An iterative sample scenario algorithm for the dynamic dispatch waves problem

Leon Lan[1*] Jasper van Doorn[1]

Niels Wouda[2] Arpan Rijal[2] Sandjai Bhulai[1]

[1] Vrije Universiteit Amsterdam, The Netherlands
[2] University of Groningen, The Netherlands

* Presenting author

July 28, 2023

Same-day delivery

[1]

[1] https://nl.wikipedia.org/wiki/Flitsbezorger

Same-day delivery

- Applications in e-commerce, meal delivery, and urban consolidation centers.
- Same-day delivery problems are dynamic vehicle routing problems with stochastic *delivery* requests: goods must be delivered from a central depot.
- 90% of the studies on same-day delivery problems have been published in the last five years.[2]

---

2 J. Zhang and T. V. Woensel (2023). "Dynamic Vehicle Routing with Random Requests: A Literature Review". en. In: *International Journal of Production Economics* 256, p. 108751

Key points of this talk

- We formulate the *dynamic dispatch waves problem* (DDWP)[3] introduced during the EURO meets NeurIPS 2022 vehicle routing competition.[4]
- We propose an iterative method based on solving sampled scenarios to tackle this problem.
- Our results show that the method is scalable and overcomes the large computational efforts often associated with sampling-based methods.

---

3 M. A. Klapp, A. L. Erera, and A. Toriello (2018). "The One-Dimensional Dynamic Dispatch Waves Problem". In: *Transportation Science* 52.2, pp. 402–415

4 W. Kool, D. Numeroso, R. Reijnen, R. R. Afshar, T. Catshoek, K. Tierney, E. Uchoa, and J. Gromicho (2022). *EURO Meets NeurIPS 2022 Vehicle Routing Competition*.

Dynamic dispatch waves problem

- Time horizon of H units divided into $\mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$ epochs.
- Each epoch $t \in \mathcal{T}$ starts at time $T_t \geq T_0 = 0$, with $T_t > T_{t-1}$.
- At the start of epoch $t \in \mathcal{T}$, a set of requests ω_t with known support Ω_t is revealed.
- A delivery request n has a location, a demand, a hard time window $[e_n, l_n]$, and a release time $r_n = T_t$.
- Unlimited fleet of homogeneous vehicles is available.

Dynamic dispatch waves problem

- In each epoch $t \in \mathcal{T}$, we have to decide which of the known requests to *dispatch*, and how to route them, and which ones to *postpone* to later epochs.
- **Goal**: deliver all requests within their time windows at minimum total traveling distance.

Markov decision process

- **State**: Define the *state* s_t as

$$s_t = \{n \in \mathcal{N}_t \mid \text{request } n \text{ not yet dispatched}\},$$

where $\mathcal{N}_t = \cup_{t'=1}^{t} \omega_{t'}$ denotes the set of known requests at epoch t.

- **Action**: Define the *action* space $\mathcal{A}(s_t)$ as

$$\mathcal{A}(s_t) = \{a_t \subseteq s_t \mid m_t \subseteq a_t\},$$

where $m_t \subseteq s_t$ denotes the set of must-dispatch requests.

- **Cost**: The direct cost $C(s_t, a_t)$ is given by the cost of routing all dispatched requests a_t, i.e., the optimal cost of the VRPTW with release times and departure time T_t.

Markov decision process

- **Transition:** The transition to the next state s_{t+1} is given by

$$s_{t+1} = (s_t \setminus a_t) \cup \omega_{t+1}.$$

- **Optimality equation:** The objective of the DDWP is to select for each epoch $t \in \mathcal{T}$ a minimum cost action, such that the following Bellman optimality conditions are satisfied:

$$V(s_t) = \begin{cases} C(s_t, s_t) & \text{if } t = |\mathcal{T}|, \\ \min\limits_{a \in \mathcal{A}(s_t)} \left[C(s_t, a) + \mathbb{E}_{\omega_{t+1}} [V(s_{t+1})] \right] & \text{otherwise.} \end{cases} \tag{1}$$

Sample scenario methods

- Main idea of sample scenario methods:
    - Sample a set of future scenarios.
    - Solve each scenario as a static and deterministic problem.
    - Use solutions to derive an action a_t.
- Advantage: Useful when decision space requires much detail.[5]
- Drawbacks: Computationally expensive to solve many large instances.

5 N. Soeffker, M. W. Ulmer, and D. C. Mattfeld (2022). "Stochastic Dynamic Vehicle Routing in the Light of Prescriptive Analytics: A Review". en. In: *European Journal of Operational Research* 298.3, pp. 801–820

Iterative sample scenario methods

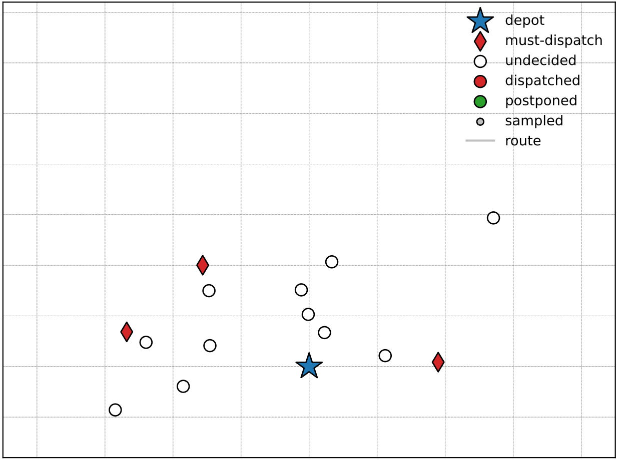
- What if we instead *incrementally* build an action to minimize computational challenges?
- Similar idea: dynamic stochastic hedging heuristic[6] for a dynamic pickup-and-delivery problem.

6 L. M. Hvattum, A. Løkketangen, and G. Laporte (2006). "Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic". In: *Transportation Science* 40.4, pp. 421–438

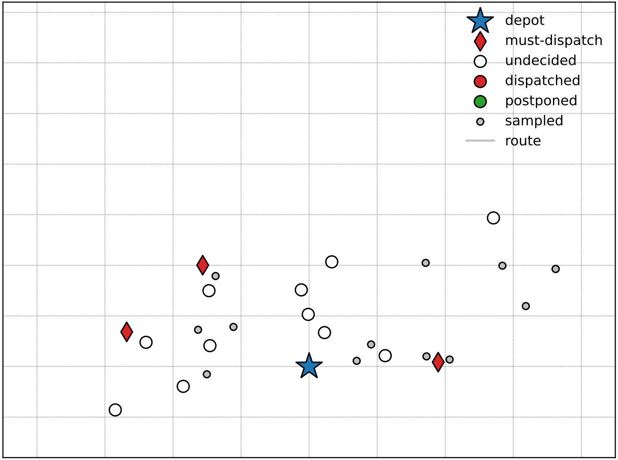
Iterative conditional dispatch

- Let d_t denote the set of dispatched requests and let p_t denote the set of postponed requests.
- **Initialize** $d_t = m_t$ and $p_t = \emptyset$.
- **Repeat** for a fixed number of iterations:
  - **Step one**: Solve $|\mathcal{S}|$ sample scenarios conditioned on d_t and p_t.
  - **Step two**: Classify undecided requests into either d_t, p_t, or leave undecided.
- **Return** action $a_t = d_t$.

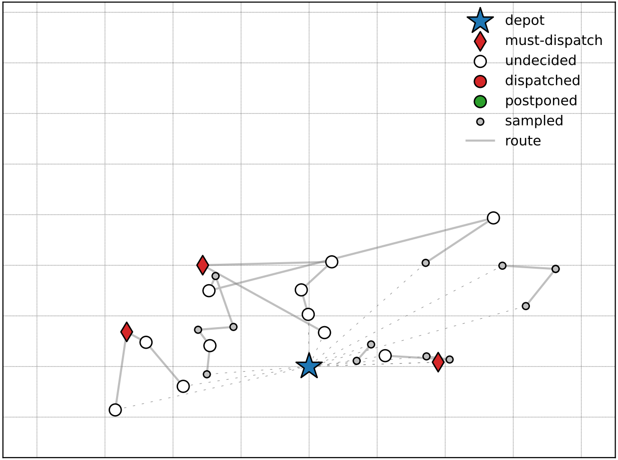
Epoch instance



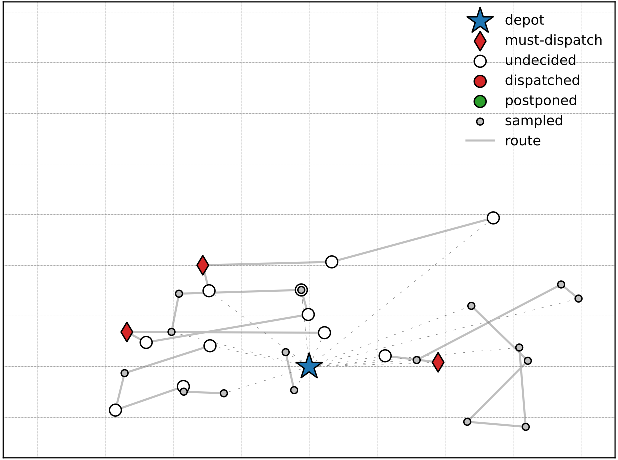
Scenario instance #1



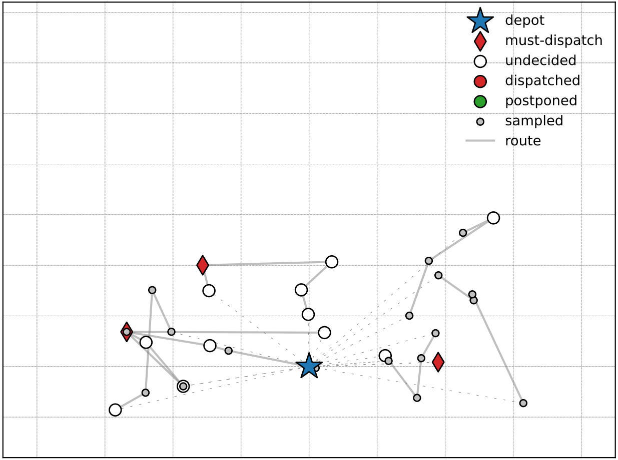
Scenario solution #1



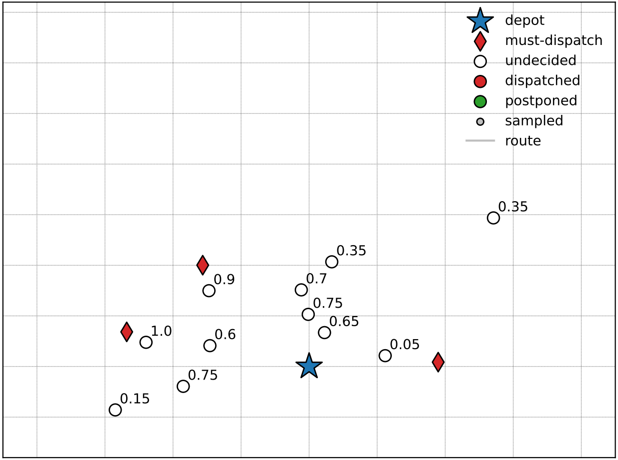
Scenario solution #2



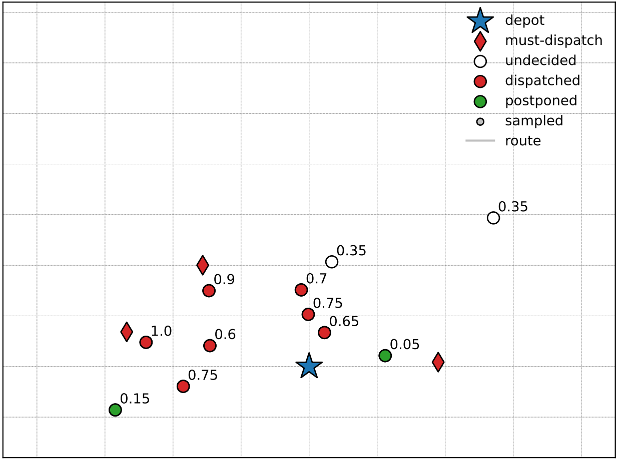
Scenario solution #3



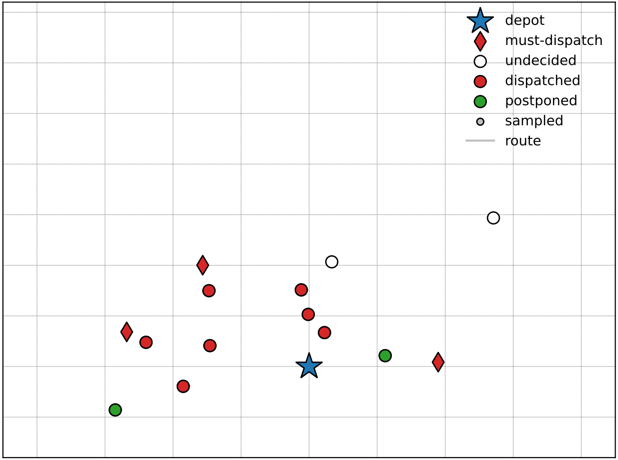
Dispatch probability



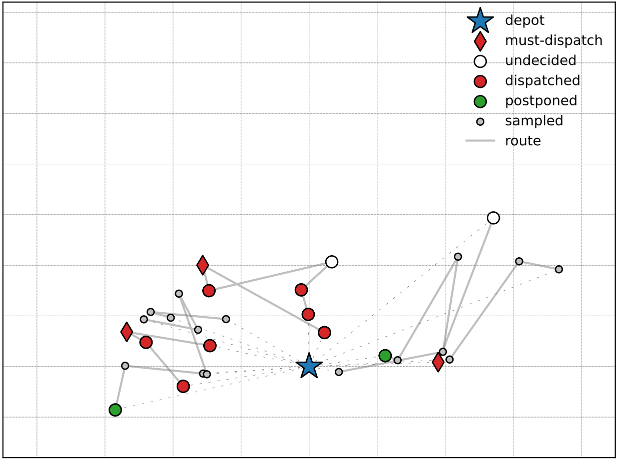
Dispatch if score ≥ 0.50 and postpone if score ≤ 0.20



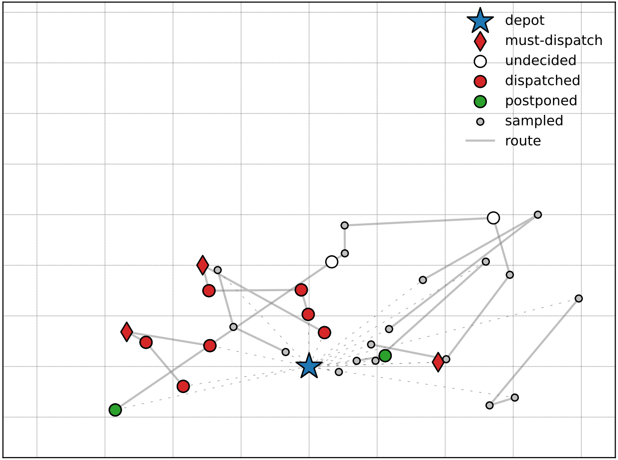
Rinse and repeat!



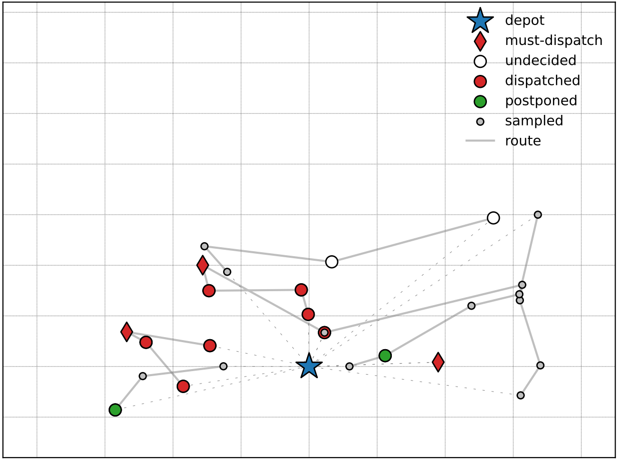
Rinse and repeat!



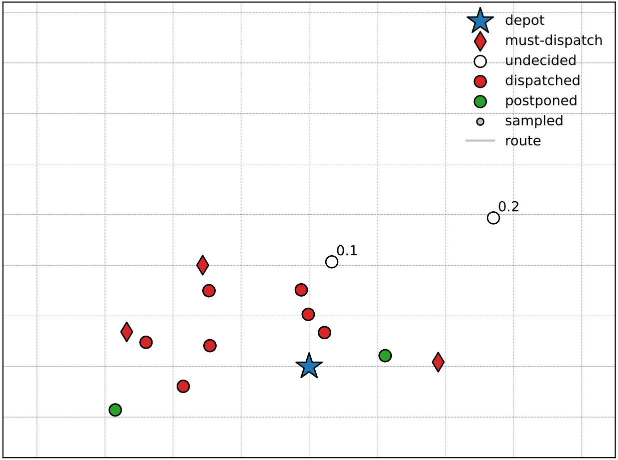
Rinse and repeat!



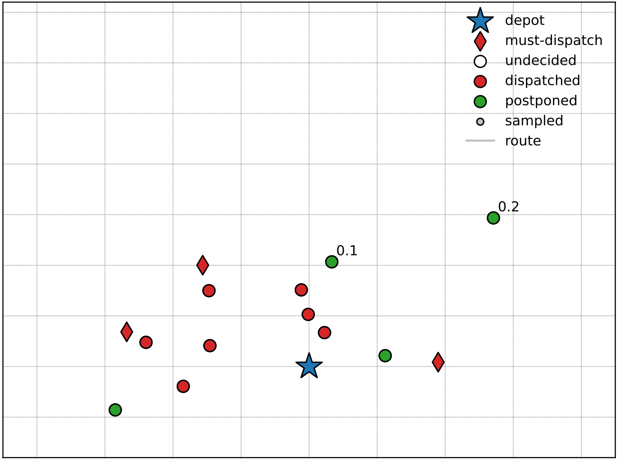
Rinse and repeat!



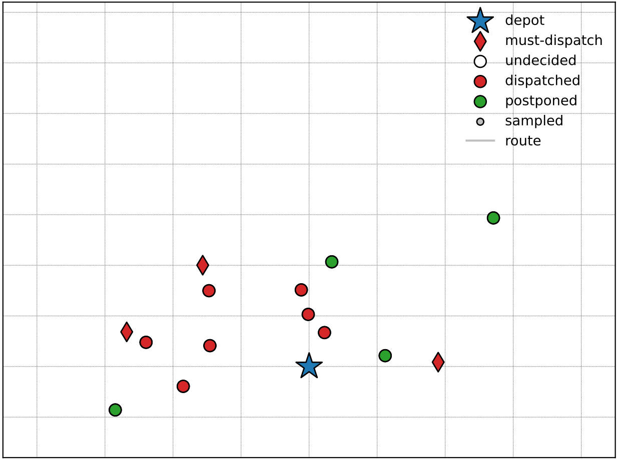
Rinse and repeat!



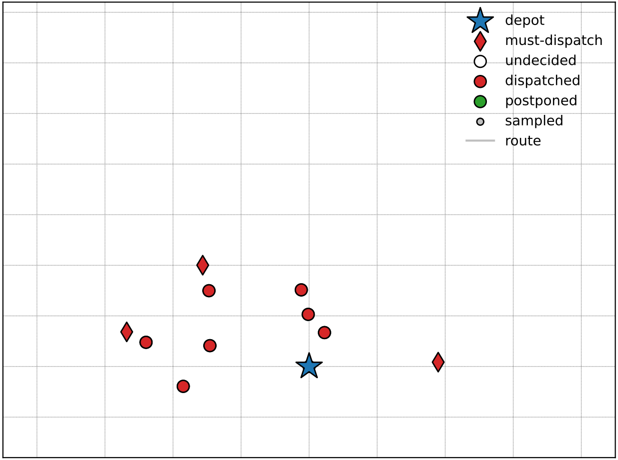
Rinse and repeat!



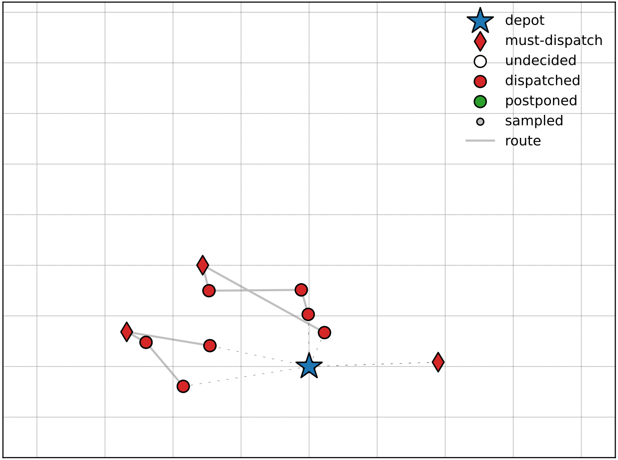
Rinse and repeat!



Dispatched requests



Solve VRPTW-RT for dispatched requests



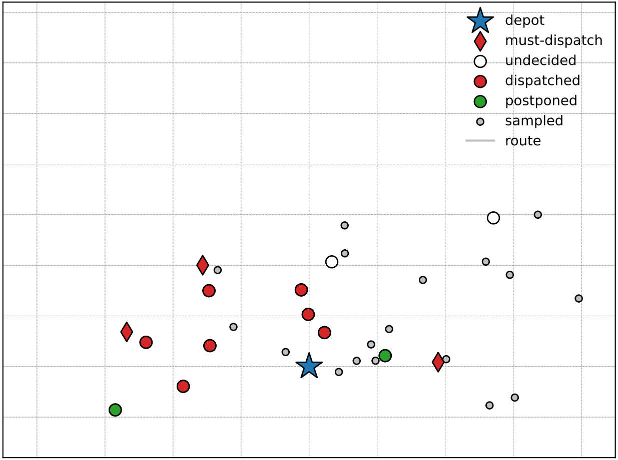
Details of solving scenarios

- Let L denote the number of lookahead epochs.
- A scenario $S \in \mathcal{S}$ is a set of requests formed by
  - the current state s_t, and
  - a sample path realization $(\tilde{\omega}_{t+1}, \tilde{\omega}_{t+2}, \ldots, \tilde{\omega}_{t+L})$.
- Note that requests in $\tilde{\omega}_{t'}$ have release times $T_{t'}$.

Details of solving scenarios

- Let L denote the number of lookahead epochs.
- A scenario $S \in \mathcal{S}$ is a set of requests formed by
  - the current state s_t, and
  - a sample path realization $(\tilde{\omega}_{t+1}, \tilde{\omega}_{t+2}, \ldots, \tilde{\omega}_{t+L})$.
- Note that requests in $\tilde{\omega}_{t'}$ have release times $T_{t'}$.
- But what about d_t and p_t? How can we "condition" on the choices that we have already made so far and prevent routes with "conflicting" requests?

Details of solving scenarios



Dispatch windows

- Instead of solving VRPTW-RT, we solve scenarios as VRPTW with dispatch windows (VRPTW-DW).
- Define a request *dispatch window* $[r_n^-, r_n^+]$, where r_n^- denotes the *earliest* time that request n can be dispatched, and r_n^+ denotes the *latest* time that it can be dispatched.

Route dispatch window feasibility

Consider a route R consisting of a sequence of requests and let θ_R denote its departure time. Then a route is dispatch window feasible if

$$\max_{n \in R}\{r_n^-\} \leq \theta_R \leq \min_{n \in R}\{r_n^+\}.$$

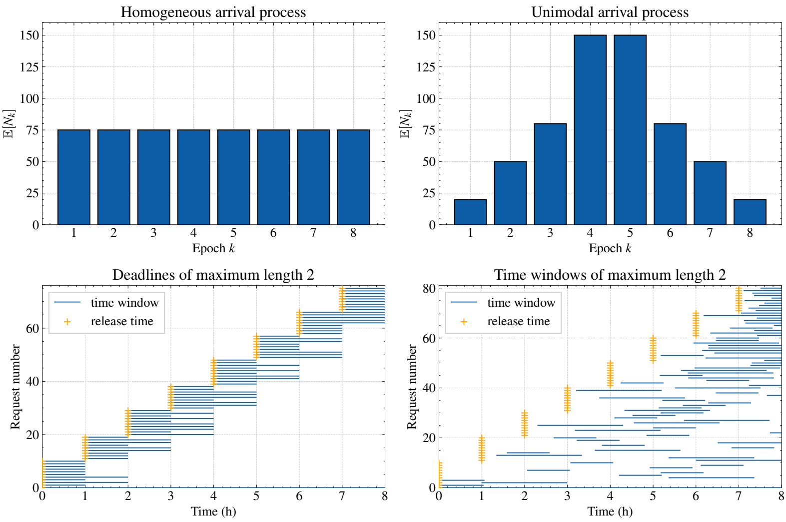
Modifying request dispatch windows

- Consider a scenario S with dispatched requests d_t and postponed requests p_t. We modify the dispatch windows as follows:
  - For dispatched requests $n \in d_t$,
  $$[r_n^-, r_n^+] = [T_t, T_t].$$
  - For postponed requests $n \in p_t$,
  $$[r_n^-, r_n^+] = [T_{t+1}, H].$$
  - For all other requests $n \in S \setminus (d_t \cup p_t)$,
  $$[r_n^-, r_n^+] = [r_n, H].$$

Benchmark instances

- Planning horizon of 8 hours, divided into 1-hour epochs.
- Gehring and Homberger (R, RC, C) instance geographies.
- 600 expected number of requests in total.
- Arrival process and time window variations (shown next slide).
- 72 different total base instances $\times$ 25 different random seeds:
  $\rightarrow$ 1800 instances to be solved

Benchmark instances



Parameter tuning

- We use PyVRP[7] to solve all VRP instances.
  - Optimized to solve scenarios well on short time limits (< 1 second).
- ICD parameters:
  - Number of iterations: 3
  - Number of scenarios per iteration: $|\mathcal{S}| = 30$
  - Number of lookahead epochs: $L = 1$
  - Dispatch threshold: 50%
  - Postponement threshold: 20%
- Epoch time limit of 120 seconds
  - 90 seconds total for scenarios → 1 second per scenario
  - 30 seconds to compute the cost of dispatching action

7 N. A. Wouda, L. Lan, and W. Kool (2023). *PyVRP: a high-performance VRP solver package [Manuscript submitted for review]*. URL: https://github.com/PyVRP/PyVRP/

Algorithms

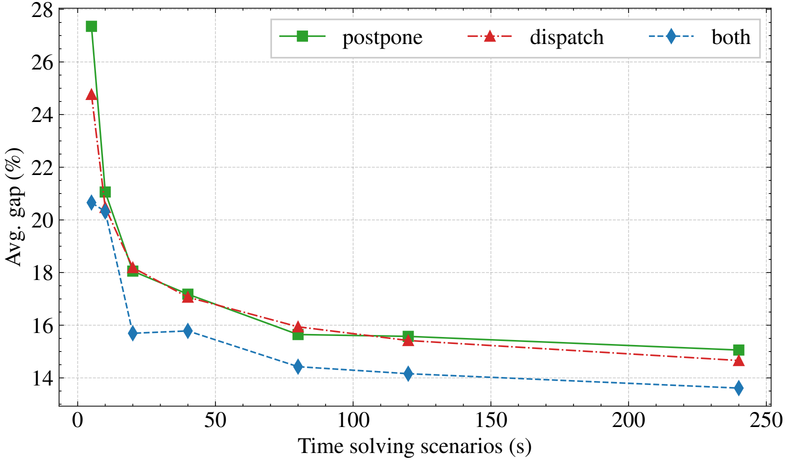
- Greedy baseline: dispatch all available requests
- Variants of ICD:
  - Postpone: only use a postponement threshold (and dispatch all not-postponed)
  - Dispatch: only use a dispatch threshold
  - Both: use both dispatch and postponement threshold
- Compare against *hindsight* solution assuming perfect information.

Results: GH instances

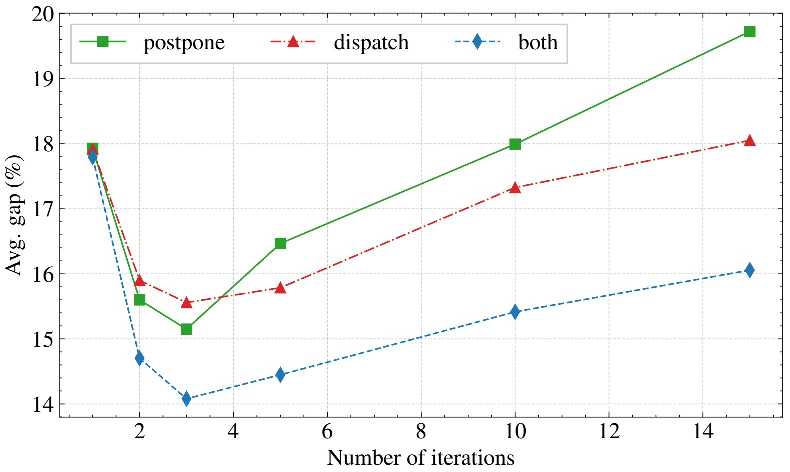
Method	greedy	postpone	dispatch	**both**
Avg. gap	36.57%	10.82%	10.13%	**9.33**%

- 27.24% improvement over greedy baseline algorithm.
- Double threshold shows benefit over single threshold.

Results: GH instances (subset)



Results: GH instances (subset)



EURO-NeurIPS 2022 Vehicle Routing Competition

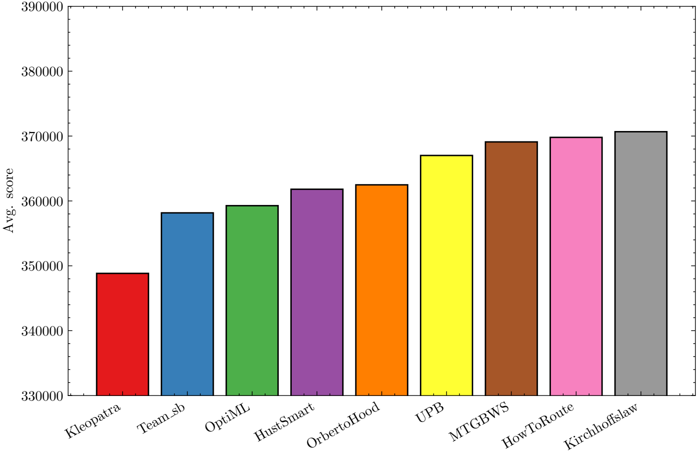


Figure: Results on EURO-NeurIPS final 100 instances

EURO-NeurIPS 2022 Vehicle Routing Competition

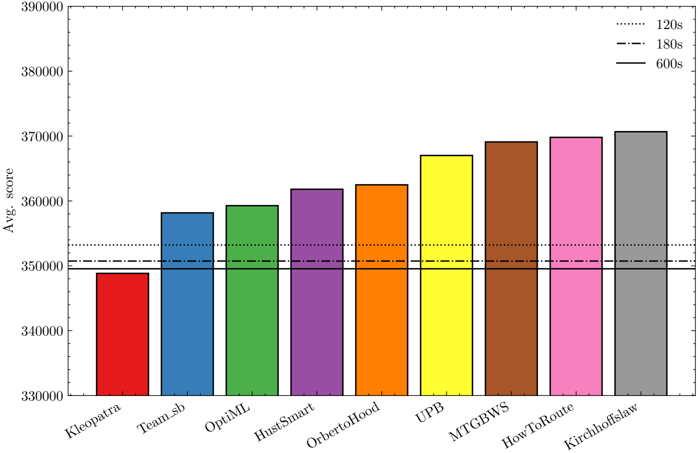


Figure: Results on EURO-NeurIPS final 100 instances

Conclusion

- We showed that ICD can achieve great performance on the DDWP, overcoming the large computational efforts often associated with sampling-based methods.
- Samples do not need to be solved optimally, but good scenario solutions lead to better overall solutions.

Future work

- **Variants of the DDWP**: limited fleet, limited route duration, maximizing the number of served requests, event-based triggered epochs
- **Solve DDWP (near-)optimally**: L-shaped method
- **Consensus functions**: hamming distance, prize-collecting

Thank you for attending!

Preprint: will be uploaded soon!
Slides: https://github.com/leonlan/slides
PyVRP: https://github.com/PyVRP/PyVRP

l.lan@vu.nl — leonlan — www.leonlan.com

Bibliography I

Hvattum, L. M., A. Løkketangen, and G. Laporte (2006). "Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic". In: *Transportation Science* 40.4, pp. 421–438.

Klapp, M. A., A. L. Erera, and A. Toriello (2018). "The One-Dimensional Dynamic Dispatch Waves Problem". In: *Transportation Science* 52.2, pp. 402–415.

Kool, W., D. Numeroso, R. Reijnen, R. R. Afshar, T. Catshoek, K. Tierney, E. Uchoa, and J. Gromicho (2022). *EURO Meets NeurIPS 2022 Vehicle Routing Competition*.

Soeffker, N., M. W. Ulmer, and D. C. Mattfeld (2022). "Stochastic Dynamic Vehicle Routing in the Light of Prescriptive Analytics: A Review". en. In: *European Journal of Operational Research* 298.3, pp. 801–820.

Wouda, N. A., L. Lan, and W. Kool (2023). *PyVRP: a high-performance VRP solver package [Manuscript submitted for review]*. URL: https://github.com/PyVRP/PyVRP/.

Bibliography II

Zhang, J. and T. V. Woensel (2023). "Dynamic Vehicle Routing with Random Requests: A Literature Review". en. In: *International Journal of Production Economics* 256, p. 108751.