UNIVERSITY OF ZAGREB
FACULTY OF TRANSPORT AND TRAFFIC SCIENCES
DEPARTMENT OF INTELLIGENT TRANSPORT SYSTEMS

Electric vehicle routing problem with single or multiple recharges

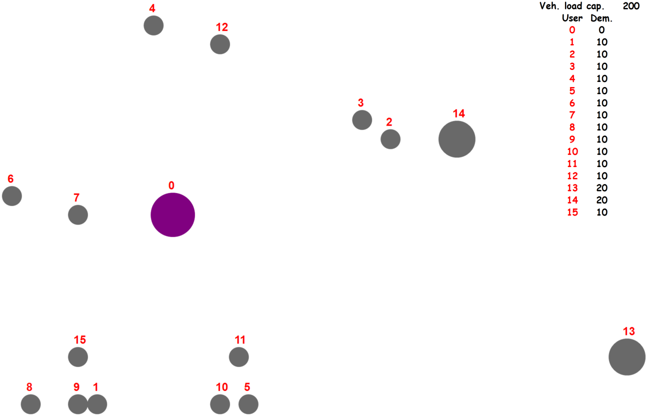
Tomislav Erdelić, Tonči Carić, Martina Erdelić, Leo Tišljarić

13th International Scientific Conference on Sustainable, Modern and Safe Transport (TRANSCOM 2019), High Tatras, Novy Smokovec - Grand Hotel Bellevue, Slovak Republic, May 29-31, 2019

May 29, 2019

Outline

What is VRP?



Veh. load cap.	200
User	Dem.
0	0
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	20
14	20
15	10

Figure: Capacitated VRP - instance *c106C15*, [1]: 15 users

What is VRP?

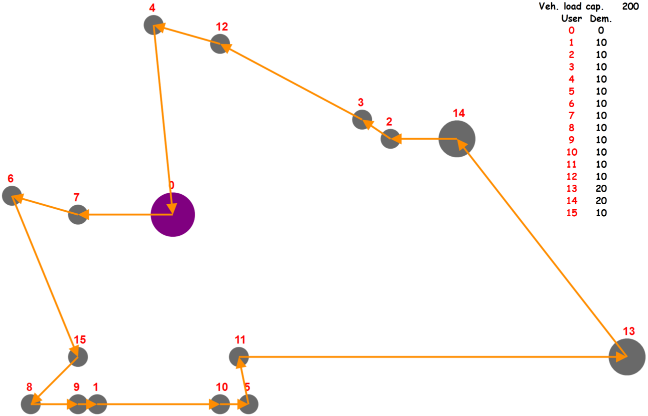
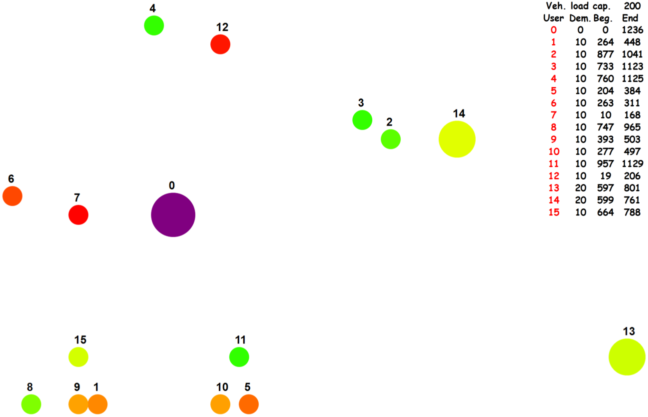


Figure: CVRP: $v = 1,\ t_d = 196.03$

What is VRP?



Veh. load cap.			200
User	Dem.	Beg.	End
0	0	0	1236
1	10	264	448
2	10	877	1041
3	10	733	1123
4	10	760	1125
5	10	204	384
6	10	263	311
7	10	10	168
8	10	747	965
9	10	393	503
10	10	277	497
11	10	957	1129
12	10	19	206
13	20	597	801
14	20	599	761
15	10	664	788

Figure: VRP with time windows - instance *c106C15*, [1]: 15 users

What is VRP?

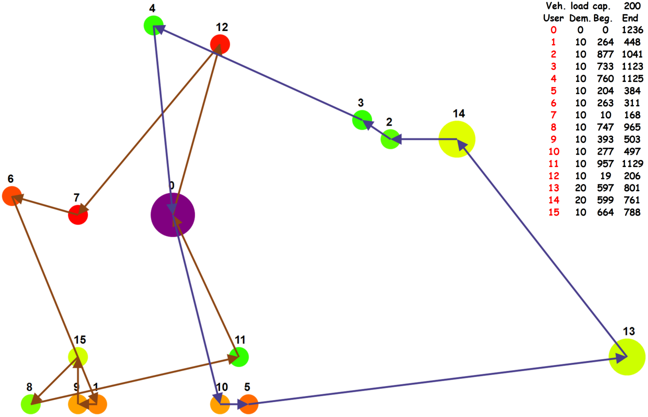


Figure: VRPTW: $v = 2,\ t_d = 274.35$

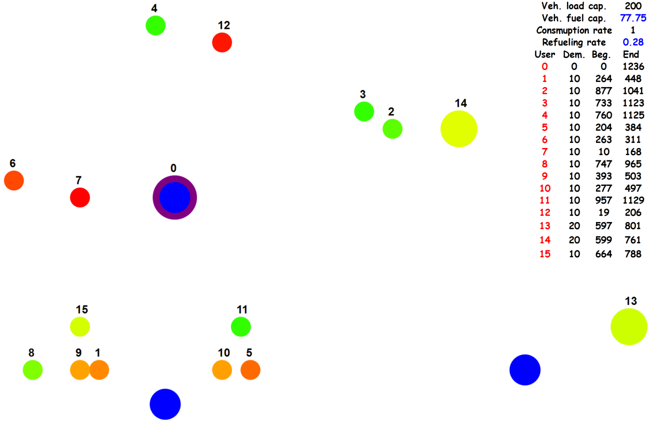
Why electric vehicles?

- Pros
  - Do not have local GHG emission
  - Produce minimal noise
  - Can be powered from renewable energy sources
  - Independent on the fluctuating fossil oil prices
  - Lower maintenance cost

Why electric vehicles?

- Pros
  - Do not have local GHG emission
  - Produce minimal noise
  - Can be powered from renewable energy sources
  - Independent on the fluctuating fossil oil prices
  - Lower maintenance cost
- Cons
  - Battery capacity → Range - 160 - 240 km → operational limitations: frequent visits to charging stations
  - Range anxiety
  - Purchase price
  - Battery lifetime and price

What is E-VRP?



Veh. load cap.			200
Veh. fuel cap.			77.75
Consumption rate			1
Refueling rate			0.28
User	Dem.	Beg.	End
0	0	0	1236
1	10	264	448
2	10	877	1041
3	10	733	1123
4	10	760	1125
5	10	204	384
6	10	263	311
7	10	10	168
8	10	747	965
9	10	393	503
10	10	277	497
11	10	957	1129
12	10	19	206
13	20	597	801
14	20	599	761
15	10	664	788

Figure: Electric VRPTW - instance *c106C15*, [1]: 15 users

What is E-VRP?

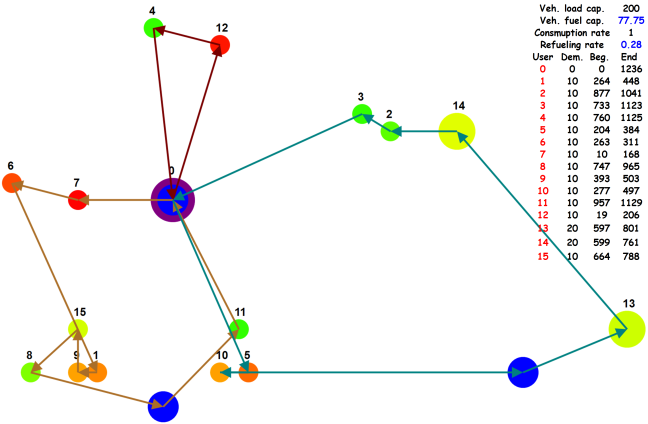


Figure: E-VRPTW: $v = 3,\ t_d = 275.13$

E-VRPTW - Mixed integer linear program

- Schneider et al. [1], formulated E-VRPTW as MILP eqs. (1-9)

Table: E-VRPTW - MILP notations

Name	Description
$V = \{1, \ldots, N\}$	Set of geographically scattered customers
F	Set of CSs for BEVs
F'	Virtual set of CSs
β	Number of virtual CSs per CS
$0,\ N+1$	Depot
$A = \{(i,j) \mid i,j \in V_{0,N+1} \cup F', i \neq j\}$	Set of arcs
$x_{ij} = \{0,1\}$	Binary variable
$C,\ Q$	Load and battery capacity
$r,\ g$	Energy consumption and recharge rate
$s_i,\ [e_i, l_i],\ i \in V_{0,N+1} \cup F'$	Service time and time window
$q_i,\ i \in V_{0,N+1} \cup F'$	Load demand
$\tau_i,\ u_i,\ y_i,\ i \in V_{0,N+1} \cup F'$	Arrival time, remaining load and bat. cap.

E-VRPTW - Mixed integer linear program

$$\min \sum_{j \in V \cup F'} x_{0j} \tag{1}$$

$$\min \sum_{i \in V_0 \cup F', j \in V_{N+1} \cup F', i \neq j} d_{ij} x_{ij} \tag{2}$$

$$(t_{ij} + s_i + l_0) x_{ij} + \tau_i - \tau_j \leq l_0, \forall i \in V_0, \forall j \in V_{N+1} \cup F', i \neq j \tag{3}$$

$$(t_{ij} + l_0 + gQ) x_{ij} - g y_i + \tau_i - \tau_j \leq l_0, \forall i \in F', \forall j \in V_{N+1} \cup F', i \neq j \tag{4}$$

$$e_j \leq \tau_j \leq l_j, \forall j \in V_{0,N+1} \cup F' \tag{5}$$

$$(q_i + C) x_{ij} + u_j - u_i \leq C, \forall i \in V_0 \cup F', \forall j \in V_{N+1} \cup F', i \neq j \tag{6}$$

$$0 \leq u_j \leq C, \forall j \in V_{0,N+1} \cup F' \tag{7}$$

$$(r d_{ij} + Q) x_{ij} + y_j - y_i \leq Q, \forall j \in V_{N+1} \cup F', \forall i \in V, i \neq j \tag{8}$$

$$0 \leq y_j + r d_{ij} x_{ij} \leq Q, \forall j \in V_{N+1} \cup F', \forall i \in 0 \cup F', i \neq j \tag{9}$$

Initial solution

Algorithm 1 *k*-Time Oriented Nearest Neighbor Heuristic ($k - TONNH$)

Open new vehicle and set the current customer i to be the depot
while there are no unserved customers **do**
    Initialize set C_1 with unrouted customers that are reachable from i according to vehicle load capacity, customer time window and depot time window
    Initialize set C_2 to be an empty set
    for each customer j in C_1 **do**
        if there is enough energy to reach j from i and then the depot from j **then**
            Add j to the C_2
        else if the segment from i to j and then from j to the depot is feasible by inserting nearest CS between the depot and i and/or between i and j and/or between j and the depot **then**
            Add j to the C_2 with appropriate placement of CS
        end if
    end for
    if C_2 is not empty **then**
        From k customers in C_2 that minimize the function $\delta_1 d_{ij} + \delta_2 t_{ij}^s + \delta_3 t_{ij}^w$ select one at random, add it to the current vehicle and set the selected customer as customer i
    else
        Close the current vehicle, open new vehicle and set the current customer i to be the depot
    end if
end while

Example - $k = 3$

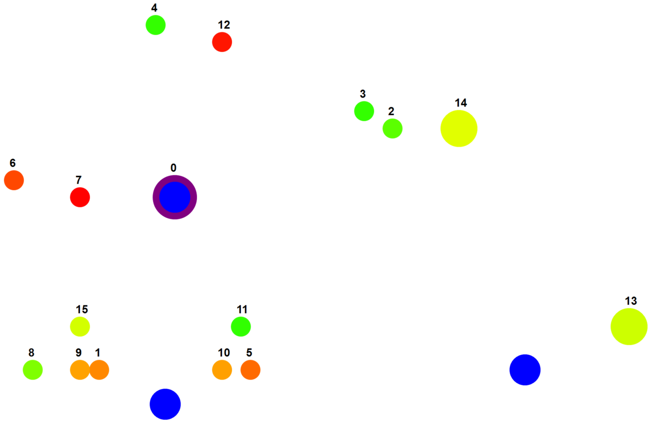


Figure: E-VRPTW - instance *c106C15*, [1]: 15 users

Example - $k = 3$

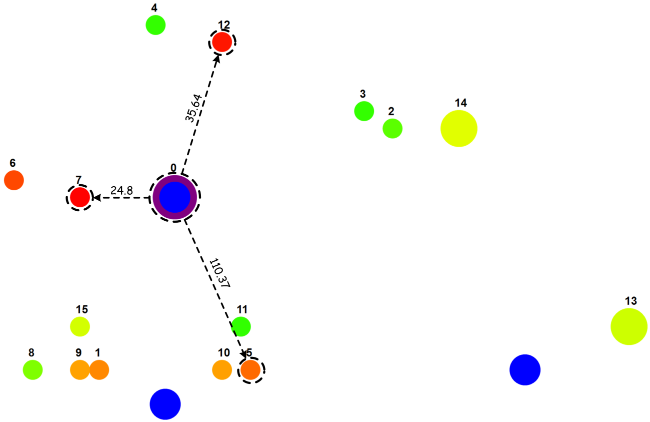


Figure: First step - possible users

Example - $k = 3$

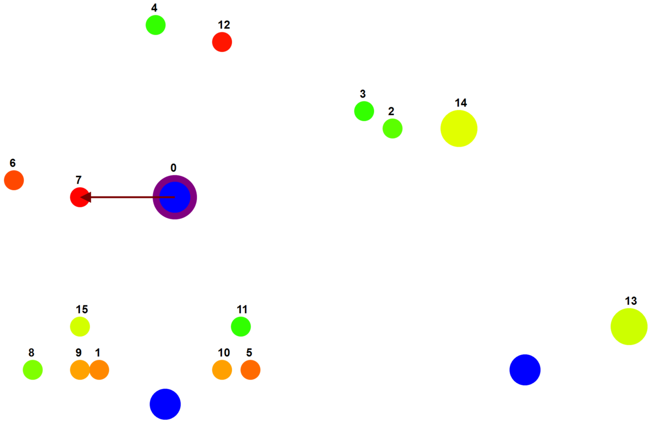


Figure: First step - user added to vehicle

Example - $k = 3$

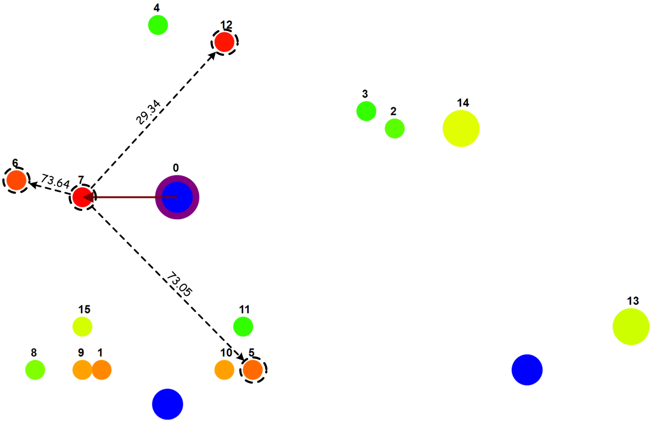


Figure: Second step - possible users

Example - $k = 3$

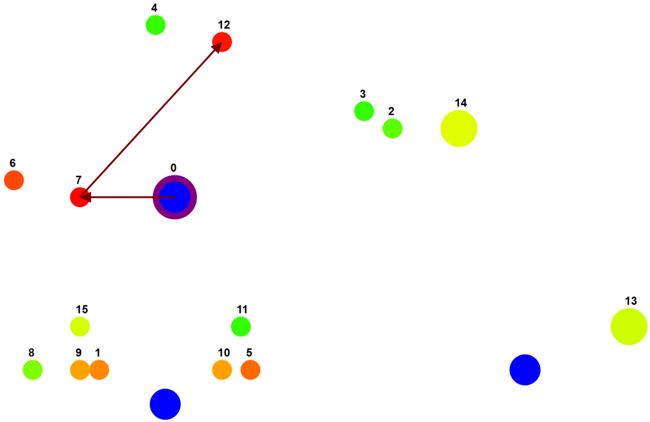


Figure: Second step - user added to vehicle

Example - $k = 3$

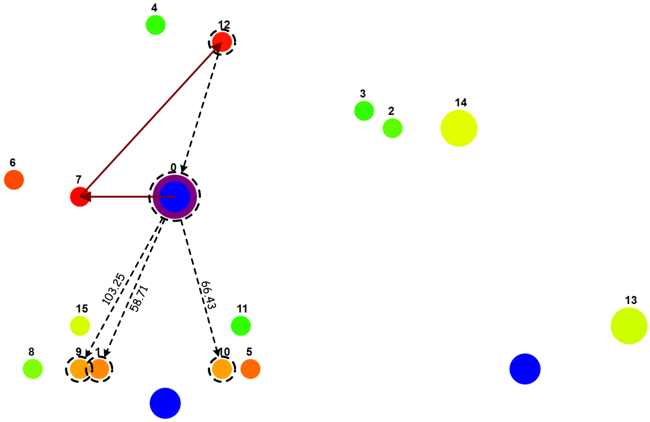


Figure: Third step - possible users

Example - $k = 3$

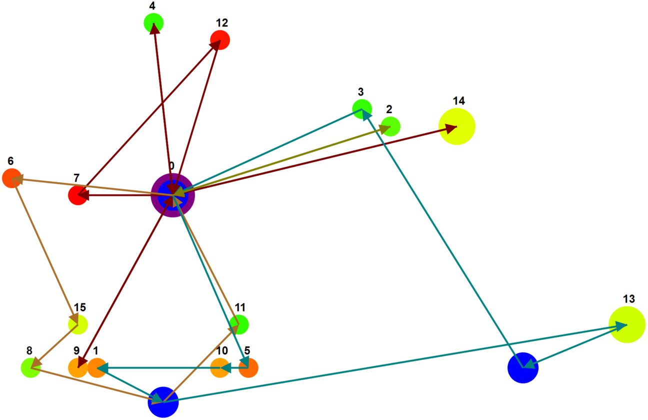


Figure: Final solution: $v = 4$, $t_d = 497.91$

Improvement heuristics

- **Adaptive Large Neighborhood Search (ALNS)**
  - Adaptive removal and insertion of users and stations in the solution - removal and insertion operators

Improvement heuristics

- **Adaptive Large Neighborhood Search (ALNS)**
  - Adaptive removal and insertion of users and stations in the solution - removal and insertion operators

Table: Removal and insertion operators

Removal			Insertion	
User	Station	Route	User	Station
random	*random*	*greedy NRR*	*greedy*	*greedy*
worst distance	*worst distance*		*regret-2*	*best*
worst time			*regret-3*	*greedy with com.*
shaw			*time-based*	
proximity-based			*zone*	
demand-based				
time-based				
zone				
Options: with preceding/succeeding station				

Improvement heuristics

- **Adaptive Large Neighborhood Search (ALNS)**
  - Adaptive removal and insertion of users and stations in the solution - removal and insertion operators
  - The operators selected in next iterations are selected by roulette wheel strategy based on their probability
  - After each iteration for each removal and insertion operator (except the GNRR) scores are added ($\sigma_3 \leq \sigma_2 \leq \sigma_1$):
    - σ_1 if new best solution is found
    - σ_3 if new solution is better than the current solution
    - σ_2 if new solution is worse than the current solution but is accepted due to the simulated annealing acceptance criteria

ALNS - algorithm

Algorithm 2 Adaptive Large Neighborhood Search (ALNS) [2]

while iteration limit not met **do**
  Every N_{RR} iterations select and perform route removal and customer insertion procedures coupled with greedy station insertion to make the solution energy feasible
  Every N_{SR} iterations select and perform station removal and insertion procedures
  Select and perform customer removal
  if partial solution is energy infeasible **then**
    Perform greedy station insertion to make the partial solution energy feasible
  end if
  if partial solution is energy feasible **then**
    Select and perform customer insertion
  end if
  if solution is feasible **then**
    Apply acceptance criteria to accept or reject the solution
  end if
  Every N_C/N_S iterations update operators weights and probabilities
end while

Example

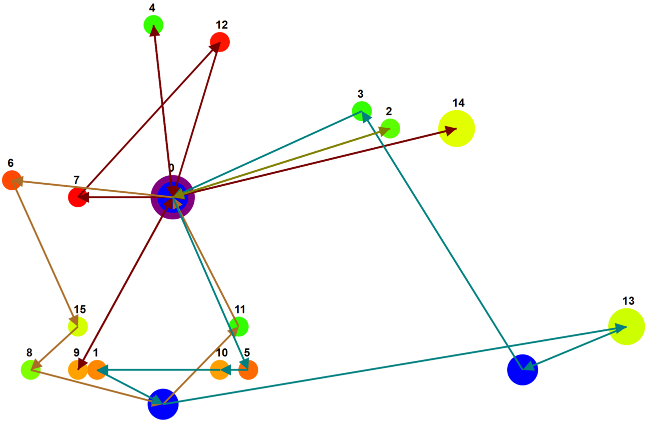


Figure: Initial solution: $M = 4$, $D = 497.91$

Example

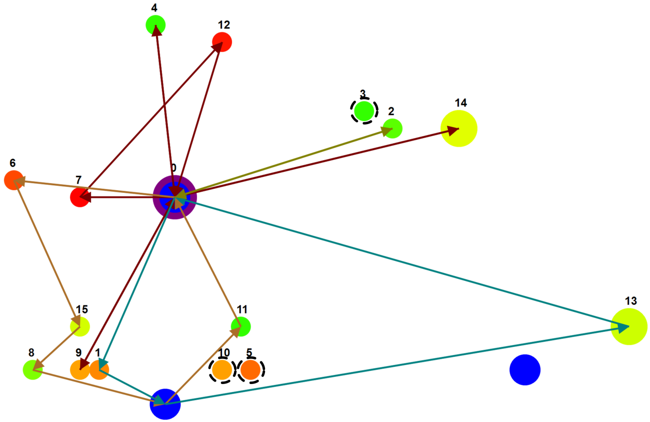


Figure: $v = 4,\ t_d = 463.27$

Example

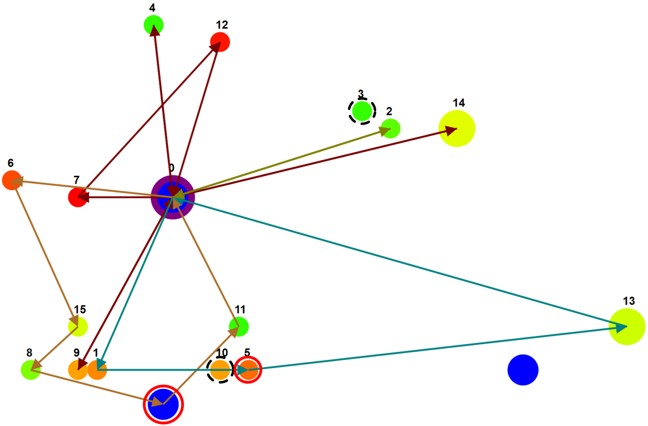


Figure: $v = 4,\ t_d = 461.7$

Example

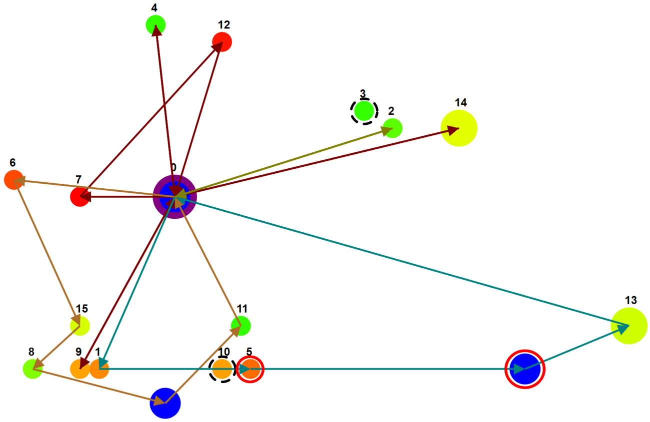


Figure: $v = 4,\ t_d = 462.47$

Example

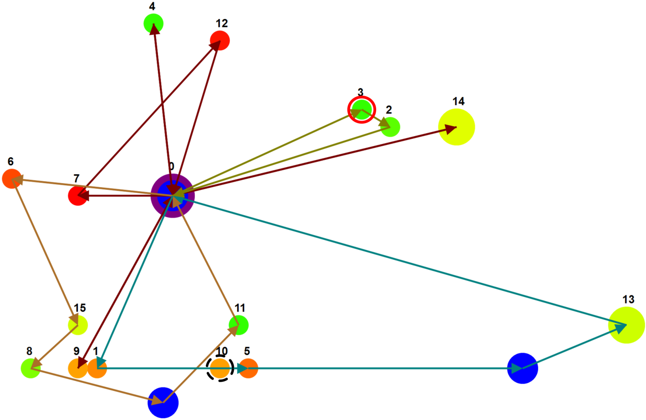


Figure: $v = 4,\ t_d = 464.09$

Example

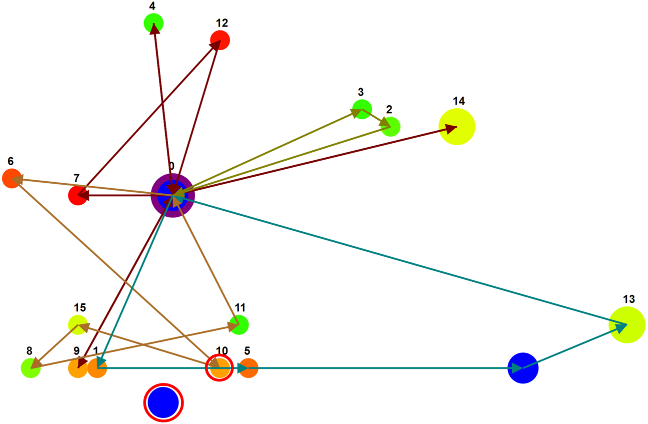


Figure: $v = 4,\ t_d = 488.59$

Example

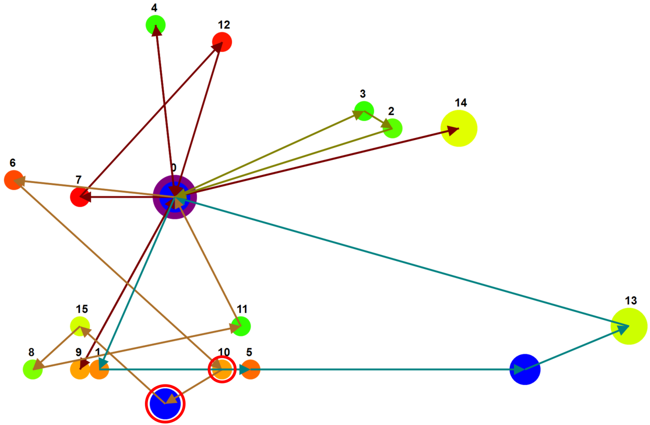


Figure: $v = 4,\ t_d = 492.71$

Benchmark instances

- E-VRPTW instances of Schneider et al. [1]
  - 36 small instances - 5, 10 and 15 users - Exact → MATLAB 2016B (MILP) & ALNS
  - 56 large instances -100 users and 21 stations - ALNS
- Parameters for ALNS are presented in article and [2]
- Observed two different recharge policies
  - **Multiple** recharge policy - allowing multiple recharges during the route, but no more than two consecutive recharges
  - **Single** recharge policy - allowing only one recharge per route

Small instances

- Number of virtual stations (β) significantly influences the problem complexity
- ALNS was able to produce high quality solutions in much lesser time

Table: Results on small E-VRPTW instances

Inst	Schneider et al. [1]			$\beta = 1$				$\beta = 2$			
	v	d	t_d	N_v	N_d	t_v	t_d	N_v	N_d	t_v	t_d
5(12)	17	2.27	29.65	12(11)	12(10)	0.06	0.04	12(12)	12(12)	94.33	11.17
10(12)	25	3.62	175.68	12(10)	12(10)	41.6	2.32	12(9)	10(8)	363.02	542.18
15(12)	32	4.33	1284.61	7(5)	7(4)	384	334.56	—	8(3)	—	540.93

Inst	Schneider et al. [1]			$ALNS_M$				$ALNS_S$			
	v	d	t_d	v	Δ_d	t_{vd}	N_s	v	Δ_d	t_{vd}	N_s
5	17	2.27	29.65	17	0.00	2.82	31	27	7.12	2.75	19
10	25	3.62	175.68	25	0.00	1.76	47	38	2.32	2.13	32
15	32	4.33	1284.61	32	−0.52	2.47	57	41	5.82	1.96	32

Large instances

- Compared to the solutions of Schneider et al. [1]
  - ALNS with multiple recharge policy produced 3.6% more vehicles with difference in total traveled distance within 1%
  - ALNS with single recharge policy produced 17% more vehicles and increased total traveled distance in average by 1.79%

Table: Results on large E-VRPTW instances

Inst	Schneider et al. [1]		$ALNS_M$				$ALNS_S$				$ALNS_M^{init}$	
	v	d	v	Δ_d	t_{vd}	N_s	v	Δ_d	t_{vd}	N_s	v	Δ_d
c1	96	9.43	99	−0.98	85.72	82	110	7.76	58.83	71	172	124.93
c2	32	5.13	32	0.00	90.05	31	32	0.27	62.08	29	44	147.88
r1	154	15.12	161	−1.48	91.42	199	84	7.23	36.92	69	236	56.72
r2	29	10.07	31	−0.91	182.15	42	34	0.22	105.04	31	42	70.69
rc1	105	11.28	108	0.43	62.21	131	—	—	—	—	178	91.52
rc2	25	9.18	26	0.35	128.74	42	34	−6.49	74.28	23	36	64.94

Conclusion & Future research

- Conclusion
  - Number of virtual stations (β) significantly influences the problem complexity when solving the problem exactly
  - The applied ALNS for multiple recharge policy produced high-quality solutions in reasonable time
  - Single recharge policy produced in average one additional vehicle per instance, but with lower number of charging stations visited - better approximates the real-life conditions
- Future research
  - Further improve the metaheuristic efficiency and computation time
  - Selection of available charging technology and partial recharge

The End

Questions?

terdelic@fpz.hr

Literature I

[1] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014. [Online]. Available: https://doi.org/10.1287/trsc.2013.0490

[2] M. Keskin and B. Çatay, "Partial recharge strategies for the electric vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 65, pp. 111 – 127, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X16000322