07/05/2022

A Column-Generation Approach for the Electric Autonomous Dial-a-Ride Problem

Anthropolis chair seminar for 2022

INTERVENANTS

SU Yue

PhD candidate

yue.su@centralesupelec.fr

Jakob Puchinger

Professor at CentraleSupélec,
Head of the Anthropolis chair

jakob.puchinger@irt-systemx.fr

Nicolas Dupin

PhD, Maître de Conférences
Associé à temps plein

nicolas.dupin@lisn.upsaclay.fr

Plan

AnthroPOLIS
HUMAN CENTERED URBAN DESIGN

SystemX
INSTITUT DE RECHERCHE TECHNOLOGIQUE

1. Introduction: E-ADARP

1.1 Introduction: E-ADARP

Background

Transport-related problems: greenhouse gas emission, congestion…

Two possible directions to address these concerns

Using battery electric vehicles

- Becoming increasing relevant in green freight transportation (no pollutant emissions and lower noise pollution)
- Limitations: higher initial investment, reduced range, potential need to recharge on the route

Providing ride-sharing services

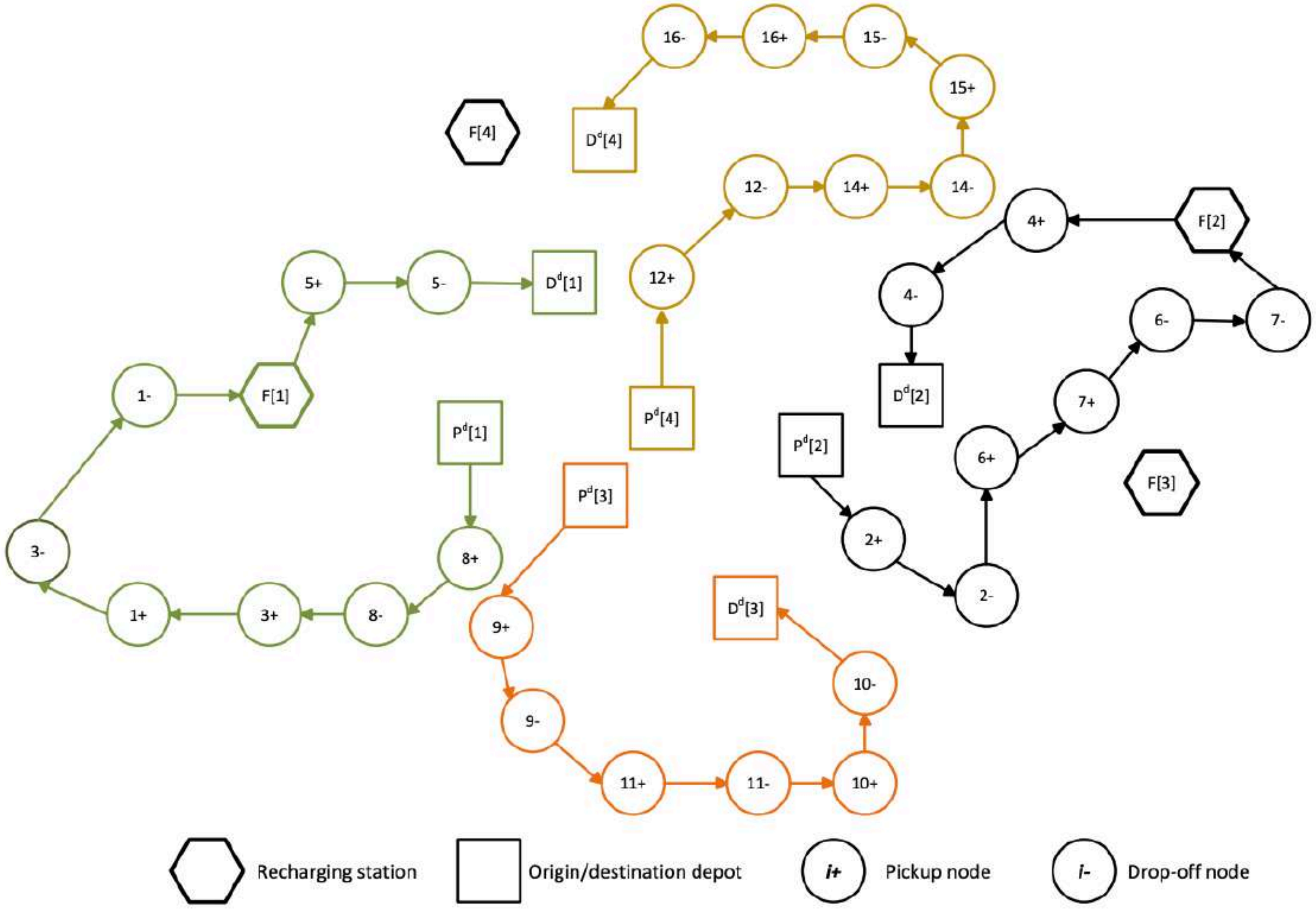
- Reduce the number of vehicles on route, benefit users with discounted price

- **Dial-a-Ride problem (DARP)**: a fleet of vehicles provides ride-sharing services to users specifying their origins, destinations, and preferred arrival time

Few studies have been done to investigate the influence of electric vehicle on DARP optimization (Bongiovanni et al., 2019)

1.2 Introduction: E-ADARP

E-ADARP example: 4 vehicles 16 requests



Illustrative example of E-ADARP

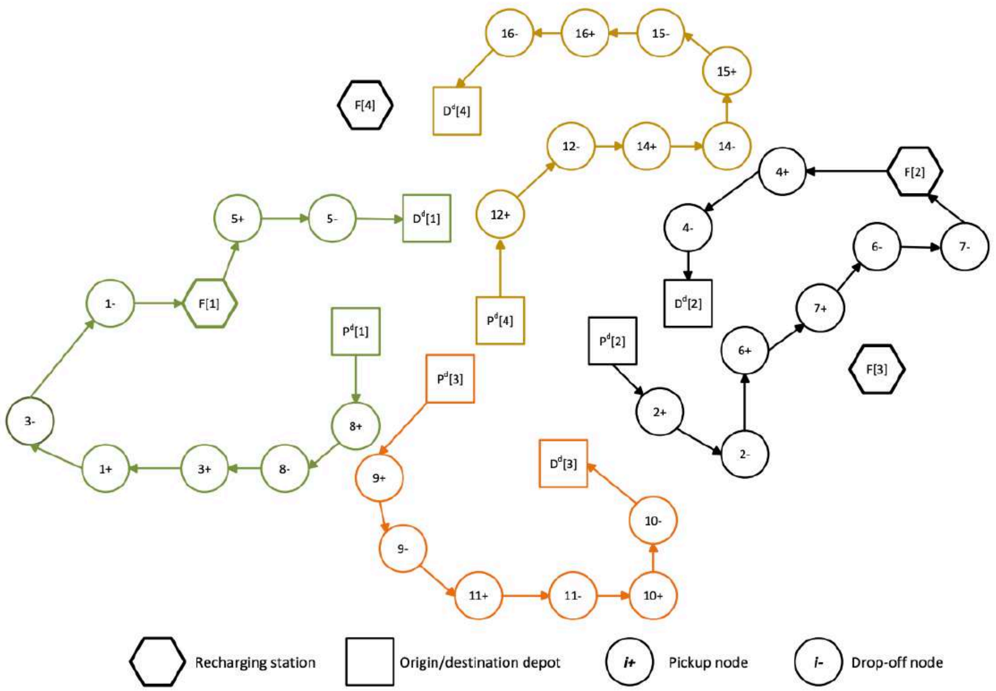
1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of DARP (differs from VRP):

- Pair and precedence constraint
  -> drop-off node should be visited after its pickup node
  -> drop-off node and the corresponding pickup node should on the same route.

- The service quality should be considered
  -> a maximum user ride time is set as constraint
  -> the delay of service begin time is possible to eliminate the unnecessary waiting time and improve service quality

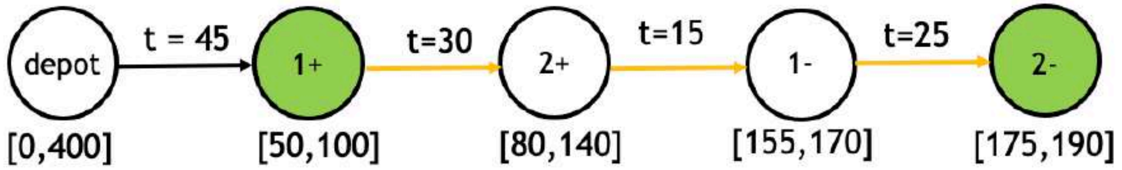


Illustrative example of DARP

1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of DARP (differs from VRP):



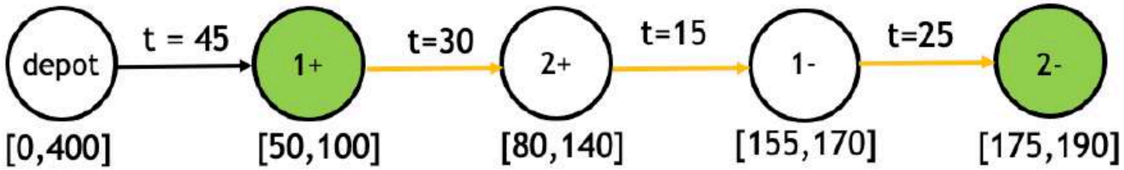
Node	depot	1+	2+	1-	2-
B	55	100	130	155(145 arrive)	180

Illustrative example of improving service quality by delaying the service begin time at previous node

1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of DARP (differs from VRP):



Node	depot	1+	2+	1-	2-
B	55	100	130	155(145 arrive)	180
B’	55	100	140	155	180

Total user ride time = 55+50=105
Total user ride time’ = 55+40=95

Illustrative example of improving service quality by delaying the service begin time at previous node

1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of DARP (differs from VRP):

- The service quality should be considered

-> a maximum user ride time is set as constraint

-> the delay of service begin time helps to improve service quality

"Eight-step" method in Cordeau and Laporte (2003)
- calculate Forward Time Slack
- improve the solution feasibility

1. Set $D_0 := e_0$.
2. Compute A_i, W_i, B_i and D_i and for each vertex v_i in the route.
3. Compute F_0.
4. Set $D_0 := e_0 + \min\{F_0, \sum_{0<p<q} W_p\}$.
5. Update A_i, W_i, B_i and D_i for each vertex v_i in the route.
6. Compute L_i for each request assigned to the route.
7. For every vertex v_j that corresponds to the origin of a request j
   (a) Compute F_j.
   (b) Set $B_j := B_j + \min\{F_j, \sum_{j<p<q} W_p\}$; $D_j := B_j + d_j$.
   (c) Update A_i, W_i, B_i and D_i, for each vertex v_i that comes after v_j in the route.
   (d) Update the ride time L_i for each request i whose destination vertex is after vertex
8. Compute changes in violations of vehicle load, route duration, time window and ri constraints.

To improve solution feasibility

1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of E-ADARP (differs from DARP):

- Weighted-sum objective function
  - total travel time for all vehicles
  - total excess user ride time for all the users

  => need to optimize the schedule!

No method has been designed to determine the optimal schedule…

“Eight-step” method in Cordeau and Laporte (2003)?

- calculate Forward Time Slack
- improve the solution feasibility
- cannot guarantee to get “optimal” schedule!

1. Set $D_0 := e_0$.
2. Compute A_i, W_i, B_i and D_i and for each vertex v_i in the route.
3. Compute F_0.
4. Set $D_0 := e_0 + \min\{F_0, \sum_{0<p<q} W_p\}$.
5. Update A_i, W_i, B_i and D_i for each vertex v_i in the route.
6. Compute L_i for each request assigned to the route.
7. For every vertex v_j that corresponds to the origin of a request j
   (a) Compute F_j.
   (b) Set $B_j := B_j + \min\{F_j, \sum_{j<p<q} W_p\}$; $D_j := B_j + d_j$.
   (c) Update A_i, W_i, B_i and D_i, for each vertex v_i that comes after v_j in the route.
   (d) Update the ride time L_i for each request i whose destination vertex is after verte
8. Compute changes in violations of vehicle load, route duration, time window and constraints.

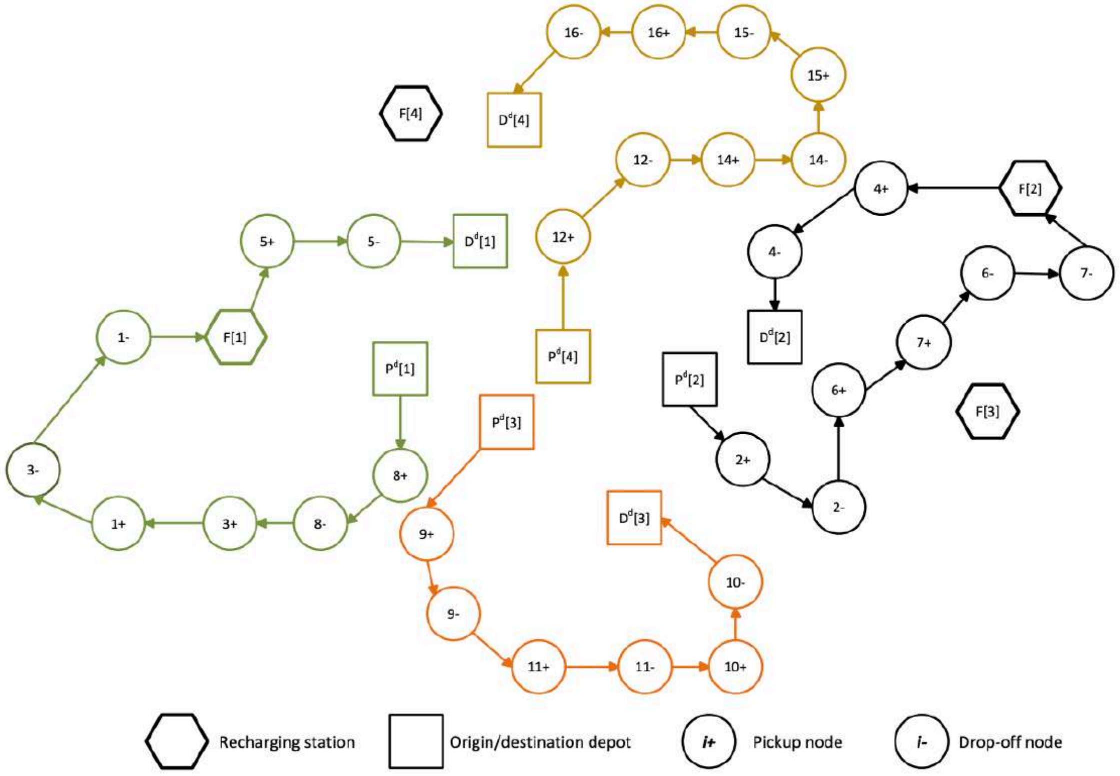
1.2 Introduction: E-ADARP

E-ADARP characteristics

Characteristics of E-ADARP (differs from DARP):

- Weighted-sum objective function
  - total travel time for all vehicles
  - total excess user ride time for all the users
- The use of EAVs
  - Detour to recharging station on the route
  - Partial recharging at recharging stations
  - No restriction on route duration
  - Limit of visit to recharging stations
  - Minimum battery level at the end of route

Three different scenarios: 10%, 40%, and 70% of total energy being kept at the end of route



Illustrative example of 4 vehicles and 16 requests

1.3 Introduction: E-ADARP

Problem definition

Objective:

Minimizing total travel time and excess user ride time: $w_1 = 0.75,\ w_2 = 0.25$

$$w_1 * \sum_{k \in K} \sum_{j \in V \setminus P^d} \sum_{i \in V} x_{i,j}^{k,r} \bullet t_{i,j}^{r} + w_2 * \sum_{i \in P^u} R_i$$

Decision Variables	Definitions
$x_{i,j}^k$	1 if vehicle k travels from location i to $j \in V$, 0 otherwise
T_i^k	Service begin time of vehicle k at location $i \in V$
L_i^k	Load of vehicle k at location $i \in V$
B_i^k	Battery load of vehicle k at location $i \in V$
E_f^k	Charging time of vehicle k at charging station $f \in F$
R_i	Excess user ride time of passenger $i \in P^u$

TAB. 1 – Decision variables for E-ADARP

1.3 Introduction: E-ADARP

Problem definition

Objective:

Minimizing total travel time and excess user ride time: $w_1 = 0.75,\ w_2 = 0.25$

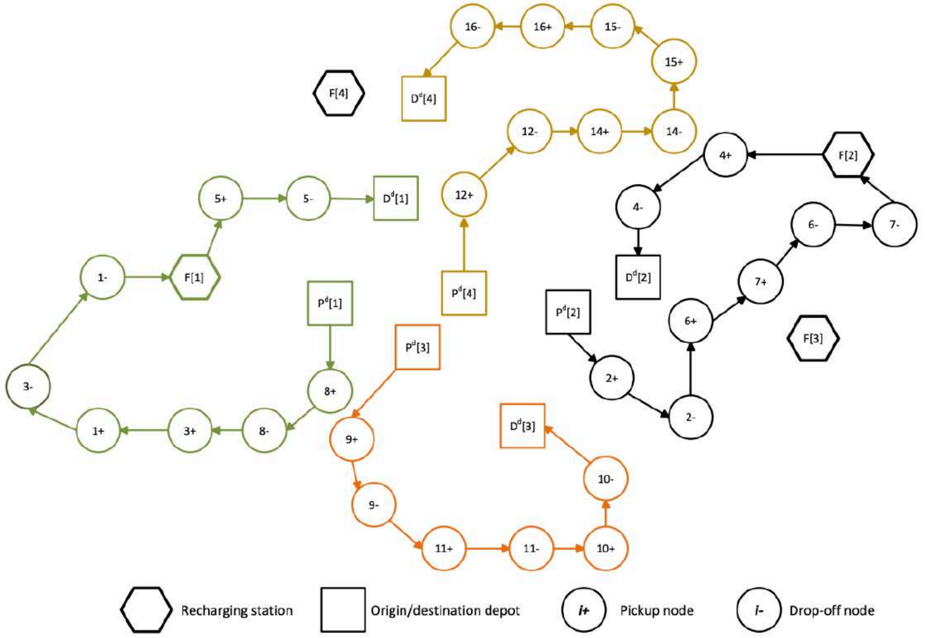
$$w_1 * \sum_{k \in K} \sum_{j \in V \setminus P^d} \sum_{i \in V} x_{i,j}^{k,r} \bullet t_{i,j}^{r} + w_2 * \sum_{i \in P^u} R_i$$

Constraints:

- Constraints on arcs and nodes
- Time window on pickup and drop-off nodes
- Capacity of vehicle (Identical)
- User maximum ride time
- Battery and charging limitation (minimum battery level at the end of route)

Given data :

- Pickup and drop-off locations for each request
- Recharging station locations



Illustrative example of 4 vehicles and 16 requests

1.3 Introduction: E-ADARP

Extended formulation (master problem)

Set covering problem (MP)

$$\text{minimize} \sum_{w\in\Omega'} c_w y_w + \sum_{i\in P^u} P_i a_i$$

$$\sum_{w\in\Omega'} \theta_{iw} y_w \geq 1 - a_i, \forall i \in P^u$$

$$\sum_{w\in\Omega'} \varphi_{fw} y_w \leq 1, \forall f \in F \cup D^d$$

$$\sum_{w\in\Omega'} y_w \leq |K|$$

$$y_w \in \{0,1\}, \ \forall w \in \Omega$$

Continuous Restricted MP (RMP)

$$\text{minimize} \sum_{w\in\Omega'} c_w y_w + \sum_{i\in P^u} P_i a_i$$

$$\sum_{w\in\Omega'} \theta_{iw} y_w \geq 1 - a_i, \forall i \in P^u$$

$$\sum_{w\in\Omega'} \varphi_{fw} y_w \leq 1, \forall f \in F \cup D^d$$

$$\sum_{w\in\Omega'} y_w \leq |K|$$

$$y_w \geq 0, \ \forall w \in \Omega'$$

2. Methodology

2.1 CG approach for E-ADARP

Column generation scheme

Continuous RMP

$$minimize \sum_{w\in\Omega'} c_w y_w + \sum_{i\in P^u} P_i a_i$$

$$\sum_{w\in\Omega'} \theta_{iw} y_w \geq 1 - a_i, \forall i \in P^u \longrightarrow \pi_i$$

$$\boxed{\sum_{w\in\Omega'} \varphi_{fw} y_w \leq 1, \forall f \in F \cup D^d} \longrightarrow \tau_f$$

$$\boxed{\sum_{w\in\Omega'} y_w \leq |K|} \longrightarrow \sigma$$

$$y_w \geq 0, \forall w \in \Omega'$$

pricing sub-problem:

$$\min_{w\in\Omega} c_w - \sum_{i\in P^u} \theta_{iw}\pi_i - \sum_{f\in F\cup D^d} \varphi_{fw}\tau_f - \sigma$$

Subject to:

- Constraints on arcs and nodes
- Time window on pickup and drop-off nodes
- Capacity of vehicle
- Maximum user ride time
- Minimum battery level constraint
- Battery capacity constraints
- Vehicle capacity constraints
- Restricted visit to recharging station

2.1 CG approach for E-ADARP

Column generation scheme

Continuous RMP

$$\textit{minimize} \sum_{w\in\Omega'} c_w y_w + \sum_{i\in P^u} P_i a_i$$

$$\sum_{w\in\Omega'} \theta_{iw} y_w \geq 1 - a_i, \forall i \in P^u$$

$$\sum_{w\in\Omega'} \varphi_{fw} y_w \leq 1, \forall f \in F \cup D^d$$

$$\sum_{w\in\Omega'} y_w \leq |K|$$

$$y_w \geq 0, \ \forall w \in \Omega'$$

Dual information

$(\pi_i \sigma), \tau_f$,

Stop criteria: no negative columns can be found

Negative-reduced-cost columns into Ω'

Pricing sub-problem:

$$\min_{w\in\Omega'} c_w - \sum_{i\in P^u} \theta_{iw}\pi_i - \sum_{f\in F\cup D^d} \varphi_{fw}\tau_f - \sigma$$

Subject to:

- Constraints on arcs and nodes
- Time window on pickup and drop-off nodes
- Capacity of vehicle
- Maximum user ride time
- Minimum battery level constraint
- Battery capacity constraints
- Vehicle capacity constraints
- Restricted visit to recharging station

2.2 Labeling algorithm to solve subproblem

Representation of partial path: how to handle excess user ride time?

$$\min_{w\in\Omega'} c_w - \sum_{i\in P^u} \theta_{iw}\pi_i - \sum_{f\in F\cup D^d} \varphi_{fw}\tau_f - \sigma$$

$$\boxed{w_1 * \sum_{k\in K}\sum_{j\in V\setminus P^d}\sum_{i\in V} x_{i,j}^{k,r} \bullet t_{i,j}^{r} + w_2 * \sum_{i\in P^u} R_i}$$

Routing cost:
(1) total travel time
(2) total excess user ride time

Total excess user ride time:

- We cannot determine the minimum excess user ride time when the delivery is not complete
- Total excess user ride time is not always increasing if extending node by node

2.2 Labeling algorithm to solve subproblem

Representation of partial path: how to handle excess user ride time?

$$\min_{w\in\Omega'} c_w - \sum_{i\in P^u} \theta_{iw}\pi_i - \sum_{f\in F\cup D^d} \varphi_{fw}\tau_f - \sigma$$

$$\boxed{w_1 * \sum_{k\in K}\sum_{j\in V\setminus P^d}\sum_{i\in V} x_{i,j}^{k,r} \bullet t_{i,j}^{r} + w_2 * \sum_{i\in P^u} R_i}$$

Routing cost:
(1) total travel time
(2) total excess user ride time

Total excess user ride time:

- We cannot determine the minimum excess user ride time when the delivery is not complete
- Total excess user ride time is not always increasing if extending node by node

Confliction in dominance rule:

- If label 1 dominates label 2 because of lower reduced cost, however after extend label1 and label 2 from current node i to the next node j, $cost'_1 > cost'_2$

We cannot extend the partial path node by node!

2.2 Labeling algorithm to solve subproblem

Representation of partial path

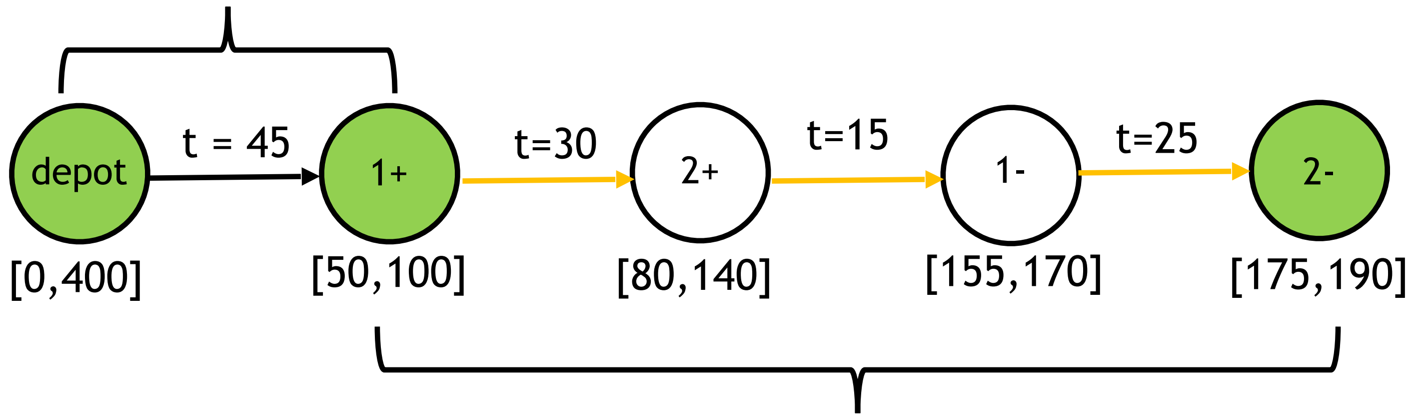
$$\mathscr{P} = \mathscr{S}_1 + \mathscr{S}_2 + \cdots + \mathscr{S}_n$$

Zero-split node:

- The node that no passenger on board at arrival/departure
- Example: depot, recharging station, pickup/ drop-off (with condition)

Segment:

Segment that starts and ends with zero-split node



2.2 Labeling algorithm to solve subproblem

Calculate the minimum excess user ride time for each segment

Important points:

- The optimal schedule is not unique for a given segment;
- We need to consider all the optimal schedules for a given segment;

Two-stage evaluation scheme:

- Stage 1: determine the latest optimal schedule $\mathscr{B}^l$ for a given segment;
- Stage 2: based on stage 1, determine the earliest optimal schedule $\mathscr{B}^e$ for a given segment;

With $\mathscr{B}^e$ and $\mathscr{B}^l$ we have all the optimal schedules for a segment!

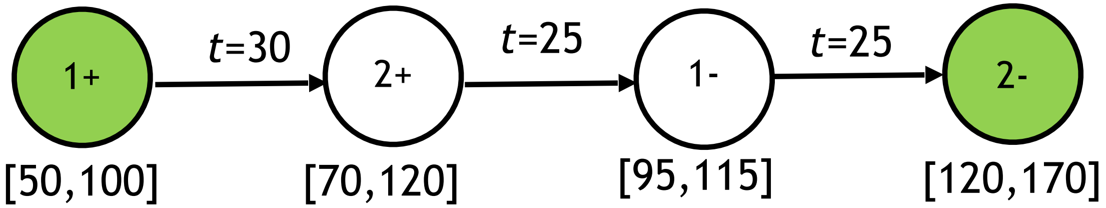
Maximum user ride time feasibility

- Checked after we obtain $\mathscr{B}^e$ and $\mathscr{B}^l$

2.2 Labeling algorithm to solve subproblem

Cost evaluation in each segment: two-stage evaluation scheme

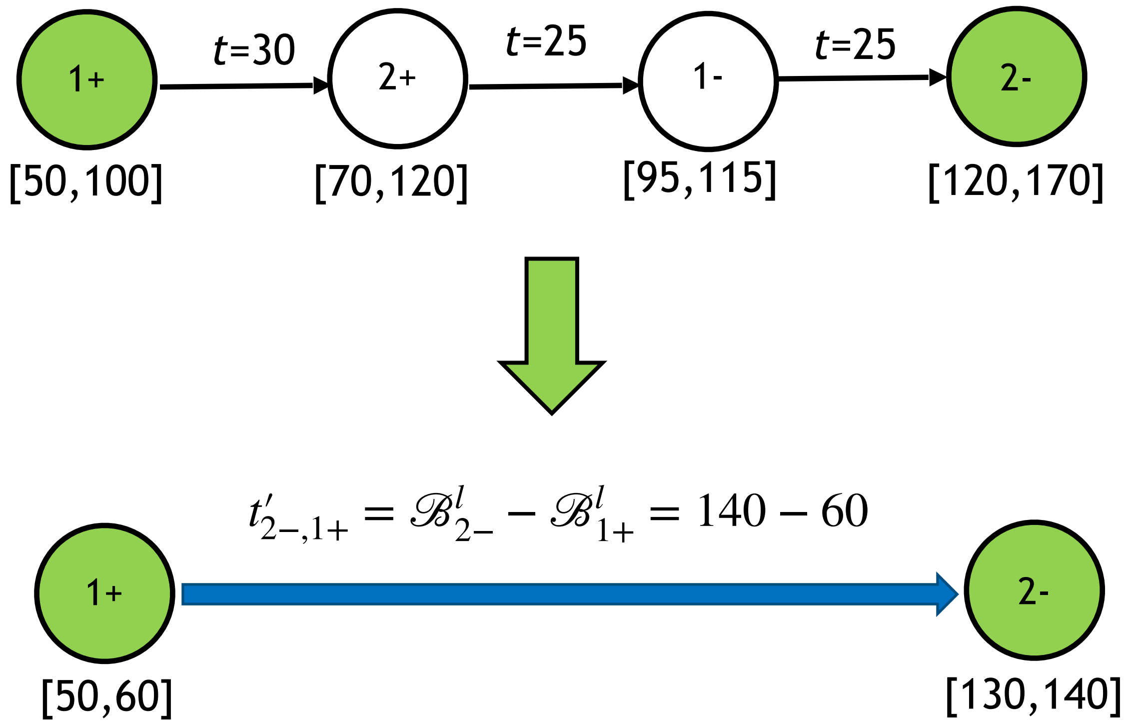
With $\mathscr{B}^e$ and $\mathscr{B}^l$ we have all the optimal schedules for a zero-split segment!



$\mathscr{B}^l$	60	90	115	140
$\mathscr{B}^\delta$	$60-\delta$	$90-\delta$	$115-\delta$	$140-\delta$
$\mathscr{B}^e$	50	80	105	130

2.2 Labeling algorithm to solve subproblem

Convert original graph to a sparser graph

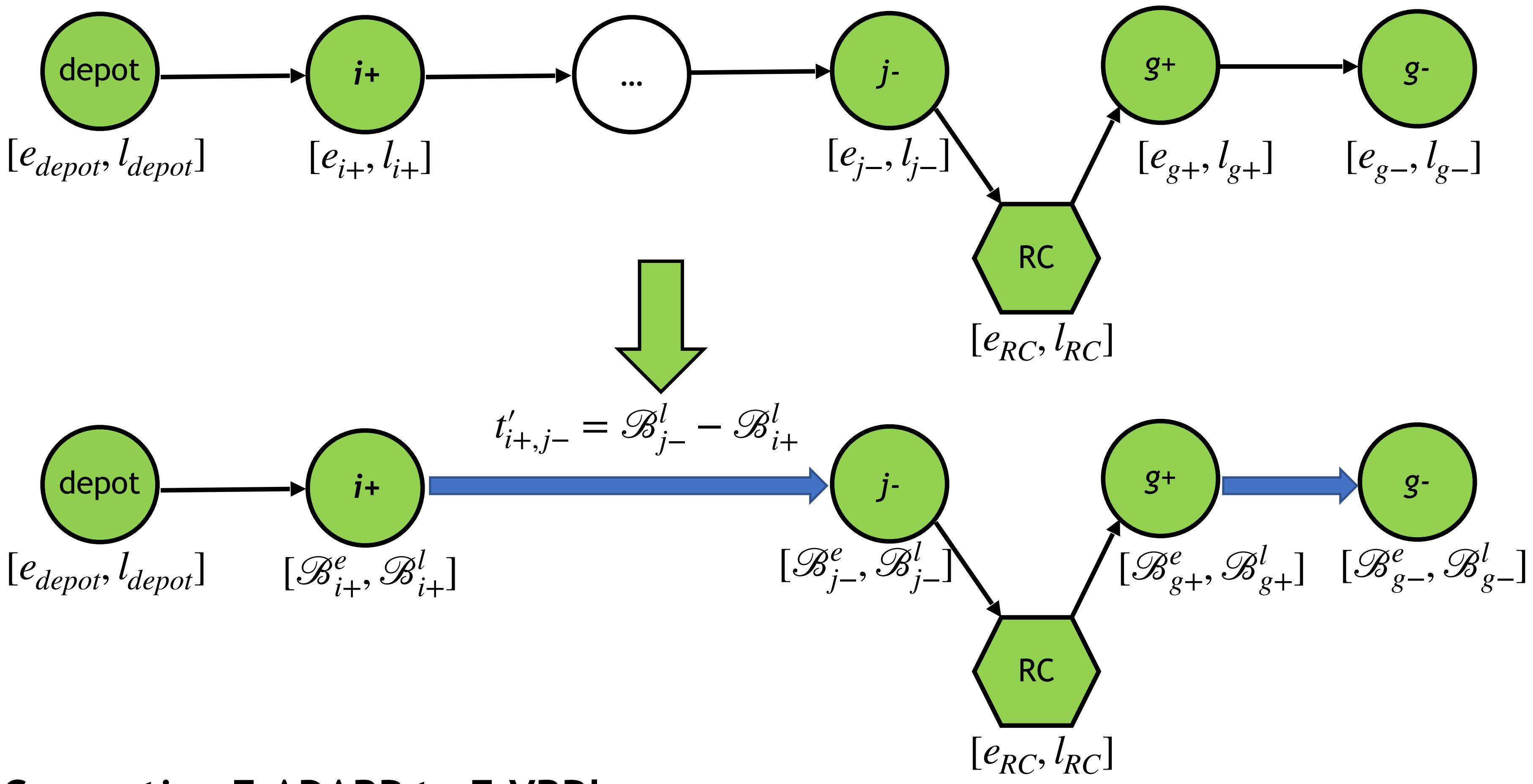


We can transform $\mathcal{S} = \{1, \cdots, m\}$ to an arc $(1, m)$

- The total travel time from 1 to m (denoted as $t'_{1,m}$) is $\mathscr{B}^l_m - \mathscr{B}^l_1$
- The time window of node 1 is $[\mathscr{B}^e_1, \mathscr{B}^l_1]$ and $[\mathscr{B}^e_m, \mathscr{B}^l_m]$
- The energy consumption 1 to m is $\sum_{i=1}^{m-1} h_{i,i+1}$

2.2 Labeling algorithm to solve subproblem

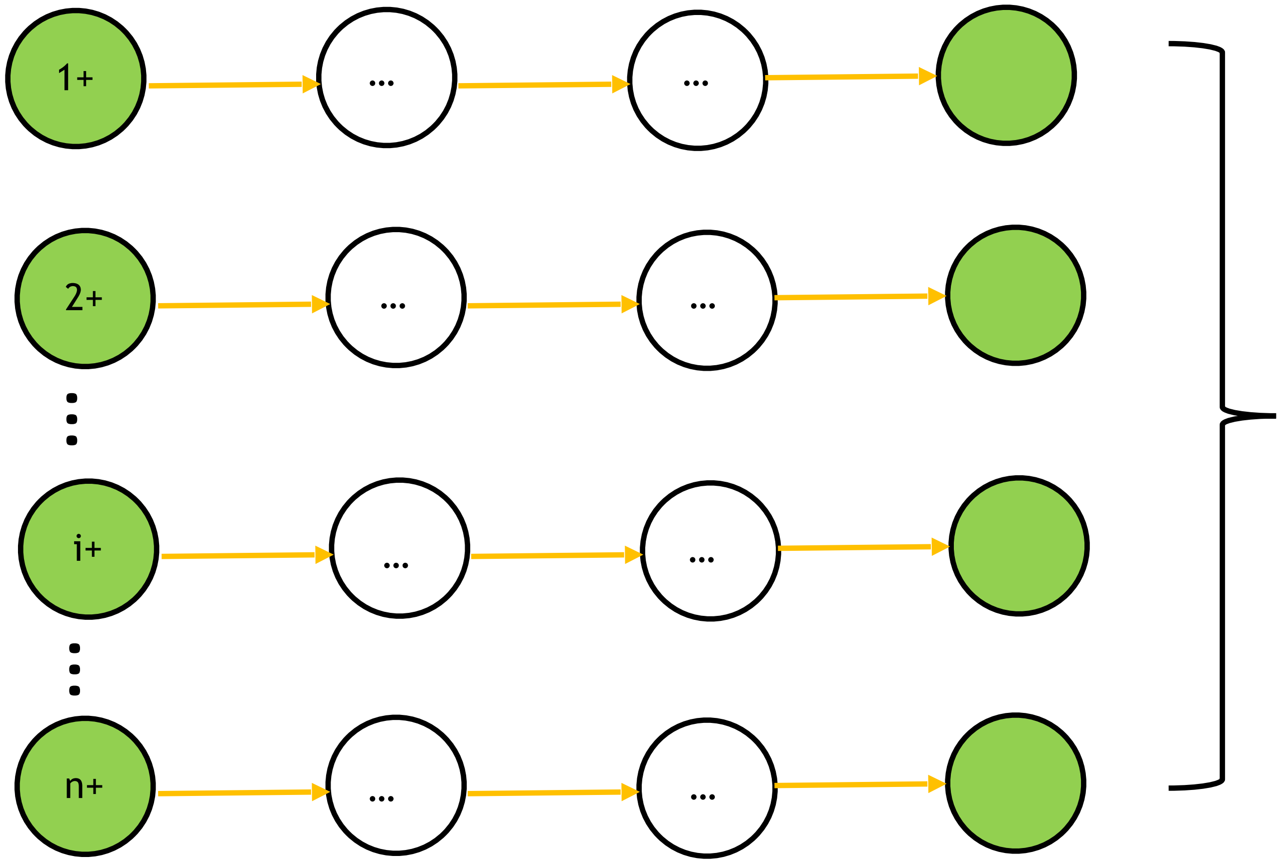
Convert original graph to a sparser graph



Converting E-ADARP to E-VRP!

2.2 Labeling algorithm to solve subproblem

Acceleration strategy: segment concatenation



All the possible sequences for *n* pickup nodes

=> Constant time feasibility checking

2.2 Labeling algorithm to solve subproblem

Label extension, feasibility checking, and dominance rules (based on Desaulniers et al., 2016)

Label definition:

$$L_i = \{T_i^{cost}, (T_j^{rch_s})_{s\in S}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}, T_i^{req}\}$$

Resources extension functions:

$$T_j^{cost} = T_i^{cost} + \bar{c}_{i,j}$$

$$T_j^{rch_s} = T_i^{rch_s} + \begin{cases} 1, & \text{if j = s} \\ 0, & \text{otherwise} \end{cases}$$

$$T_j^{tMin} = \begin{cases} \max\{\mathscr{B}_j^e, T_i^{tMin} + t'_{i,j}\}, & \text{if} T_i^{rch} = \varnothing \\ \max\{\mathscr{B}_j^e, T_i^{tMin} + t'_{i,j}\} + Z_{i,j}, & \text{otherwise} \end{cases}$$

$$T_j^{tMax} = \begin{cases} \min\{\mathscr{B}_j^l, \max\{\mathscr{B}_j^e, T_i^{tMin} + T_i^{rtMax} + t'_{i,j}\}\}, & \text{if} i \in S \\ \min\{\mathscr{B}_j^l, \max\{\mathscr{B}_j^e, T_i^{tMax} + t'_{i,j}\}\}, & \text{otherwise} \end{cases}$$

$$T_j^{rtMax} = \begin{cases} T_i^{rtMax} + h'_{i,j}, & \text{if} T_i^{rch} = \varnothing \\ \min\{H, \max\{0, T_i^{rtMax} - S_{i,j}\} + h'_{i,j}\}, & \text{otherwise} \end{cases}$$

$$T_j^{req} = T_i^{req} \cup U_n(T_j^{tMin})$$

2.2 Labeling algorithm to solve subproblem

Label extension, feasibility checking, and dominance rules

Dominance rule: Let $L^k = \{T_i^{cost}, (T_j^{rch_s})_{s \in S}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}, T_i^{req}\}, k \in 1{,}2$ are two labels

Assume that the partial path associated to L^1 and L^2 are $\mathscr{P}_1$ and $\mathscr{P}_2$, respectively, and $\mathscr{P}_1$, $\mathscr{P}_2$ end at the same nodeLabel 1 dominates label 2 if:

$$T_1^r \leqslant T_2^r, \forall r \in \{cost, rch, tMin\}$$

$$T_1^{req} \subseteq T_2^{req}$$

$$T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}) \leqslant T_2^{rtMax} - (T_2^{tMax} - T_2^{tMin})$$

$$T_1^{rtMax} - (T_2^{tMax} - T_2^{tMin}) \leqslant T_2^{rtMax}$$

3. Computational Results

3.1 Benchmark instances

Existing instances from literature

Adapted instances from standard DARP instances in Cordeau (2006) B&C solution

- With up to 5 vehicles and 50 requests. The largest instance solved optimally by state-of-the-art method **(B&C)** is with 5 vehicles and 40 requests
- Type-a instances

Real-world ride-sharing dataset of Uber Technologies B&C solution

- With up to 5 vehicles and 50 requests
- Type-u instances

Adapted instances from DARP instances in Ropke, Cordeau, and Laporte(2007) DA solution

- With up to 8 vehicles and 96 requests.
- Type-r instances

3.2 Computational results

On type-a instances under low energy restriction (10% energy kept at the end of route)

$\gamma = 0.1$	CG with labeling algorithm				B&P results [b]		Bongiovanni et al., (2019) results [a]			
Instance	I-RMP	*LB*	*LB%*	CPU(s)	*Obj*	N_{node}	*Obj'*	*LB'*	*LB'%*	CPU(s)
a2-16	**237.38***	**237.38**	0	23.11	237.38*	1	237.38*	237.38	0	1.2
a2-20	**279.08***	**279.08**	0	140.63	279.08*	1	279.08*	279.08	0	4.2
a2-24	**346.21***	**346.21**	0	274.17	346.21*	1	346.21*	346.21	0	9.0
a3-18	**236.82***	**236.82**	0	14.21	236.82*	1	236.82*	236.82	0	4.8
a3-24	**274.80***	**274.80**	0	116.37	274.80*	1	274.80*	274.80	0	13.80
a3-30	**413.27***	**413.27**	0	351.47	413.27*	1	413.27*	413.27	0	102
a3-36	**481.17***	**481.17**	0	884.61	481.17*	1	481.17*	481.17	0	106.80
a4-16	**222.49**	222.38	0.05%	7.48	222.49*	9	222.49*	222.49	0	3.6
a4-24	**310.84***	**310.84**	0	28.76	310.84*	1	310.84*	310.84	0	31.2
a4-32	**393.96***	**393.96**	0	311.25	393.96*	1	393.96*	393.96	0	612
a4-40	**453.84***	**453.84**	0	763.99	453.84*	1	453.84*	453.84	0	517.2
a4-48	**554.54***	**554.54**	0	2148.51	554.54*	1	554.54	526.96	5.04%	7200
a5-40	416.79	413.48	0.25%	318.57	414.51*	5	414.51*	414.51	0	1141.8
a5-50	**559.17***	**559.17**	0	1521.55	559.17*	1	559.17	531.15	5.01%	7200
Avg			**0.02%**	**493.19**					0.72%	1210.54

12/14 instances solved optimally at the root node
Lower bound improved by 0.7% on average, 2 lower bounds improved (maximum: 5.04%)
Computational time decreased 59.26% on average

3.2 Computational results

On type-a instances under medium energy restriction (40% energy kept at the end of route)

$\gamma = 0.4$	I-RMP	*LB*	*LB*%	CPU(s)	*Obj*	N_{node}	Obj'	LB'	LB'%	CPU(s)
a2-16	**237.38***	**237.38**	0	40.44	237.38*	1	237.38*	237.38	0	1.8
a2-20	**280.70***	**280.70**	0	134.19	280.70*	1	280.70*	280.70	0	49.8
a2-24	**347.04****	**347.04**	0	535.49	347.04*	1	*348.04**	*348.04*	-0.29%	25.2
a3-18	**236.82***	**236.82**	0	22.75	236.82*	1	236.82*	236.82	0	4.2
a3-24	**274.80***	**274.80**	0	80.98	274.80*	1	274.80*	274.80	0	16.8
a3-30	**413.34****	**413.34**	0	484.56	413.34*	1	*413.37**	*413.37*	-0.01%	99
a3-36	**483.83****	481.85	0.41%	1273.45	482.75*	11	*484.14**	*484.14*	-0.06%	306.6
a4-16	**222.49**	222.38	0.05%	6.49	222.49*	13	222.49*	222.49	0	5.4
a4-24	**311.03***	**311.03**	0	43.16	311.03*	1	311.03*	311.03	0	39.6
a4-32	**394.26***	**394.26**	0	276.85	394.26*	1	394.26*	394.26	0	681.6
a4-40	**453.84***	**453.84**	0	681.77	453.84*	1	453.84*	453.84	0	417.6
a4-48	**554.60***	**554.60**	0	6648.38	554.60*	1	554.60	529.22	4.58%	7200
a5-40	415.25	413.48	0.25%	307.83	414.51*	5	414.51*	414.51	0	1221
a5-50	**559.51***	**559.51**	0	1984.53	559.51*	1	560.50	428.91	23.34%	7200
Avg			**0.05%**	**894.35**					2.02%	1233.47

11/14 instances solved optimally at the root node, 4 new optimal solutions identified
Lower bound improved by 1.97% on average, maximum improvement: 23.34%
Computational time decreased 27.49% on average

3.2 Computational results

On type-a instances under high energy restriction (70% energy kept at the end of route)

$\gamma = 0.7$	I-RMP	LB	$LB\%$	CPU(s)	Obj	N_{node}	Obj'	LB'	$LB'\%$	CPU(s)
a2-16	**240.66***	**240.66**	0	56.13	240.66*	1	240.66*	240.66	0	5.4
a2-20	**293.27**	**292.73**	0.18%	1108.84	293.27*	3	NA	287.17	2.08%	7200
a2-24	**353.18****	**353.18**	0	1057.37	353.18*	1	*358.21**	*358.21*	-1.42%	961.2
a3-18	**240.58***	**240.58**	0	38.57	240.58*	1	240.58*	240.58	0	48
a3-24	**275.97****	**275.97**	0	175.73	275.97*	1	*277.72**	*277.72*	-0.63%	152.4
a3-30	**426.40**	**422.57**	0.56%	1171.62	424.93*	7	NA	417.06	1.85%	7200
a3-36	500.57	**491.80**	0.45%	2191.55	-	-	494.04	485.91	1.65%	7200
a4-16	**223.13***	**223.13**	0	8.36	223.13*	1	223.13*	223.13	0	67.2
a4-24	318.24	316.24	0.13%	84.28	316.65*	3	*318.21**	*318.19*	-0.49%	1834.8
a4-32	**397.87**	**396.84**	0.26%	734.65	397.87*	55	430.07	387.99	2.48%	7200
a4-40	**467.72**	**466.96**	0.16%	3252.36	-	-	NA	443.62	5.15%	7200
a4-48	NA	*476.54*	NA	7200	-	-	NA	524.92	NA	7200
a5-40	**426.38**	**417.25**	2.14%	4461.07	-	-	447.63	405.99	4.78%	7200
a5-50	NA	*176.58*	NA	7200	-	-	NA	522.37	NA	7200
Avg			**0.32%**	**2052.90**					1.70%	4333.50

5/14 instances solved optimally at the root node, 7 new best solutions
Lower bound improved by 1.31% on average, maximum improvement of 4.99%
Computational time decreased 52.63% on average

3.2 Computational results

On type-u instances under low energy restriction (10% energy kept at the end of route)

$\gamma=0.1$	CG+labeling algorithm				Bongiovanni et al., (2019) results [a]			
Instance	I-RMP	*LB*	*LB%*	CPU(s)	*Obj'*	*LB'*	*LB'%*	CPU(s)
u2-16	**57.61**	57.08	0.92%	43.61	57.61*	57.61	0	21
u2-20	**55.59***	**55.59**	0	248.18	55.59*	55.59	0	9.6
u2-24	91.36	90.55	0.79%	1143.90	91.27*	91.27	0	432
u3-18	**50.74***	**50.74**	0	36.56	50.74*	50.74	0	10.8
u3-24	**67.56***	**67.56**	0	106.95	67.56*	67.56	0	130.2
u3-30	**76.75***	**76.75**	0	998.10	76.75*	76.75	0	438
u3-36	104.39	103.94	0.10%	3016.06	104.04*	104.04	0	1084.8
u4-16	**53.58***	**53.58**	0	12.89	53.58*	53.58	0	48
u4-24	**89.83***	**89.83**	0	51.34	89.83*	89.83	0	13.2
u4-32	**99.29***	**99.29**	0	434.30	99.29*	99.29	0	1158.6
u4-40	**133.11***	***131.11***	0	3385.92	133.11*	133.11	0	185.4
u4-48	**148.08**	**147.02**	0.72%	7200	148.30	134.48	9.18%	7200
u5-40	**121.86***	**121.86**	0	1631.99	121.86	114.12	6.35%	7200
u5-50	**142.82**	**142.75**	0.05%	7100.90	143.10	132.69	7.09%	7200
Avg			**0.18%**	1815.05			1.62%	1795.11

10/14 instances solved optimally at the root node, 2 new best solutions
Lower bound improved by 1.44% on average (maximum improvement 8.46%)

3.2 Computational results

On type-u instances under medium energy restriction (40% energy kept at the end of route)

$\gamma = 0.4$	I-RMP	*LB*	*LB*%	CPU(s)	*Obj'*	*LB'*	*LB'*%	CPU(s)
u2-16	**57.65***	**57.65**	0	53.17	57.65*	57.65	0	25.8
u2-20	56.61	56.06	0.50%	407.33	56.34*	56.34	0	12
u2-24	91.62	90.80	0.91%	1140.23	91.63*	91.63	0	757.2
u3-18	**50.74***	**50.74**	0	45.20	50.74*	50.74	0	13.8
u3-24	**67.56***	**67.56**	0	109.48	67.56*	67.56	0	220.8
u3-30	**76.75***	**76.75**	0	912.78	76.75*	76.75	0	336.6
u3-36	**104.06***	**104.06**	0	4922.65	104.06*	104.06	0	2010
u4-16	**53.58***	**53.58**	0	13.63	53.58*	53.58	0	44.4
u4-24	**89.83***	**89.83**	0	56.83	89.83*	89.83	0	28.2
u4-32	**99.29***	**99.29**	0	696.80	99.29*	99.29	0	2667.6
u4-40	**133.78****	133.37	0.31%	2186.06	*133.91**	*133.91*	-0.10%	2653.2
u4-48	**147.63**	***146.37***	0.85%	7200	NA	133.86	9.33%	7200
u5-40	**121.96***	**121.96**	0	1249.12	122.23	112.58	7.69%	7200
u5-50	**142.84**	**142.75**	0.06%	7200	143.14	134.09	6.13%	7200
Avg			**0.19%**	**1870.95**			1.66%	2169.26

9/14 instances solved optimally at the root node, 4 new best solutions
Lower bound improved by 1.47% on average (maximum improvement 8.48%)

3.2 Computational results

On type-u instances under high energy restriction (70% energy kept at the end of route)

$\gamma = 0.7$	I-RMP	LB	$LB\%$	CPU(s)	Obj'	LB'	$LB'\%$	CPU(s)
u2-16	60.01	58.48	1.20%	276.21	59.19*	59.19	0	338.4
u2-20	**56.86***	**56.86**	0	2247.60	56.86*	56.86	0	72
u2-24	**92.17**	**91.48**	0.75%	2056.31	NA	90.83	1.45%	7200
u3-18	**50.99**	50.94	0.10%	65.43	50.99*	50.99	0	24
u3-24	68.44	68.00	0.57%	192.63	68.39*	68.39	0	400.2
u3-30	**77.41****	77.33	0.10%	1202.98	*78.14**	*78.14*	-0.94%	3401.4
u3-36	106.45	**105.46**	0.31%	7200	105.79	104.37	1.34%	7200
u4-16	**53.87***	**53.87**	0	33.77	53.87*	53.87	0	88.8
u4-24	**89.96***	**89.96**	0	82.30	89.96*	89.96	0	22.8
u4-32	**99.50***	**99.50**	0	4559.29	99.50*	99.50	0	2827.2
u4-40	**136.36**	**134.56**	1.32%	2881.51	NA	133.01	2.46%	7200
u4-48	**185.16**	*-216.07*	NA	7200	NA	132.49	NA	7200
u5-40	**124.01**	**122.82**	0.96%	4774.56	NA	109.28	11.88%	7200
u5-50	216.07	*132.79*	8.01%	7200	144.36	133.33	7.64%	7200
Avg			**1.03%**	**2855.19**			1.98%	3598.2

4/14 instances solved optimally at the root node, 5 new best solutions
Lower bound improved by 0.95% on average (maximum improvement: 10.92%)

3.2 Computational results

On type-r instances under different energy restrictions

$\gamma = 0.1$	CG with labeling algorithm			DA 30 runs	
Instance	I-RMP	*LB*	CPU(s)	BKS	$\overline{CPU}$(s)
r5-60	**687.80**	**682.27**	10251.21	691.84	1314.30
r6-48	**506.45***	**506.45**	1109.06	515.01	567.88
r6-60	**692.25**	**689.31**	2389.21	697.67	875.03
r6-72	**761.34**	***748.90***	18000	780.72	1446.90
r7-56	**613.19**	**611.97**	1402.51	617.67	538.80
r7-70	**760.23**	**753.56**	7275.55	777.86	926.47
r7-84	975.26	***638.59***	18000	906.67	1507.28
r8-64	**632.22***	**632.22**	3246.47	654.98	612.44
r8-80	**788.99***	**788.99**	14563.82	814.64	982.31
r8-96	1,329.75	***651.32***	18000	1073.05	1580.94
Avg			9423.78		981.36
$\gamma = 0.4$	I-RMP	*LB*	CPU(s)	BKS	$\overline{CPU}$(s)
r5-60	833.61	**682.76**	15555.25	719.68	1861.60
r6-48	**506.45***	**506.45**	1195.55	509.71	709.77
r6-60	**689.46**	**689.32**	4109.06	703.57	1157.06
r6-72	NA	*-303.92*	18000	836.96	2412.51
r7-56	**612.17**	**611.97**	1496.33	618.46	693.68
r7-70	**759.27**	**753.56**	10380.54	782.84	1207.86
r7-84	1081.76	*-404.31*	18000	1001.10	2090.81
r8-64	**632.22***	**632.22**	3030.01	646.15	764.26
r8-80	**788.99**	***780.81***	18000	833.13	1415.64
Avg			9974.08		1368.13
#opt	*#bestub*	*#bestlb*	$\overline{CPU}$(s)	*#bestub*	$\overline{CPU}$(s)
5	14	17	9698.93	5	1174.74

3.2 Computational results

Performance of algorithm

Performance of CG algorithm on type-a and –u instances

- 50 out of 84 solved optimally at the root node
- Very small deviation (0.31% on average) noticed at the root node => a relatively small number of nodes to search in the B&P tree
- Significant improvement on lower bound: 40 equal lower bounds and 24 better lower bounds. 1.3% increase on lower bound on average
- 22 new best integer solutions reported, 15 are newly-identified optimal solutions
- Shorter computational time than B&C: 30% decrease on average

Performance of CG algorithm on type-r instances

- 14 better integer solutions, 5 are optimal solutions
- 17 first-reported lower bounds

4. Conclusion & extension

4.1 Conclusion & Extension

Conclusion

- New representation of partial path and novel schedule optimization method;
- Efficient CG algorithm with customized labeling algorithm for E-ADARP;
- Significant improvement on lower bound quality and solution quality to state-of-the-art works;
- CG algorithm is proved to be capable of handling large-sized instances

Future works

- Extend CG with labeling algorithm to related topic such as E-PDPTW;
- The novel scheduling optimization method can be used in a multiple-objective optimization context;
- Adapt CG with labeling algorithm to handle dynamic DARP, with requests being renewed

Reference

1. C. Bongiovanni, M. Kaspi, N. Geroliminis, The electric autonomous dial-a-ride problem, Transportation Research Part B: Methodological 122 (2019) 436–456

2. J.-F. Cordeau, A branch-and-cut algorithm for the dial-a-ride problem, Operations Research 54 (3) (2006)573–586
3. J.-F. Cordeau, G. Laporte, 2003 A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological 37(6):579–5

4. G. Desaulniers, F. Errico, S. Irnich, M. Schneider, Exact algorithms for electric vehicle-routing problems with time windows, Operations Research 64 (6) (2016) 1388–1405.

5. S. Ropke, J.-F. Cordeau, G. Laporte, Models and branch-and-cut algorithms for pickup and delivery problems with time windows, Networks: An International Journal 49 (4) (2007) 258–272.
6. Y. Su, J. Puchinger, N. Dupin, 2021 A deterministic annealing local search for the electric autonomous dial-a-ride problem .

Thank you for your attention

Questions ?