Solving Large-Scale Dynamic Vehicle Routing Problems with Stochastic Requests

Alexandre Florio

VeRoLog Webinar, March 31, 2022

Solving Large-Scale Dynamic Vehicle Routing with Stochastic Requests in Real-Time

Joint work with Jian Zhang, Kelin Luo and Tom Van Woensel

https://arxiv.org/abs/2202.12983

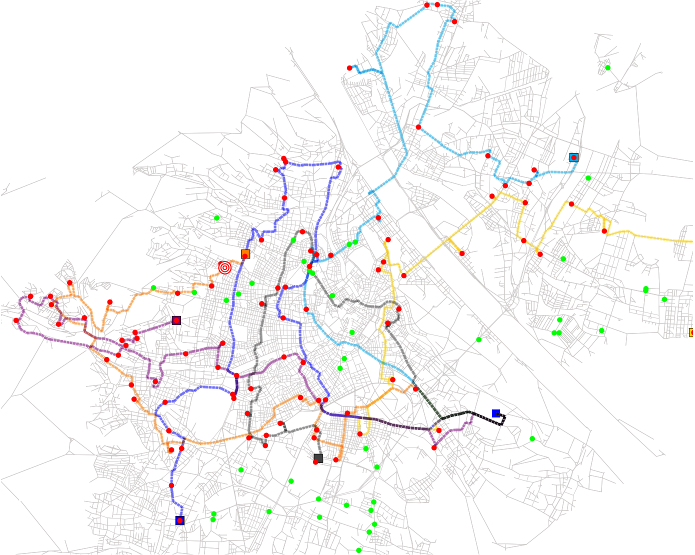
https://github.com/amflorio/dvrp-stochastic-requests

https://youtu.be/D57xNfU73as

General Setting & Motivation

- Dynamic VRP with stochastic requests (DVRPSR)
  - Static (planned) and dynamic (on-demand) requests
  - Applications: technician routing, package collection (first-mile, courier or marketplace sellers), field service management
- **Request:** pickup (or service) location and service time
  - Requests may be rejected
- Multiple drivers/vehicles
- Vehicles must return to the depot by the given deadline
- **Goal:** serve as many requests as possible

General Setting & Motivation



Related Literature

Literature	Problem setting				Initial plan		Online policy	
	RT	Req.	Veh.	Graph	Method	Ant.	Method	Ant.
Bent & Van Hentenryck (2004)	×	100	n/a	Synth.	MSA	✓	MSA	✓
Chen & Xu (2006)	×	100	n/a	Synth.	CGBH	×	CGBH	×
Hvattum et al. (2006)	×	130	n/a	Synth.	DSHH	✓	DSHH	✓
Ichoua et al. (2006)	✓	240	6	Synth.	TS	✓	TS	✓
Thomas (2007)	×	50	1	Synth.	GRASP	×	Heuristics	✓
Azi et al. (2012)	✓	144	5	Synth.	n/a		ALNS	✓
Ferrucci & Bock (2015, 2016)	×	150	12	SSSN	TS	✓	TS	✓
Klapp et al. (2018a)	×	40	1	Synth.	B&C	✓	Rollout	✓
Ulmer et al. (2018a)	×	100	1	Synth.	CI	×	VFA	✓
Ulmer et al. (2018b)	×	100	1	Synth.	CI	×	VFA	✓
Ulmer et al. (2019)	×	100	1	Synth.	CI	×	VFA+rollout	✓
van Heeswijk et al. (2019)	×	400	n/a	Synth., SSSN	CW	×	VFA, rollout	✓
Voccia et al. (2019)	×	192	13	Synth.	n/a		MSA	✓
Ulmer (2020)	✓	180	3	Synth.	n/a		VFA	✓
Ulmer & Thomas (2020)	✓	50	1	Synth.	n/a		VFA	✓
This paper	✓	**947**	**20**	**LSSN**	**PbCGBH**	✓	**PbPs**	✓

(For a review: Soeffker, Ulmer, Mattfeld (2021, EJOR): 10.1016/j.ejor.2021.07.014)

Methodology

- Problem formulation: sequential stochastic optimization model
  - Decisions must be taken “real-time”, each time a request arrives
- Key contribution: approximation of the reward-to-go (potential) by relaxed knapsack models
  - Stochastic lookahead policy with **zero** tunable parameters
- First decision: initial plan: column generation-based petal heuristic
- **Remaining decisions: potential-based scheduling policy**

Notation

$\mathcal{G} = (\mathcal{V}, \mathcal{A})$	Street network graph
$\mathcal{V}$	Road junctions or intersections
$\mathcal{A}$	Road segments
$0 \in \mathcal{V}$	Depot
$t(i, j)$	Duration of the fastest path from node i to node j
$[0, U]$	Service period
K	Fleet size

Notation

$\mathcal{D}(u)$	(Ordered) set of dynamic requests up to instant $u \in [0, U]$
(u, i, d)	Request, where $u \in [0, U]$ is the arrival/release time, $i \in \mathcal{V}$ is the location and $d \in \mathbb{R}_{>0}$ is the service duration
$T \equiv \lvert\mathcal{D}(U)\rvert$	R.v. indicating the total number of dynamic requests

We assume the spatiotemporal probability distribution of dynamic requests can be sampled from (e.g., data is available)

$\omega = \{r_1^\omega, \dots, r_{T_\omega}^\omega\}$ Sample path of the request arrival process

MDP Model: Decision Epochs and States

- Decision epoch: moment when a decision must be made
- In our setting, we have:
  - Initial decision: setup initial routes to serve static/planned requests
  - Following decisions: each time a dynamic request arrives, decide whether to accept or reject the request and, if accept, how to serve it
- Hence, $1 + T$ decision epochs, where T is unknown

MDP Model: Decision Epochs and States

Definition (Route)

A route consists of a sequence of nodes θ that starts and ends at the depot, where each node is associated with a (possibly empty) set of scheduled requests

Definition (Budget of a route θ that started at instant τ)

The budget of a route θ that started at instant τ, $b(\tau, \theta)$, is the slack time relative to the end of the service period (given by U)

MDP Model: Decision Epochs and States

Definition (Vehicle state)

State of vehicle $k \in \{1, \ldots, K\}$ is given by

$$V_k = \begin{cases} \emptyset & \text{if } k \text{ is idle (stationed at the depot)} \\ (\tau_k, \theta_k) & \text{if } k \text{ started to travel along route } \theta_k \text{ at instant } \tau_k \end{cases}$$

Definition (State)

State S_0 (initial state) represents all parameters of the problem

State S_t, $t \in \{1, \ldots, T\}$, is a $(K+1)$-tuple $S_t = (V_1, \ldots, V_K, r_t)$, where V_k are vehicle states, and $r_t = (u_t, i_t, d_t)$ is the t-th element of $\mathcal{D}(U)$

Online Decisions: Scheduling Policies

At each state S_t, $t \in \{1, \dots, T\}$, a scheduling policy prescribes:

1. **Acceptance** decision: accept or reject request r_t
2. **Assignment** decision: if r_t is accepted, assign r_t to a vehicle k
3. **Routing** decision: when r_t is assigned to vehicle k, define how route θ_k is adjusted (in case $V_k = (\tau_k, \theta_k)$) or initialized (in case $V_k = \emptyset$) to accommodate r_t

Online Decisions: Routing Policies

Consider a request $r = (u, i, d)$ and a vehicle state $V = (\tau, \theta)$

Definition (Cheapest Insertion (CI) Routing Policy ρ_{CI})

Routing policy ρ_{CI} inserts r into θ by cheapest insertion, creating a new route $\rho_{\text{CI}}(\theta, r)$

$\rightarrow$ Polynomial time

Definition (Reoptimization Routing Policy ρ_{R})

Routing policy ρ_{R} inserts r into θ, creating a new route $\theta' = \rho_{\text{R}}(\theta, r)$ such that the budget $b(\tau, \theta')$ is maximized

$\rightarrow$ NP-hard

Online Decisions: State Transitions

- Let $\mathcal{X}_t$ be the set of possible decisions when at state S_t
- A policy π maps each possible state S_t to a decision $X^\pi(S_t) \in \mathcal{X}_t$
- State transition:

$$S_{t+1} = S^M(S_t, x_t, r_{t+1})$$

- Decisions must be taken in "real-time" (i.e., in seconds, not minutes)

MDP Model: Rewards and Objective

Rewards:

$$R(S_t, x) = \begin{cases} 1 & \text{if } x \in \mathcal{X}_t \text{ is an `accept' decision} \\ 0 & \text{otherwise} \end{cases}$$

Objective:

$$\max_{\pi \in \Pi, y \in \mathcal{F}} \mathbb{E}\left[\mathbb{E}\left[\sum_{t=1}^{T} R(S_t, X^{\pi}(S_t)) \middle| S_1 \right] \right]$$

where Π is the set of feasible policies and $\mathcal{F}$ is the set of feasible initial plans

Methodology

Online decisions:

- **Potential-based policy (PbP)**
- **Simplified PbP (S-PbP)**

Offline decisions:

- Myopic plan
- **Potential-based plan**

Potential of a State

Given a policy π, the potential of a state S_t is the expected reward-to-go:

$$\Phi_\pi(S_t) = \mathbb{E}\left[\sum_{t'=t}^{T} R(S_{t'}, X^\pi(S_{t'}))\middle| S_t\right] \qquad t \neq 0$$

Given an approximation function $\hat{\Phi}_{\pi^*}(S_t) \approx \Phi_{\pi^*}(S_t)$ and a set of candidate decisions $\tilde{\mathcal{X}}_t$, we prescribe decision

$$X^\pi(S_t) = \underset{x_t \in \tilde{\mathcal{X}}_t}{\arg\max}\ R(S_t, x_t) + \mathbb{E}\left[\hat{\Phi}_{\pi^*}(S_{t+1})\middle| x_t\right]$$

Towards an Approximation Model: Challenges

Goal:

- Estimate the potential $\Phi_{\pi^*}(S_t)$ of the optimal policy π^* for any given state S_t

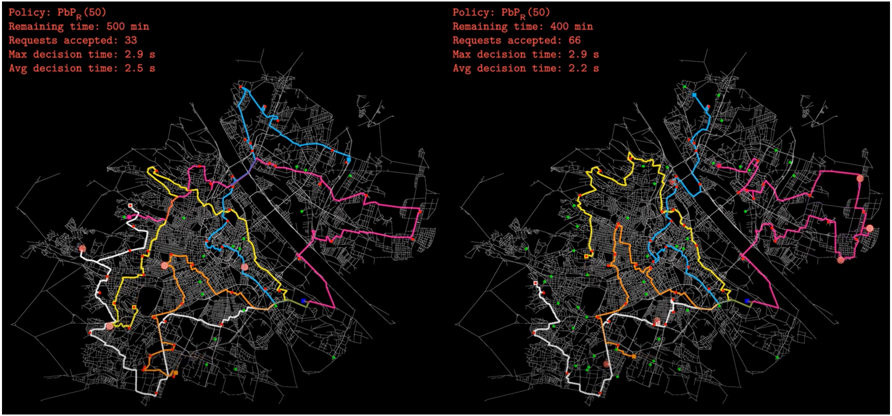
What we have available:

- We can sample trajectories $\Omega = \{\omega_1, \ldots, \omega_H\}$ from the spatiotemporal request distribution

Two main challenges must be overcome:

- How to estimate the minimum budget required for serving a (stochastic) future request?
- How to handle the "competition" of vehicles to serve requests?

Towards an Approximation Model: Challenges



Towards an Approximation Model: Two Remarks

Remark 1 (Late Depot Arrival)

Since the goal is to maximize the number of accepted requests, drivers return to the depot at an instant near the end of the service period

Remark 2 (Request Order Preservation)

The relative order of scheduled requests does not change (CI policy), or changes only slightly (reopt. policy), when a new request is assigned to a route

Modeling Insight: Effective Vehicle Speed

Remarks 1 and 2 suggest the concept of **effective speed**:

Definition (Effective Vehicle Speed)

At instant u, the effective speed of a vehicle k with state $V_k = (\tau_k, \theta_k)$ is the speed such that k arrives at the depot exactly at instant U, provided that V_k never changes

To predict the location of vehicle k at a future instant $u' > u$, we simulate route θ_k under the effective speed for an amount of time $u' - u$

Modeling Insight: Assignment Costs

We now have a good prediction of where vehicle k will be at any future instant u'. Next, we determine:

$\mathcal{V}_{u_t}(V_k, u')$ (Predicted) set of nodes along θ_k not yet traversed by vehicle $V_k = (\tau_k, \theta_k)$ at instant u'

Finally: the min cost (or budget consumption) when assigning a future request $r' = (u', i', d')$ to vehicle k is approximated by:

$$c_{u_t}(V_k, r') = \min_{j \in \mathcal{V}_{u_t}(V_k, u')} t(j, i') + t(i', j) + d'$$

Multiple-Knapsack Approximation of the Potential

Given H sample paths $\Omega = \{\omega_1, \ldots, \omega_H\}$ for the remaining horizon:

$$\mathbb{E}\left[\hat{\Phi}_{\pi^*}(S_{t+1})\middle| x_t\right] \approx \frac{1}{H} \sum_{\omega \in \Omega} \phi^{\omega}(S_{t+1}|x_t)$$

where

$$\begin{aligned}
\phi^{\omega}(S_{t+1}|x_t) = \max \quad & \sum_{k \in \overline{K}} \sum_{r \in \omega} z_{kr} & \\
\text{s.t.} \quad & \sum_{r \in \omega} c_{u_t}(V_k, r) z_{kr} \leq b(\tau_k, \theta_k) & k \in \overline{K} \\
& \sum_{k \in \overline{K}} z_{kr} \leq 1 & r \in \omega \\
& 0 \leq z_{kr} \leq 1 & k \in \overline{K}, r \in \omega
\end{aligned}$$

Potential-based Policy

Given a routing policy $\rho \in \{\rho_{\mathrm{CI}}, \rho_{\mathrm{R}}\}$. Upon arrival of $r_t = (u_t, i_t, d_t)$:

1. Initialize a set of candidate decisions $\tilde{\mathcal{X}}_t = \{x_t^-\}$ with the 'reject' decision x_t^- related to request r_t
2. If it is feasible to serve r_t with vehicle k under the given routing policy, add the corresponding decision x_t^k to set $\tilde{\mathcal{X}}_t$
3. Compute $\hat{\Phi}_{\pi^*}(S_{t+1}|x_t)$ for each $x_t \in \tilde{\mathcal{X}}_t$, and select the decision with highest total (expected) reward

Potential-based Policy: Discussion

Pros:

- Accuracy: MPE of $\pm 2.5\%$ on most instances, from $u = 0$
- Fast: requires the solution of $H(K+1)$ linear programs (LPs)
- **Zero** tunable parameters

Cons:

- Not **so** fast: for very large K and very high request rate, it takes a while to solve all LPs
- The simplified PbP (S-PbP) trades-off accuracy by efficiency

What sort of policy is that?

- `http://tinyurl.com/Powelllookaheadpolicies`
- Stochastic lookahead policy with sampling, stage aggregation, latent variables and policy approximation

Benchmark Policies

- Greedy policies: GP_{CI} and GP_R
  - Accept all feasible requests; with and without reoptimization
- Rollout: R_{CI}-$GP_{CI}(H)$ and R_R-$GP_{CI}(H)$
  - H sample paths, GP_{CI} as base policy
- Policy function approximation: PFA_{CI} and PFA_R
  - Single parameter (trained offline) trades off immediate reward and reward-to-go
- Rollout: R_{CI}-$PFA_{CI}(H)$ and R_R-$PFA_{CI}(H)$
  - H sample paths, PFA_{CI} as base policy

Computational Study

- Real street network of Vienna (16,080 nodes and 36,424 arcs)
- Service period: 10 hours
- Number of vehicles: $K \in \{2, 3, 5, 6, 10, 12, 20\}$
- Request rate (per minute): $\Lambda \in \{0.2, 0.4, 0.8, 1.5\}$
  - Smallest instances larger than most instances from previous works
- Degree of dynamism: $\eta \in \{75\%, 85\%, 90\%, 95\%\}$
- Three spatiotemporal request distributions:
  - Uniform, time-independent (UTI)
  - Clustered, time-independent (CTI)
  - Clustered, time-dependent (CTD)

Instances, Policies and Offline Planners Simulated

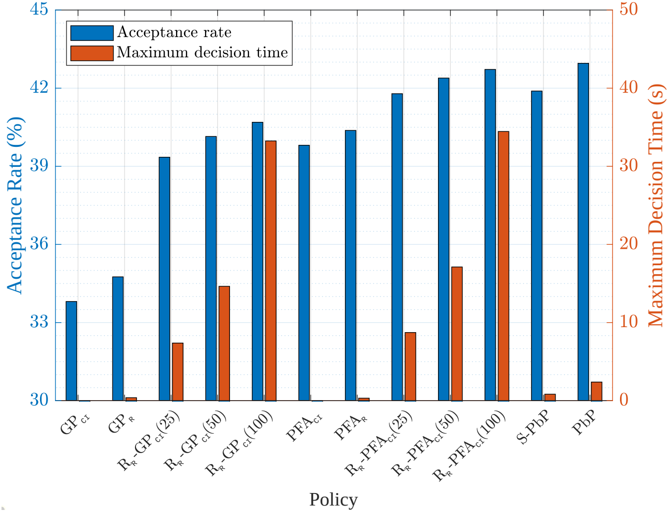
Instance parameters					Algorithms		Simulations
Λ	η	K	Dist.	Scen.	Offline	Online	
0.2	0.75	2, 3	UTI CTI CTD	5	MY PB	$\mathrm{GP}_{\mathrm{CI}}$, GP_{R}, R_{CI}-$\mathrm{GP}_{\mathrm{CI}}$(10, 25, 50, 100), R_{R}-$\mathrm{GP}_{\mathrm{CI}}$(10, 25, 50, 100), $\mathrm{PFA}_{\mathrm{CI}}$, $\mathrm{PFA}_{\mathrm{R}}$, R_{R}-$\mathrm{PFA}_{\mathrm{CI}}$(25, 50, 100)*, S-PbP, PbP	12,270
0.4	0.85	3, 5	UTI CTI CTD	5	MY PB	$\mathrm{GP}_{\mathrm{CI}}$, GP_{R}, R_{CI}-$\mathrm{GP}_{\mathrm{CI}}$(10, 25, 50, 100), R_{R}-$\mathrm{GP}_{\mathrm{CI}}$(10, 25, 50, 100), $\mathrm{PFA}_{\mathrm{CI}}$, $\mathrm{PFA}_{\mathrm{R}}$, R_{R}-$\mathrm{PFA}_{\mathrm{CI}}$(25, 50, 100)*, S-PbP, PbP	12,270
0.8	0.90	6, 12	UTI CTI CTD	5	PB	$\mathrm{GP}_{\mathrm{CI}}$, GP_{R}, $\mathrm{PFA}_{\mathrm{CI}}$, $\mathrm{PFA}_{\mathrm{R}}$, R_{CI}-$\mathrm{GP}_{\mathrm{CI}}$(10), R_{R}-$\mathrm{GP}_{\mathrm{CI}}$(10), R_{R}-$\mathrm{PFA}_{\mathrm{CI}}$(10), S-PbP, PbP	3,210
1.5	0.95	10, 20	UTI	1**	PB	$\mathrm{GP}_{\mathrm{CI}}$, GP_{R}, $\mathrm{PFA}_{\mathrm{CI}}$, $\mathrm{PFA}_{\mathrm{R}}$, R_{CI}-$\mathrm{GP}_{\mathrm{CI}}$(10), R_{R}-$\mathrm{GP}_{\mathrm{CI}}$(10), R_{R}-$\mathrm{PFA}_{\mathrm{CI}}$(10), S-PbP, PbP	214

Potential-based vs Myopic Plans

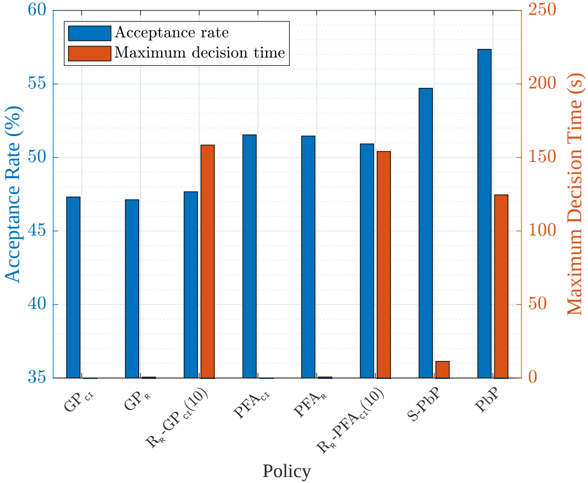
Average number of accepted requests:

Policy	Myopic Plan	Potential-based Plan	Diff
GP_{CI}	54.1	61.2	13.0%
GP_{R}	54.9	62.7	14.1%
R_{CI}-GP_{CI}(10)	63.0	66.2	5.1%
R_{CI}-GP_{CI}(25)	65.8	69.4	5.4%
R_{CI}-GP_{CI}(50)	67.3	71.1	5.6%
R_{CI}-GP_{CI}(100)	68.2	72.1	5.8%
R_{R}-GP_{CI}(10)	63.4	67.2	6.1%
R_{R}-GP_{CI}(25)	66.2	70.7	6.7%
R_{R}-GP_{CI}(50)	67.7	72.2	6.8%
R_{R}-GP_{CI}(100)	68.5	73.3	7.0%
S-PbP(50)	72.0	75.2	4.4%
PbP(50)	**72.9**	**77.2**	**5.9%**
PFA_{CI}	61.3	71.8	17.1%
PFA_{R}	62.1	72.6	16.9%
Avg	66.4	72.4	9.0%

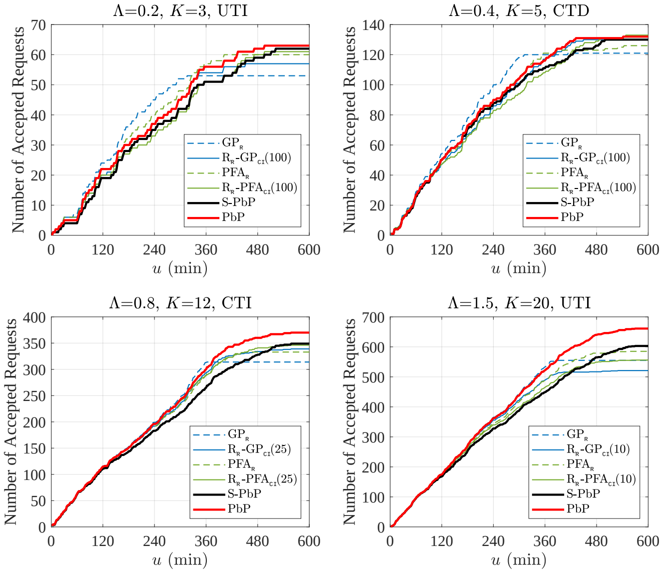
Policy Comparison ($\Lambda \in \{0.2, 0.4\}$)



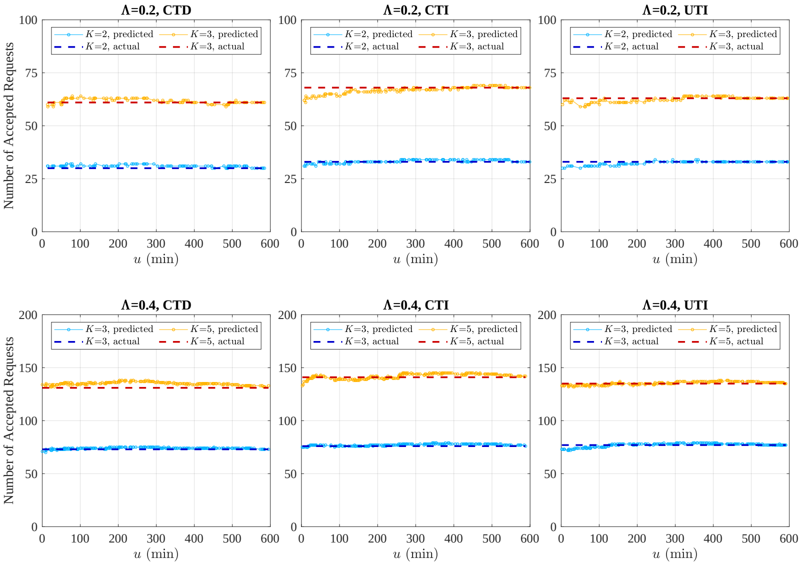
Policy Comparison ($\Lambda \in \{0.8, 1.5\}$)



Acceptance Profiles



Multiple-Knapsack Potential Approximations



Conclusions

- Main contributions and takeaways:
  - Expected reward-to-go can be accurately and efficiently approximated by knapsack models
  - Accurate potential approximations enable high-performing scheduling policies, which outperform classical ADP methods traditionally used for DVRPs such as rollout algorithms and PFA
  - Coverage of the service area is more important than budget alone

- Possible extensions and future research:
  - Vehicle capacity, time windows, (self-imposed) time window assignments
  - Pickup and delivery
  - Reassignment of requests among vehicles (more complex)

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754462

Computational experiments were performed on the Dutch national e-infrastructure with the support of SURF Cooperative

Geographical data for Vienna are copyrighted to OpenStreetMap contributors and available at http://openstreetmap.org

Questions & Discussions