Letting a Large Neighborhood Search for an Electric Dial-A-Ride Problem Fly: On-The-Fly Charging Station Insertion

Maria Bresich[1] Günther R. Raidl[1] Steffen Limmer[2]

[1]Institute of Logic and Computation, TU Wien, Austria
{mbresich,raidl}@ac.tuwien.ac.at
[2]Honda Research Institute Europe, Germany
steffen.limmer@honda-ri.de

GECCO 2024, Melbourne, Australia
July 14 – 18, 2024

TU WIEN Informatics ac ALGORITHMS AND COMPLEXITY GROUP

HRI Honda Research Institute EU

Dial-A-Ride Services

- On-demand, public transit services
  + Cost efficient, environment-friendlier, customizable service
  – Ride-sharing, detours

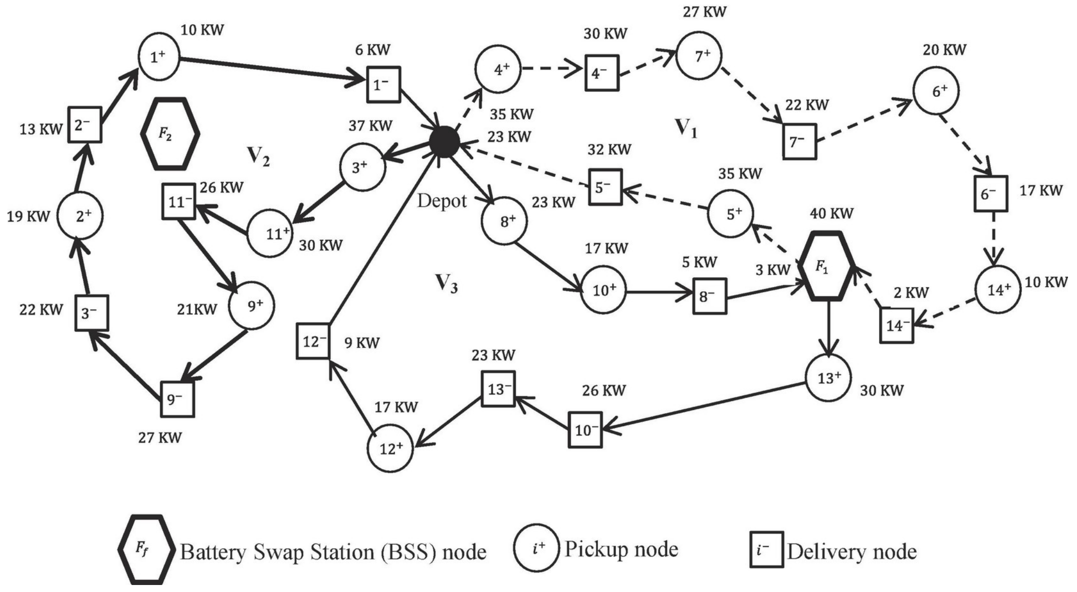
- Dial-A-Ride Problem (DARP)
  - Planning and scheduling of DAR services
  - Combinatorial optimization problem
  - NP-hard

Electric Autonomous DARP (E-ADARP)

Bongiovanni et al. (2019)

Definition

- Given:
  - *n* transportation requests
  - *m* electric autonomous vehicles
- Task:
  - Design *m* vehicle routes serving all requests
  - Minimize total travel and excess ride time
  - Satisfy constraints



Example from Masmoudi et al. (2018).

E-ADARP Constraints I

- Start and end at a depot
- Pickup and drop-off of request in same route
- Respect time windows and planning horizon
- User ride time ≤ maximum ride time
  - → Consideration of user inconvenience

E-ADARP Constraints II

Battery management:

- Minimal battery levels at destination depots
- Only unoccupied vehicles can charge
- $\leq$ 1 visit per charging station (CS)

Simplifying assumption:

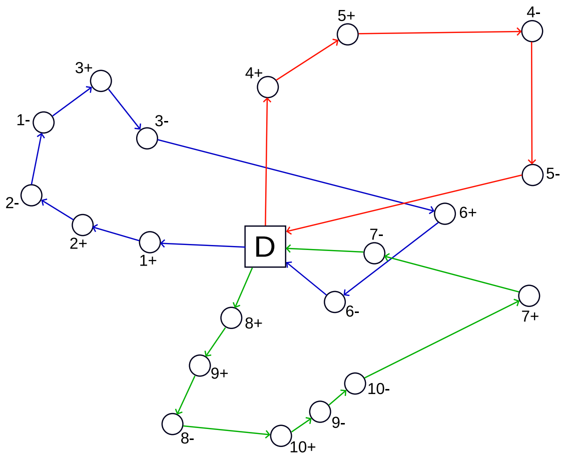
- Charging: linear increase of state-of-charge

Related Work

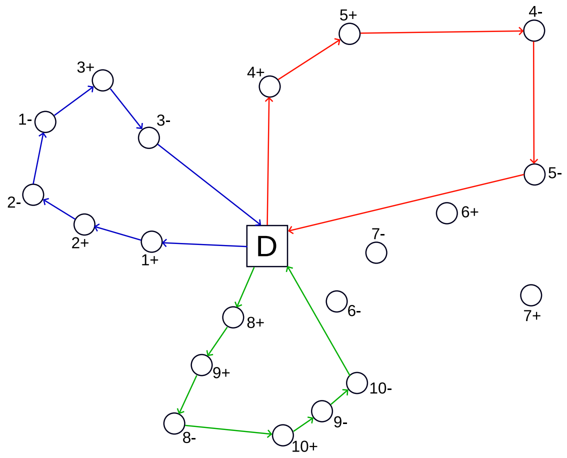
Static E-ADARP:

- Bongiovanni et al. (2019):
  - mixed integer linear programming (MILP) formulations
  - branch-and-cut algorithm
- Su et al. (2023):
  - deterministic annealing (DA) algorithm
- Limmer (2023):
  - bilevel large neighborhood search (BI-LNS)

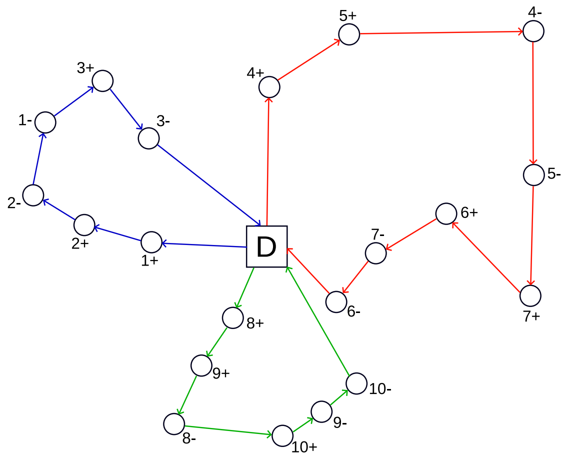
Large Neighborhood Search (LNS)



Initial solution.



Destroyed solution.



Repaired solution.

Destroy operator: random removal (Ropke and Pisinger, 2006)

LNS: Repair Operators

Greedy heuristic:

- Cheapest insertion over all unserved requests and routes

Time window order based heuristic:

- Non-decreasing order of the start of pickup time windows
- Cheapest insertion over all routes

Random order based heuristic:

- Random order
- Cheapest insertion over all routes

Solving Approaches

2 approaches for:

- Insertion and optimization of charging stops
- Efficient scheduling and evaluation

Direct Approach:

- Destroy and repair operators for charging stops
- Labeling algorithm

On-the-fly (OTF) approach: evaluation with OTF insertion of charging stops

- MILP formulation
- Time-efficient heuristic

On-The-Fly Approach

- Route evaluation procedure:
  - Checks feasibility
  - Inserts charging stops
  - Determines charging durations

OTF Heuristic: Workflow I

- Remove charging stops from route
- Identify fragments and depots
- Forward pass:
  - Earliest service start times and waiting durations
  - Battery levels of vehicle
  - Latest stop i^{ch} before which charging is necessary
- Backward pass:
  - Latest service start times and backward-waiting durations

OTF Heuristic: Workflow II

- Charging not necessary: terminate → feasible route
- Charging necessary
  - Go backwards from i^{ch}
  - Check each possible position and charging station
- No feasible insertion: terminate → infeasible route
- Feasible insertions: select and insert stop
- Update all affected data
- Repeat

Computational Study

- Implementation in Julia 1.10.0
- Gurobi 10.0.3 (single-threaded)
- Runs per instance: 30
- Time limit: 300 s
- Memory limit: 20 GB
- Intel Xeon E5-2640 with 2.4 GHz

Benchmark Instances

- 2 data sets:
  - 14 Cordeau[1,3] instances: 2 – 5 vehicles, 16 – 50 requests
  - 10 Ropke[2,4] instances: 5 – 8 vehicles, 48 – 96 requests
  - → Enhanced with E-ADARP features
- Minimum end battery level ratio: $\gamma = 0.7$
- 2 modes for CS visits:
  - 1 visit per CS: $n_S = 1$
  - Unlimited visits per CS: $n_S = \infty$

[1] Cordeau (2006)
[2] Ropke et al. (2007)
[3] Bongiovanni et al. (2019)
[4] Limmer (2023)

Results: Cordeau

Instance	MILP		DA			BI-LNS		Direct		OTF MILP		OTF Heuristic	
	RT [min]	Obj	RT mean [min]	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean
						$n_s = 1$							
a2-16	0.09	**240.66**	1.60	**240.66**	**240.66**	242.83	245.50	**240.66**	242.19	**240.66**	240.84	**240.66**	**240.66**
a2-20	120.00	NA	2.88	**293.27**	294.11	NA	NA	**293.27**	297.60	**293.27**	294.98	**293.27**	**293.27**
a2-24	16.02	358.21	3.44	**353.18**	NA	356.99	363.04	**353.18**	358.85	**353.18**	353.45	**353.18**	**353.18**
a3-18	0.80	**240.58**	0.97	**240.58**	**240.58**	242.49	246.13	244.35	244.39	**240.58**	240.98	**240.58**	**240.58**
a3-24	2.54	277.72	2.06	**275.97**	277.43	277.52	277.52	277.43	278.42	**275.97**	277.46	**275.97**	**275.97**
a3-30	120.00	NA	1.30	**424.93**	436.20	432.27	436.56	**424.93**	430.07	**424.93**	430.32	**424.93**	**426.12**
a3-36	120.00	**494.04**	2.09	**494.04**	502.27	496.75	500.84	**494.04**	497.86	**494.04**	502.35	**494.04**	**497.18**
a4-16	1.12	**223.13**	0.52	**223.13**	**223.13**	**223.13**	223.95	**223.13**	**223.13**	**223.13**	223.37	**223.13**	**223.13**
a4-24	30.58	318.21	0.90	**316.65**	318.31	319.37	321.10	318.31	319.89	**316.65**	319.17	**316.65**	**316.65**
a4-32	120.00	430.07	1.19	**397.87**	405.85	401.97	402.59	**397.87**	399.26	**397.87**	399.98	**397.87**	**397.87**
a4-40	120.00	NA	1.91	479.02	NA	471.72	478.93	**467.72**	480.89	471.78	487.42	**467.72**	**474.47**
a4-48	120.00	NA	2.74	582.22	NA	579.71	588.48	**575.35**	582.71	580.49	591.36	575.62	**579.63**
a5-40	120.00	447.63	1.63	424.26	436.94	420.20	423.88	420.01	425.32	425.93	433.02	**418.75**	**421.16**
a5-50	120.00	NA	2.64	603.24	NA	593.71	602.30	**589.61**	597.55	596.54	612.00	**589.61**	**596.09**
						$n_s = \infty$							
a2-16			1.99	**240.66**	**240.66**	242.44	242.44	241.32	241.34	**240.66**	**240.66**	**240.66**	**240.66**
a2-20			5.27	286.32	288.89	290.33	291.23	288.39	290.50	**285.86**	285.95	**285.86**	**285.86**
a2-24			5.96	354.38	374.68	354.53	356.89	**350.32**	355.40	**350.32**	350.33	**350.32**	**350.32**
a3-18			1.10	**238.82**	**238.82**	241.95	242.46	241.13	241.13	**238.82**	239.55	240.03	240.03
a3-24			2.50	**275.20**	**275.20**	277.52	278.02	276.27	276.62	**275.20**	276.05	**275.20**	**275.20**
a3-30			2.85	415.71	417.07	419.16	426.30	416.89	421.54	**413.45**	415.85	**413.45**	**414.08**
a3-36			5.72	484.85	487.91	490.26	492.79	484.07	492.96	**483.08**	490.60	484.49	**486.98**
a4-16			0.52	**222.49**	**222.49**	**222.49**	223.57	**222.49**	**222.49**	**222.49**	223.09	**222.49**	**222.49**
a4-24			1.18	315.98	317.99	316.51	318.38	**315.40**	316.48	**315.40**	316.00	315.98	**315.98**
a4-32			2.06	**394.94**	401.82	396.64	397.98	**394.94**	**394.94**	**394.94**	395.84	**394.94**	**394.94**
a4-40			3.77	458.52	467.60	461.16	461.91	458.63	462.73	**457.76**	465.13	457.88	**458.67**
a4-48			6.72	568.08	575.96	568.01	570.80	565.04	569.97	**561.15**	565.94	561.38	**564.54**
a5-40			2.50	419.33	425.29	418.79	421.06	415.96	417.24	**415.88**	425.07	**415.88**	**416.43**
a5-50			5.88	579.15	588.98	571.37	575.49	571.13	576.96	573.37	586.47	**567.61**	**571.48**

Results: Ropke

Instance	DA RT mean [min]	DA Obj min	DA Obj mean	BI-LNS Obj min	BI-LNS Obj mean	BI-LNS Feas	Direct Obj min	Direct Obj mean	Direct Feas	OTF MILP Obj min	OTF MILP Obj mean	OTF MILP Feas	OTF Heuristic Obj min	OTF Heuristic Obj mean	OTF Heuristic Feas
$n_s = 1$															
a5-60	8.46	NA	NA	NA	NA	0/10	NA	NA	0/30	NA	NA	0/30	NA	NA	0/30
a6-48	8.37	NA	NA	519.55	522.50	10/10	517.60	528.24	30/30	523.49	537.45	30/30	**517.12**	**521.62**	30/30
a6-60	5.45	NA	NA	733.45	742.02	9/10	719.78	737.77	30/30	736.52	750.14	10/30	**714.16**	**731.45**	30/30
a6-72	9.84	NA	NA	NA	NA	0/10	NA	NA	0/30	NA	NA	0/30	NA	NA	0/30
a7-56	3.69	NA	NA	649.11	669.71	10/10	**635.63**	656.45	30/30	656.86	675.46	25/30	636.56	**649.37**	30/30
a7-70	8.51	NA	NA	NA	NA	0/10	820.90	**840.02**	26/30	NA	NA	0/30	**816.64**	840.59	30/30
a7-84	13.18	NA	NA	NA	NA	0/10	NA	NA	0/30	NA	NA	0/30	NA	NA	0/30
a8-64	20.12	NA	NA	646.82	652.38	10/10	645.69	656.21	30/30	652.65	674.25	30/30	**639.06**	**651.33**	30/30
a8-80	14.47	NA	NA	854.85	863.74	10/10	842.23	863.08	30/30	869.19	888.37	8/30	**837.79**	**862.75**	30/30
a8-96	14.35	NA	NA	NA	NA	0/10	NA	NA	0/30	NA	NA	0/30	NA	NA	0/30
$n_s = \infty$															
a5-60	8.21	708.54	723.73	697.87	709.11	10/10	693.79	703.04	30/30	695.81	704.15	30/30	**686.36**	**692.75**	30/30
a6-48	8.07	509.76	525.10	511.04	514.53	10/10	508.22	510.85	30/30	509.87	519.44	30/30	**508.10**	**509.46**	30/30
a6-60	4.83	697.57	711.52	699.70	705.56	10/10	694.19	700.72	30/30	699.25	708.04	30/30	**689.95**	**695.13**	30/30
a6-72	9.57	796.19	826.48	788.34	801.80	10/10	780.68	796.83	30/30	782.32	799.84	30/30	**769.12**	**779.49**	30/30
a7-56	3.53	625.91	641.82	627.34	633.38	10/10	619.63	627.64	30/30	623.45	632.08	30/30	**617.12**	**621.51**	30/30
a7-70	8.00	781.56	800.35	777.69	785.49	10/10	763.22	773.36	30/30	769.84	783.03	30/30	**757.66**	**765.36**	30/30
a7-84	11.75	915.61	938.49	900.98	916.93	10/10	892.31	906.41	30/30	900.77	920.03	30/30	**888.40**	**897.33**	30/30
a8-64	21.50	649.93	668.48	645.62	648.60	10/10	634.36	646.57	30/30	654.75	663.89	30/30	**632.95**	**641.17**	30/30
a8-80	12.41	843.26	865.90	815.06	825.74	10/10	813.44	823.82	30/30	821.29	837.30	30/30	**801.08**	**814.87**	30/30
a8-96	13.45	1097.76	1136.43	1072.77	1091.06	10/10	1058.08	1073.35	30/30	1074.02	1091.18	30/30	**1048.87**	**1060.22**	30/30

Conclusions & Outlook

Conclusion

- Charging stops important for route quality
- All approaches competitive
- OTF heuristic overall best approach
  - 23/48 new best solutions

Future Work

- Very large-scale instances
- Machine learning-supported operators
- Dynamic E-ADARP

Discussion & Questions

Thank you!

Maria Bresich[1]: mbresich@ac.tuwien.ac.at
Günther R. Raidl[1]: raidl@ac.tuwien.ac.at
Steffen Limmer[2]: steffen.limmer@honda-ri.de

[1]Institute of Logic and Computation, TU Wien, Austria
[2]Honda Research Institute Europe, Offenbach/Main, Germany

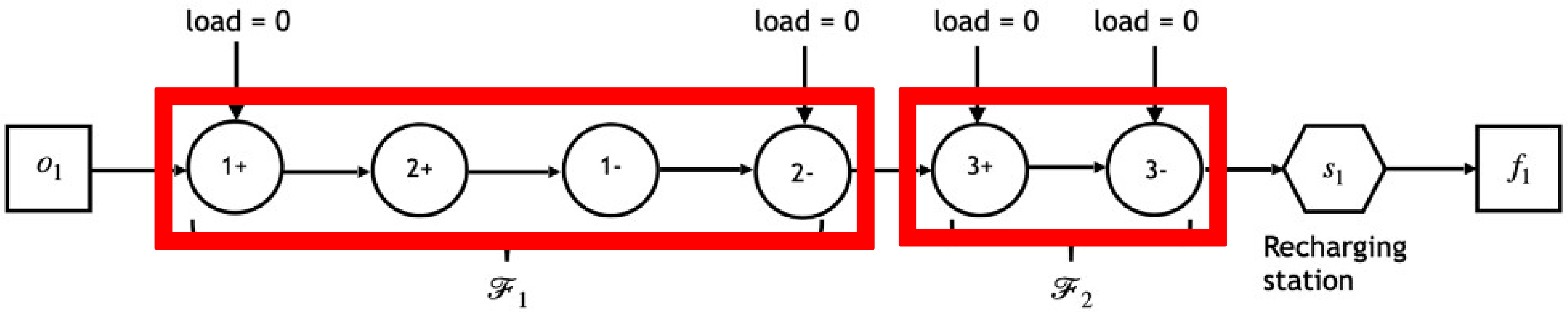
References I

- Claudia Bongiovanni, Mor Kaspi, and Nikolas Geroliminis. The electric autonomous dial-a-ride problem. Transportation Research Part B: Methodological, 122:436–456, 2019.
- Jean-François Cordeau. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. Operations Research, 54(3):573–586, 2006.
- Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological, 37(6):579–594, 2003.
- Sin C. Ho, Wai Y. Szeto, Yong-Hong Kuo, Janny M. Y. Leung, Matthew Petering, and Terence W. H. Tou. A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological, 111:395–421, 2018.
- Steffen Limmer. Bilevel large neighborhood search for the electric autonomous dial-a-ride problem. Transportation Research Interdisciplinary Perspectives, 21:100876, 2023.

References II

- Mohamed A. Masmoudi, Manar Hosny, Emrah Demir, Konstantinos N. Genikomsakis, and Naoufel Cheikhrouhou. The dial-a-ride problem with electric vehicles and battery swapping stations. Transportation Research Part E: Logistics and Transportation Review, 118:392–420, 2018.
- David Pisinger and Stefan Ropke. Large neighborhood search. In Michel Gendreau and Jean-Yves Potvin, editors, Handbook of Metaheuristics, pages 399–419. Springer US, Boston, MA, 2010. ISBN 978-1-4419-1665-5.
- Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science, 40(4):455–472, 2006.
- Stefan Ropke, Jean-François Cordeau, and Gilbert Laporte. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks, 49(4):258–272, 2007.
- Yue Su, Nicolas Dupin, and Jakob Puchinger. A deterministic annealing local search for the electric autonomous dial-a-ride problem. European Journal of Operational Research, 309(3):1091–1111, 2023.

Battery-Restricted Fragments Su et al. (2023)



Route with battery-restricted fragments, taken from Su et al. (2023).

- Subsequence of only pickup and drop-off nodes
- Vehicle arrives and leaves empty
- Minimum user excess ride time: exact calculation with a linear program (LP)

Battery-Restricted Fragments: Usage

Efficient route cost computation:

- Sum of fragments' minimum user excess ride times

Su et al. (2023):

- Preprocessing: enumerate and evaluate all feasible fragments
  - Time-consuming, memory-demanding

Our approach:

- Evaluate fragments on 1st encounter
  - → Better scalability
- Improve charging stop insertion

OTF Heuristic

- Feasibility with 1 charging stop:
  - Optimal insertion
  - Linear run time: $O(|R|)$
    - R … fragment-based route
- Multiple charging stops:
  - Best insertion heuristic
  - Run time: $O(|R| * |S'| * n_{charging})$
    - S' … set of available CSs
    - $n_{charging}$ … number of inserted CSs