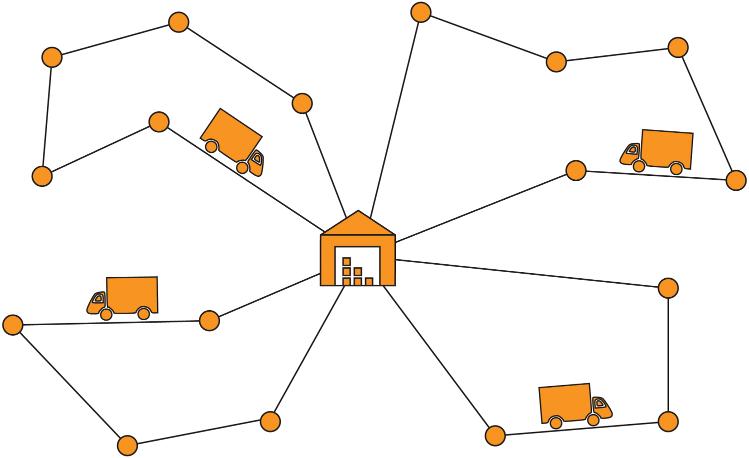
PyVRP and ICD: Results from the EURO Meets NeurIPS 2022 Vehicle Routing Competition

Leon Lan[1] Niels Wouda[2] Wouter Kool[3]
Jasper van Doorn[1] Arpan Rijal[2] Sandjai Bhulai[1]

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

April 16, 2024

Vehicle routing problems



The status quo of routing research

- Exact methods
  - Optimal solutions up to 100 nodes on many variants
  - Limit is around 300 nodes
  - State-of-the-art algorithm VRP Solver[1]
- Heuristic methods
  - Less than 0.1% optimality gap up to 300 nodes within a few minutes
  - Scales easily up to 5K nodes
  - Hybrid Genetic Search (HGS)[2] solves over 20 VRP variants

[1] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck (2020). "A generic exact solver for vehicle routing and related problems". *Mathematical Programming*

[2] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins (2014). "A unified solution framework for multi-attribute vehicle routing problems: electric companion". *European Journal of Operational Research*

What are open issues in routing research?

1. More realistic variants
2. Simpler methods
3. Routing software

Competitions driving routing research

- DIMACS 2021 Implementation Challenge: Vehicle Routing[3]
- Amazon 2021 Last Mile Routing Challenge[4]
- EURO Meets NeurIPS 2022 Vehicle Routing Competition[5]

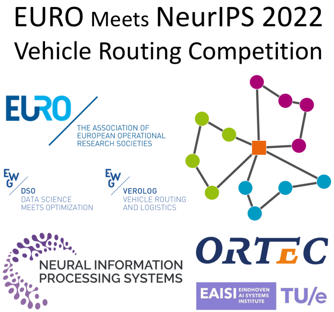
[3] http://dimacs.rutgers.edu/programs/challenge/vrp/

[4] https://routingchallenge.mit.edu/

[5] https://euro-neurips-vrp-2022.challenges.ortec.com/

EURO Meets NeurIPS 2022 Vehicle Routing Competition

Overview of challenge



- Main organizer: Wouter Kool[6]
- 1 July - 31 October 2022
- Daily updated leaderboard
- Over 50 teams participated
- Top 10 teams went to the finals

6 W. Kool, L. Bliek, D. Numeroso, Y. Zhang, T. Catshoek, K. Tierney, T. Vidal, and J. Gromicho (2022). "The EURO Meets NeurIPS 2022 Vehicle Routing Competition". *Proceedings of the NeurIPS 2022 Competitions Track*. PMLR

EURO Meets NeurIPS 2022 Vehicle Routing Competition

Track details

Static track

- **Problem**: Vehicle routing problem with time windows (VRPTW)
- Extends the DIMACS 2021 VRPTW challenge
- State-of-the-art baseline provided: HGS-VRPTW[7]

Dynamic track

- **Problem**: Dynamic same-day delivery problem based on VRPTW
- State-of-the-art unknown
- "Weak" baselines provided: greedy, random, etc.

[7] W. Kool, J. O. Juninck, E. Roos, K. Cornelissen, P. Agterberg, J. van Hoorn, and T. Visser (2022). "Hybrid Genetic Search for the Vehicle Routing Problem with Time Windows: a High-Performance Implementation".

This talk discusses our two main contributions

Part 1: PyVRP

- An open-source and state-of-the-art VRP solver[8]
- Ranked 1st on static VRP track

Part 2: Iterative conditional dispatch (ICD)

- A simple sampling method for dynamic VRPs[9]
- Earlier version ranked 3rd on dynamic VRP track

---

[8] N. A. Wouda, L. Lan, and W. Kool (2024). "PyVRP: A High-Performance VRP Solver Package". *INFORMS Journal on Computing*

[9] L. Lan, J. van Doorn, N. A. Wouda, A. Rijal, and S. Bhulai (2024). "An iterative sample scenario approach for the dynamic dispatch waves problem". *Transportation Science*

PyVRP: An open-source, state-of-the-art VRP solver

1. **Open-source**: Github, MIT license, open to contributions
2. **High performance**: state-of-the-art, Python and C++, rigorous benchmarking
3. **Easy to use**: model-and-run, extensive documentation, many features

PyVRP is open-source

Github, MIT license, community

- Open-source at www.github.com/PyVRP/PyVRP
- Liberal MIT license
- Community contributions welcomed

PyVRP is performant

Built on top of the state-of-the-art, Python and C++

- Built on top of HGS-CVRP[10]
- Key ideas: genetic algorithm hybridized with local search
- Python interface with performance-critical parts in C++

[10] T. Vidal (2022). "Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood". *Computers & Operations Research*

PyVRP is performant

Rigorous benchmarking

PyVRP v0.8.0 compared to other solvers. Gaps with respect to the best-known solution.

	PyVRP	OR-Tools	HGS	ILS-SP
CVRP	0.22%	5.23%	0.11%	0.66%
VRPTW	0.58%	10.86%	0.32%	
PCVRPTW	0.39%	13.24%		
MDVRPTW	0.99%		1.95%	
VRPB	0.35%			1.09%

PyVRP is easy to use

Model-and-run

Not needed to know implementation details

PyVRP is easy to use

Model-and-run

Not needed to know implementation details

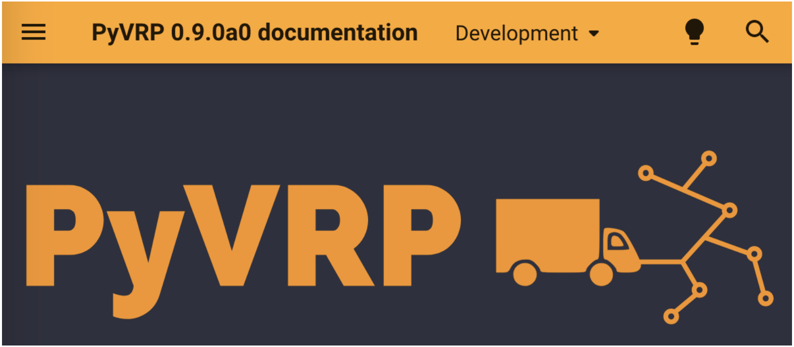
```
import pyvrp

model = pyvrp.Model()

model.add_vehicle_type(num_available=1, capacity=10)
model.add_depot(x=0, y=0)

for idx in range(1, 10):
    model.add_client(x=idx, y=idx, delivery=1)

for frm in model.locations:
    for to in model.locations:
        distance = abs(frm.x - to.x) + abs(frm.y - to.y)
        model.add_edge(frm, to, distance=distance)

result = model.solve(pyvrp.stop.MaxRuntime(10))
```

PyVRP is easy to use

Extensive documentation

Documentation at `pyvrp.org`

- Example notebooks
- API reference



PyVRP is easy to use

Many features

PyVRP can solve up to 13 known VRP variants

1. Capacitated Vehicle Routing Problem (CVRP)
2. Asymmetric Vehicle Routing Problem (AVRP)
3. Vehicle Routing Problem with Time Windows (VRPTW)
4. Team Orienteering Problem (TOP)
5. Vehicle Routing Problem with Profits (VRPP)
6. Open Vehicle Routing Problem (OVRP)
7. Multi-Depot Vehicle Routing Problem (MDVRP)
8. Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)
9. Vehicle Routing Problem with Backhauls (VRPB)
10. Distance-Constrained Vehicle Routing Problem (DCVRP)
11. Generalized Vehicle Routing Problem (GVRP)
12. Vehicle Routing Problem with Multiple Time Windows (VRPMTW)
13. Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

Part 1: PyVRP - Conclusion

- We presented an open-source, state-of-the-art vehicle routing problem solver.
- PyVRP is easy to use and we hope to help researchers and practitioners to build on this.

Part 2: ICD

Let's move on to part two...

Same-day delivery

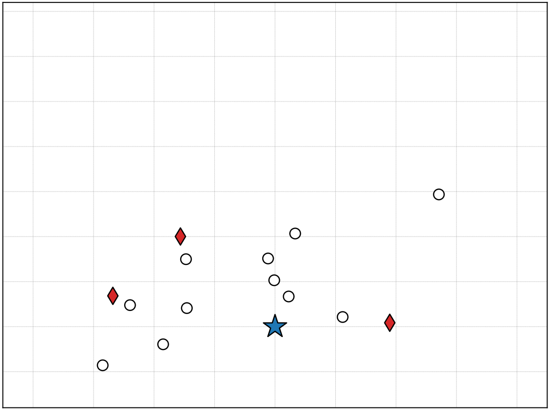
Same-day delivery

- Same-day delivery problems are dynamic vehicle routing problems with stochastic *delivery* requests.
- 90% of the studies on same-day delivery problems have been published in the last five years.[11]

[11] J. Zhang and T. V. Woensel (2023). "Dynamic vehicle routing with random requests: A literature review". *International Journal of Production Economics*

Dynamic dispatch waves problem

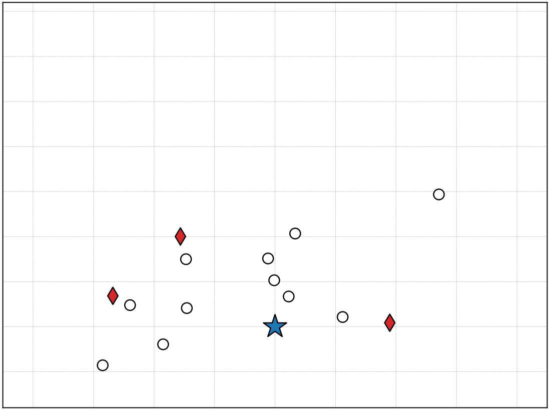
Illustrative example



Blue star is the depot. White circles are undecided requests. Red diamonds are must-dispatch.

Dynamic dispatch waves problem

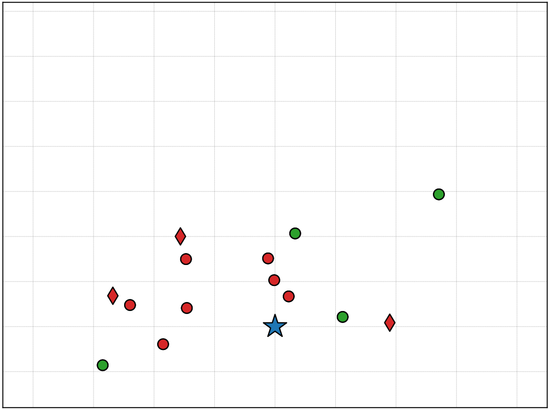
Illustrative example



Decide which requests to dispatch or to postpone. Requests are have time windows, which define their dispatch flexibility.

Dynamic dispatch waves problem

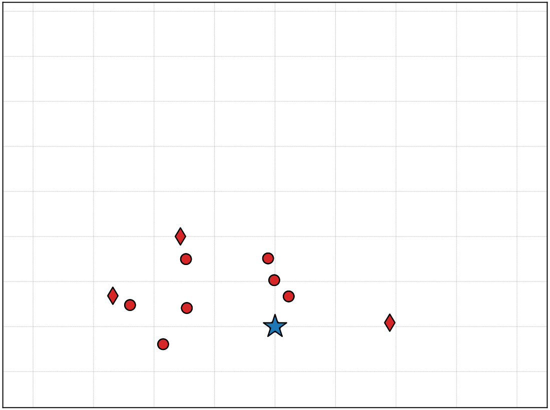
Illustrative example



Decision made. Red means dispatched, green means postponed.

Dynamic dispatch waves problem

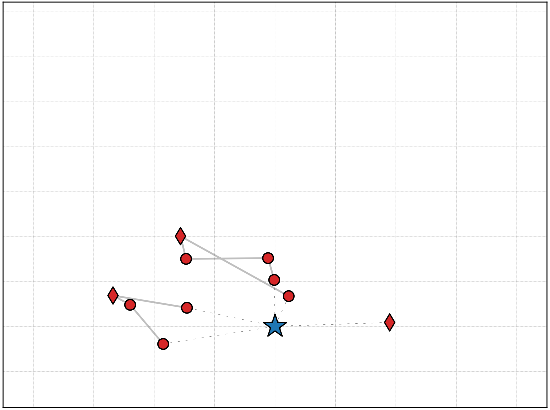
Illustrative example



Given the dispatched requests, the next step is to determine good routes.

Dynamic dispatch waves problem

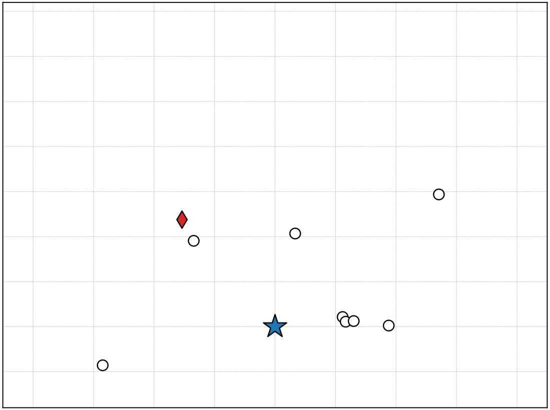
Illustrative example



The routing solution.

Dynamic dispatch waves problem

Illustrative example



New decision moment. The set of requests is formed by new arrivals and previously postponed requests.

Dynamic dispatch waves problem

Problem description

- Requests have demand, service times, and time windows
- Unlimited homogeneous vehicles, no multi-trips
- Arrival process distribution known or data is available
- Decide in each epoch $t \in \{1, \ldots, T\}$:
  - The requests to dispatch, and
  - The corresponding routes.
- **Goal**: deliver all requests on time with minimum total distance

Sample scenario methods

- **Main idea**: sample future request realizations, solve each realization, and derive a good action.
- **Big drawback:** Computationally expensive to solve many, large, NP-hard VRPs!
- What if we instead *incrementally* build an action to minimize computational challenges?

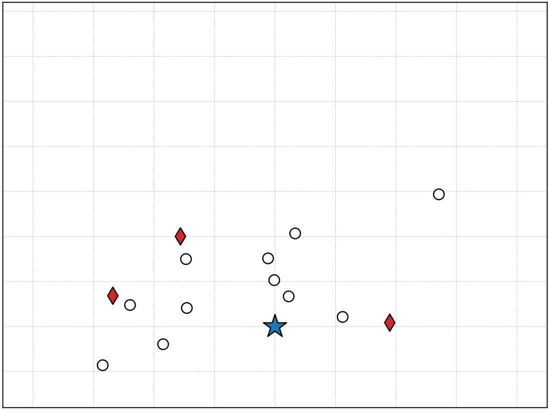
Iterative conditional dispatch

Algorithm outline

- d_t: The set of *dispatched* requests in epoch t
- p_t: The set of *postponed* requests in epoch t
- **Initialize** d_t to must-dispatch and p_t empty
- **Repeat** for a fixed number of iterations:
  - **Step 1**: Solve sample scenarios conditioned on d_t and p_t
  - **Step 2**: Classify undecided requests as d_t, p_t, or leave undecided
- **Return** dispatched requests d_t

Iterative conditional dispatch

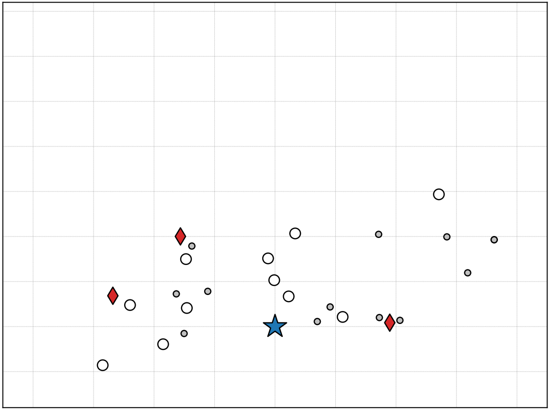
Illustrative example



Available requests at some epoch t, say 12:00.

Iterative conditional dispatch

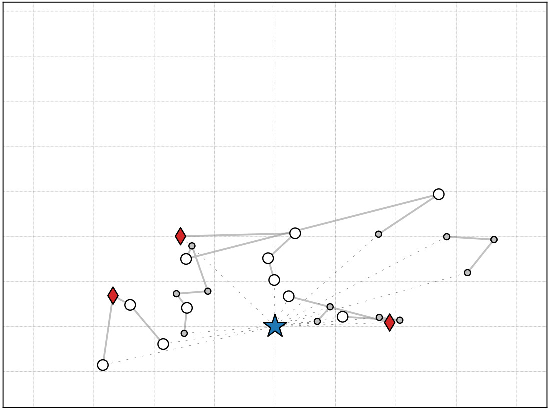
Illustrative example



A sample scenario instance including requests from 13:00.

Iterative conditional dispatch

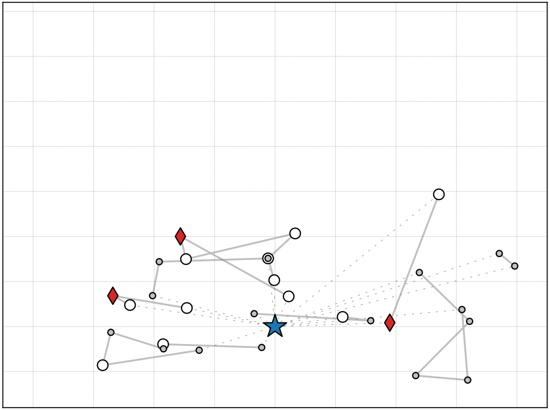
Illustrative example



Solve the scenario instance (VRPTW with release times). Note that red requests are never paired with grey requests.

Iterative conditional dispatch

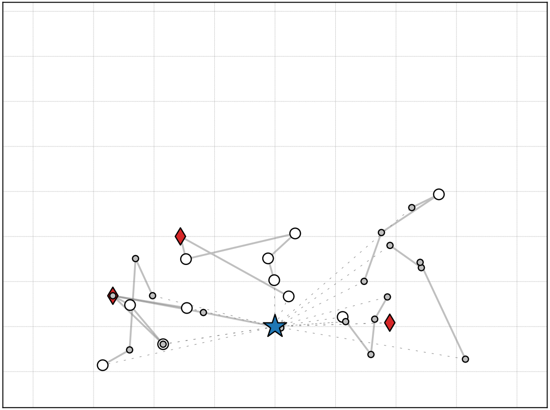
Illustrative example



Solve another scenario instance.

Iterative conditional dispatch

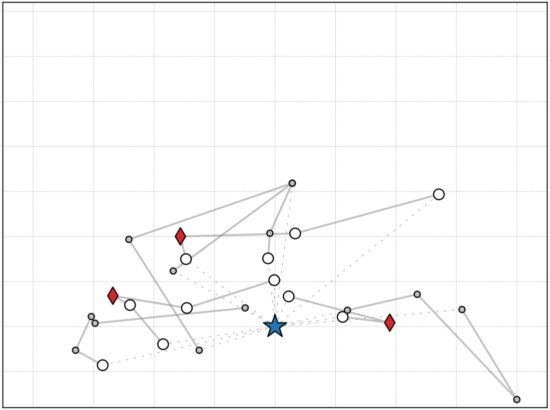
Illustrative example



And another one... (30x in total)

Iterative conditional dispatch

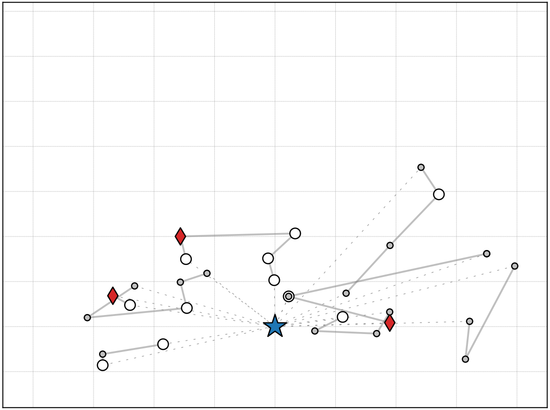
Illustrative example



And another one... (30x in total)

Iterative conditional dispatch

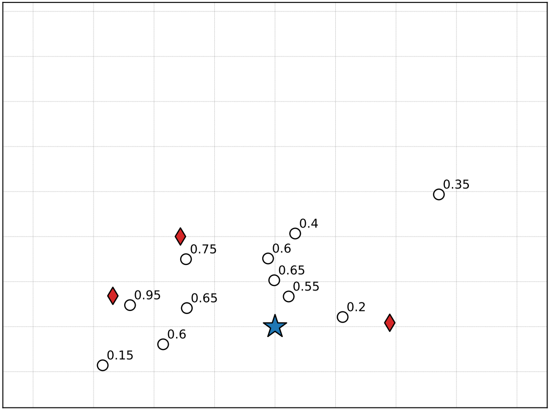
Illustrative example



And another one... (30x in total)

Iterative conditional dispatch

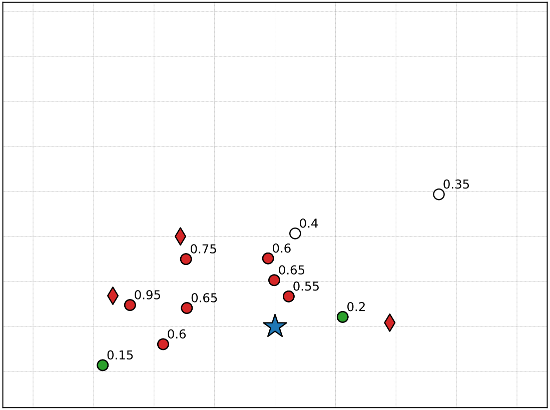
Illustrative example



Compute the dispatch scores: how frequently was each request dispatched?

Iterative conditional dispatch

Illustrative example

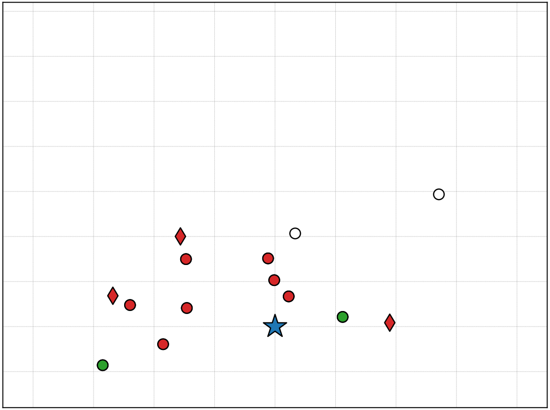


Dispatch if score ≥ 0.5, postpone if score ≤ 0.2.

Leave some undecided.

Iterative conditional dispatch

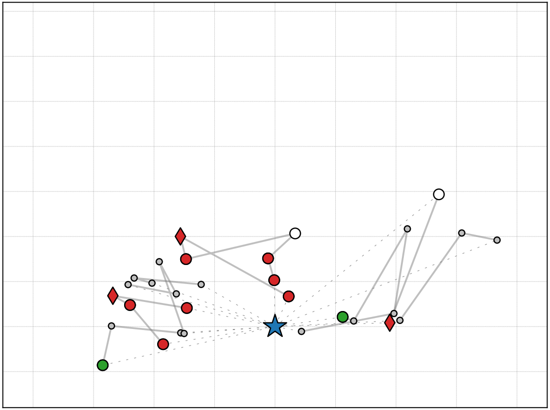
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

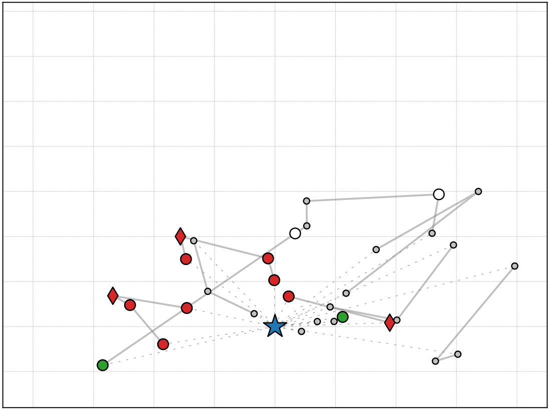
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

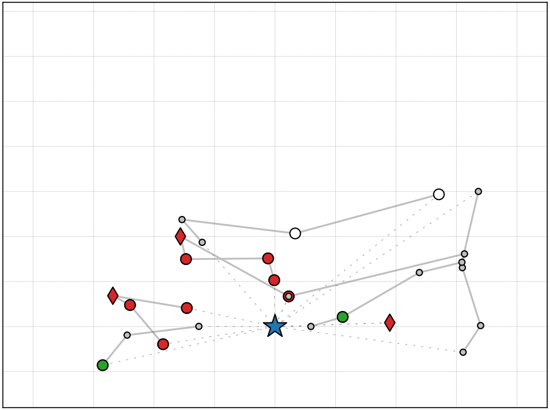
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

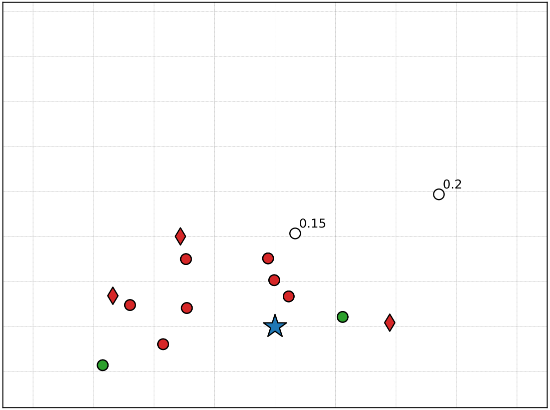
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

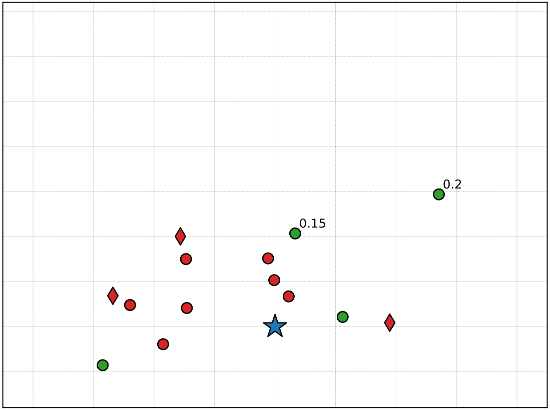
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

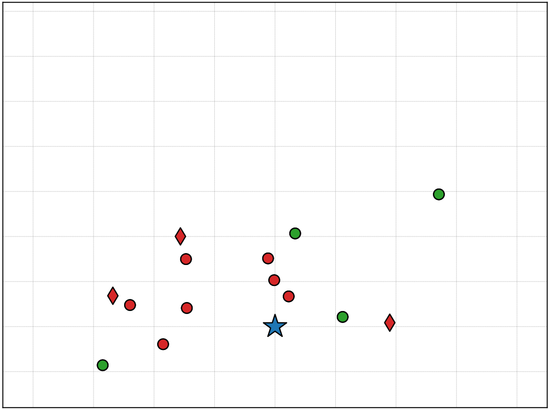
Illustrative example



Rinse and repeat!

Iterative conditional dispatch

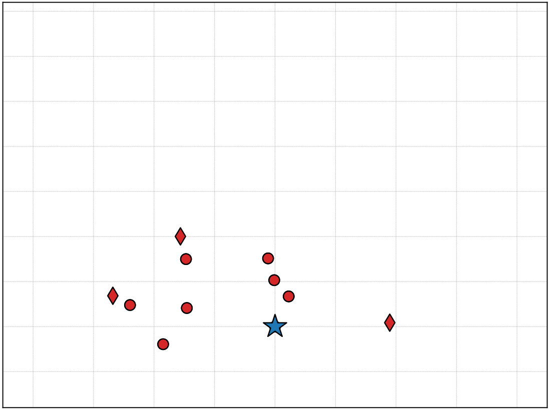
Illustrative example



All requests classified as dispatched or postponed.

Iterative conditional dispatch

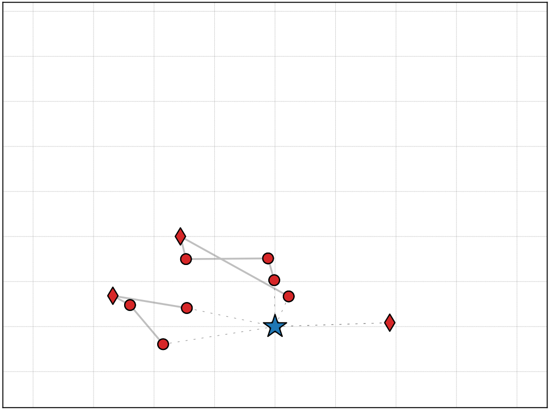
Illustrative example



The requests assigned for dispatch.

Iterative conditional dispatch

Illustrative example



Final routing solution.

Fixing decisions with dispatch windows

Key challenges

1. How to translate *dynamic* decisions (dispatched or postponed requests) to the *static* VRPTW?
2. How to make this fast and incorporate in our static solver?

Fixing decisions with dispatch windows

Dispatch windows

Define a request *dispatch window* $[r_i^-, r_i^+]$, where r_i^- denotes the *earliest* time that request i can be dispatched, and r_i^+ denotes the *latest* time that it can be dispatched.

$$[r_i^-, r_i^+] = \begin{cases} [T_t, T_t] & \text{for dispatched requests } i \in d_t, \\ [T_{t+1}, H] & \text{for postponed requests } i \in p_t, \\ [r_i, H] & \text{for undecided requests,} \end{cases}$$

where H denotes the planning horizon.

Example

Example For example, $[12, 12]$ means a request has to be dispatched at time 12:00, while $[13, H]$ means a request can only be dispatched after time 13:00.

Fixing decisions with dispatch windows

Vehicle routing problem with dispatch windows

Route dispatch window feasibility

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

$$\max_{i \in R}\{r_i^-\} \leq \theta_R \leq \min_{i \in R}\{r_i^+\}.$$

From release dates to dispatch windows

We solve scenarios as VRPTW with dispatch windows to condition on the dispatched requests d_t. Extending solvers to deal with dispatch windows takes little effort (details in paper).

Instances

Top 10 teams are evaluated on 100 final instances.

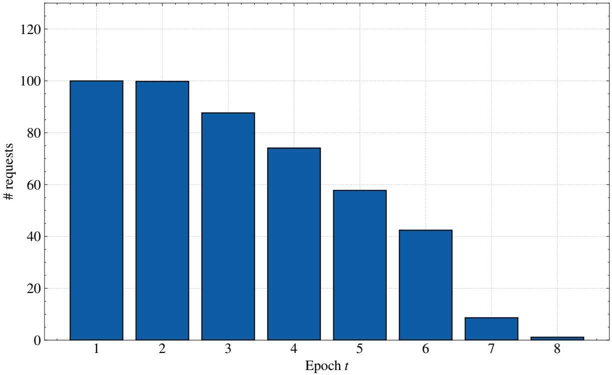
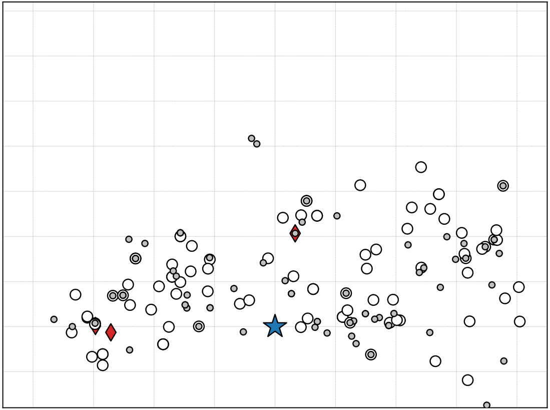


Figure: Average arrival process of a final's instance.

Impression of a scenario instance...



Parameter settings

ICD parameters:

- 3 iterations, 30 scenarios, 50% dispatch, 20% postpone threshold
- 90 seconds for scenarios → 1 second per scenario
- 30 seconds to compute the cost of final routing solution

Computational details:

- Single core (can be parallelized trivially)
- Time limits: 120s (competition), 180s, and 600s

EURO-NeurIPS results

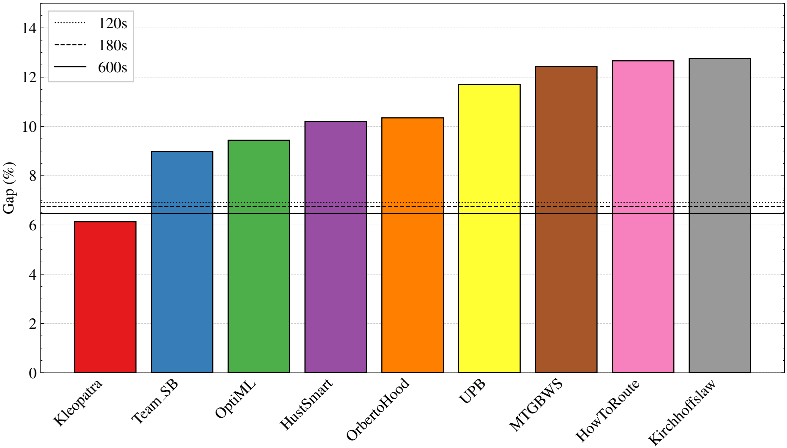


Figure: Gap w.r.t. hindsight solutions on EURO-NeurIPS final 100 instances. Dashed lines represent ICD with different time limits.

Part 2: ICD - Conclusion

- We showed that ICD can achieve great performance on the DDWP, overcoming the large computational efforts often associated with sampling-based methods.
- ICD is conceptually simple and very easy to implement, and shows great promise for extension to other dynamic VRPs.
- Having an easily extensible static VRP solver was crucial in developing ICD.

What are open issues in routing research?

- More realistic variants
- Simpler methods
- Routing software

What are open issues in routing research?

- More realistic variants
- Simpler methods
- Routing software
- Machine learning!

Thank you KAIST!

https://www.leonlan.com/kaist2024

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

Bibliography I

Kool, W., L. Bliek, D. Numeroso, Y. Zhang, T. Catshoek, K. Tierney, T. Vidal, and J. Gromicho (2022). "The EURO Meets NeurIPS 2022 Vehicle Routing Competition". *Proceedings of the NeurIPS 2022 Competitions Track*. ISSN: 2640-3498. PMLR.

Kool, W., J. O. Juninck, E. Roos, K. Cornelissen, P. Agterberg, J. van Hoorn, and T. Visser (2022). "Hybrid Genetic Search for the Vehicle Routing Problem with Time Windows: a High-Performance Implementation".

Lan, L., J. van Doorn, N. A. Wouda, A. Rijal, and S. Bhulai (2024). "An iterative sample scenario approach for the dynamic dispatch waves problem". *Transportation Science*. Forthcoming.

Pessoa, A., R. Sadykov, E. Uchoa, and F. Vanderbeck (2020). "A generic exact solver for vehicle routing and related problems". *Mathematical Programming*.

Vidal, T. (2022). "Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood". *Computers & Operations Research*.

Vidal, T., T. G. Crainic, M. Gendreau, and C. Prins (2014). "A unified solution framework for multi-attribute vehicle routing problems: electric companion". *European Journal of Operational Research*.

Bibliography II

Wouda, N. A., L. Lan, and W. Kool (2024). "PyVRP: A High-Performance VRP Solver Package". *INFORMS Journal on Computing.*

Zhang, J. and T. V. Woensel (2023). "Dynamic vehicle routing with random requests: A literature review". *International Journal of Production Economics.*