Learning to Handle Constraints in Routing Problems via a Construct-and-Refine Framework

Anonymous Author(s)
Affiliation
Address
email

Abstract

VRPs with complex real-world constraints remain challenging for both traditional OR solvers and neural methods. Despite leveraging GPU acceleration and reduced domain knowledge, neural solvers have largely been limited to simplified VRP variants, but struggle with complex constrained ones. We first rethink the existing popular single-paradigm neural solvers and identify paradigm-inherent limitations: construction solvers suffer from inflexible stepwise feasibility, and improvement solvers easily get stuck in infeasible searches with long runtimes. However, these paradigms are naturally complementary: construction efficiently provides strong initial solutions that help improvement rapidly reach feasible, high-quality solutions. Motivated by this, we propose Construct-and-Refine (CaR), the first generic neural framework for efficient constraint handling, compatible with existing construction and improvement solvers. To promote synergistic paradigm integration, we introduce a joint training framework with bespoke losses to generate diverse, high-quality, (near)-feasible solutions that are refined by a light improvement process (e.g., only 10 steps down from 5k). We also present the first study of a shared encoder for cross-paradigm representation learning in handling complex constraints. We evaluate CaR on two challenging VRP variants, TSP with Time Windows (TSPTW), where feasibility masking is intractable, and CVRP with Backhaul, Time Windows, and Duration Limit (CVRPBLTW), where masking is tractable but overly restrictive. Our experiments, as shown in Table 1, demonstrate that CaR achieves superior feasibility, solution quality, and efficiency compared to both traditional and neural state-of-the-art solvers.

Table 1: Results on constrained VRPs. Best are **bolded**. L2C: Learning to Construct; L2I: Learning to Improve.

	Method	#Params	Paradigm §	n=50 Obj. ↓	n=50 Gap ↓	n=50 Infsb% ↓	n=50 Time	n=100 Obj. ↓	n=100 Gap ↓	n=100 Infsb% ↓	n=100 Time
TSPTW	LKH3	/	I	25.611	◇	0.12%	7h	46.858	◇	0.13%	1.4d
	OR-Tools†	/	I	25.763	-0.001%	65.72%	2.4h	46.424	0.026%	97.45%	12m
	Greedy-C	/	C	26.394	1.534%	72.55%	4.5s	51.945	9.651%	99.85%	11.4s
	POMO	1.25M	L2C	/	/	100.00%	4s	/	/	100.00%	14s
	POMO*	1.25M	L2C	26.222	1.635%	37.27%	4s	47.249	1.959%	38.22%	14s
	POMO* + PIP (greedy)	1.25M	L2C	25.657	0.177%	2.67%	7s	47.372	1.223%	6.96%	32s
	NeuOpt-GIRE *‡ ($T = 5$k)	0.69M	L2I	25.617	0.028%	0.02%	11.6m	46.913	**0.123%**	**0.02%**	30m
	CaR-POMO ($T_{\mathrm{R}} = 20$)	1.64M	L2(C+I)	25.614	0.014%	0.01%	51s	47.001	0.406%	2.34%	2.1m
	CaR-PIP ($T_{\mathrm{R}} = 20$)	1.64M	L2(C+I)	25.612	**0.005%**	**0.00%**	52s	46.923	0.146%	**0.02%**	2.4m
CVRPBLTW	OR-Tools	/	I	14.677	◇	0.00%	1.7h	25.342	◇	0.00%	3.5h
	POMO	1.25M	L2C	15.999	9.169%	0.00%	2s	27.046	7.004%	0.00%	4s
	POMO+EAS+SGBS*	1.25M	L2C	15.156	3.263%	0.00%	10.3m	25.558	0.854%	0.00%	1h
	NeuOpt-GIRE *‡ ($T = 5$k)	0.69M	L2I	14.201	**-1.163%**	27.30%	5.5m	24.237	-0.533%	41.20%	15m
	POMO*	1.25M	L2C	14.873	2.310%	0.00%	2s	24.592	-1.645%	0.00%	4s
	CaR (k-opt) ($T_{\mathrm{R}} = 20$)	1.64M	L2(C+I)	14.844	2.114%	0.00%	8s	24.585	-1.724%	0.00%	17s
	CaR (R&R) ($T_{\mathrm{R}} = 20$)	1.72M	L2(C+I)	14.601	0.463%	**0.00%**	10s	24.400	**-2.448%**	**0.00%**	19s

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.