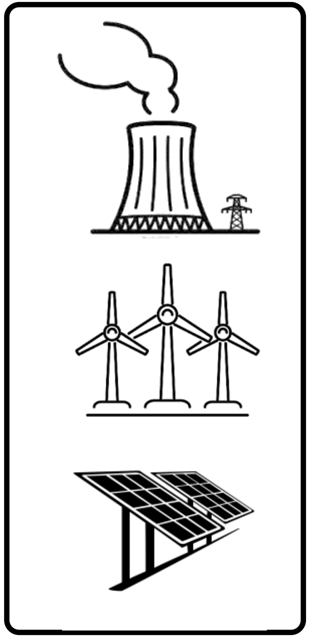
Decision Making in EV/Mobility Operations

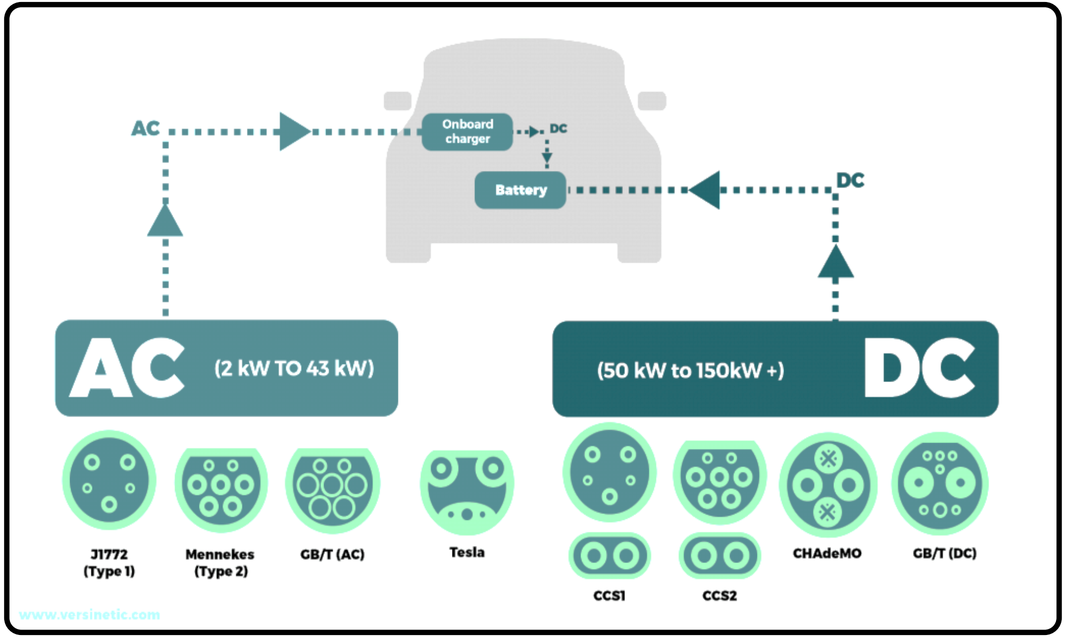
(A Tutorial on Optimization)

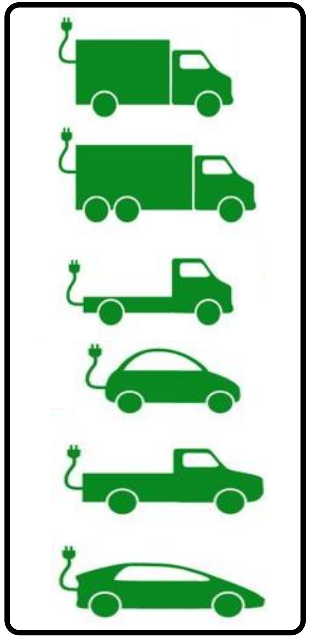
ACM India Summer School 2025: AI for Social Good
Hosted by IIT Gandhinagar

Prasant Misra
W: https://sites.google.com/site/prasantmisra/
W: https://www.linkedin.com/in/prasantmisra/

Complexity of Managing *Electric* Mobility Operations







Key Issues:

- Difference in price by source – wholesale/retail
- Differences in price based on time of day/year

Key Issues:

- Mix of charging station types
- Variable usage based on location and micro demand factors
- Mixed ownership – private, enterprise, fleet operator, public

Key Issues:

- Differing range based on vehicle type
- Differing access to charging points
- Differing ownership

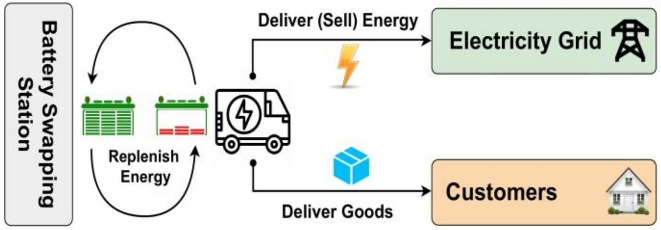
Electric Mobility Challenges for Fleet Operator

WHEN to charge? HOW much to charge? HOW fast to charge?

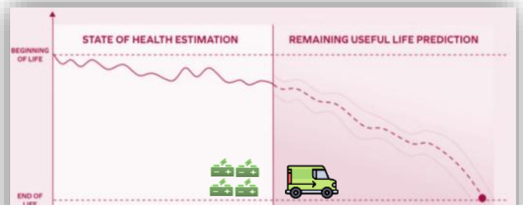
FIND available chargers when charging demand is high?

HOW to make money from (captive) chargers when not in use?

WHEN/WHERE can EVs take part in value-added services & for HOW long?



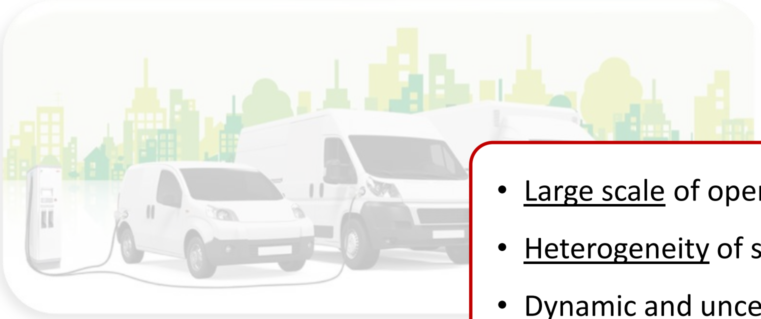
ESTIMATE the remaining useful life of the EV battery pack?



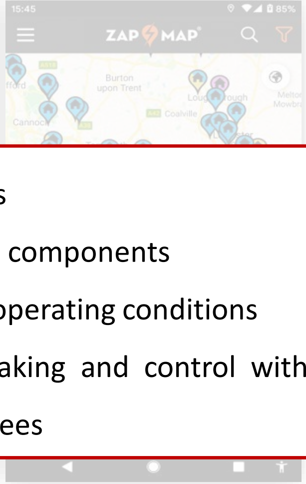
HOW to operate a mixed fleet? (ICEVs & EVs)

Why are these problems complex and hard?

WHEN to charge? HOW much to charge? HOW fast to charge?



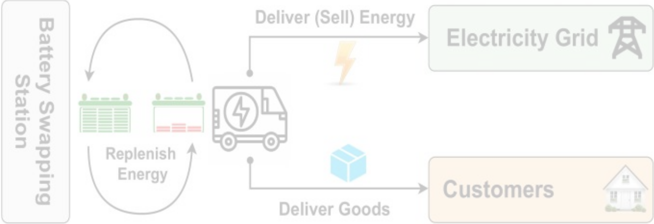
FIND available chargers when charging demand is high?



HOW to make money from (captive) chargers when not in use?

- Large scale of operations
- Heterogeneity of system components
- Dynamic and uncertain operating conditions
- Goal-driven decision making and control with time-bounded task completion guarantees

WHEN/WHERE can EVs take p… added services & for HOW…



ESTIMATE the remaining useful life of the EV battery pack?

HOW to operate a mixed fleet? (ICEVs & EVs)

Agent-based L2O Approach to Electric Vehicle Routing Problem *with* Vehicle-to-Grid Supply

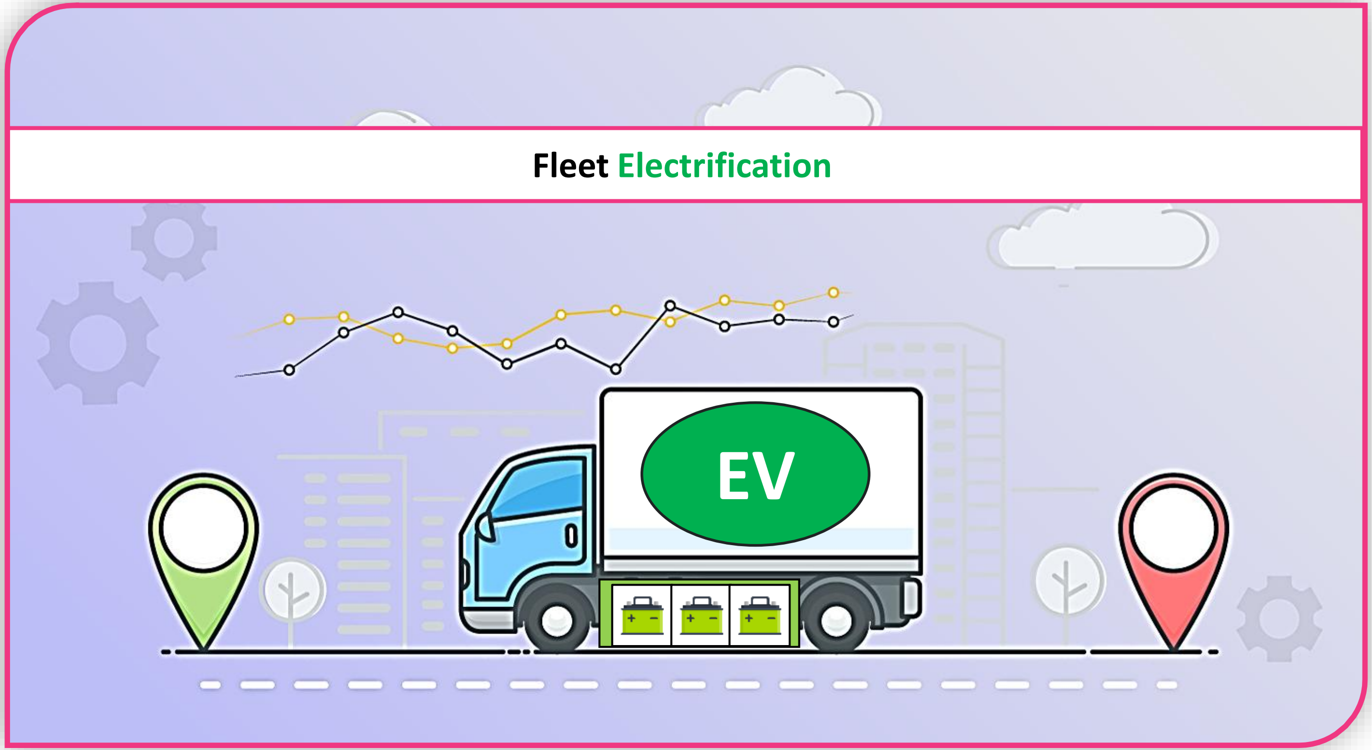
A. Narayanan, **P. Misra**, A. Ojha, V. Bandhu, S. Ghosh, and A. Vasan, "A Reinforcement Learning Approach for Electric Vehicle Routing Problem with Vehicle-to-Grid Supply" *AMAAS (ALA WKSHP)*, 2022.

Business Problem

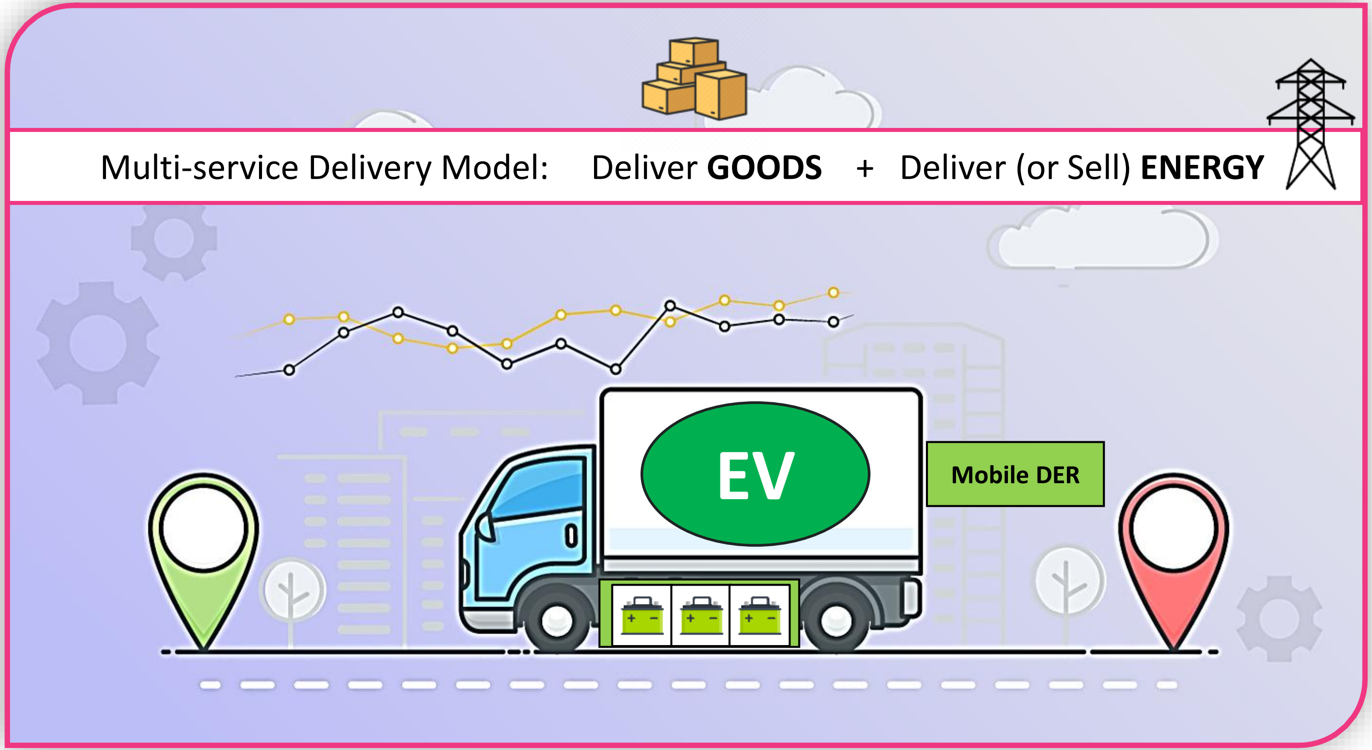
Last-mile delivery is the **MOST** expensive **(> 50% of the overall shipping cost)** part of the logistic and e-commerce process!

Business Objective: Reduce Operations Cost

(Emerging) Business Solution

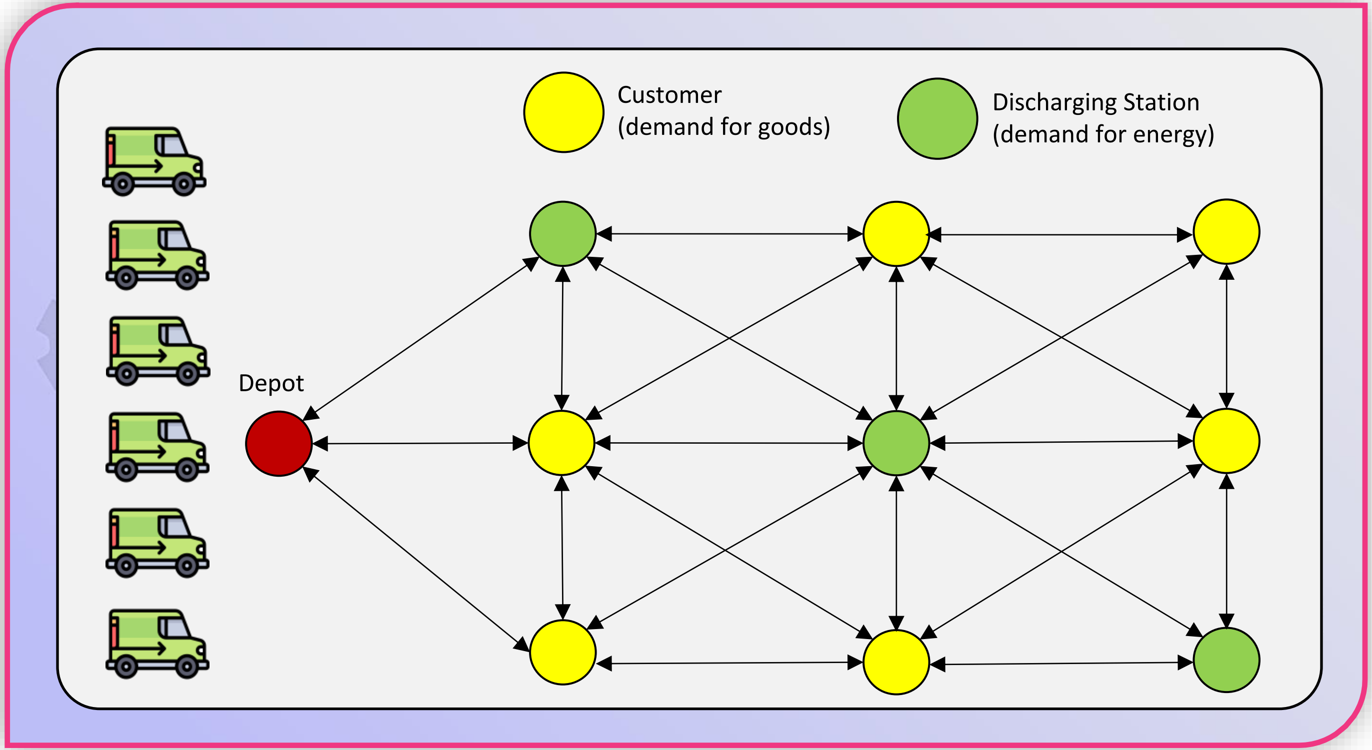


(New Proposal) Business Solution

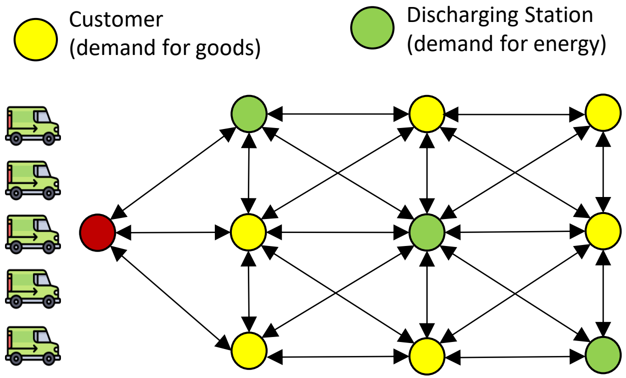


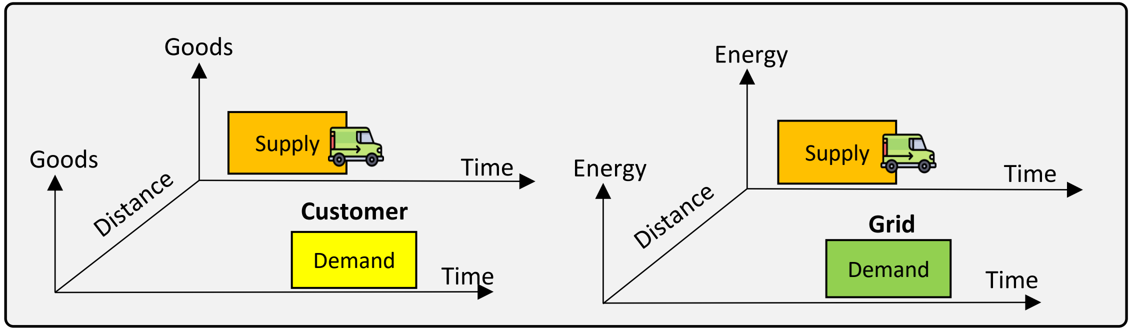
Research Problem

Design a **scalable EV routing algorithm** that reduces the fleet-level trip cost with multi-service delivery



Solution Approach





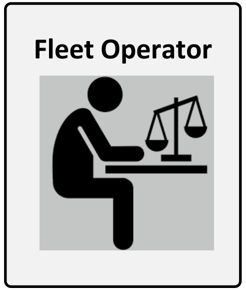
Challenges

- Supply needs to match the Demand both in space and time
- VRP becomes even harder with EV related constraints
- Difficult to scale to large problem instances using existing techniques

Approach

Agent based L2O that learns routing policies

Problem Description



Given

- Customer order fulfillment list (from fulfillment center)
- Peak energy demand periods (from grid)
- Starting state-of-charge (SOC) of all EVs in the fleet (Q)

Constraint

- (Mandatory) EVs must complete all customer deliveries
- (Optional) EVs can sell power to the grid, where possible
- The entire trip (depot-depot) must be managed within Q; without recharging

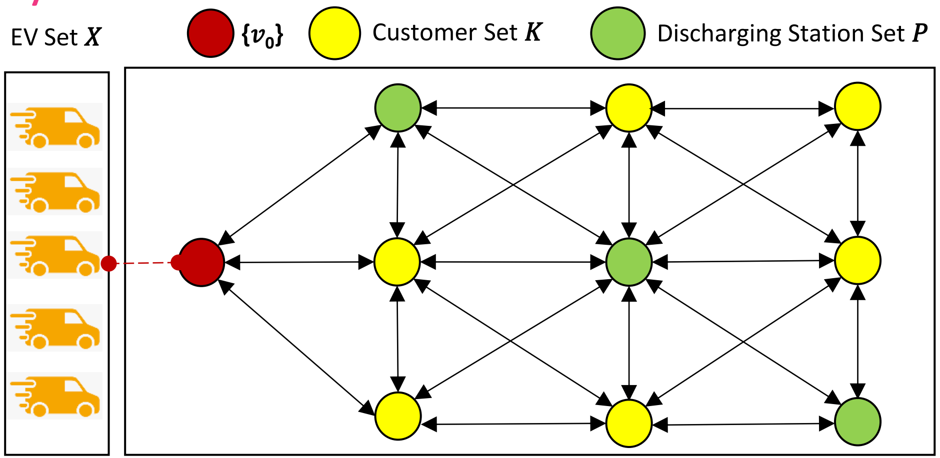
Assumption

- All EVs are charged to Q at the depot, before trip commencement

Obtain

- Routing plan optimizing cost of fleet operations

System Model



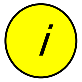
Complete Undirected Graph $G = (V, E)$

- Set of nodes $V = \{v_0\} \cup K \cup P$
- Set of edges $E = \{ (i, j) : i, j \in V, i \neq j \}$

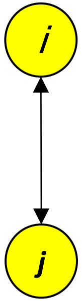
- $|X| = u$
- $|K| = m$
- $|P| = n$

Decision Variables	
α_{ij}	indicates, if edge ij is traversed by an EV (binary)
γ_i	service time at discharging station node i
τ_i	time-of-arrival at node i
Θ_i	remaining battery capacity on arrival at node i
λ_i	remaining cargo on arrival at node i

Parameters	
C	carrying capacity of each EV
Q	starting state-of-charge (SoC) of each EV



c_i	demand (of goods) at node i
s_i	service time at node i
e_i	earliest start of service at node i
l_i	latest start of service at node i



d_{ij}	distance between nodes i and j
t_{ij}	travel time between nodes i and j
H	charge consumption rate (kWh/km)
b_{ij}	energy consumed in travelling between i and j (= H . d_{ij})

R	discharging rate of each discharging station
G_i^1	start time – grid demand at node i
G_i^2	end time – grid demand at node i

Optimization Problem

Objective: Minimize the Trip Cost of the EV Fleet

$$\mathrm{M} = min\ (\ Y_1.\boxed{\sum_{u \in X} d_{ij}.\alpha_{ij}^{u}} + Y_2.\boxed{\sum_{u \in X}\sum_{i \in V} \alpha_{0i}^{u}} - Y_3.\boxed{\sum_{u \in X}\sum_{i \in P} \gamma_i^u * \sum_{j \in V} \alpha_{ij}^{u}}\ )$$

Total distance travelled by EVs

Total EVs used in a trip

Total time spend by EVs at discharging stations

Constraints

[C1]: Ensure every customer is visited exactly once, while making it **optional** to visit any of the discharging stations

$$\boxed{\sum_{u \in X}\sum_{j \in V, i \neq j} \alpha_{ij}^{u} = 1 \qquad \forall i\ \in K}$$

Optimization Problem

[C2]: Ensure flow conservation (at each node: # incoming edges = # outgoing edges)

$$\sum_{j \in V, i \neq j} \alpha_{ij}^{u} = \sum_{k \in V, i \neq j} \alpha_{ji}^{u} \qquad \forall j \in V, \forall u \in X$$

[C3]: Ensure time feasibility of arcs leaving customers and the depot

$$\tau_i^u + (t_{ij} + s_i) . \alpha_{ij}^u - l_0 . \left(1 - \alpha_{ij}\right) \leq \tau_j^u \qquad \forall i \in K, \forall i \in V, \forall u \in X$$

[C4]: Ensure time feasibility of arcs leaving discharging stations and the depot

$$\tau_i^u + (t_{ij} + \gamma_i^u) . \alpha_{ij}^u - l_0 . \left(1 - \alpha_{ij}\right) \leq \tau_j^u \qquad \forall i \in P, \forall i \in V, \forall u \in X$$

[C5]: Ensure that each customer node is visited within its time window

$$e_i . \sum_{j \in V} \alpha_{ij}^u \leq \tau_i^u \leq l_i . \sum_{j \in V} \alpha_{ij}^u \qquad \forall i \in K, \forall u \in X$$

Optimization Problem

[C6]: Ensure that discharge service time aligns with the grid peak demand period, if discharging stations are visited

$$g_i^1 . \sum_{j \in V} \alpha_{ij}^u \leq \tau_i^u \leq g_i^2 . \sum_{j \in V} \alpha_{ij}^u \qquad \forall i \in P, \forall u \in X$$

[C7]: Ensure remaining charge (energy) feasibility for arcs leaving customers and the depot

$$0 \leq \theta_j^u \leq \theta_i^u - (H . d_{ij}) . \alpha_{ij}^u + Q \, (1 - \alpha_{ij}^u) \qquad \forall i \in K, \forall j \in V, \forall u \in X, i \neq j$$

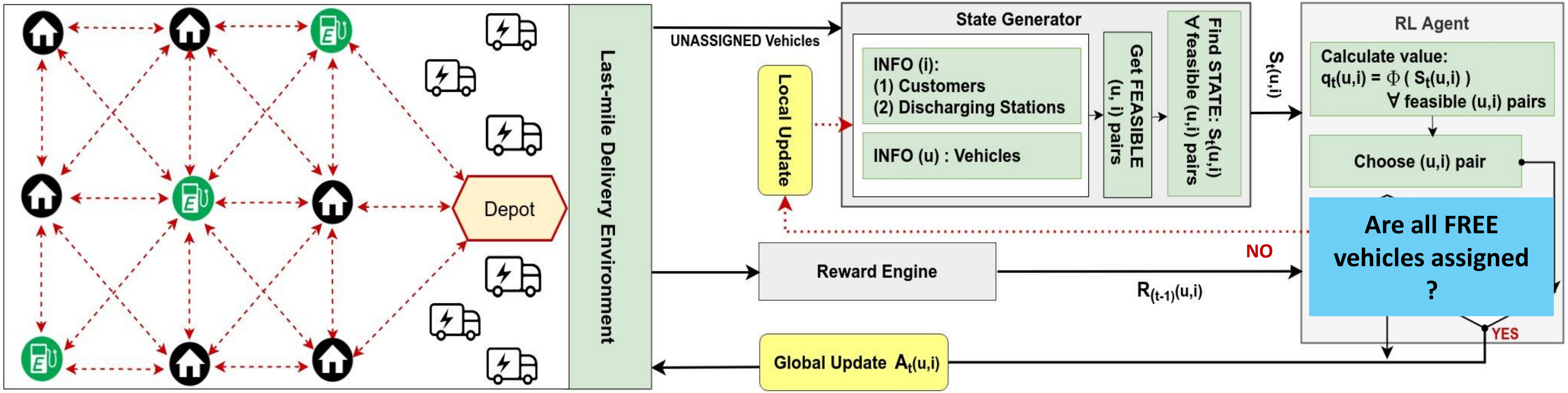
[C8]: Ensure remaining charge (energy) feasibility for arcs leaving discharging stations and the depot

$$0 \leq \theta_j^u \leq \theta_i^u - (H . d_{ij} + R . \gamma_i^u) . \alpha_{ij}^u + Q \, (1 - \alpha_{ij}^u) \qquad \forall i \in P, \forall j \in V, \forall u \in X, i \neq j$$

[C9]: Ensure all customer demands are fulfilled

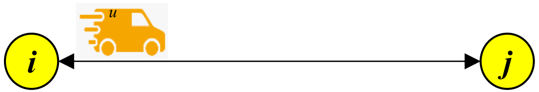
$$0 \leq \lambda_j^u \leq \lambda_i^u - c_i . \alpha_{ij} - C . (1 - \alpha_{ij}^u) \qquad \forall i, j \in V, \forall u \in X$$

L2O Agent Representation



System decision := find routes that minimize the trip cost of the fleet, subject to constraints
Routing decision := which vehicle u should be assigned to which node i ?

Masking scheme for finding FEASIBLE (Vehicle -> Node) pairs



- The earliest arrival time at node j violates the time window constraint **[C5, C6]**
- Node j is a customer; and the current SoC of the vehicle cannot support the complete trip from node i to node j and back to the depot **[C7]**
- Node j is a discharging station; and the current SoC of the vehicle cannot support the complete trip from node i to node j and back to the depot; as well as the discharge operation at node j **[C8]**
- Node j is a customer with unfulfilled demand that is either nil or exceeds the remaining carrying load of vehicle u **[C9]**

L2O Agent Representation

Routing decision := from all feasible (u, i) pairs, which pair is the best choice?

State $S_t(u, i)$

State Variables		Normalizing Factor	
b_{ij}	energy consumed in travelling between nodes i and j (proxy for distance travelled between nodes i and j)	E	energy required to travel the diagonal length of the graph
z_i	energy spent at node i (0, if at customer; else z_i)	E	
I_{depo}	flag: indicates if vehicle u *is* starting from the depot	-	
I_{cust}	flag: indicates if node i is a customer	-	
w_u^i	wait time of vehicle u at node i before it can start service	T	Decision time horizon

Action $A_t(u, i)$

- Assign *vehicle* u $\rightarrow$ *node* i
- If *vehicle* u is BUSY; perform local update
  (i) remove assigned *node* i from the service list;
  (ii) update distance and time
- If the assignment of all FREE vehicles is done, perform global update and get reward

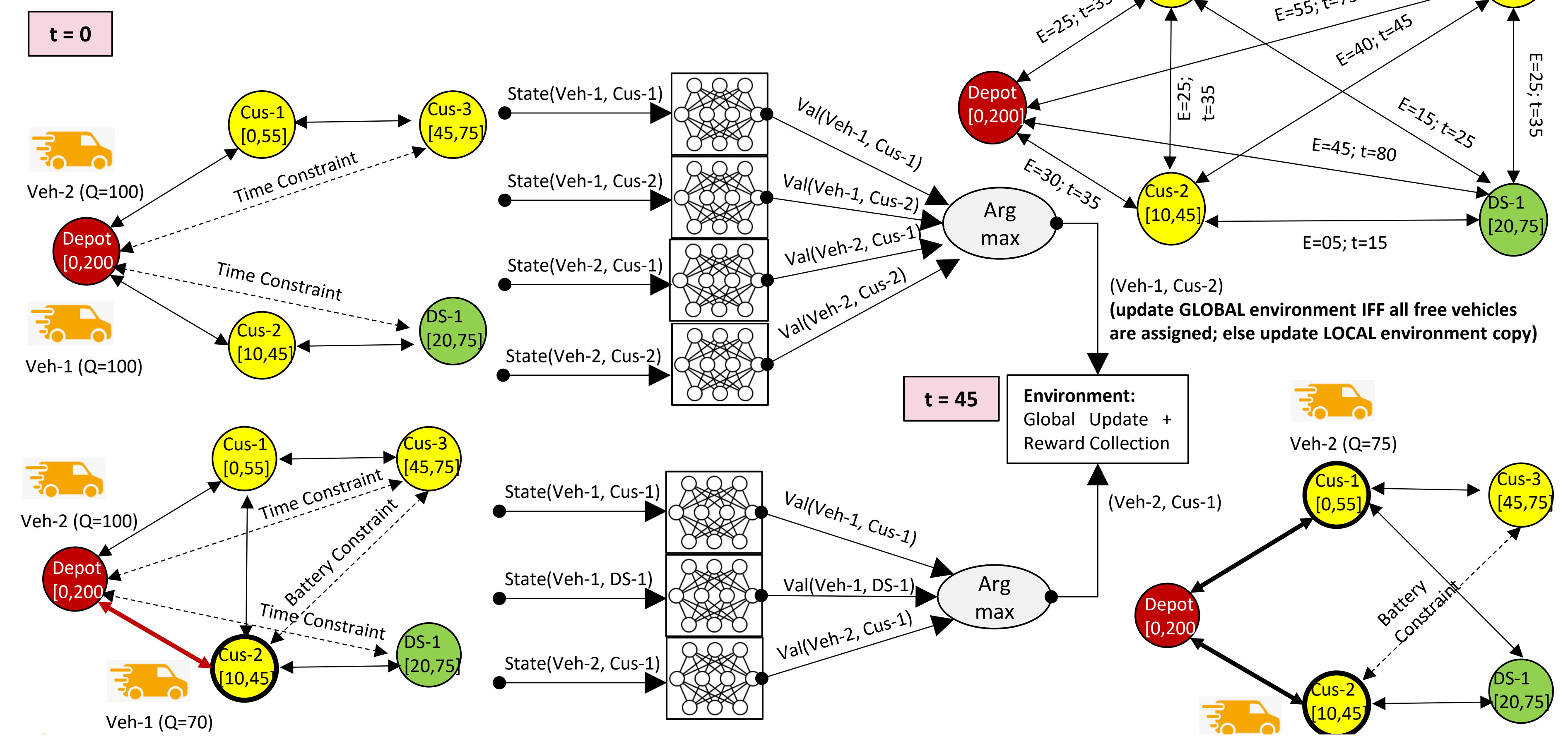
Reward $R_t(u, i)$

$-(A_1 * b_{ij})$	$A_1 = 0.15$	-VE reward for choosing longer route segments
$+(A_2 * z_i)$	$A_2 = 0.001$	+VE reward for visiting discharging station node
$+(A_3 * I_{cust})$	$A_3 = 0.15$	+VE reward for visiting customer node
$-(A_4 * w_u^i)$	$A_1 = 0.15$	-VE reward for assignments that lead to waiting time
$-(A_5 * I_{depo})$	$A_1 = 0.55$	-VE reward for sending new vehicles from the depot

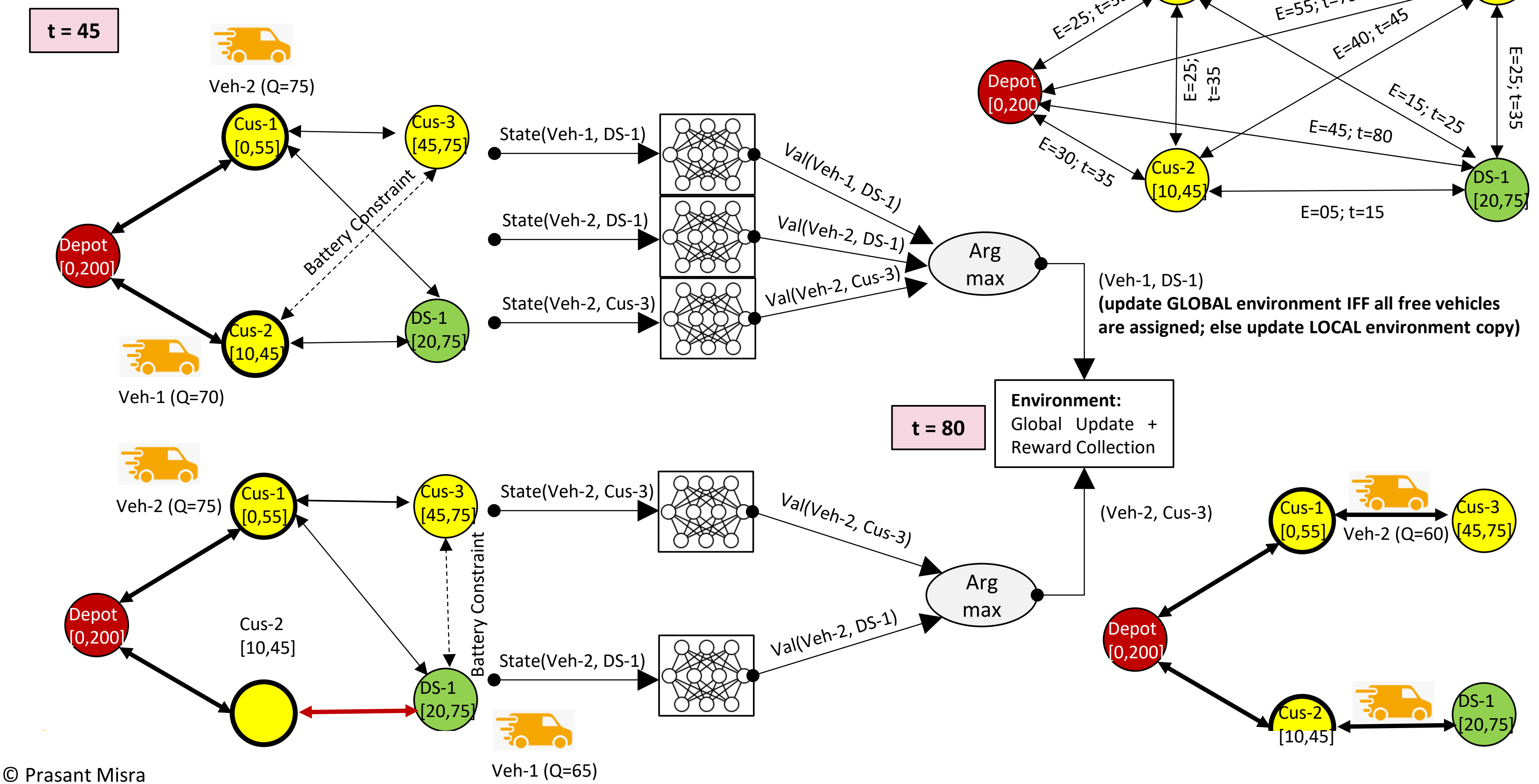
L2O Agent Training Algorithm

1) Initialize the neural network with weights ϕ
2) Initialize batch size β, replay buffer B
3) :**FOR** <*Episode*> = 1 :**TO** <*Total-Num-Episodes*> :**DO**
   i. Randomly choose data instance from training set
   ii. Initialize environment
   iii. :**WHILE** *t* < *T* :**DO**
      a. Create a copy of the environment for local updates
      b. :**WHILE** <*free-vehicle*> is *unassigned* :**DO**
         i. Find feasible (vehicle u, node i) pairs $\forall\, u$ (whether free or busy)
            a. :**IF** no feasible pairs found, then **BREAK**
         ii. Calculate $q_t\,(u, i) = \phi(S_t(u, i))$ $\forall$ feasible (vehicle u, node i) pairs, regardless of the current state of each vehicle u (busy or free)
         iii. Choose (u, i) pair with highest q_t value (using ϵ-greedy assignment)
         iv. Perform local update on the environment copy
      c. Execute new (u, i) assignments in the global environment and get reward $R_t(u, i)$
      d. Add $[S_t(u, i); R_t(u, i); q_t(u, i)]$ to replay buffer
   iv. Delete oldest entries in B if size exceeds buffer capacity
   v. Draw β samples from B
   vi. Update ϕ by minimizing MSE loss between $q_t(u, i)$ and $R_t\,(u, i)$

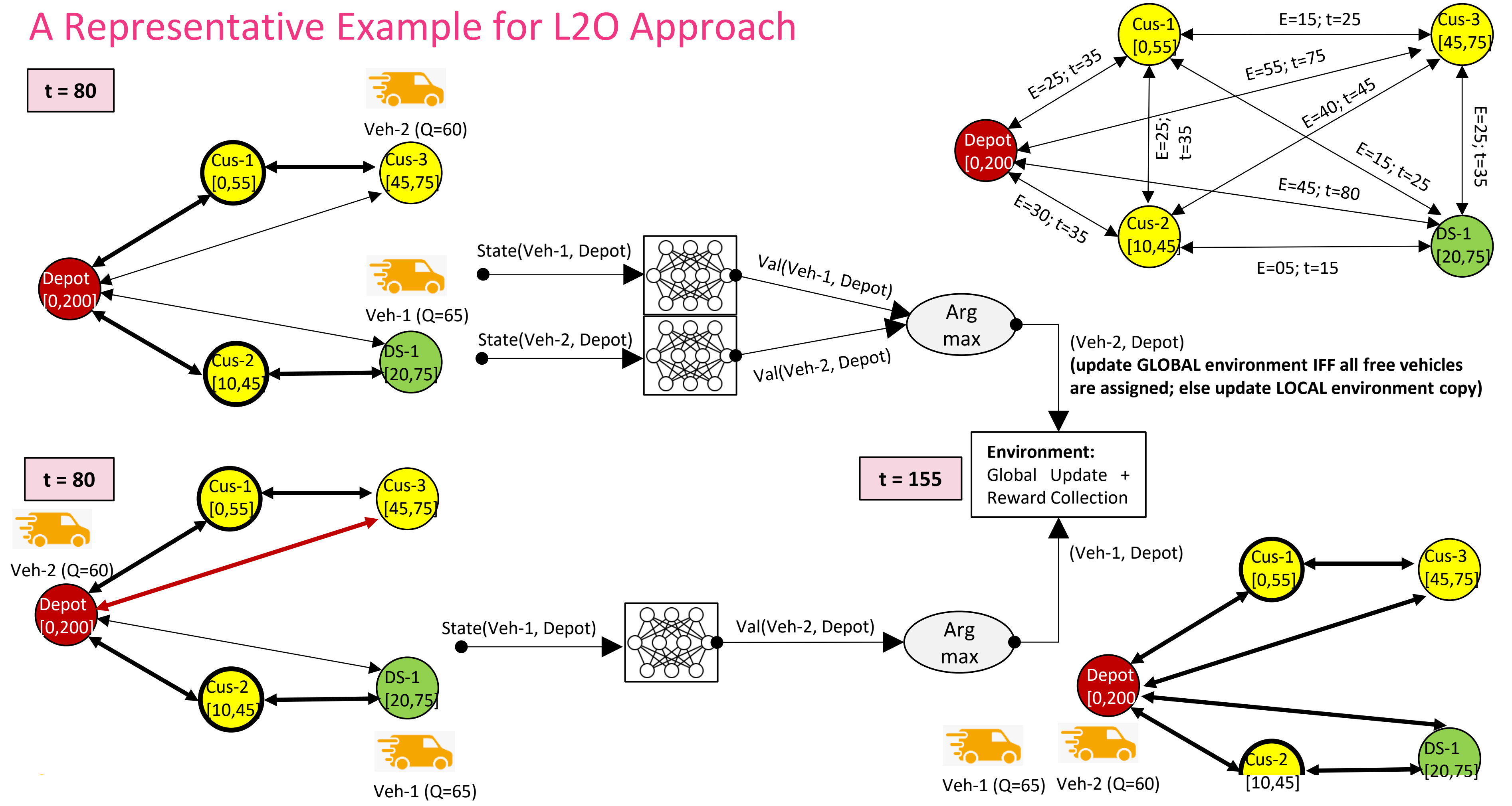
A Representative Example for L2O Approach



A Representative Example for L2O Approach



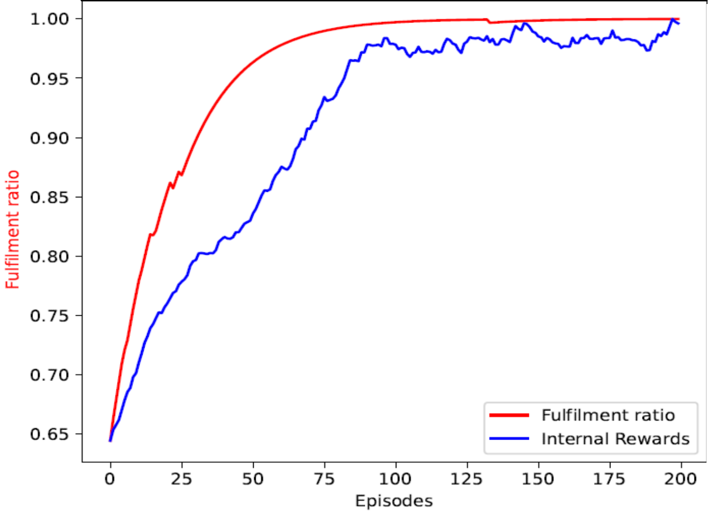
A Representative Example for L2O Approach



Evaluation

- Implementation framework: PyTorch
- Training: 200 episodes (random combination of customers; discharge stations, vehicles)
- Testing: Solomon Dataset [6-10 instances per dataset]
  - Clustered (CL)
  - Random (RA)
  - Random Clustered (RC)

RL Training



Neural Network Architecture				
Layer #	**Description**	**Neurons**	**Justification**	**Activation Func**
1	Input Layer	5	5 state variables	-
2	FC	12		ReLU
3	FC	6		ReLU
4	FC	3		ReLU
4	Output Layer	1		-

Hyperparameters			
Parameter	**Value**	**Parameter**	**Value**
Optimizer	Adam	Exploration factor	Linear decay from 1 to 0 over 75 episodes
Batch Size (β)	16	Exploration policy	ε-greedy
Replay Buffer Size (B)	5000		
Learning Rate	0.001	Training Epochs per Episode	Min -> # customers

Evaluation

Table 4: Performance comparison on specific instances of Solomon datasets: MILP vs. GA vs. RL

DataSet	MILP_d	GA_d	RL_d	MILP_v	GA_v	RL_v	MILP_ed	GA_ed	RL_ed	MILP_t	GA_t	RL_t	MILP_cost	GA_cost	RL_cost
CL101	214.56	214.71	250.22	3	3	3	180	180	90	7068	68.10	3.46	268.42	268.43	291.99
CL201	218.60	219.58	290.56	2	2	3	90	90	270	3029	64.81	3.52	189.06	189.10	248.81
RA105	620.44	556.81	632.62	5	5	5	40	30	20	1078	69.58	3.53	521.10	521.33	526.49
RA109	504.19	460.52	634.91	4	4	5	40	30	30	31417	68.41	3.48	415.18	416.11	524.09
RC101	478.56	462.15	488.604	4	4	4	40	30	20	2608	65.39	3.56	414.27	416.17	419.58
RC106	346.23	346.50	367.23	3	3	3	30	30	20	7308	45.01	3.42	310.25	310.26	313.47
RC102	352.65	352.74	368.17	3	3	3	30	30	20	89857	43.46	3.44	310.48	310.48	313.50
RC105	465.09	412.37	489.84	4	4	4	40	30	20	89835	88.47	3.53	413.80	414.40	419.62

Table shows specific instances where the MILP formulation converges within a reasonable amount of time

GA accuracy is as a good as MILP

Table 3: Performance comparison on Solomon datasets: GA vs. RL | #C: num. customers | #S: num. discharging stations | *_d: distance travelled | *_v: num. vehicles used | *_ed: energy discharged | *_t: compute time (seconds) | *_cost: $\mathcal{M}$ (Eq.(1))

Dataset	#C	#S	GA_d	RL_d	GA_v	RL_v	GA_ed	RL_ed	GA_t	RL_t	GA_cost	RL_cost	GA > RL cost (%)	$\frac{\text{GA\_t}}{\text{RL\_t}}$
CL1_25	22	3	219.47	259.54	2.56	3.22	160.00	160.00	53.55	3.43	228.30	284.90	19.87	15.61
CL2_25	22	3	209.58	281.73	1.75	3.00	67.50	168.75	68.62	3.50	168.86	230.5	26.74	19.6
RA1_25	22	3	453.84	707.70	4.25	6.16	23.33	65.83	73.23	3.46	442.98	498.76	11.18	21.16
RA2_25	22	3	368.8[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	222.10	299.10	25.74	22.52
RC1_25	22	3	351.10	562.01	3.25	4.75	30.00	73.75	56.06	3.45	335.88	366.47	8.35	16.25
RC2_25	22	3	322.09	595.21	2.13	3.50	23.75	136.25	82.61	3.50	221.86	250.69	11.50	23.6
CL1_50	45	5	431.08	607.83	5.00	6.22	360.00	510.00	180.38	8.66	435.10	496.46	12.36	20.83
CL2_50	45	5	337.15	427.51	2.00	3.38	0.00	123.75	180.83	8.82	215.56	339.57	36.52	20.5
RA1_50	45	5	797.9[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	755.22	919.95	17.91	35.10
RA2_50	45	5	628.19	1085.32	5.75	6.27	46.36	205.46	408.88	8.91	390.22	533.4	26.84	45.89
RC1_50	45	5	718.44	1066.00	6.25	8.62	47.50	92.50	212.31	8.44	649.98	815.06	20.25	25.16
RC2_50	45	5	602.01	1124.18	3.88	6.00	46.25	136.25	263.72	7.55	404.37	508.25	20.44	34.93
CL1_100	90	10	773.39	1089.53	9.00	11.22	160.00	270.00	307.42	31.45	904.02	1068	15.35	9.78
CL2_100	90	10	545.19	751.99	3.00	4.50	0.00	33.75	269.99	34.37	324.73	461.23	29.60	7.86
RA1_100	90	10	1253.[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1302.78	1564.10	16.71	25.5
RA2_100	90	10	925.22	1403.01	5.46	7.72	79.09	175.45	1466.74	33.74	568.48	715.07	20.50	42.22
RC1_100	90	10	1438.64	2019.25	12.88	15.62	93.75	258.75	806.64	30.98	1338.50	1526.46	12.31	26.03
RC2_100	90	10	1060.76	1636.08	6.63	9.13	92.50	233.75	1025.32	34.61	689.12	927.33	25.69	29.63

Mean Optimality Gap between GA and RL: 17.23% (std. dev. 7.98%)

Mean Optimality Gap between GA and RL: 22.38% (std. dev. 8.35%)

Mean Optimality Gap between GA and RL: 20.02% (std. dev. 6.58%)

1. **Accuracy:** GA is better than RL by an average of 19.8% (range 8.3% - 36.52%)
2. **Execution Time:** RL is faster than GA by an average of 24 times

Agent-based L2O Approach to Electric Vehicle Routing Problem *with* Vehicle-to-Grid Supply and Battery Swapping

Ajay Narayanan, Prasant Misra, Ankush Ojha, Abhinav Gupta, Supratim Ghosh, and Arunchandar Vasan. 2023. Agent-based Learning Approach to Electric Vehicle Routing Problem with Vehicle-to-Grid Supply and Battery Swapping. In Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD) (CODS-COMAD '23). Association for Computing Machinery, New York, NY, USA, 185–193

Table 4: Performance comparison on Bangalore city case-study
#m: #customers | #n: #discharging stations | #o: #battery swapping stations | ∗_t: compute time (s) | ∗_c: cost $\mathcal{M}$ (Eq.(1))

Dataset	#m	#n	#o	H1_t	H2_t	GA_t	Qk_t	LA_t	H1_c	H2_c	GA_c	Qk_c	LA_c	LA > GA cost (%)	$\frac{GA\_t}{LA\_t}$
BLR_25	25	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	29.87	28.65
BLR_50	50	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	11.48	21.29
BLR_100	100	10	5	3.63	3.25	1036.45	63.46	66.46	7400.48	8349.70	4690.47	6165.95	5933.53	20.95	15.60

Mean Optimality Gap between GA and RL: 18.79% (std. dev. 4.58%)

Final Remarks

Summary

- We model the electric vehicle routing problem with constraints on loading capacity; time window; vehicle-to-grid energy supply (CEVRPTW-D) ; and formulate the multi-objective optimization problem to minimize the trip cost of the fleet.
- We design a value-based L2O algorithm by defining the (state, action) space, and engineer the reward signal for the agent to find the cost-effective delivery routes.
- We design and implement a genetic algorithm (GA) metaheuristic to derive optimal results for CEVRPTW-D.
- Using Solomon datasets, we evaluate and compare the computation speed and solution accuracy of the proposed model against GA and MILP.

Key Finding

Agent based L2O is 24 times faster than the GA and MILP baselines in terms of solutioning speed, but with ≈ 20% decrease in solution quality