GA approach are orders of magnitude smaller than MILP in terms of testing time (provided we consider the training time as onetime activity for each input dataset). From cost perspective, w.r.t the three AM based RL approaches, we find that *DA_PSL* and *DA_NL* perform better than AM approach.

7. Results on stochastic scenarios with feature data uncertainty

The key takeaways are that AM based RL approach performs close to the MILP_RO solutions w.r.t number of vehicles, and solution cost. In terms of solution cost, AM based RL approach gives a solution cost, which is −9% (below the MILP_RO solution) to 17% (above MILP_RO solution) for small and medium graph node problems. But for large graph node problems, AM based RL approach outperforms MILP_RO solution by 59% to 63%. The results obtained using the considered three AM based RL approaches are inconclusive. We observe that there is no clear winner amongst the three AM based RL approaches i.e., depending on the problem size (number of VRP nodes), the best RL approach varies. From cost perspective, we observe that Min-Max AM based *GEN* scenario lags *BOOT* scenarios for small and medium graph node problems. But, we find that *GEN* scenario outperforms *BOOT* scenario for large graph node problems.

A deeper look at the results is shown for RL in figure 3, where we plot the relative cost on the y-axis, and the relative number of vehicles required on the x-axis. RL requires at least as many vehicles as the baseline MILP and MILP_RO based solutions for most times. We also find good number of problem scenarios wherein RL

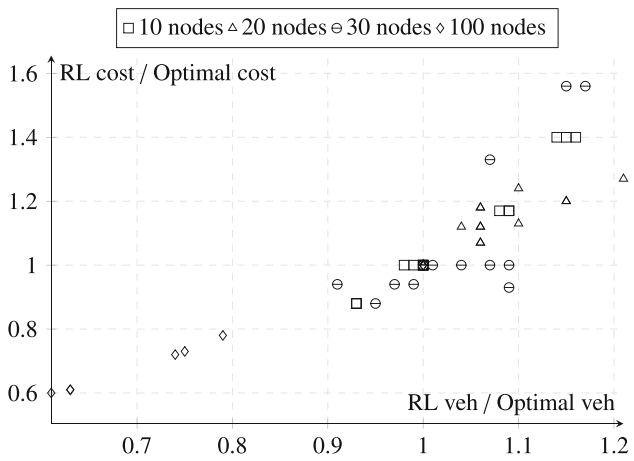

Figure 3. Scatter plot of relative costs and vehicle counts for RL in comparison with MILP baselines.

outperforms the baseline solutions w.r.t to vehicle counts and/or costs.

One common finding w.r.t AM based RL approaches is that the training times follows the the following pattern:

DA_NL training-time > *DA_PSL* training-time > *AM* training-time

This pattern is consistent across without uncertainty scenarios and Stochastic scenarios. It is primarily due to the additional dynamic encoder calls for each partial-solution tour and node level triggers being used by the dynamic encoder-decoder framework.

8. Conclusion and future work

In this work, we have proposed the following methods to solve a specific type of VRP problem viz., Capacitated Simultaneous Delivery and Pickup with Time window (VRPSDPTW): a) deterministic MILP approach, b) MILP based robust optimization (MILP_RO) approach solution using box and ellipsoidal uncertainty methodology to handle feature data with and without uncertainty, and c) we have extended and used a dynamic attention model (AM) based DRL algorithm to train and handle feature data, both in scale and uncertainty aspects.

We have used real-life public data sets in our experiments to validate our VRP approaches on data instances with uncertain feature data. We have analysed the results for solution objectives, data uncertainty or variability impacts, and solution scalability. We understand that MILP and MILP_RO based approach produces good quality solutions for smaller or medium sized datasets but gives sub-optimal solution for higher-order datasets. This is where the likes of AM based DRL or other DRL approaches help resolve and mitigate practical problem space, which typically is of a higher-order scale. Based on our experiments, we find that the AM based DRL approach yields promising and acceptable quality results for higher-order practical, real-life data sets. Also, we plan to train the RL model for better robustness inclusion by training the baseline model for a greater period (higher-order epochs), thereby accounting for more variability to the input problem space. We hope that this will produce better quality solutions. Another area that we would like to further focus is to use multi-agent DRL methods and account for the dynamic data evolution paradigm.

Appendix I. MILP experimentation run: with and without additional constraints

Table 9. Comparsion of MILP with and without additional constraints.

Dataset Instances	MILP without Additional constraints			MILP with Additional constraints			Gap (%)
	$\#V$	Cost	Gap%	$\#V$	Cost	Gap%	(MAE%)
2_1	2	236.69	0	2	236.69	0	0
2_2	2	198.09	0	2	198.09	0	0
2_3	1	167.33	0	1	182.01	0	8.06
2_4	3	184.57	19.36	2	157.91	3.82	16.88
2_5	3	186.45	0	3	186.97	0	0.28
2_6	1	216.09	0	1	230.46	0	6.24
2_7	2	139.79	0	2	198.45	0	29.6
2_8	4	462.78	0	4	475.73	0	2.72
2_9	3	325.93	67.48	3	333.93	65.35	2.4
2_10	3	202.0	23.02	3	206.50	26.79	2.18
2_11	3	205.27	0	3	205.27	0	0
2_12	6	665.76	14.15	6	684.80	18.46	2.78
2_100	–	–	–	–	–	–	–

Appendix II. DRL parameters

Parameter	Value
Samples	128000
Batches	128
Embedding dimension	128
Optimizer Learning rate	0.00001
Rollout samples	100
Gradient norm clipping	1
Feed-forward-hidden neurons	512
Number of heads in MHA	8
Num of layers	3
Optimizer	Adam
Warmup Exponential Beta	0.8
Number of warmup epochs	1
Gradient norm clipping	1
Validation batch size	100
Validation set size	10000
Tanh Clipping	10
Activation	Relu

Appendix III. GA parameters

Parameter	Value
Population size	2000
Crossover probability	0.85
Mutation probability	0.05
Number of Generations	50

Acknowledgements

We would like to Thank our colleague Rajesh Jayaprakash and anonymous reviewers for their comments, thus helping us improve this work.

References

[1] Kool W, Van Hoof H, and Welling M 2019 Attention, learn to solve routing problems. *Int. Conf. Learn. Rep.*

[2] Peng B and Wang J 2020 A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems. arXiv:2002.03282

[3] Fan J 2011 The vehicle routing problem with simultaneous pickup and delivery based on customer satisfaction. *Procedia Eng.* 15: 5284–5289,

[4] Wang H F and Chen Y Y 2012 A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Comput. Ind. Eng.* 62(1): 84–95

[5] Liu R, Xie X, Augusto V, and Rodriguez C 2013 Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *Eur. J. Oper. Res.* 230(3): 475–486

[6] Mu D, Wang C, Zhao F and Sutherland J W 2016 Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm. *Int. J. Ship. Transp. Log.* 8(1): 81–106

[7] Madankumar S and Rajendran C 2019 A mixed integer linear programming model for the vehicle routing problem with simultaneous delivery and pickup by heterogeneous vehicles, and constrained by time windows. *Sādhanā* 44(2)

[8] Hornstra R P , Silva A, Roodbergen K J, and Coelho L C 2020 The vehicle routing problem with simultaneous pickup and delivery and handling costs. *Comput. Oper. Res.* 115(104858)

[9] Olgun B, ı Koç Ç and Altıparmak F 2020 A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery. *Comput. Ind.Engi* (107010)

[10] Lu D and Gzara F 2019 The robust vehicle routing problem with time windows: Solution by branch and price and cut. *Eur. J. Oper. Res.* 275(3): 925–938

[11] Munari P, Moreno A, De La Vega J, Alem D, Gondzio J, and Morabito R 2019 The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. *Transp. Sci.*, 53(4): 1043–1066

[12] De La Vega J, Munari P and Morabito R 2019 Robust optimization for the vehicle routing problem with multiple deliverymen. *Central Eur. J. Oper. Res.* 27(4): 905–936

[13] Gounaris C E, Repoussis P P, Tarantilis C D, Wiesemann W, and Floudas C A 2016 An adaptive memory programming framework for the robust capacitated vehicle routing problem. *Transp. Sci.* 50(4): 1239–1260

[14] Nasri M, Metrane A, Hafidi I and Jamali A 2020. A robust approach for solving a vehicle routing problem with time windows with uncertain service and travel times. *Int. J. Ind. Eng. Comput.* 11(1): 1–16

[15] Subramanyam A, Repoussis P, and Gounaris C 2020 Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. *Inform. J. Computi.* 32(3): 661–681

[16] De La Vega J, Munari P, and Morabito R 2020 Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. *Comput. Oper. Res.* 124(105062)

[17] Balcik B and Yanıkoğlu İ 2020 A robust optimization approach for humanitarian needs assessment planning under travel time uncertainty. *Euro. J. Oper. Res.* 282(1): 40–57

[18] Bertsimas D and Sim M 2004 The price of robustness. *Oper. Res.* 52(1): 35–53

[19] Ben-Tal A and Nemirovski A 2000 Robust solutions of linear programming problems contaminated with uncertain data. *Math. program.* 88(3): 411–424

[20] Arulkumaran K, Deisenroth M P, Brundage M, and Bharath A A 2017 A brief survey of deep reinforcement learning. *IEEE Signal Process. Mag.* 34:26–38

[21] Nazari M, Oroojlooy A, Snyder L.V , and Takáč M 2018 Reinforcement learning for solving the vehicle routing problem. *Adv. Neural Inf. Process. Syst.* 9860–9870

[22] Ilya Sutskever, Oriol Vinyals, and Quoc.V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014

[23] Ma Y, Li J, Cao Z, Song W, Zhang L, Chen Z, and Tang J 2021 Learning to iteratively solve routing problems with dual-aspect collaborative transformer. *Adv. Neural Inf. Process. Syst.* 34

[24] Hao Lu, Xingwen Zhang, and Shuang Yang. A learning-based iterative method for solving vehicle routing problems. In: *International Conference on Learning Representations*, 2019

[25] Xin L, Song W, Cao Z, and Zhang J 2021 Neurolkh: Combining deep learning model with lin-kernighan-helsgaun heuristic for solving the traveling salesman. *Neural Inf. Process. Syst. (NeurIPS)*

[26] Li S, Yan Z, and Wu C 2021 Learning to delegate for large-scale vehicle routing. *Adv. Neural Inf. Processi. Syst.* 34

[27] Balaji B. 2019 Orl: Reinforcement learning benchmarks for online stochastic optimization problems. arXiv:1911.10641

[28] Xin L, Song W, Cao Z, and Zhang J 2021 Step-wise deep learning models for solving routing problems. *IEEE Trans. Ind. Inf.*, pp 4861–4871

[29] Kwon Y-D, Choo J, Kim B, Yoon I, Gwon Y, and Min S 2020 Pomo: Policy optimization with multiple optima for reinforcement learning. arXiv preprint arXiv:2010.16011

[30] Joe W and Lau H C 2020 Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. *Int. Conf. Autom. Plann. Scheduling (ICAPS)*, 2020

[31] Bai R 2021 Analytics and machine learning in vehicle routing research. arXiv:2102.10012

[32] Li J, Xin L, Cao Z, Lim A, Song W, and Zhang J 2021 Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. *IEEE Trans. Intelli. Transp. Syst.*

[33] Li J, Ma Y, Gao R, Cao Z, Lim A, Song W, and Zhang J 2021 Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Trans. Cybern.*

[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Aldan Gomez N, Kaiser L, and Polosukhin I 2017 Attention is all you need. *Neural Inf Process. Syst.* 5998—6008

[35] Williams R.J 1992 Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8: 229–256

[36] VRP_dataset. Cvrppdtw breedam instances 2021 http://www.bernabe.dorronsoro.es/vrp/, 2021. [Online; accessed 14-April 2021]