Electric Autonomous Dial-A-Ride Problem

Charging, Scheduling, and Evaluation

Maria Bresich

PhD-Seminar

January 15, 2024

Outline

Recap

- Electric Autonomous Dial-A-Ride Problem (E-ADARP)
- Scheduling Problem
- Routing & Scheduling in E-ADARP
- Solving Approaches: Scheduling

Project Update

- Charging, Scheduling, and Evaluation
- Approach 1: Separation
- Approach 2: Combination - On-The-Fly Insertion
- Results

Recap

Electric Autonomous Dial-A-Ride Problem (E-ADARP)

Bongiovanni et al. (2019)

Definition (Static Electric Autonomous DARP)

Given: n users with transportation requests from a pickup to a drop-off location, a fleet of m electric autonomous vehicles

Task: Design m vehicle routes serving all requests, s.t. the total travel time and excess ride time of all users are minimized and certain constraints are satisfied.

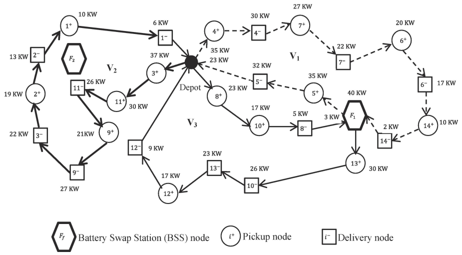

Figure: Example of E-DARP taken from Masmoudi et al. (2018).

Scheduling Problem

Occurs in routing problems where time is relevant

Time windows: a time window $[e_i, l_i]$ for a location i determines

- the earliest time e_i when service can take place
- the latest time l_i when service can take place

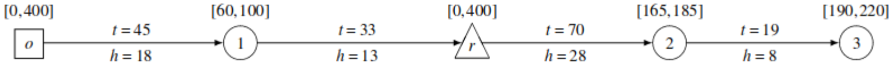

Figure: Route with time windows (taken from Desaulniers et al. (2016)).

Optimization Problem: Given a route (a sequence of locations), find a feasible schedule satisfying all time windows and maximum user ride time constraints and minimizing route duration.

$\rightarrow$ Needed for feasibility testing and route evaluation

Routing and Scheduling in DARP Cordeau and Laporte (2003)

Service related constraints:

- Time windows for pickup and drop-off locations
- Maximum user ride time

Scheduling:

- Satisfy constraints
- Minimize route duration

Goals: determine

- Routing: sequence of pickup and drop-off locations
- Departure time from the depot
- Service start time at each location

Additional goals:

- Routing: visits to charging stations
- Time for (partial) recharging
- Minimize user excess ride time

Delays (sometimes) beneficial

Solving Approaches: Scheduling

Exact:

- Linear programming (LP) (Bongiovanni et al. (2023))
- Labeling algorithm (Su et al. (2023))

Heuristic:

- 8-step scheduling procedure (Cordeau and Laporte (2003))
- Modified 8-step scheduling procedure (Parragh et al. (2009))
- 4-step scheduling heuristic (Molenbruch et al. (2017))
- Scheduling and battery management heuristic (Bongiovanni et al. (2023))

Problem: incorrect infeasibility declarations

Project Update

Solving Approach: Large Neighborhood Search (LNS)

- Destroy and repair operators for requests
- Insertion and optimization of charging stops
- Efficient scheduling and evaluation

} 2 approaches

- Destroy and repair operators for requests
- Insertion and optimization of charging stops
- Efficient scheduling and evaluation

} 2 approaches

Charging, scheduling, and evaluation:

- Separation:
  - Destroy and repair operator for charging visits
  - Labeling algorithm
- Combination: evaluation with on-the-fly insertion of charging visits
  - MILP formulation
  - Time-efficient heuristic

Charging, Scheduling, and Evaluation

Approach 1: Separation I

Destroy operator: random removal

- Remove randomly selected charging stops from routes

Repair operator: greedy heuristic (similar to Keskin and Çatay (2016))

- Time windows violated or no charging station available: remove stops
- Otherwise:
  - Determine first stop with negative battery level
  - Determine and insert nearest available charging station

  → Feasible: terminate

  → Time window infeasible: try preceding arcs

  → Battery infeasible: repeat
- Allows multiple charging stops in between successive fragments
- Combination with repair operator for requests

Charging, Scheduling, and Evaluation

Approach 1: Separation II

Labeling algorithm: (Su et al. (2023))

- Minimizes excess user ride time
- Checks feasibility regarding time window and battery constraints
- Linear time complexity
- Computes bounds for time values
- Does not compute concrete time values
  - → Solves the decision problem
  - → Charging times can be extracted
  - → Other times have to be derived

Problem: incorrect infeasibility declarations

→ Solved!

Approach 2: Combination

On-the-fly charging stop insertion:

- LNS does not deal with charging stops
- Evaluation procedure:
  - Inserts charging stops as needed
  - Determines times (waiting, service, charging)

Assumptions:

- ≤ 1 charging stops in between successive fragments or depots
- No time windows for charging stations

Allows consideration of:

- Limited number of usages of each charging station
- Individual charging speeds α_s at charging stations $s \in S$

  $\rightarrow$ not possible with labeling algorithm!

On-The-Fly Charging Stop Insertion

Mixed Integer Linear Program (MILP)

Benefits:

- Optimal insertions: stations, positions
- Allows multiple charging stops per route

Drawbacks:

- Computationally expensive
- Slow

On-The-Fly Charging Stop Insertion

Time-Efficient Heuristic I

Idea:

- Remove charging stops from route
- Determine remaining stops: fragments and depots
- Forward pass:
  - Earliest service start times t^{early} and waiting durations d^{wait}
  - Battery levels of vehicle
  - Latest stop i^{ch} before which charging is necessary
- Backward pass:
  - Latest service start times t^{late} and backward-waiting durations $d^{\mathrm{wait\_back}}$

On-The-Fly Charging Stop Insertion

Time-Efficient Heuristic I

Idea (continued):

- Charging not necessary: terminate
- Charging necessary:
  - Go backwards from i^{ch}
  - Check each possible position and charging station
- No feasible insertion: terminate with infeasible route
- Feasible insertions: select and insert stop
  - If available: feasible tour and minimum detour length
  - Otherwise: most energy charged
- Update all affected service and waiting times and battery levels
- Repeat

On-The-Fly Charging Stop Insertion

Time-Efficient Heuristic II

- Allows multiple charging stops per route
- Feasibility with 1 charging stop possible:
  - Optimal insertion
  - Linear run time: $O(|R|)$
- Multiple charging stops necessary:
  - Best insertion heuristic (similar to Keskin and Çatay (2016))
  - Run time: $O(|R| * n_{\text{charging}})$
  - n_{charging} ... number of inserted charging stations

Option: if improvement, apply MILP once at the end for optimal insertion

- Only small further improvements
- Massive slowdown

→ Discarded!

Cordeau Instances

Table: Results on Cordeau (type-a) instances with limited CS visits and $\gamma = 0.7$. Note that the best known result for a5-50 is 587.38 from another configuration (5) of operators + labeling.

Instance	e-ADARP2 (Bongiovanni et al. (2019))		DA (Su et al. (2023))			BI-LNS (Limmer (2023))		Operators + Labeling (4)		OTF insertion MILP		OTF insertion 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
a2-16	0.09	**240.66**	1.6	**240.66**	240.66	242.83	245.5	**240.66**	241.91	**240.66**	240.77	**240.66**	240.66
a2-20	120	NA	2.88	**293.27**	294.11	NA	NA	**293.27**	296.08	**293.27**	20236.6	**293.27**	293.27
a2-24	16.02	358.21	3.44	**353.18**	-	356.99	363.04	**353.18**	359.98	**353.18**	353.45	**353.18**	353.18
a3-18	0.8	**240.58**	0.97	**240.58**	240.58	242.49	246.13	244.35	244.37	**240.58**	240.65	**240.58**	240.58
a3-24	2.54	277.72	2.06	**275.97**	277.43	277.52	277.52	277.43	278.08	**275.97**	277.62	**275.97**	275.97
a3-30	120	NA	1.3	**424.93**	436.2	432.27	436.56	**424.93**	429.6	425.98	432.54	**424.93**	426.75
a3-36	120	**494.04**	2.09	**494.04**	502.27	496.75	500.84	**494.04**	497.98	**494.04**	501.93	**494.04**	496.49
a4-16	1.12	**223.13**	0.52	**223.13**	223.13	**223.13**	223.95	**223.13**	223.13	**223.13**	223.32	**223.13**	223.13
a4-24	30.58	318.21	0.9	**316.65**	318.31	319.37	321.1	318.31	319.87	**316.65**	318.42	**316.65**	316.65
a4-32	120	430.07	1.19	**397.87**	405.85	401.97	402.59	**397.87**	399.31	**397.87**	398.89	**397.87**	397.87
a4-40	120	NA	1.91	479.02	-	471.72	478.93	**467.72**	480.25	474.27	491.94	**467.72**	472.82
a4-48	120	NA	2.74	582.22	-	579.71	588.48	**574.51**	583.02	578.01	589.63	575.1	579.28
a5-40	120	447.63	1.63	424.26	436.94	420.2	423.88	**418.75**	427.27	424.75	434.61	**418.75**	420.6
a5-50	120	NA	2.64	603.24	-	593.71	602.3	590.68	3912.35	596.77	13866.27	**589.61**	596.03

Results II

Cordeau Instances

Table: Results on Cordeau (type-a) instances with unlimited CS visits and $\gamma = 0.7$.

Instance	DA (Su et al. (2023))			BI-LNS (Limmer (2023))		Operators + Labeling (4)		OTF insertion MILP		OTF insertion heuristic	
	RT mean [min]	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean
a2-16	1.99	**240.66**	240.66	242.44	242.44	241.32	241.38	**240.66**	240.66	**240.66**	240.66
a2-20	5.27	286.32	288.89	290.33	291.23	290.47	290.81	**285.86**	285.88	**285.86**	285.86
a2-24	5.96	354.38	374.68	354.53	356.89	**350.32**	357.42	**350.32**	350.33	**350.32**	350.32
a3-18	1.1	**238.82**	238.82	241.95	242.46	241.13	241.13	**238.82**	239.14	240.03	240.03
a3-24	2.5	**275.2**	275.2	277.52	278.02	276.27	276.64	**275.2**	275.84	**275.2**	275.2
a3-30	2.85	415.71	417.07	419.16	426.3	416.89	422.51	**413.45**	415.56	**413.45**	414
a3-36	5.72	484.85	487.91	490.26	492.79	484.07	493.63	**483.08**	489.05	484.49	486.5
a4-16	0.52	**222.49**	222.49	**222.49**	223.57	**222.49**	222.49	**222.49**	222.71	**222.49**	222.49
a4-24	1.18	315.98	317.99	316.51	318.38	**315.4**	316.88	**315.4**	316.25	315.98	315.98
a4-32	2.06	**394.94**	401.82	396.64	397.98	**394.94**	395.44	**394.94**	396.04	**394.94**	394.94
a4-40	3.77	458.52	467.6	461.16	461.91	458.98	465.57	459.23	465.7	**457.88**	459.7
a4-48	6.72	568.08	575.96	568.01	570.8	566.05	569.55	**561.15**	567.54	561.97	565.13
a5-40	2.5	419.33	425.29	418.79	421.06	415.96	417.71	418.18	426.43	**415.88**	416.58
a5-50	5.88	579.15	588.98	571.37	575.49	569.64	577.61	573.32	583.09	**567.61**	572.82

Results III

Ropke Instances

Table: Results on Ropke instances with limited CS visits and $\gamma = 0.7$. Not shown best known results: 774.21 for a5-60 (improved BI-LNS), 712.86 for a6-60 (OTF heuristic), and 631.34 for a7-56 (configuration (4) of operators + labeling).

Instance	BI-LNS (Limmer (2023))		Operators + Labeling (6)		OTF insertion MILP		OTF insertion heuristic	
	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean
a5-60	NA	NA	100001	100001.13	100001	100002.93	100001	100001
a6-48	519.55	522.5	**515.14**	528.62	527.97	541.53	516.76	521.55
a6-60	733.45	742.02	723.05	739.84	743.79	93384.17	**719.84**	731.57
a6-72	NA	NA	**100001**	100001.53	100003	100004.33	**100001**	100001
a7-56	649.11	669.71	**633.59**	658.25	657.88	33785.54	635.68	650.65
a7-70	NA	NA	827.04	60337.82	100001	100002.43	**815.79**	843.79
a7-84	NA	NA	100002	100002.47	100003	100006.07	**100001**	100001.73
a8-64	646.82	652.38	644.37	663.2	661.4	678.38	**639.06**	649.99
a8-80	854.85	863.74	**839.13**	867.43	100001	100001.93	839.52	859.54
a8-96	NA	NA	100006	100007.17	100009	100012.3	**100005**	100005.93

Results IV

Ropke Instances

Table: Results on Ropke instances with unlimited CS visits and $\gamma = 0.7$. Not shown best known results: 769.01 for a6-72, 616.51 for a7-56, 800.76 for a8-80, and 1044.67 for a8-96 (OTF heuristic).

Instance	DA (Su et al. (2023))			BI-LNS (Limmer (2023))		Operators + Labeling (6)		OTF insertion MILP		OTF insertion heuristic	
	RT mean [min]	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean
a5-60	8.21	708.54	723.73	697.87	709.11	695.74	705.28	698.69	706.42	**687.1**	694.04
a6-48	8.07	509.76	525.1	511.04	514.53	508.22	512.73	512.43	522.86	**508.1**	508.9
a6-60	4.83	697.57	711.52	699.7	705.56	696.12	702.47	702.41	710.76	**690.47**	694.35
a6-72	9.57	796.19	826.48	788.34	801.8	784.73	800.47	782.15	805.2	**771.14**	779.92
a7-56	3.53	625.91	641.82	627.34	633.38	620.45	630.68	626.25	637.28	**616.97**	620.74
a7-70	8	781.56	800.35	777.69	785.49	762.47	777.3	770.18	790.37	**757.34**	764.08
a7-84	11.75	915.61	938.49	900.98	916.93	891.58	909.54	909.72	935.44	**879.06**	893.06
a8-64	21.5	649.93	668.48	645.62	648.6	641.65	648.9	651.73	670.19	**632.61**	640.04
a8-80	12.41	843.26	865.9	815.06	825.74	813.95	826.89	826.01	844.76	**804.93**	813.4
a8-96	13.45	1097.76	1136.43	1072.77	1091.06	1048.77	1083.08	1081.76	1106.08	**1045.53**	1057.53

Results V

Large Instances

Table: Results on large instances with unlimited CS visits and $\gamma = 0.7$.

Instance	BI-LNS (5) (Limmer (2023))		BI-LNS (10) (Limmer (2023))		BI-LNS (15) (Limmer (2023))		OTF insertion heuristic	
	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean	Obj min	Obj mean
a180-3600	29793.51	29977.8	29397.03	29546.14	**29177.96**	29312.55	35113.43	35434.73
a200-4000	32803.99	33161.17	32390.78	32558.8	**32013.56**	32263.01	38842.84	39167.02
a220-4400	36365.84	37523.84	35627.8	35746.51	**35259.06**	35428.04	42333.38	42618.49
a240-4800	41357.57	42150.46	38496.01	38817.89	**38270.98**	38460.95	46336.67	46644.66
a260-5200	NA	NA	41890.96	42191.5	**41472.11**	41745.98	50083.3	50388.35

Discussion & Questions

Thank you!

References I

Claudia Bongiovanni, Mor Kaspi, and Nikolas Geroliminis. The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological*, 122:436–456, April 2019.

Claudia Bongiovanni, Nikolas Geroliminis, and Mor Kaspi. A ride time-oriented scheduling algorithm for dial-a-ride problems. 3333(arXiv:2211.07347), Apr 2023. arXiv:2211.07347 [cs, math].

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, July 2003.

Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, Dec 2016.

Merve Keskin and Bülent Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127, April 2016.

Steffen Limmer. Bilevel large neighborhood search for the electric autonomous dial-a-ride problem. *Transportation Research Interdisciplinary Perspectives*, 21:100876, Sep 2023.

Mohamed Amine 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, October 2018.

Yves Molenbruch, Kris Braekers, An Caris, and Greet Vanden Berghe. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Operations Research*, 77:58–71, Jan 2017.

Sophie N. Parragh, Karl F. Doerner, Richard F. Hartl, and Xavier Gandibleux. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks*, 54(4):227–242, 2009.

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, Sep 2023.

(Modified) Labeling Algorithm I Su et al. (2023)

Each node $i \in \mathcal{R}$ of a route $\mathcal{R}$ has a label with 4 resource attributes:

$$L_i := \{T_i^{rch}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}\}$$

- $T_i^{rch_s}$: list of charging stations $s \in S$ visited up to i
- T_i^{tMin}: earliest service start time at i assuming minimum recharges
- T_i^{tMax}: earliest service start time at i assuming maximum recharges
- T_i^{rtMax}: maximum charging time to fully recharge at i assuming minimum recharges

New initial label: $\{\emptyset, w_1^{\text{start}}, w_1^{\text{start}}, \frac{Q-B_1}{\alpha}\}$

(Modified) Labeling Algorithm II Su et al. (2023)

Use resource extension functions (REFs) to compute the succeeding label L_j from the previous label L_i:

$$T_j^{\mathrm{tMin}} = \begin{cases} \max(w_j^{\mathrm{start}}, T_i^{\mathrm{tMin}} + d_i + t_{i,j}) & \text{if } T_i^{\mathrm{rch}} = \emptyset \\ \max(w_j^{\mathrm{start}}, T_i^{\mathrm{tMin}} + d_i + t_{i,j}) + Z_{i,j} & \text{otherwise} \end{cases} \quad (1)$$

$$T_j^{\mathrm{tMax}} = \begin{cases} \min(w_j^{\mathrm{end}}, \max(w_j^{\mathrm{start}}, T_i^{\mathrm{tMin}} + T_i^{\mathrm{rtMax}} + d_i + t_{i,j})) & \text{if } i \in S \\ \min(w_j^{\mathrm{end}}, \max(w_j^{\mathrm{start}}, T_i^{\mathrm{tMax}} + d_i + t_{i,j})) & \text{otherwise} \end{cases} \quad (2)$$

$$T_j^{\mathrm{rtMax}} = \begin{cases} T_i^{\mathrm{rtMax}} + (\beta_{i,j} + b_j)/\alpha & \text{if } T_i^{\mathrm{rch}} = \emptyset \\ \min(H, \max(0, T_i^{\mathrm{rtMax}} - S_{i,j}) + (\beta_{i,j} + b_j)/\alpha) & \text{otherwise} \end{cases} \quad (3)$$

(Modified) Labeling Algorithm III Su et al. (2023)

A route $\mathcal{R}$ is feasible if and only if $\forall j \in \mathcal{R}$, the label L_j satisfies:

$$T_j^{tMin} \leq w_j^{\text{end}} \tag{4}$$

$$T_j^{tMin} \leq T_j^{tMax} \tag{5}$$

$$T_j^{rch_s} \leq 1 \tag{6}$$

$$T_j^{rtMax} \leq H \tag{7}$$