

A greedy randomized destroy and repair heuristic for the dial-a-ride problem with transfers

Sébastien Janssens

Delft University of Technology, The Netherlands

January 21, 2020

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions
 - Description through practical example

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions
 - Description through practical example
- ③ Results

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions
 - Description through practical example
- ③ Results
 - Summary

Outline

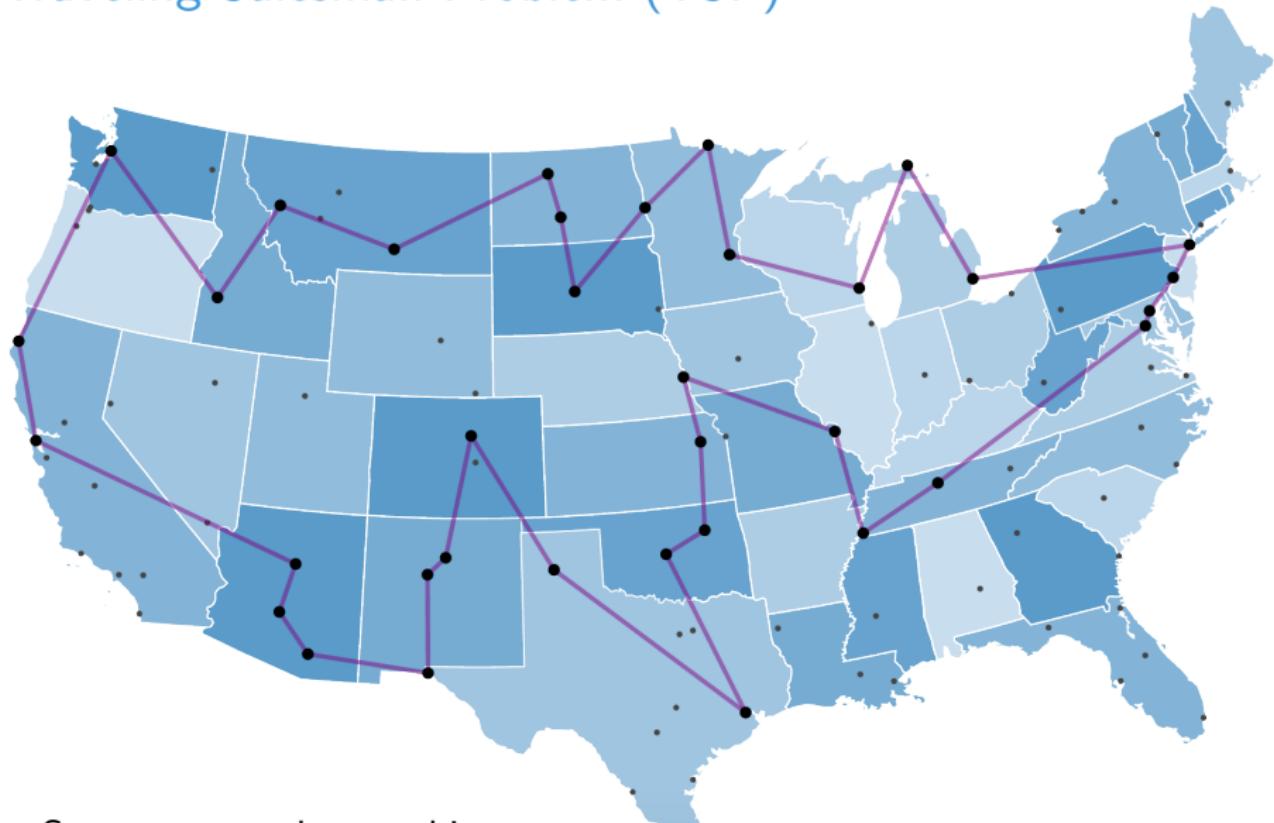
- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions
 - Description through practical example
- ③ Results
 - Summary
 - Impact of transfers on solution robustness

Outline

- ① Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)
 - Similar problems and definition of the DARP-T
 - Generated instances
 - Problem characteristics
- ② Algorithm description
 - Definitions
 - Description through practical example
- ③ Results
 - Summary
 - Impact of transfers on solution robustness
- ④ Conclusion

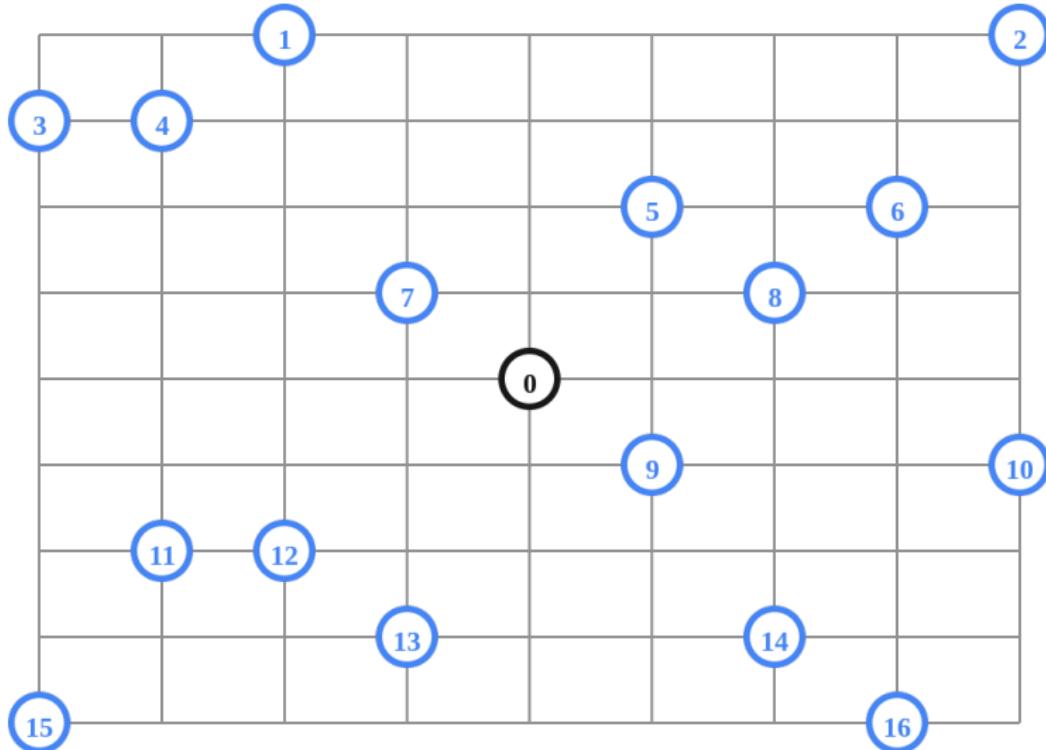
Introduction to the Dial-a-Ride Problem with Transfers (DARP-T)

Traveling Salesman Problem (TSP)



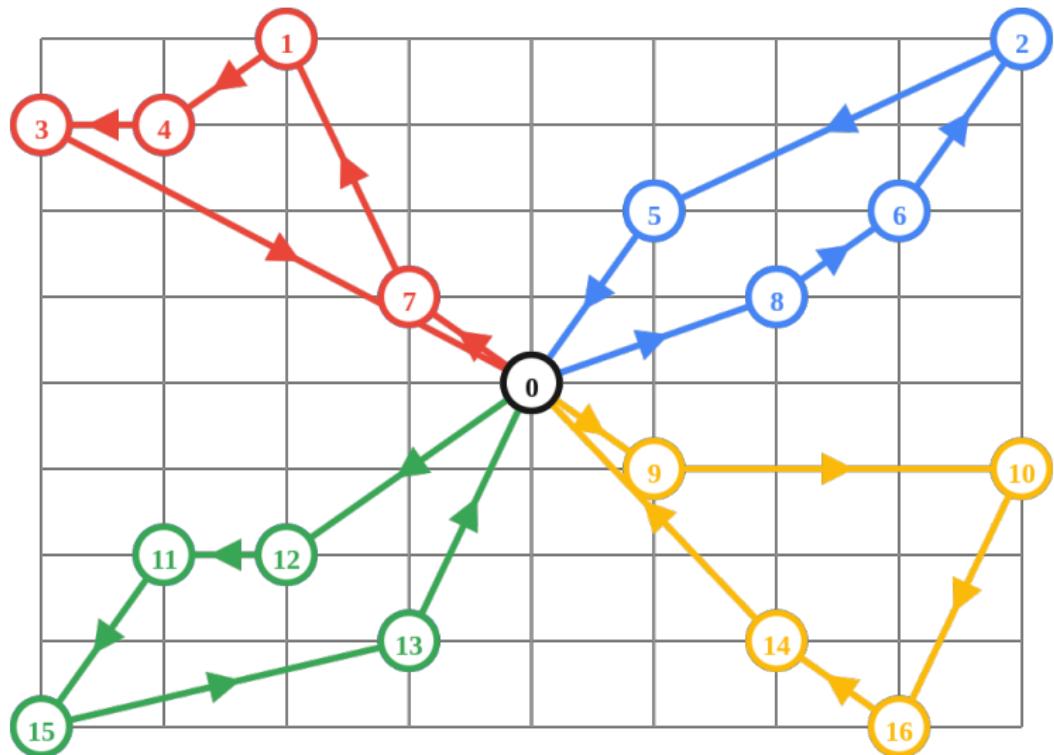
Source: examples.gurobi.com

Vehicle Routing Problem (VRP)



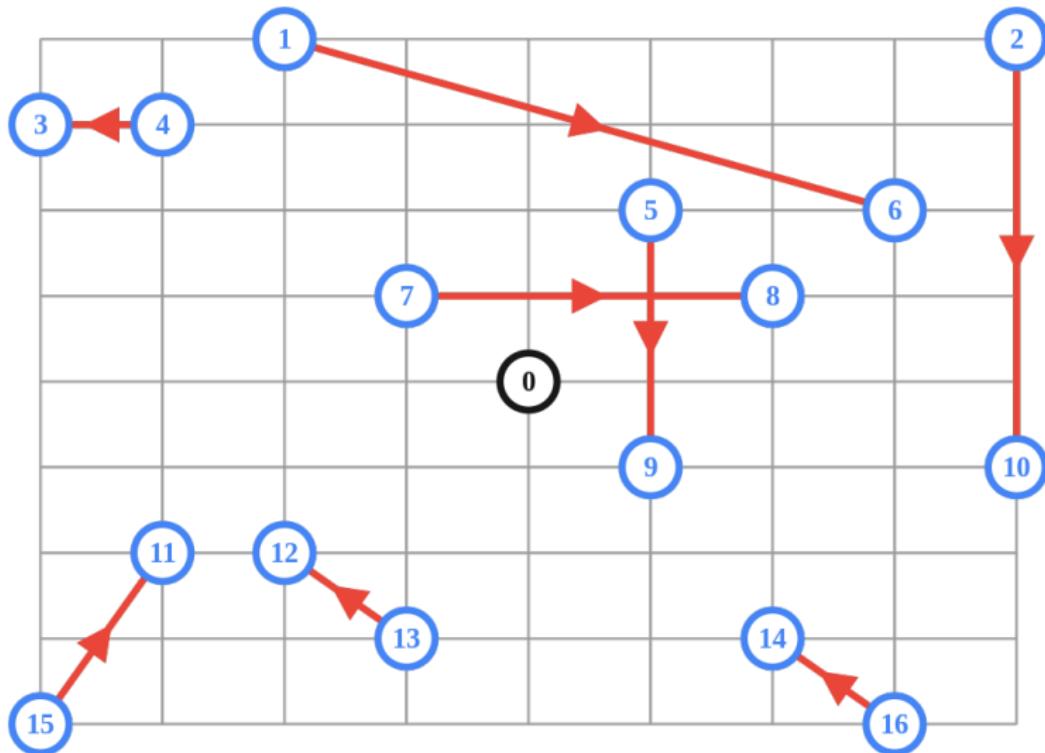
Source: developers.google.com

Vehicle Routing Problem (VRP)



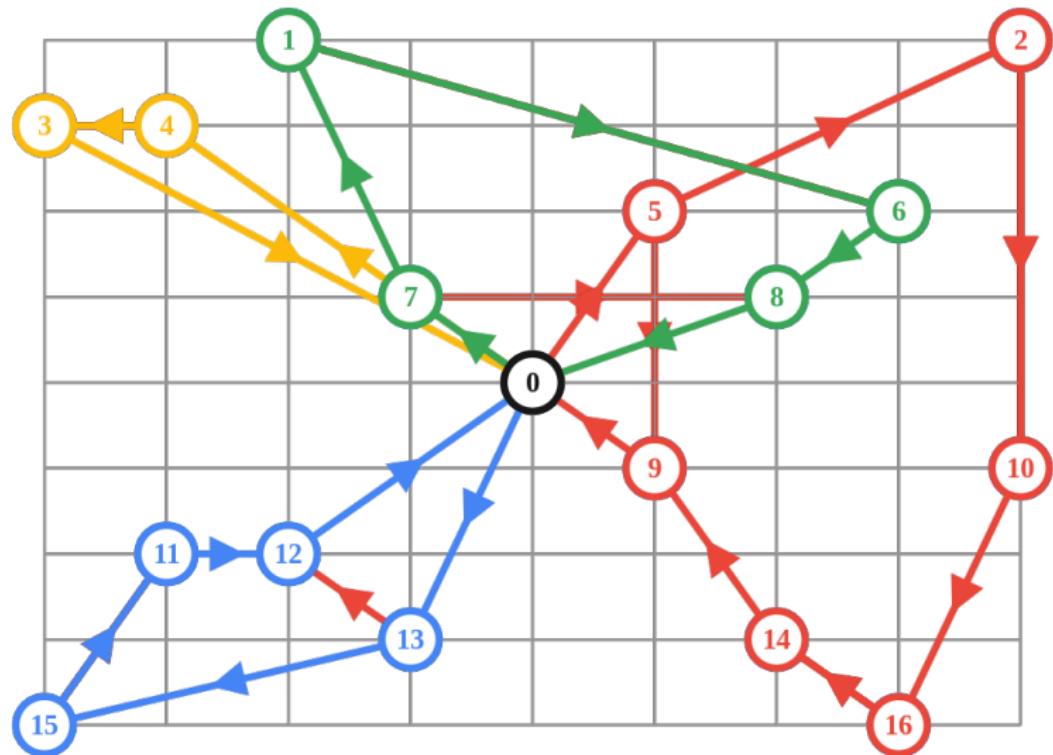
Source: developers.google.com

Pick-up and Delivery Problem (PDP)



Source: developers.google.com

Pick-up and Delivery Problem (PDP)



Source: developers.google.com

Dial-a-Ride Problem (DARP)

- PDP with user inconvenience constraints

Dial-a-Ride Problem (DARP)

- PDP with user inconvenience constraints
- In the future, self-driving vehicles could provide an on-demand transportation service

Dial-a-Ride Problem (DARP)

- PDP with user inconvenience constraints
- In the future, self-driving vehicles could provide an on-demand transportation service
- Allowing vehicles to exchange passengers results in the Dial-a-Ride Problem with Transfers (DARP-T)

Dial-a-Ride Problem (DARP)

- PDP with user inconvenience constraints
- In the future, self-driving vehicles could provide an on-demand transportation service
- Allowing vehicles to exchange passengers results in the Dial-a-Ride Problem with Transfers (DARP-T)
- Goals of the thesis:
 - Develop a heuristic that solves the DARP-T

Dial-a-Ride Problem (DARP)

- PDP with user inconvenience constraints
- In the future, self-driving vehicles could provide an on-demand transportation service
- Allowing vehicles to exchange passengers results in the Dial-a-Ride Problem with Transfers (DARP-T)
- Goals of the thesis:
 - Develop a heuristic that solves the DARP-T
 - Determine the (dis-)advantages of transfers

Formulation of the DARP-T

- Set of nodes

Formulation of the DARP-T

- Set of nodes
- Set of requests R (of different number of passengers) that have to travel from a pick-up node to a drop-off node inside a demanding time window

Formulation of the DARP-T

- Set of nodes
- Set of requests R (of different number of passengers) that have to travel from a pick-up node to a drop-off node inside a demanding time window
- Set of capacitated vehicles V allowed to transport several requests simultaneously, exchange requests at any node, and park at any node

Formulation of the DARP-T

- Set of nodes
- Set of requests R (of different number of passengers) that have to travel from a pick-up node to a drop-off node inside a demanding time window
- Set of capacitated vehicles V allowed to transport several requests simultaneously, exchange requests at any node, and park at any node
- Find a set of vehicle routes that minimizes

$$\frac{\sum_{v \in V} \text{time } v \text{ spent driving}}{\sum_{r \in A} \text{travel time of } r} + 0.1(|R| - |A|)$$

with A the set of served requests

Instances generated

- Set of nodes is the set of the 100 most populated cities in the Netherlands

Instances generated

- Set of nodes is the set of the 100 most populated cities in the Netherlands
- Vehicles starting at random locations with capacity of 2, 4 or 6

Instances generated

- Set of nodes is the set of the 100 most populated cities in the Netherlands
- Vehicles starting at random locations with capacity of 2, 4 or 6
- 150, 500 or 1000 randomly generated requests of 1, 2 or 3 passengers traveling between two randomly selected nodes inside a four-hour timescale

Instances generated

- Set of nodes is the set of the 100 most populated cities in the Netherlands
- Vehicles starting at random locations with capacity of 2, 4 or 6
- 150, 500 or 1000 randomly generated requests of 1, 2 or 3 passengers traveling between two randomly selected nodes inside a four-hour timescale
- Average request travel time of one hour

Instances generated

- Set of nodes is the set of the 100 most populated cities in the Netherlands
- Vehicles starting at random locations with capacity of 2, 4 or 6
- 150, 500 or 1000 randomly generated requests of 1, 2 or 3 passengers traveling between two randomly selected nodes inside a four-hour timescale
- Average request travel time of one hour
- Time-windows competitive with public transport (on average, 29 minutes longer than the travel time and 1.52 times the travel time)

Example of 150-request instance

Example of 1000-request instance

Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option

Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option

Note that 360^{1000}

Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option
- Large instances of an NP-hard problem \implies heuristic

Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option
- Large instances of an NP-hard problem \implies heuristic
- Transfers introduce a very large number of new possibilities:

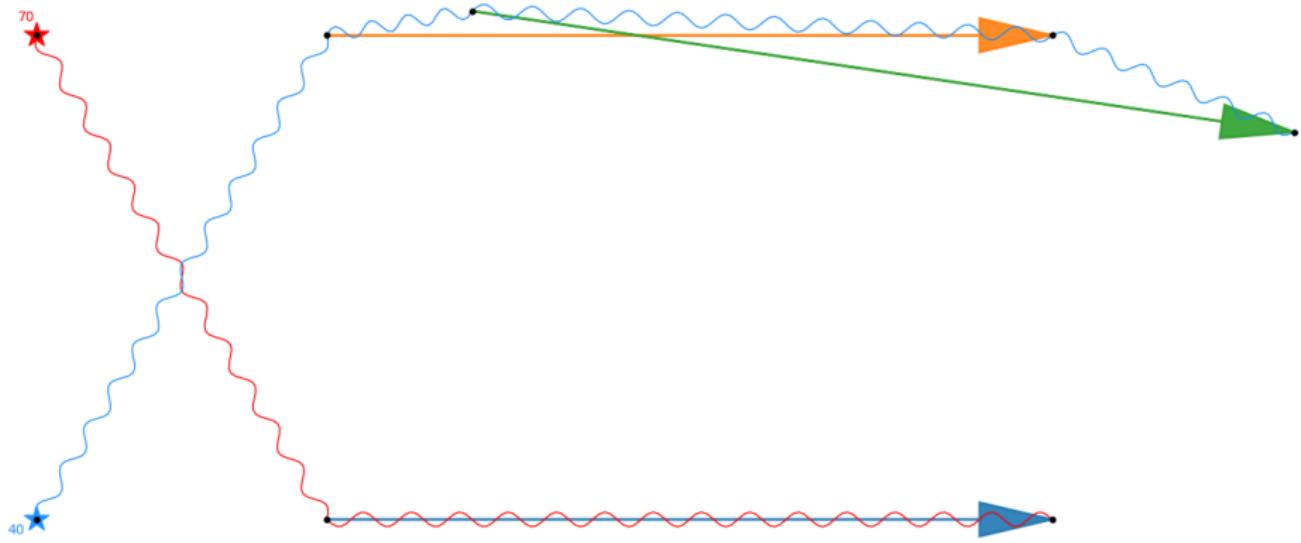
Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option
- Large instances of an NP-hard problem \implies heuristic
- Transfers introduce a very large number of new possibilities:
 - vehicles involved in transfer and requests they bring
 - location and time of transfer
 - splitting of the requests between the vehicles at the transfer

Problem characteristics

- There are k^n ways to split a set of n requests into k (possibly empty) distinct vehicles \implies brute-force is not an option
- Large instances of an NP-hard problem \implies heuristic
- Transfers introduce a very large number of new possibilities:
 - vehicles involved in transfer and requests they bring
 - location and time of transfer
 - splitting of the requests between the vehicles at the transfer
- Transfers together with time windows (and capacity) constraints make solution very vulnerable to change

Problem characteristics



Design philosophy of the algorithm

Feasibility at all times

We progressively build a solution with the constraint that the solution we are building must known to be completable to a feasible solution

Design philosophy of the algorithm

Feasibility at all times

We progressively build a solution with the constraint that the solution we are building must known to be completable to a feasible solution

Greediness

Because the solution space is expensive to explore, we do not consider moves that are unlikely to be part of a good solution

Algorithm description

Definitions

Definition

A **route** is a sequence of (node, time, action) tuples associated with a vehicle. Actions can be wait, pick up, transfer, or drop off.

Definitions

Definition

A **route** is a sequence of (node, time, action) tuples associated with a vehicle. Actions can be wait, pick up, transfer, or drop off.

There are different types of routes:

- An **entire route** is a route that starts at the initial position and time of its associated vehicle

Definitions

Definition

A **route** is a sequence of (node, time, action) tuples associated with a vehicle. Actions can be wait, pick up, transfer, or drop off.

There are different types of routes:

- An **entire route** is a route that starts at the initial position and time of its associated vehicle
- A **complete route** is a route that is empty (i.e., no request inside the vehicle) at its last node (unit)

Definitions

Definition

A **route** is a sequence of (node, time, action) tuples associated with a vehicle. Actions can be wait, pick up, transfer, or drop off.

There are different types of routes:

- An **entire route** is a route that starts at the initial position and time of its associated vehicle
- A **complete route** is a route that is empty (i.e., no request inside the vehicle) at its last node (unit)
- An **incomplete route** is a route that is not empty at its last node (unit)

Definitions

Definition

A **route** is a sequence of (node, time, action) tuples associated with a vehicle. Actions can be wait, pick up, transfer, or drop off.

There are different types of routes:

- An **entire route** is a route that starts at the initial position and time of its associated vehicle
- A **complete route** is a route that is empty (i.e., no request inside the vehicle) at its last node (unit)
- An **incomplete route** is a route that is not empty at its last node (unit)
- A **certificate of feasibility** is a complete route associated with an incomplete route proving that the requests inside the vehicle at the end of the incomplete route can be feasibly dropped-off.

Definitions

Definition

A **solution** is a list of complete, entire routes (one per vehicle).

Definitions

Definition

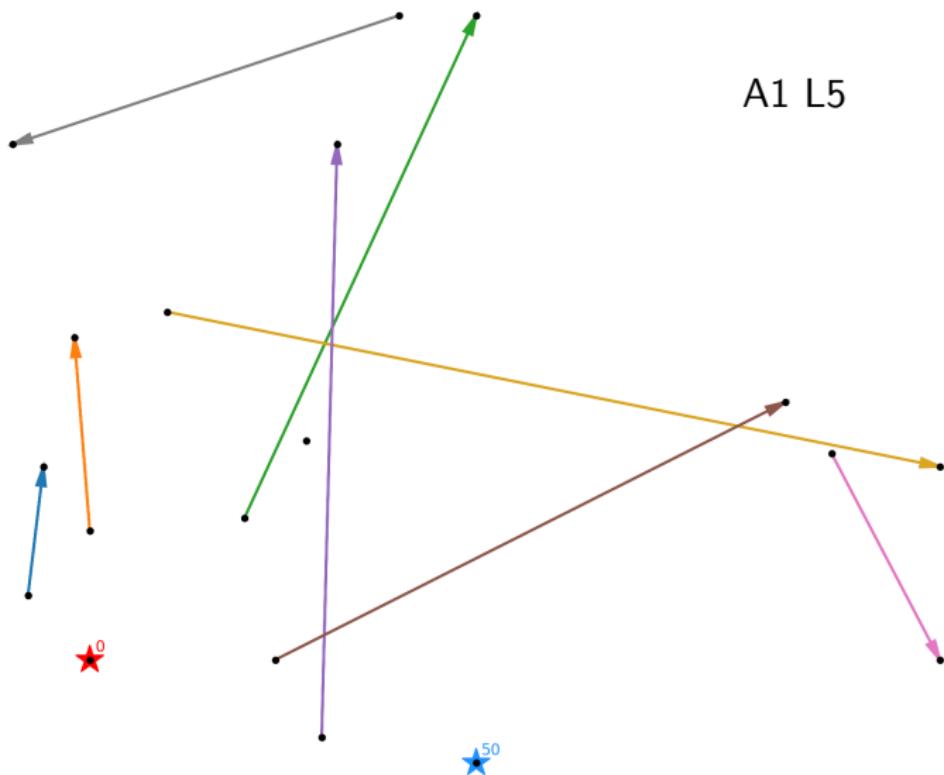
A **solution** is a list of complete, entire routes (one per vehicle).

Definition

A **partial solution** is a list of possibly incomplete, entire routes (one per vehicle). Solutions are gradually built and the partial solutions are the intermediary steps in the construction of a solution.

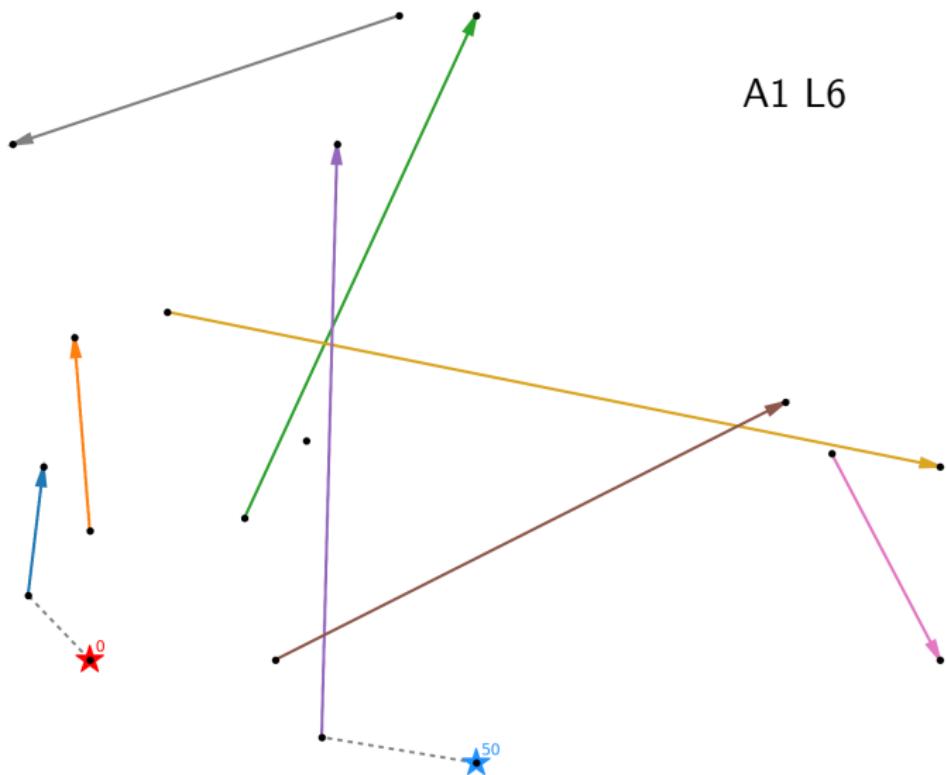
Description through practical example

A1 L5

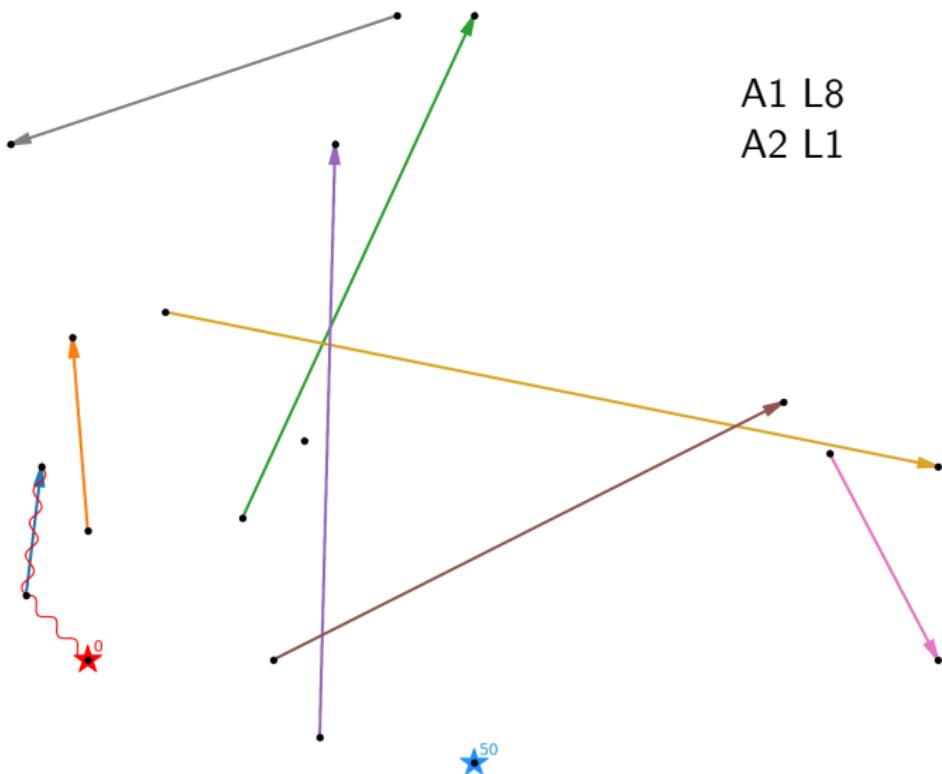


Description through practical example

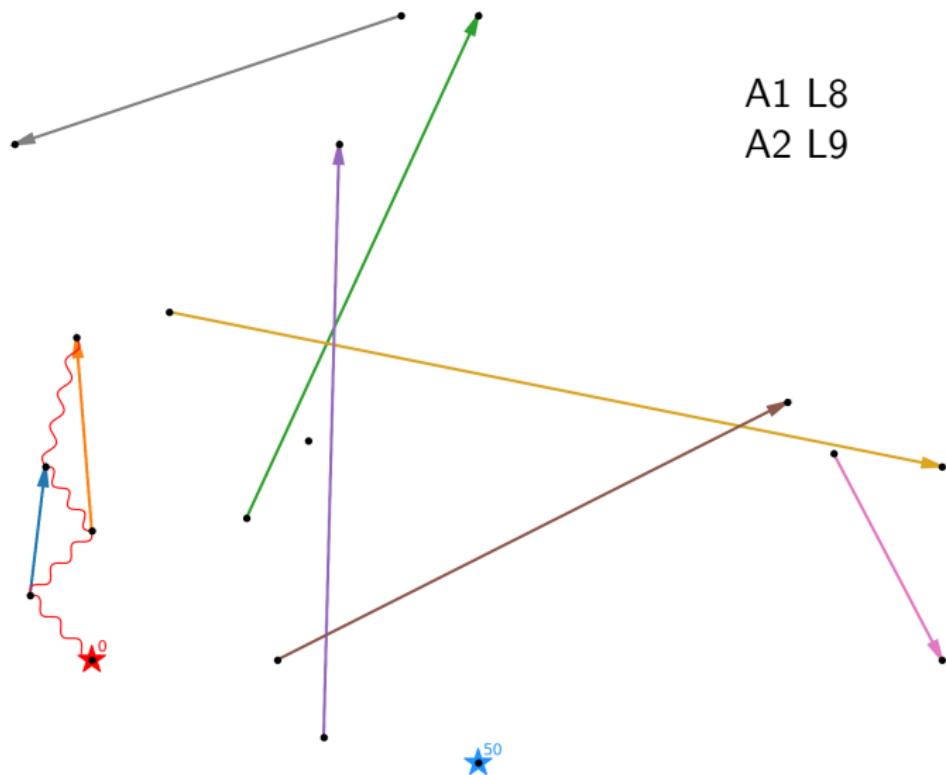
A1 L6



Description through practical example

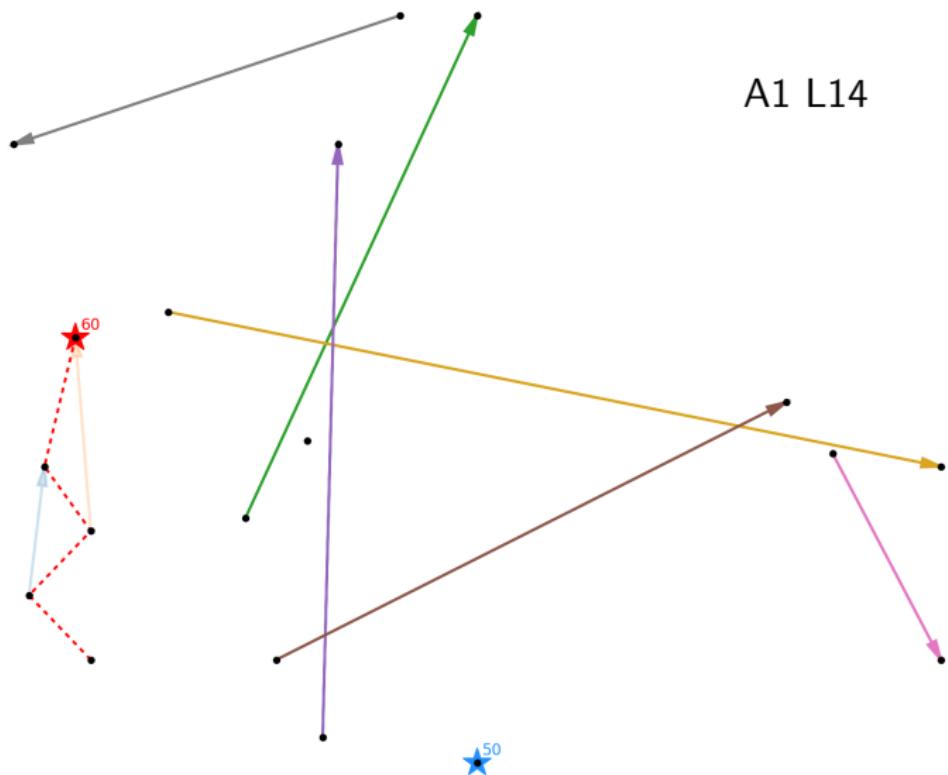


Description through practical example



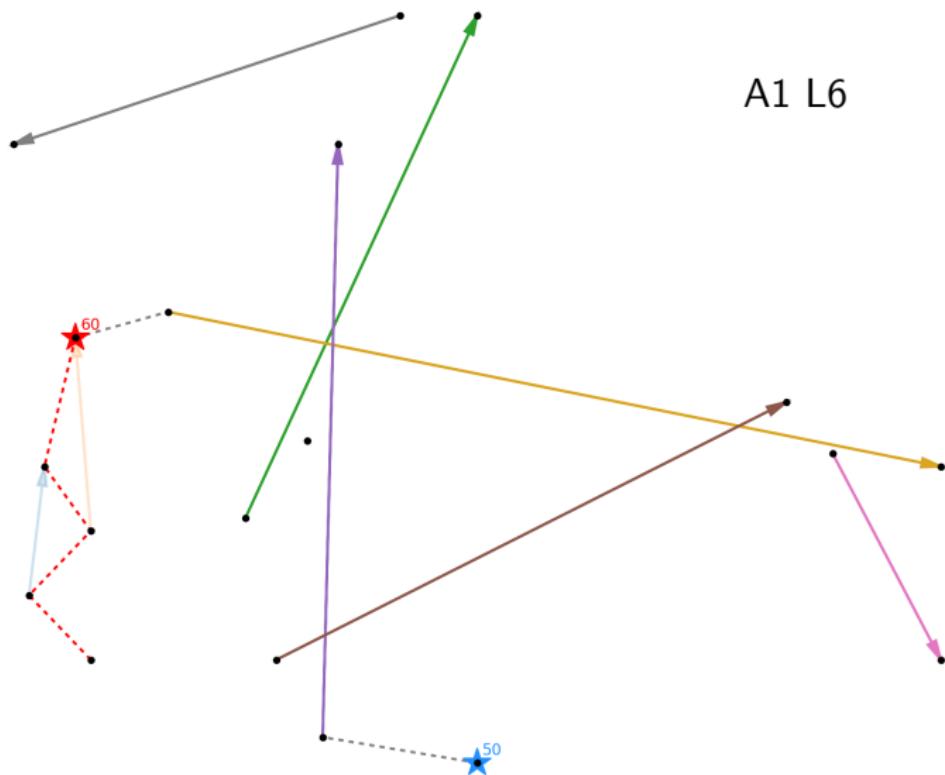
Description through practical example

A1 L14

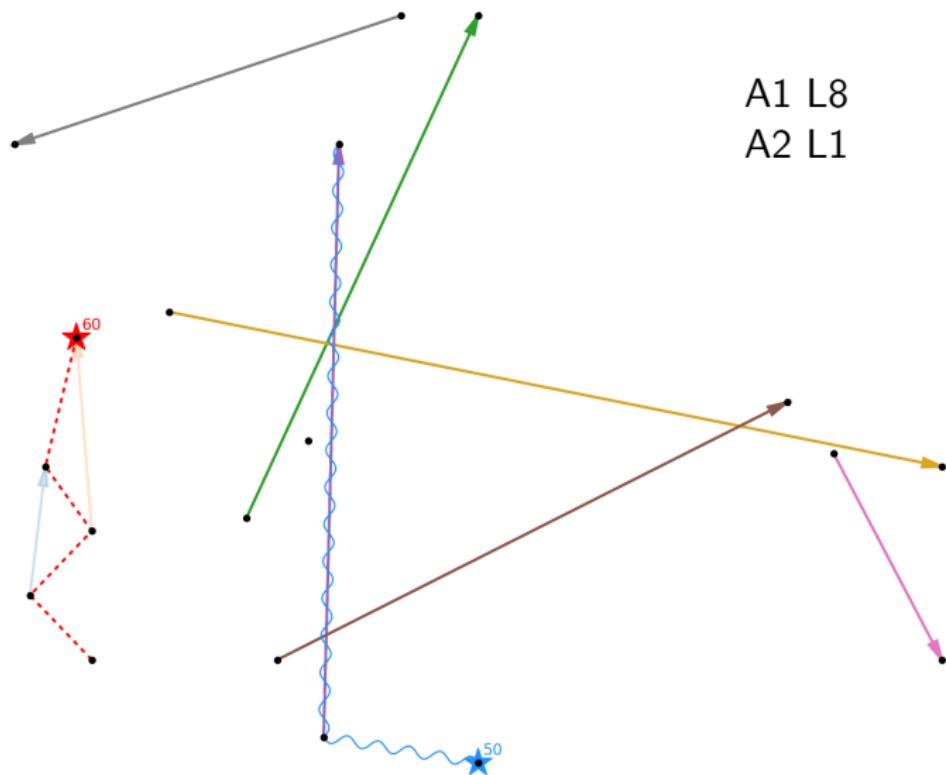


Description through practical example

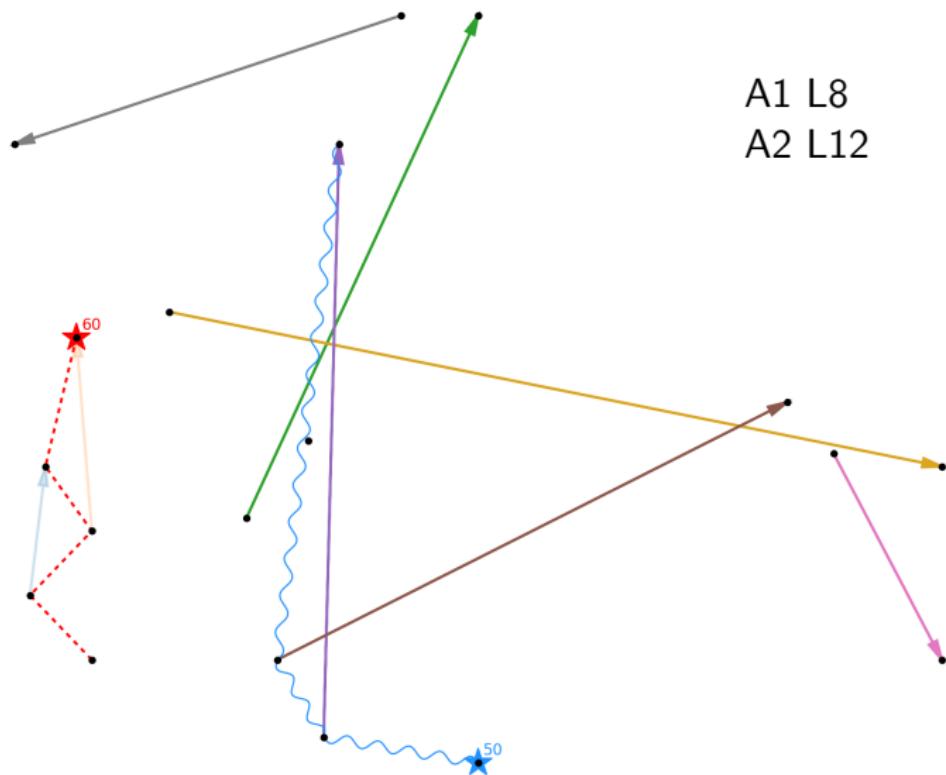
A1 L6



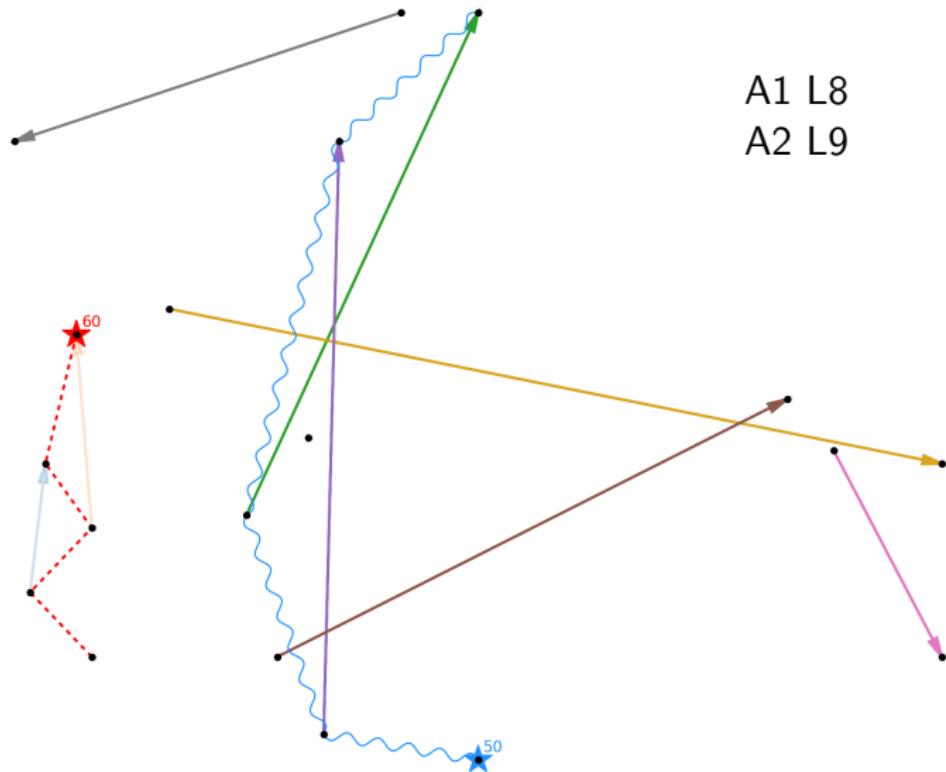
Description through practical example



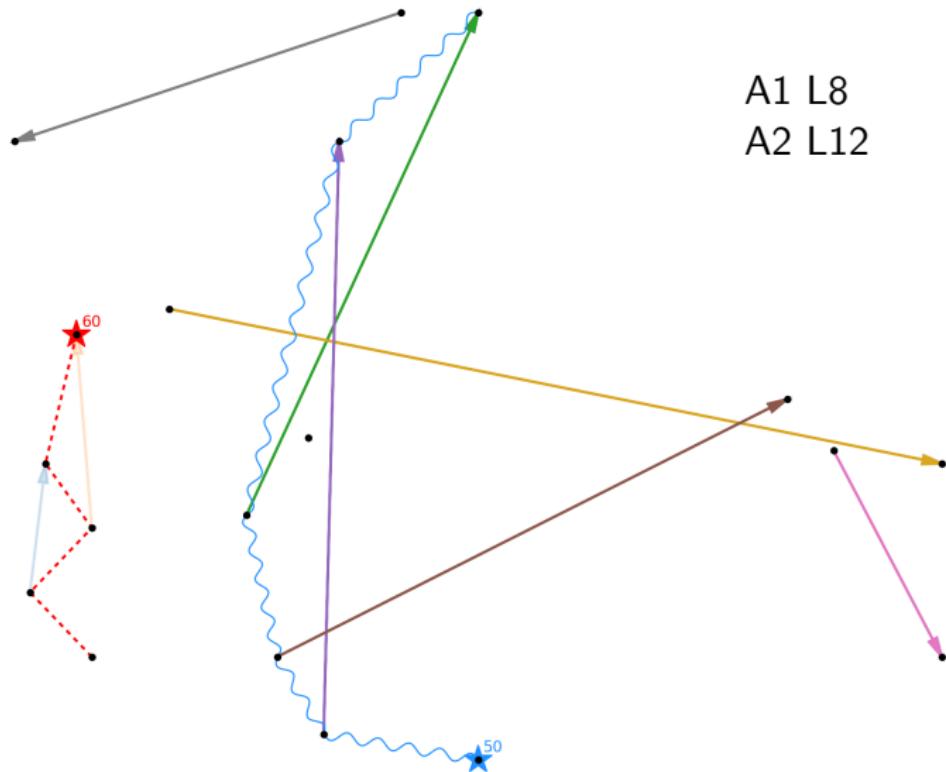
Description through practical example



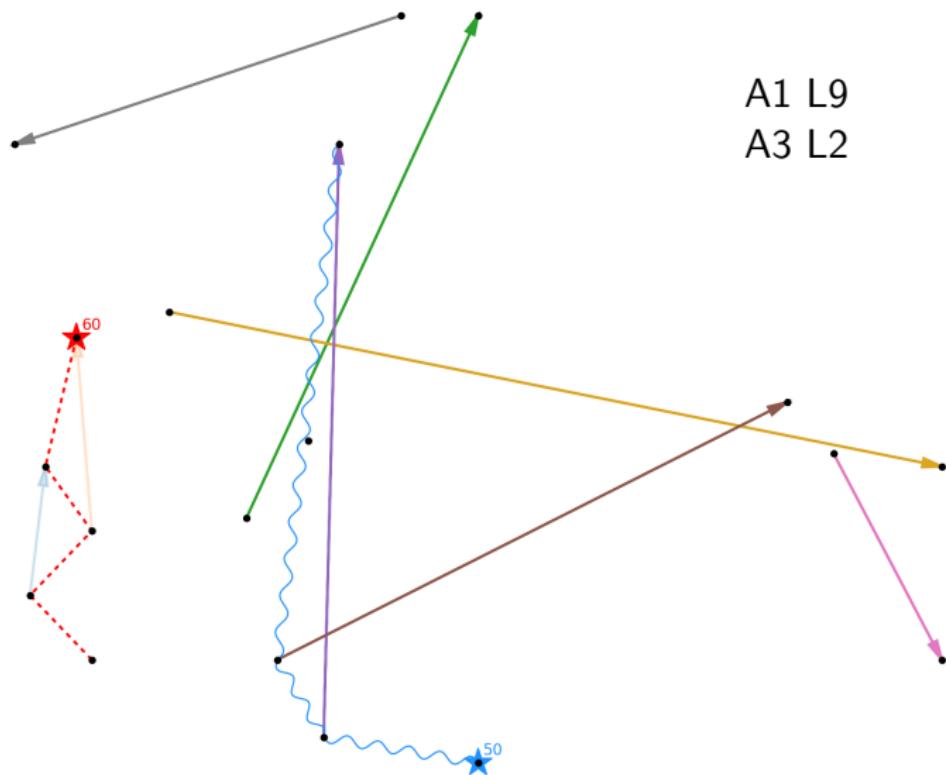
Description through practical example



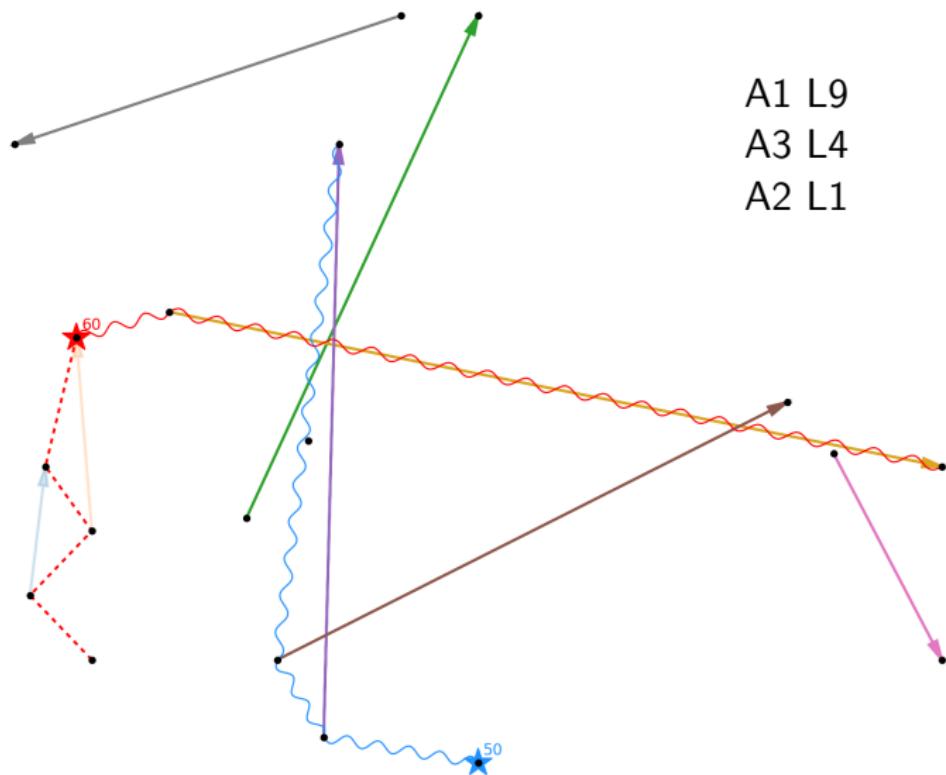
Description through practical example



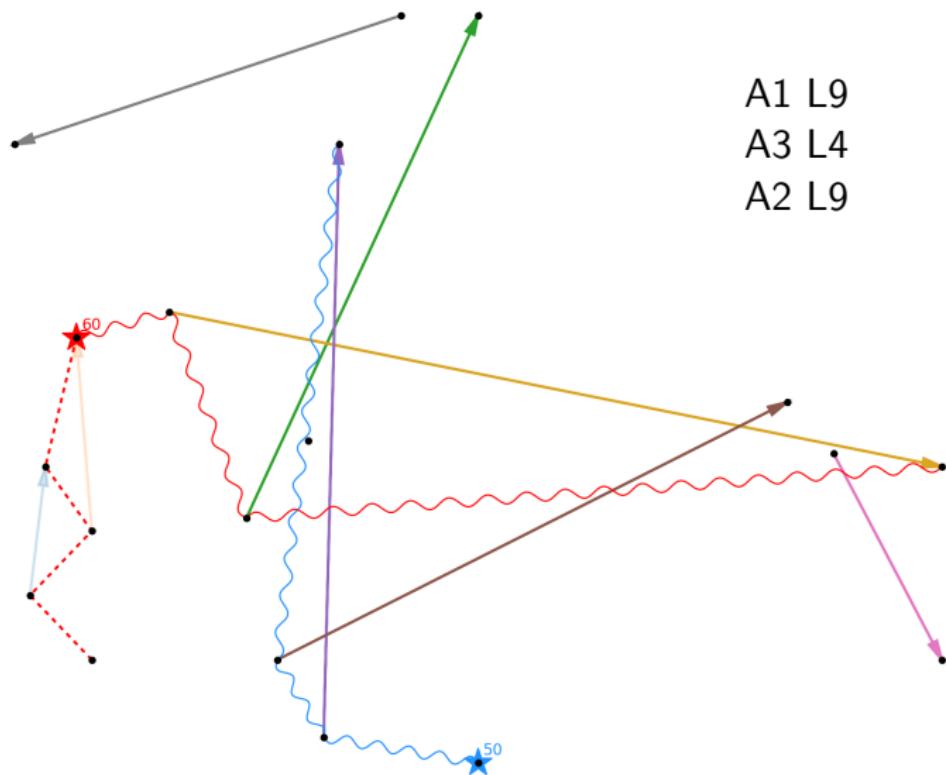
Description through practical example



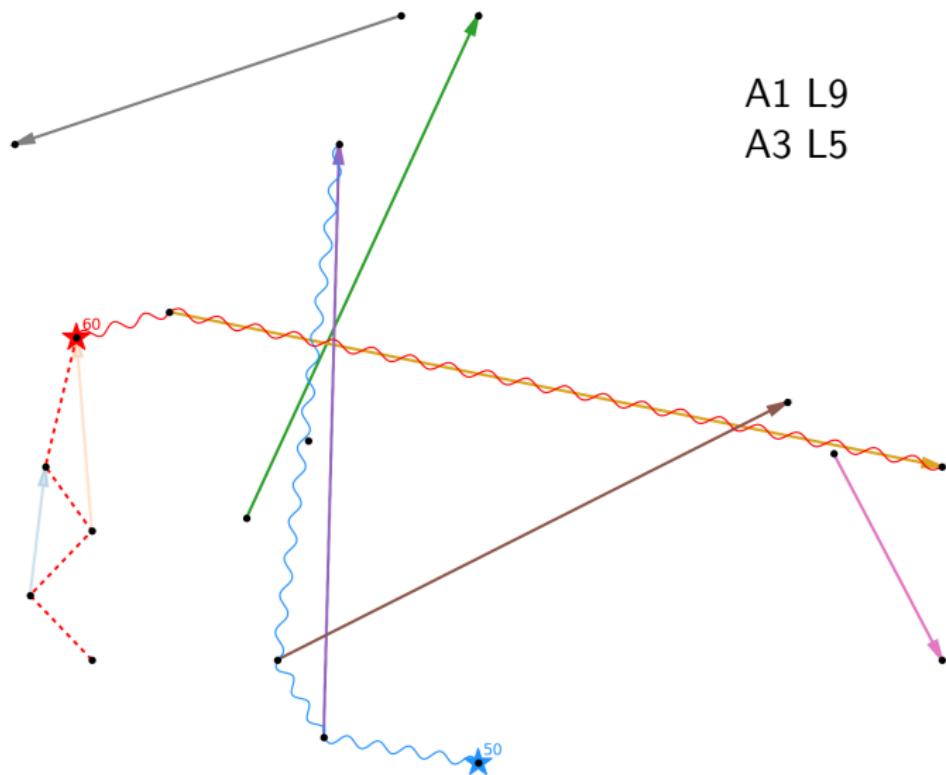
Description through practical example



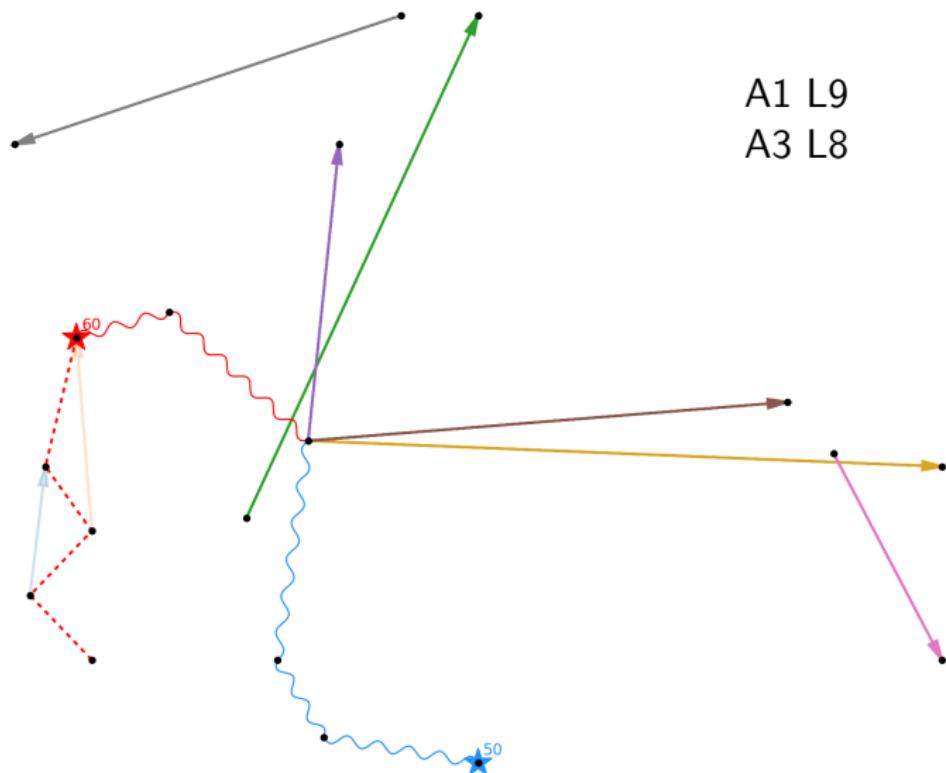
Description through practical example



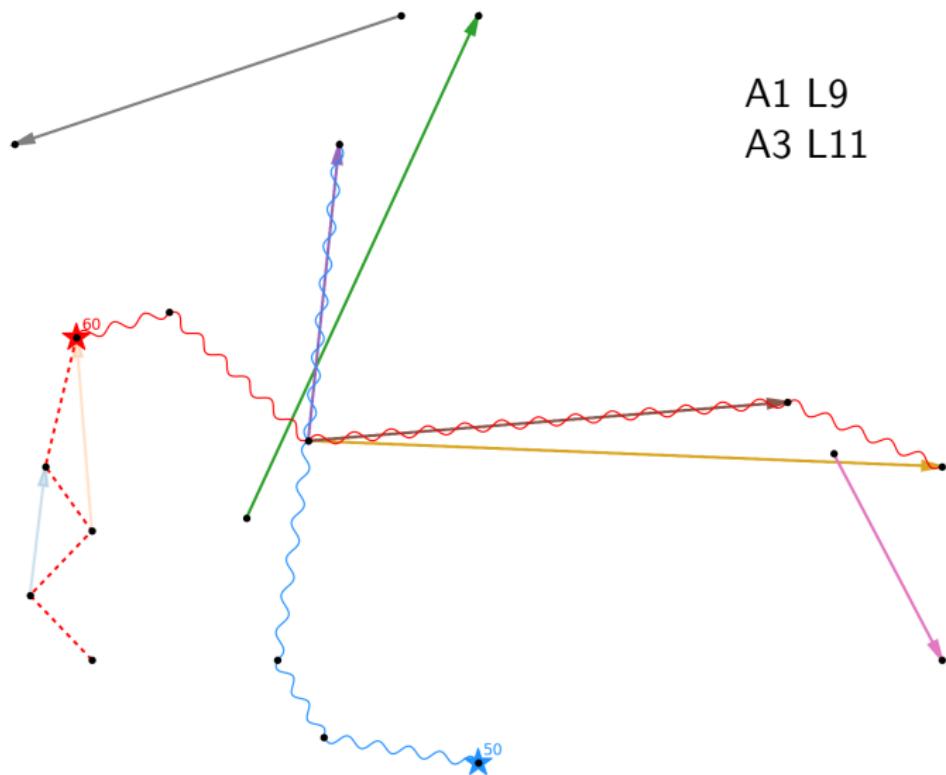
Description through practical example



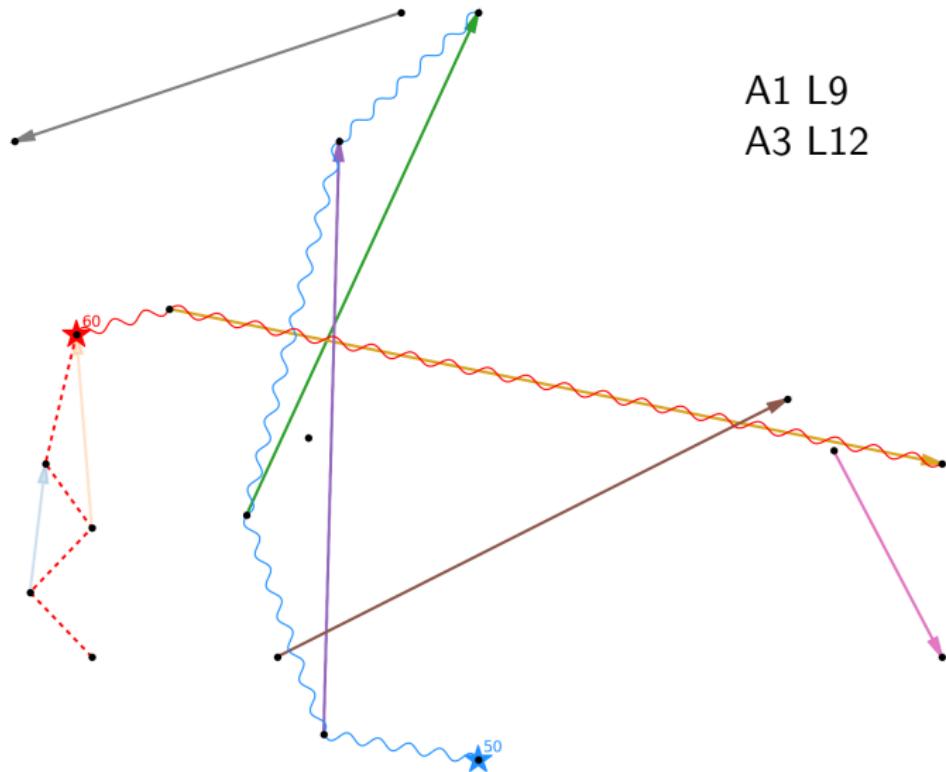
Description through practical example



Description through practical example

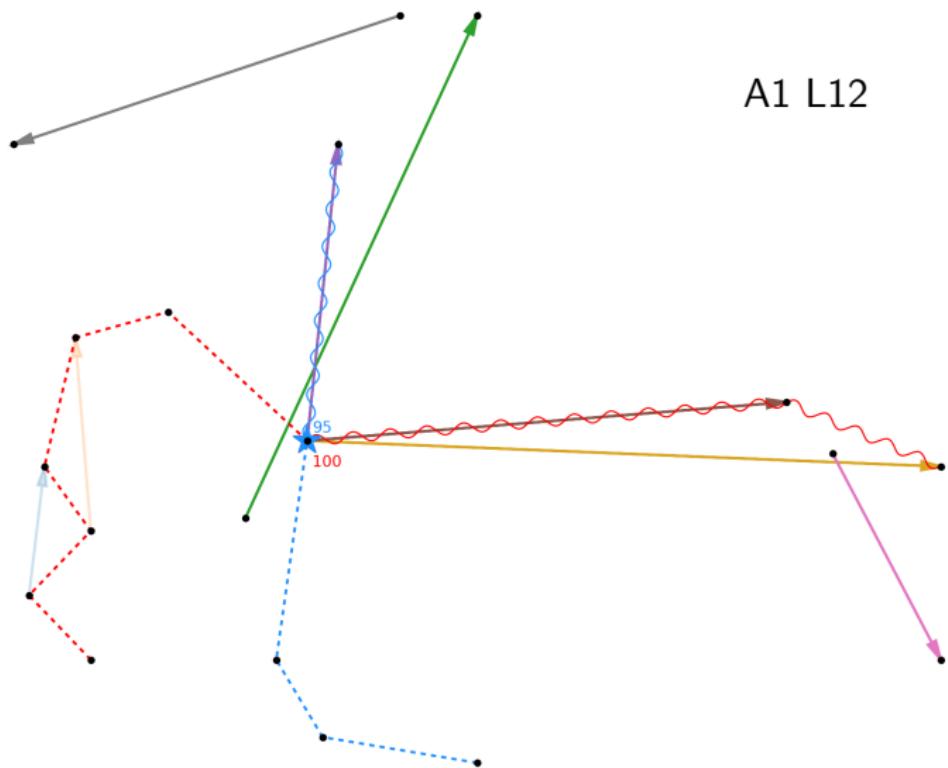


Description through practical example

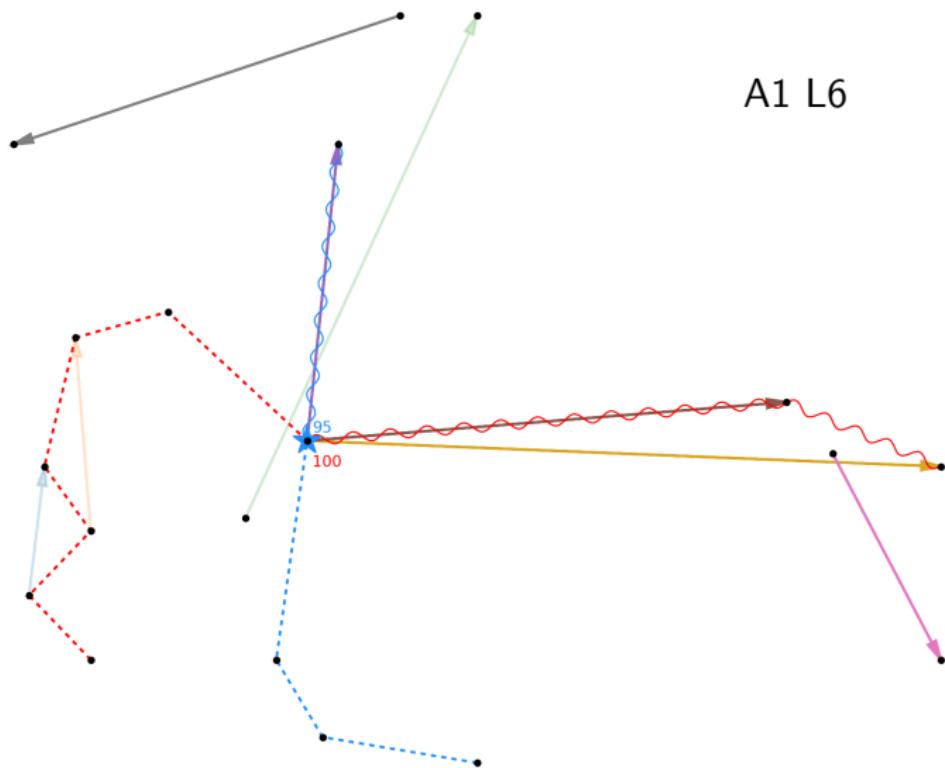


Description through practical example

A1 L12

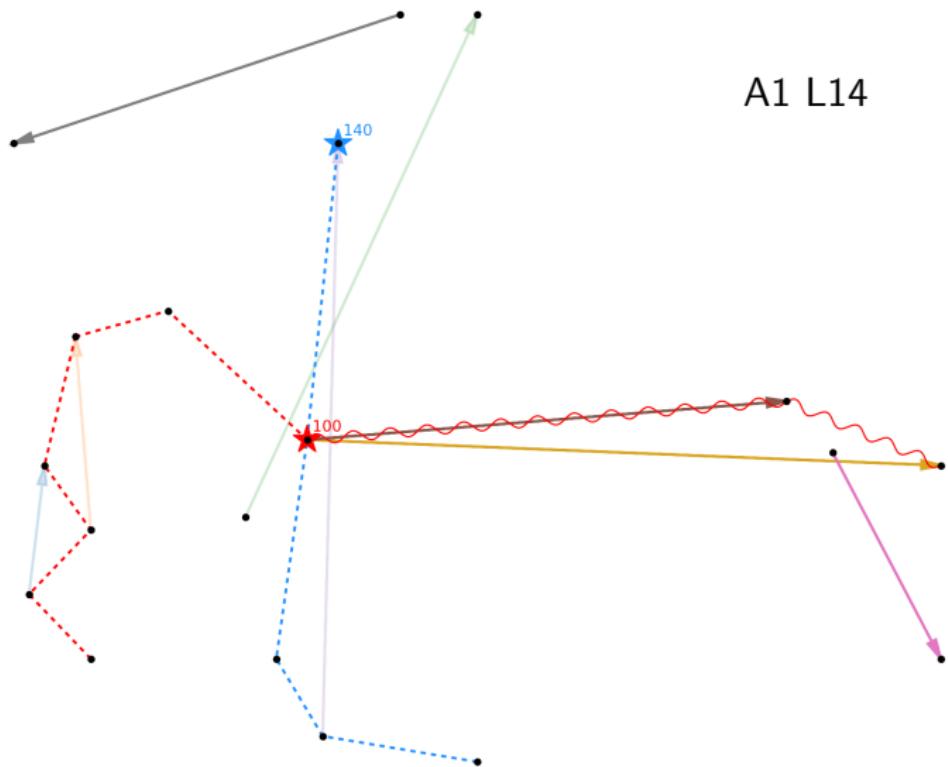


Description through practical example

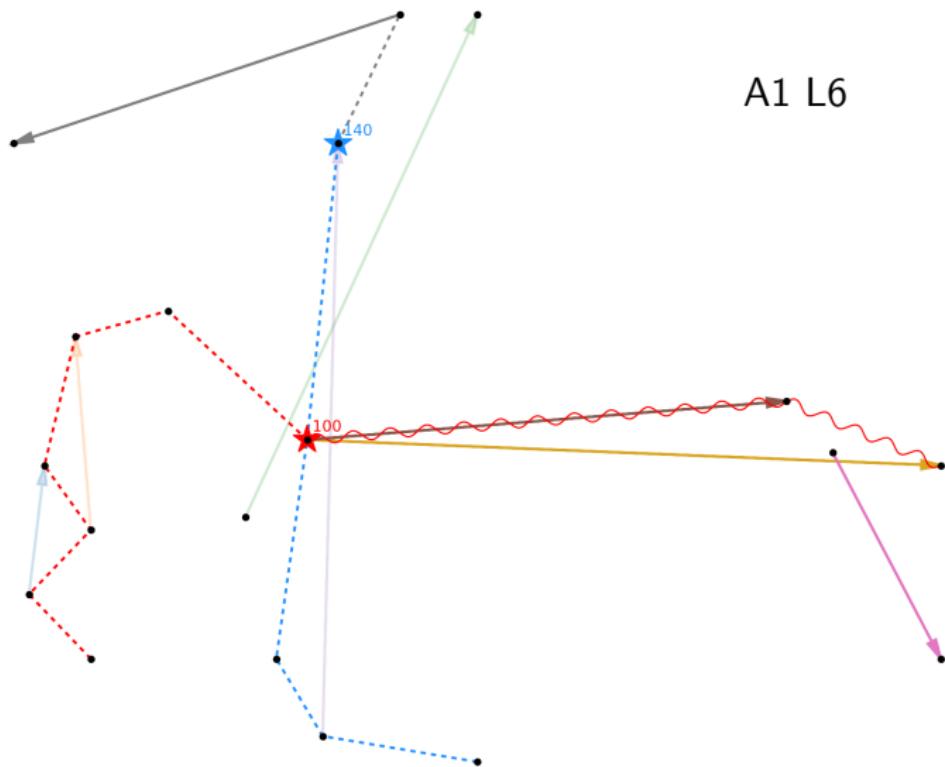


Description through practical example

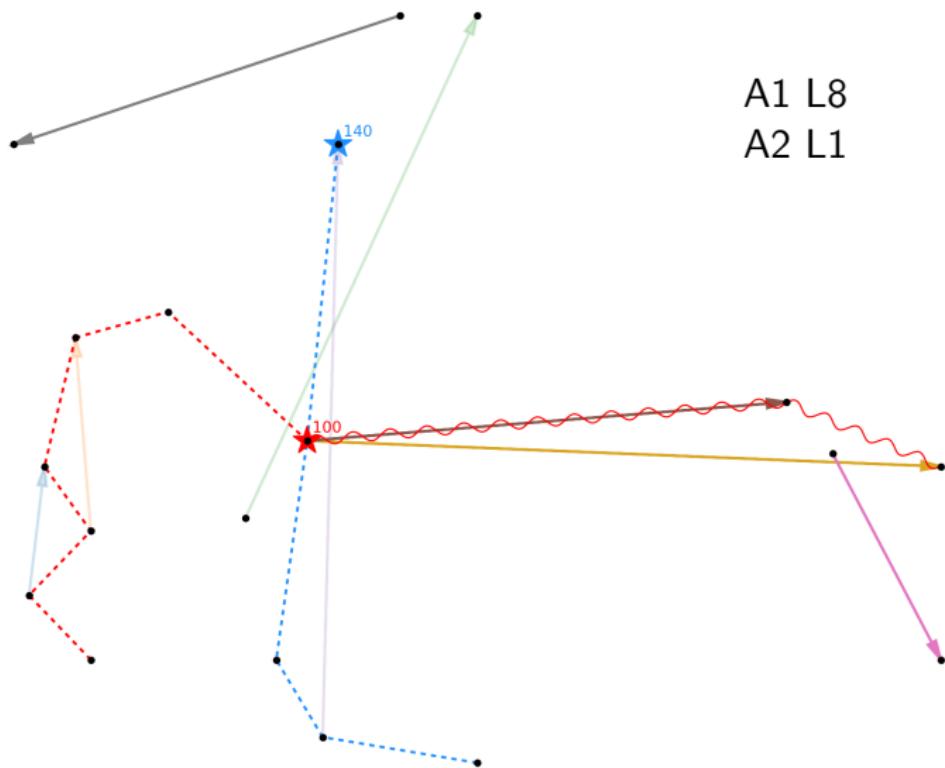
A1 L14



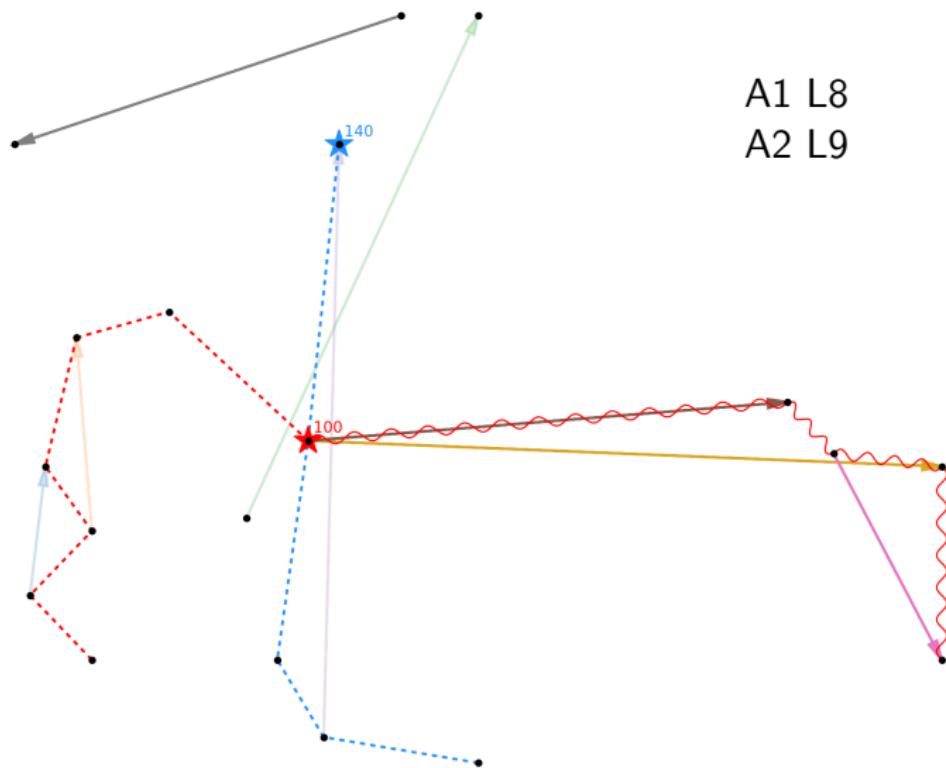
Description through practical example



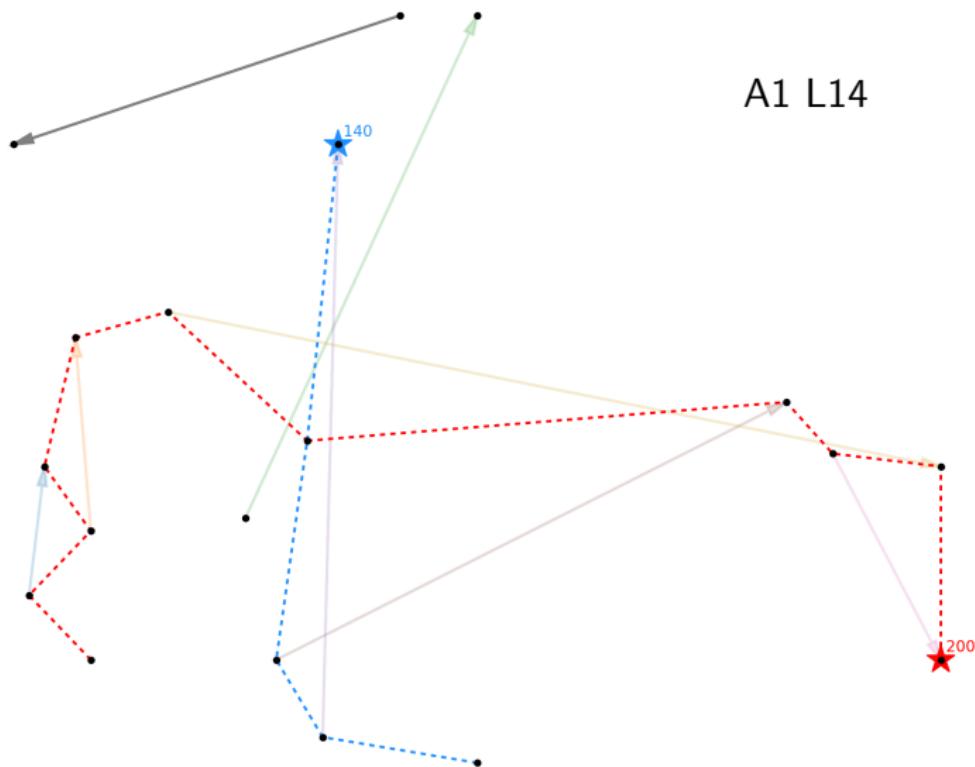
Description through practical example



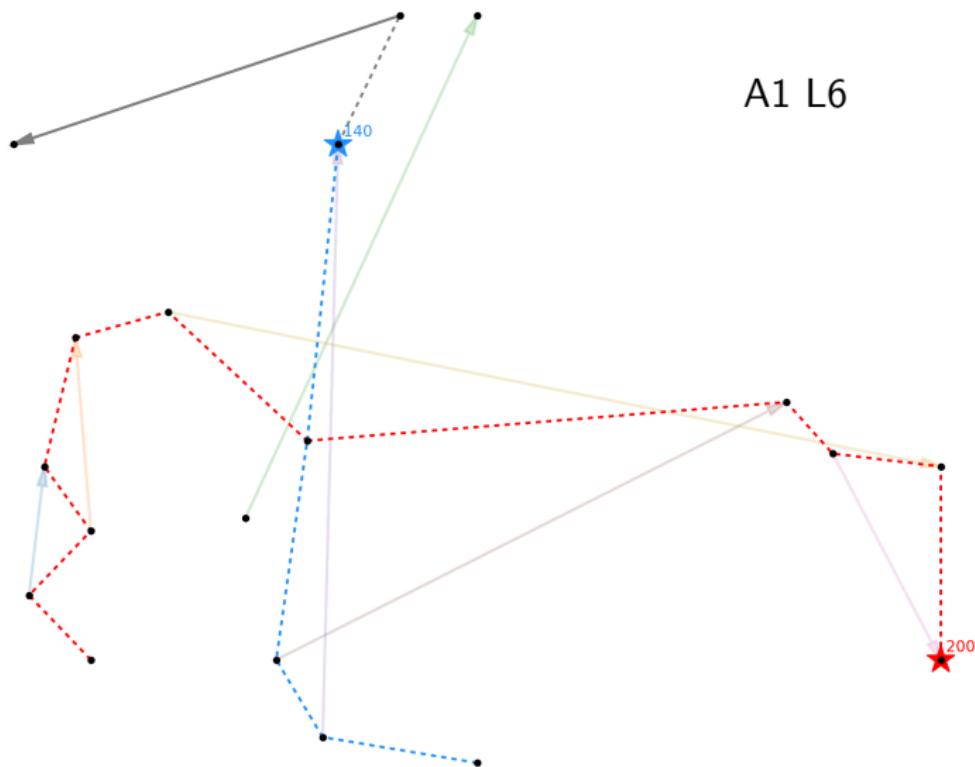
Description through practical example



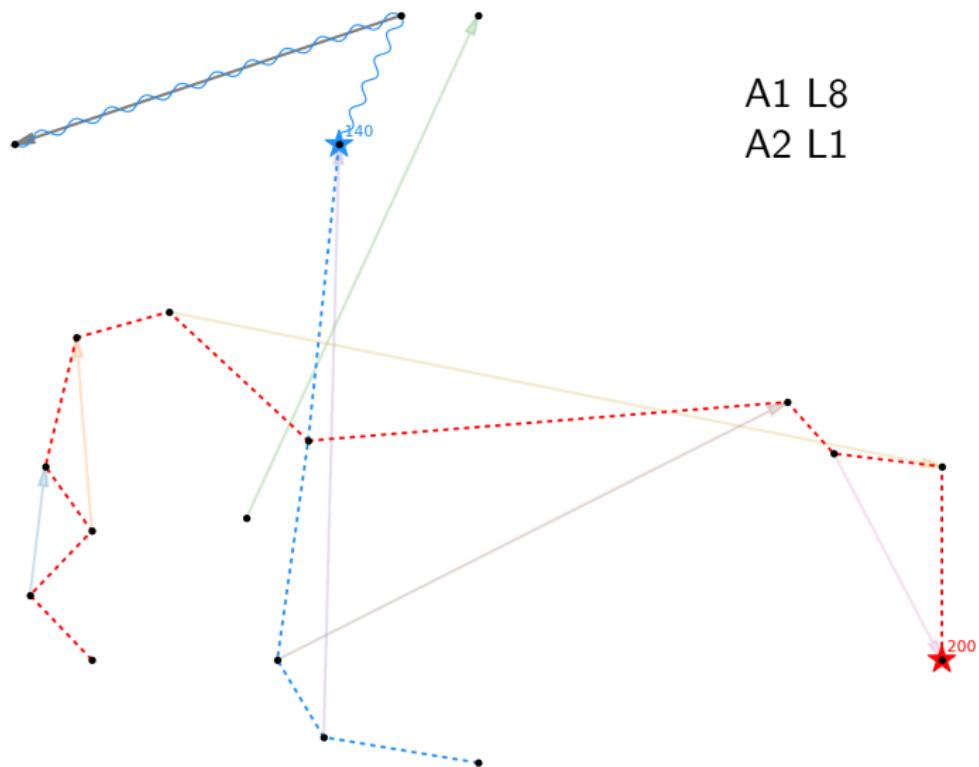
Description through practical example



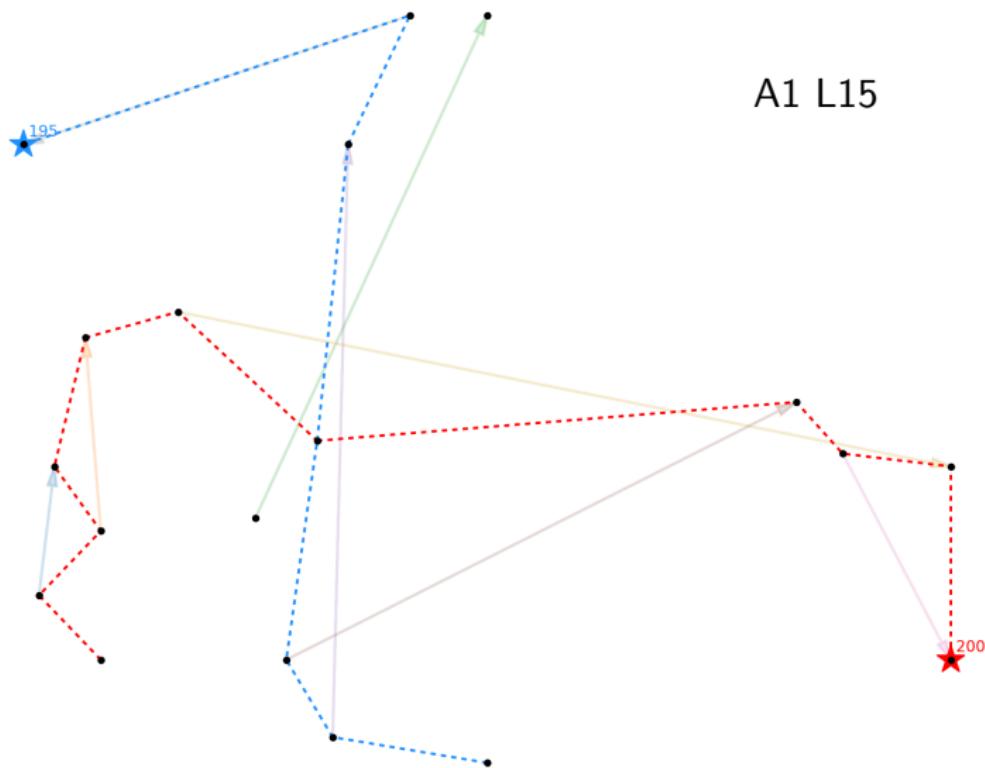
Description through practical example



Description through practical example



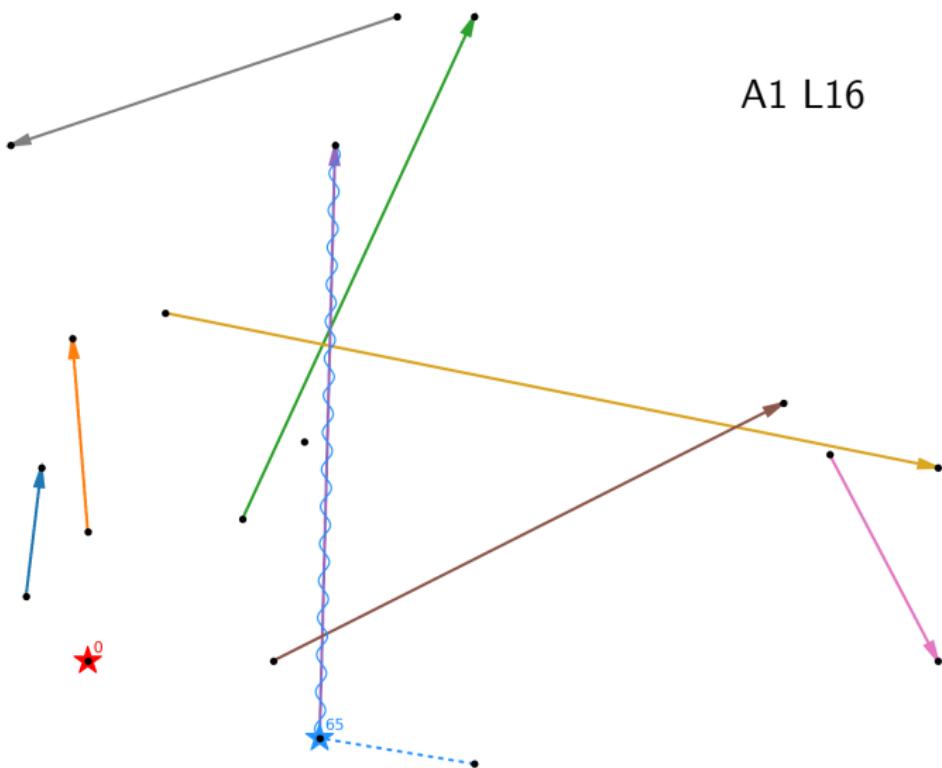
Description through practical example



A1 L15

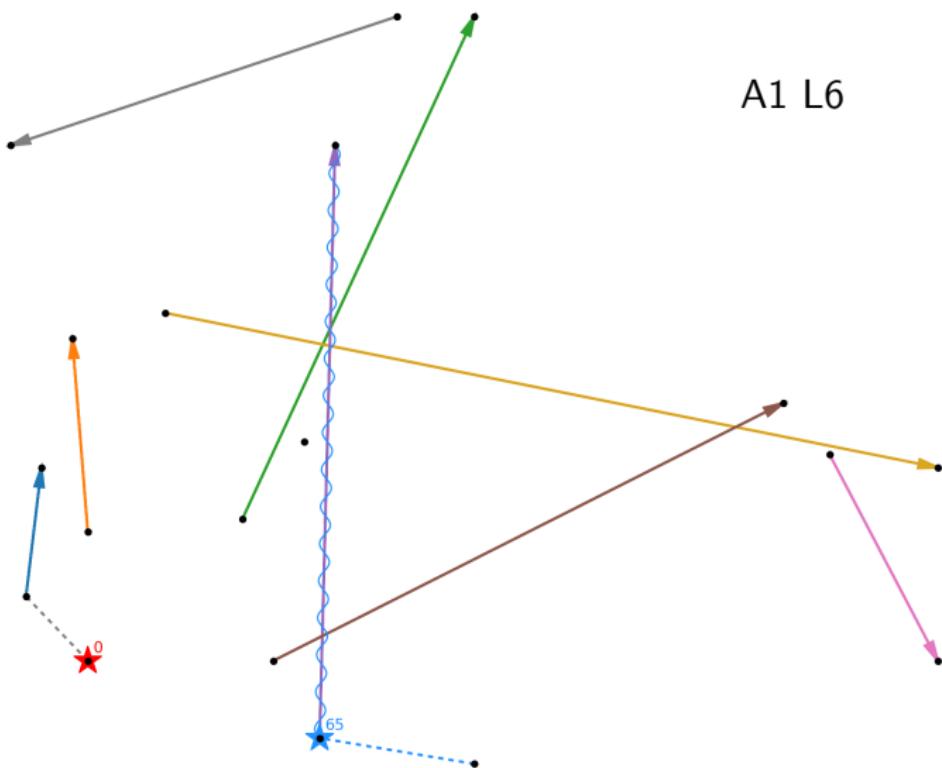
Description through practical example

A1 L16

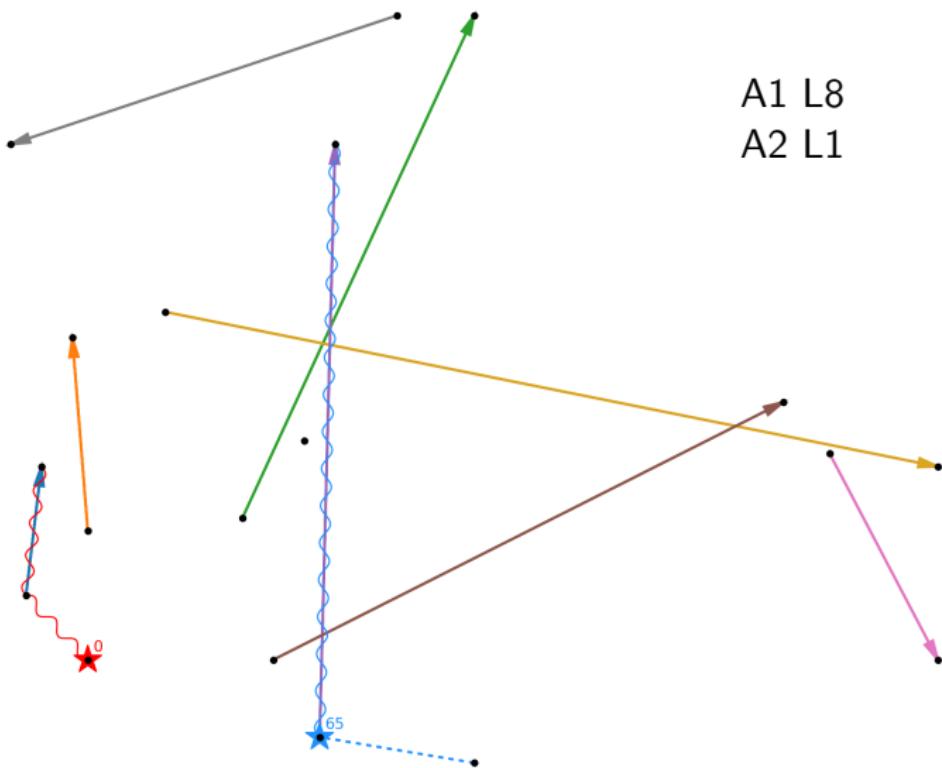


Description through practical example

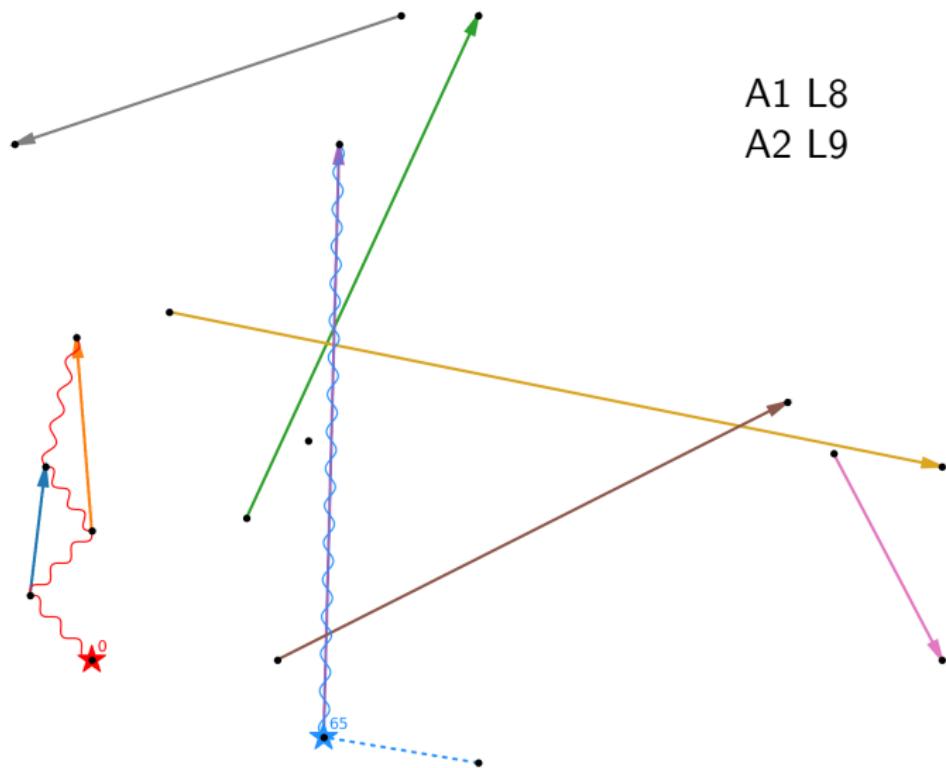
A1 L6



Description through practical example

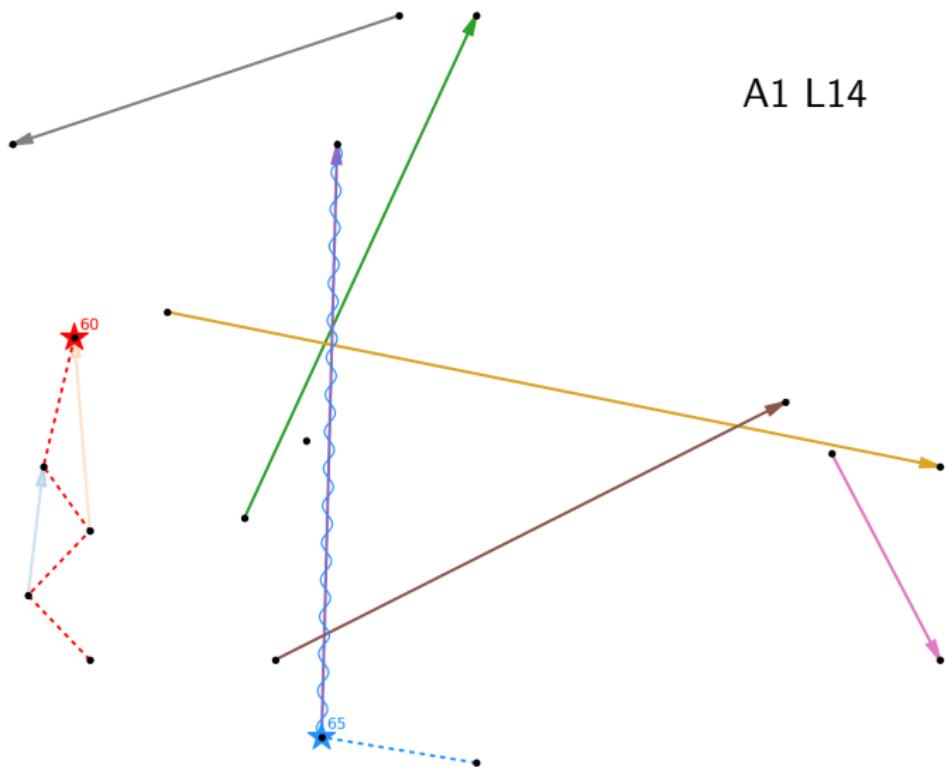


Description through practical example



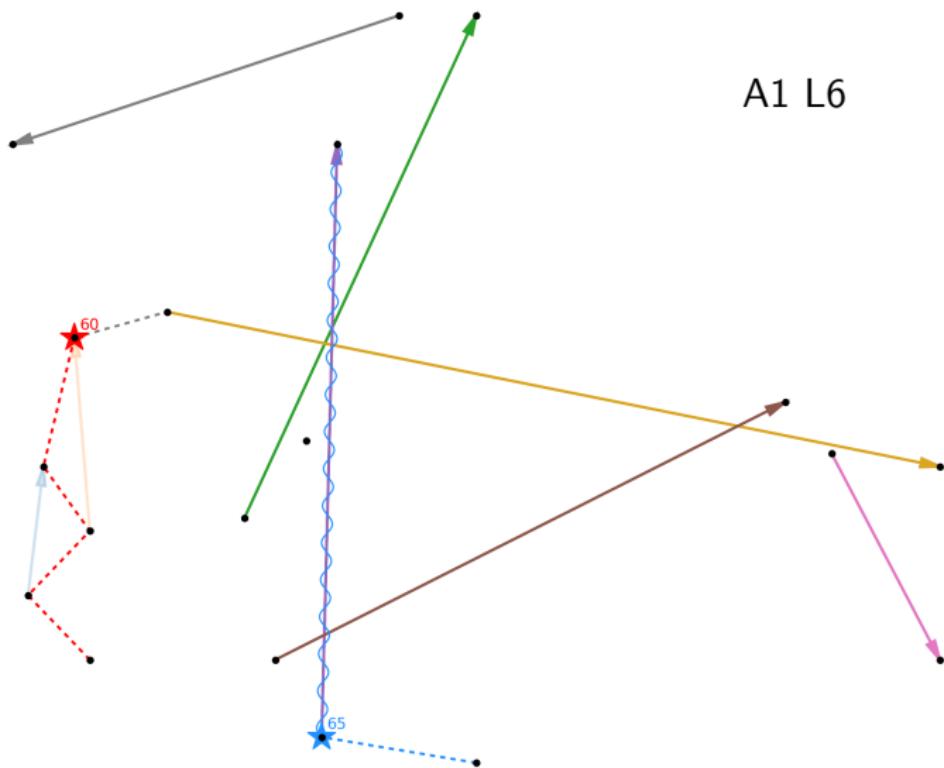
Description through practical example

A1 L14

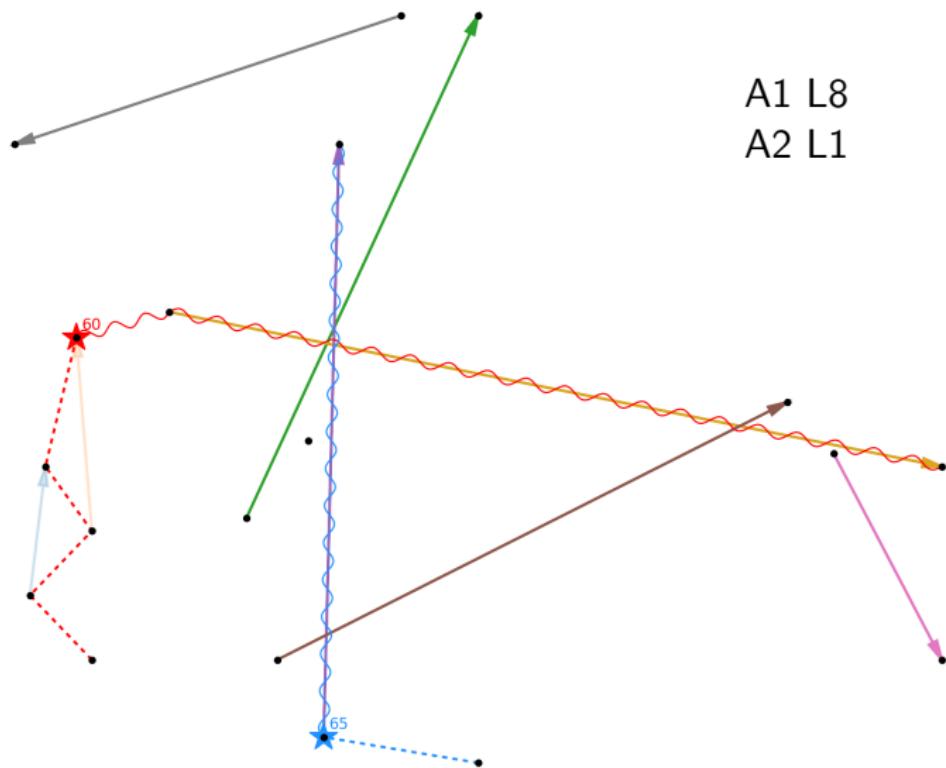


Description through practical example

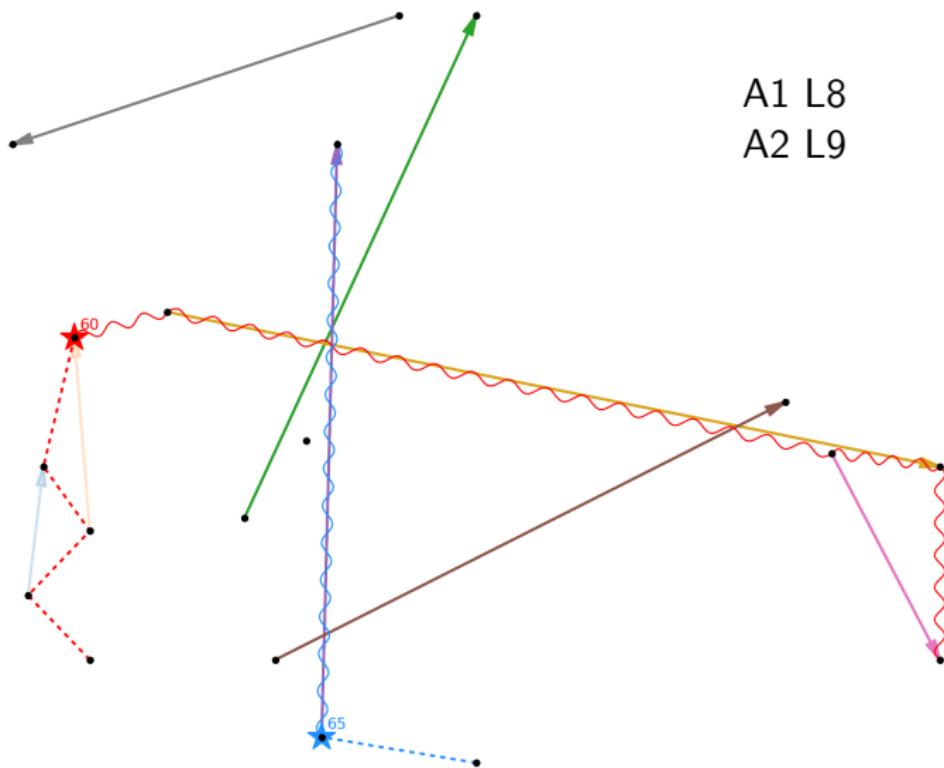
A1 L6



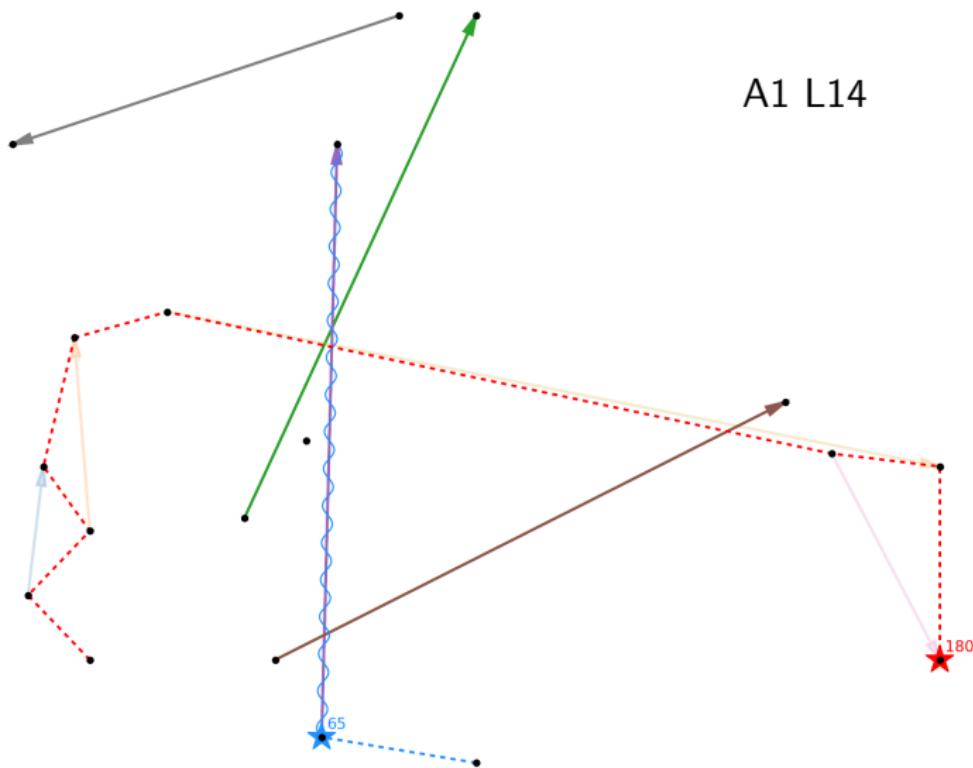
Description through practical example



Description through practical example

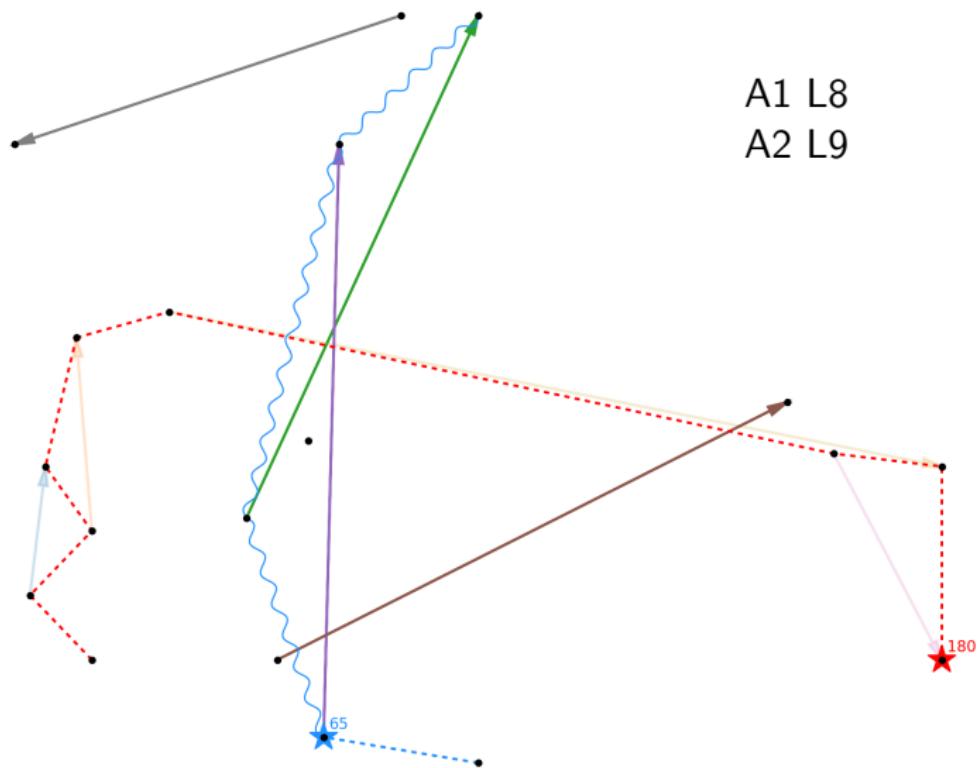


Description through practical example

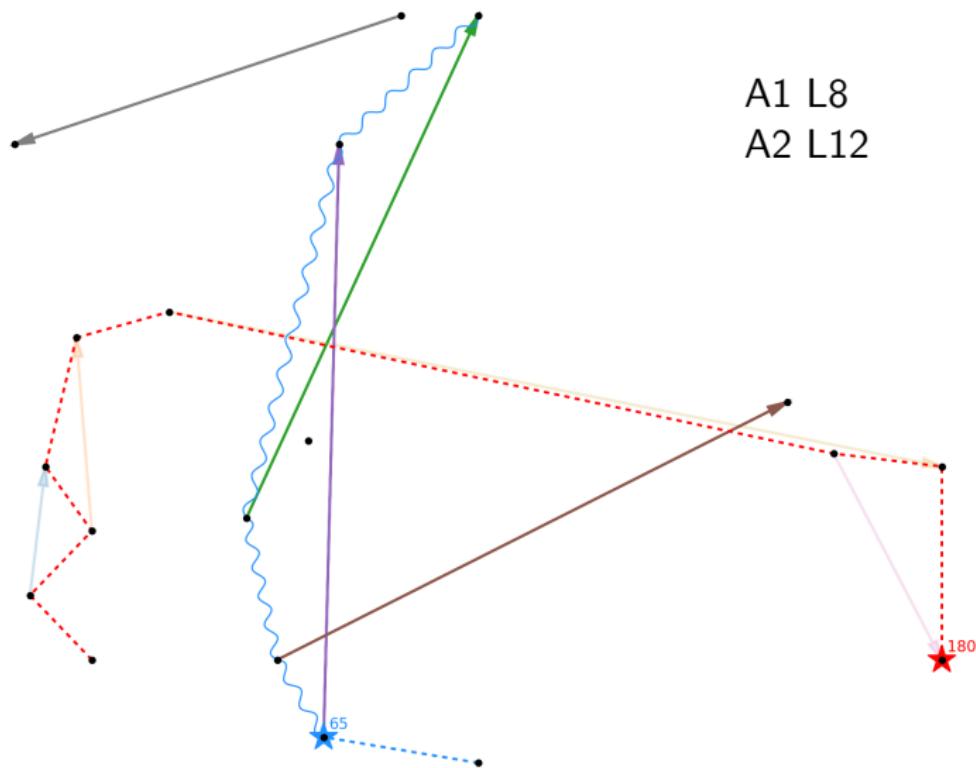


A1 L14

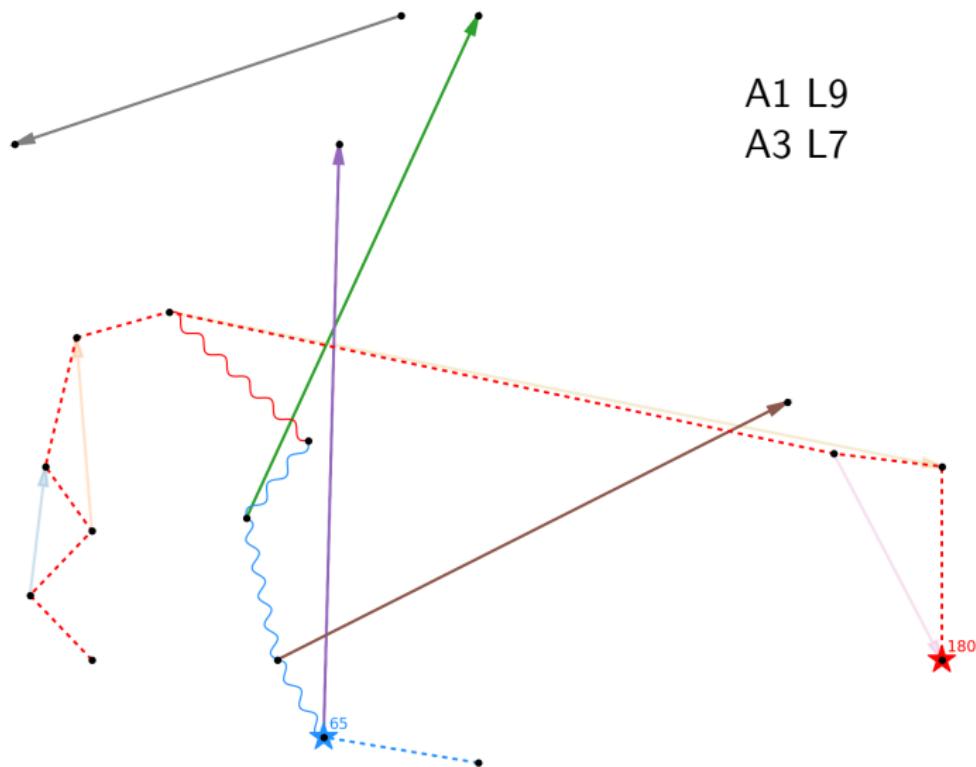
Description through practical example



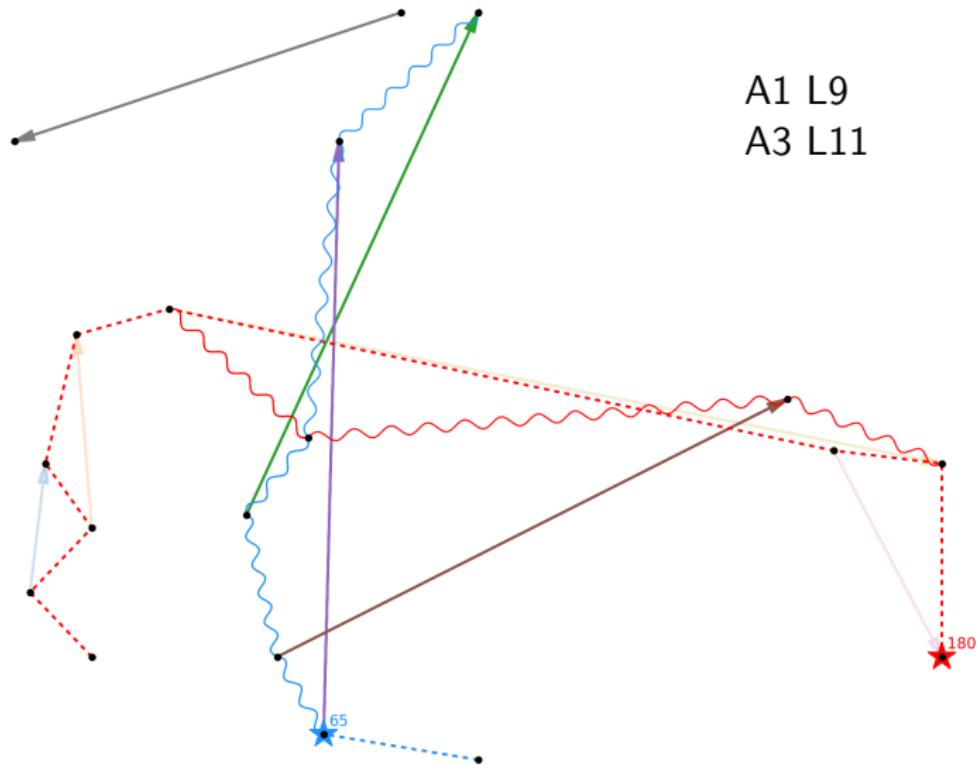
Description through practical example



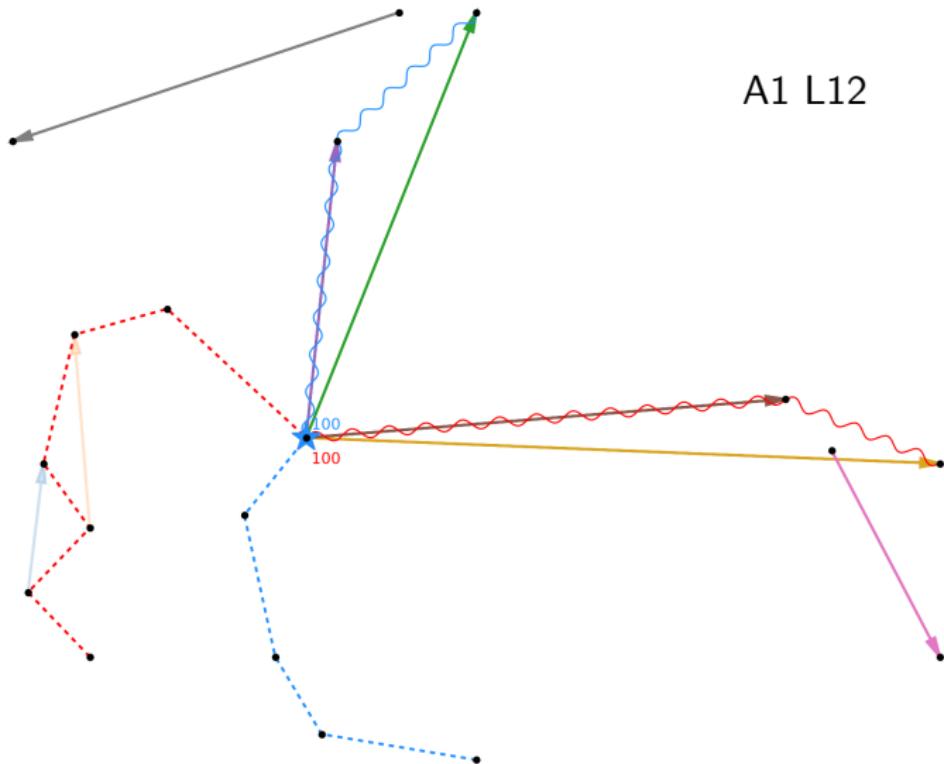
Description through practical example



Description through practical example

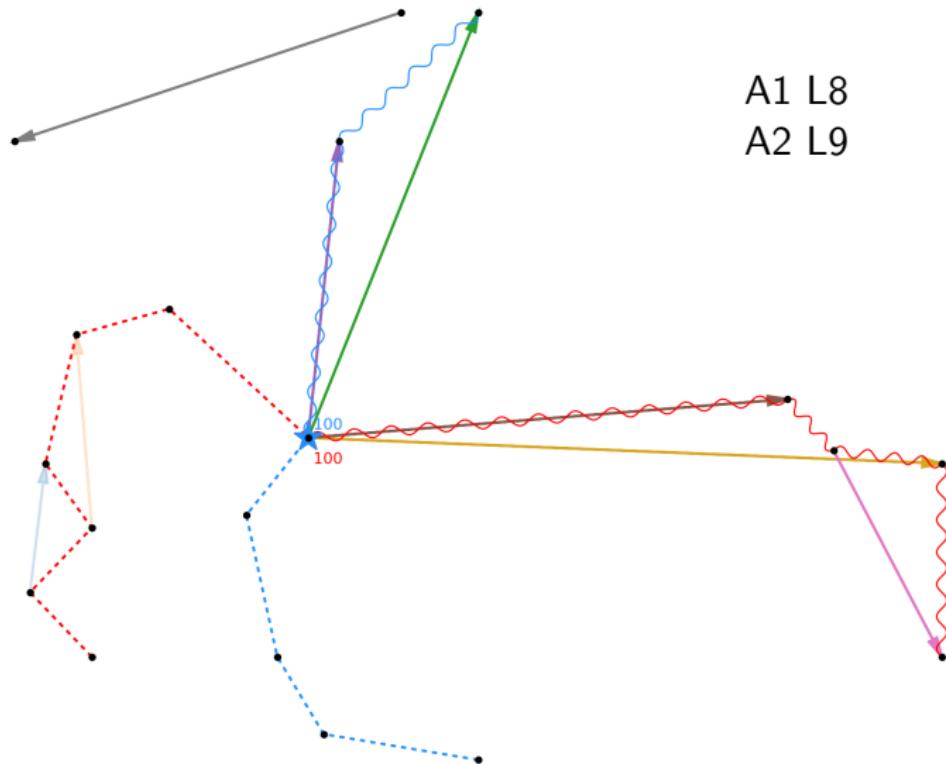


Description through practical example

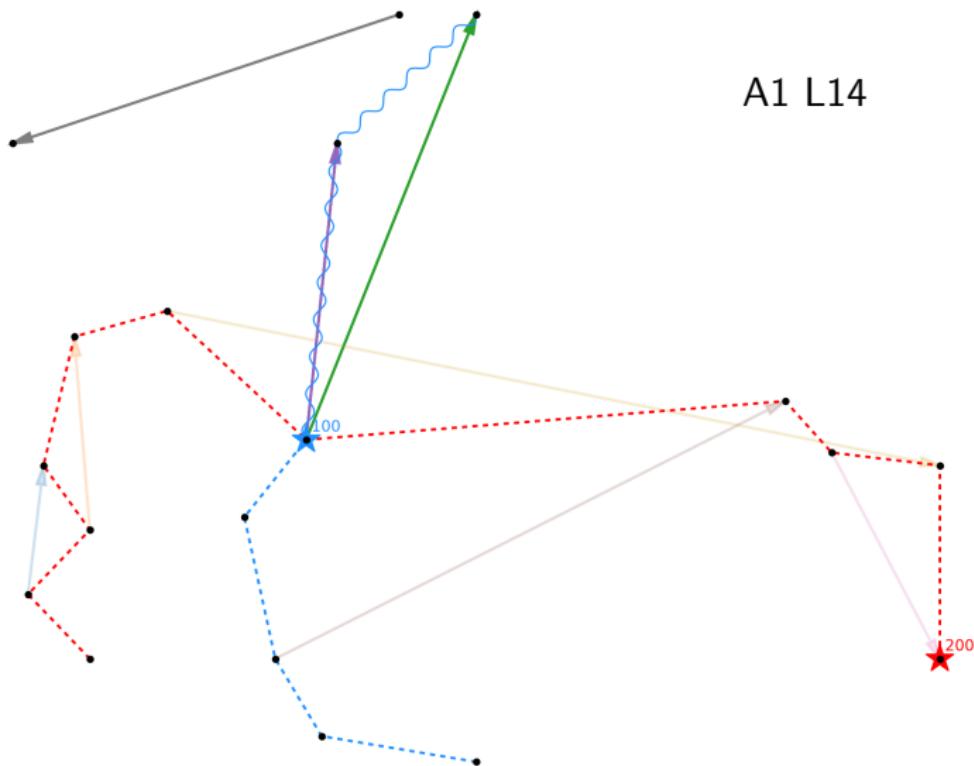


A1 L12

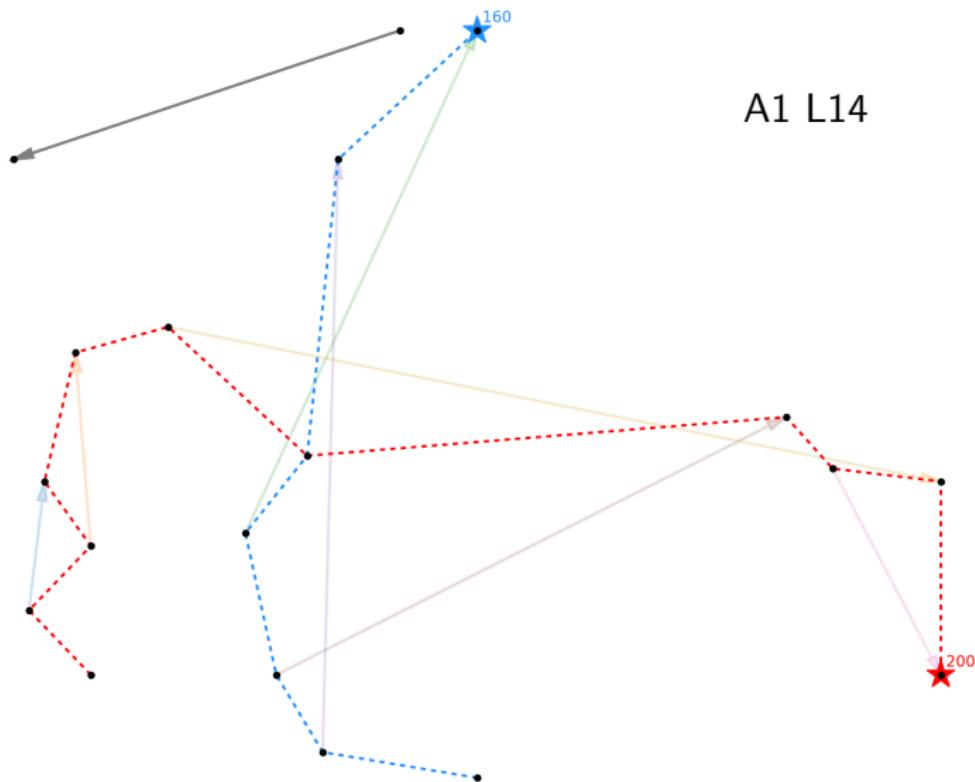
Description through practical example



Description through practical example

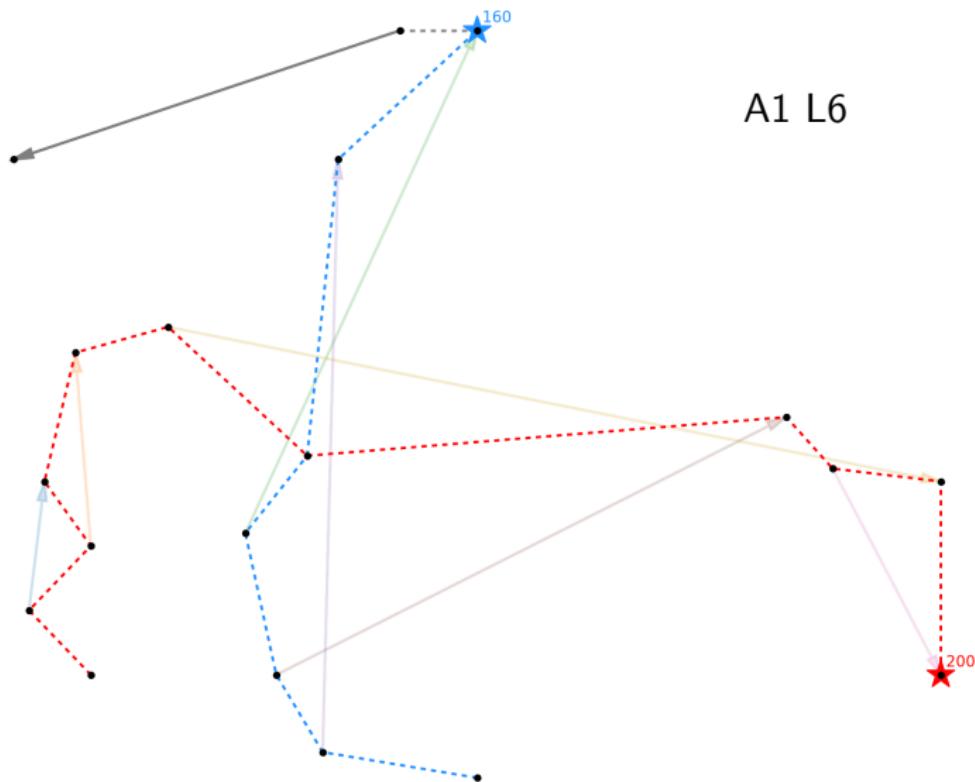


Description through practical example



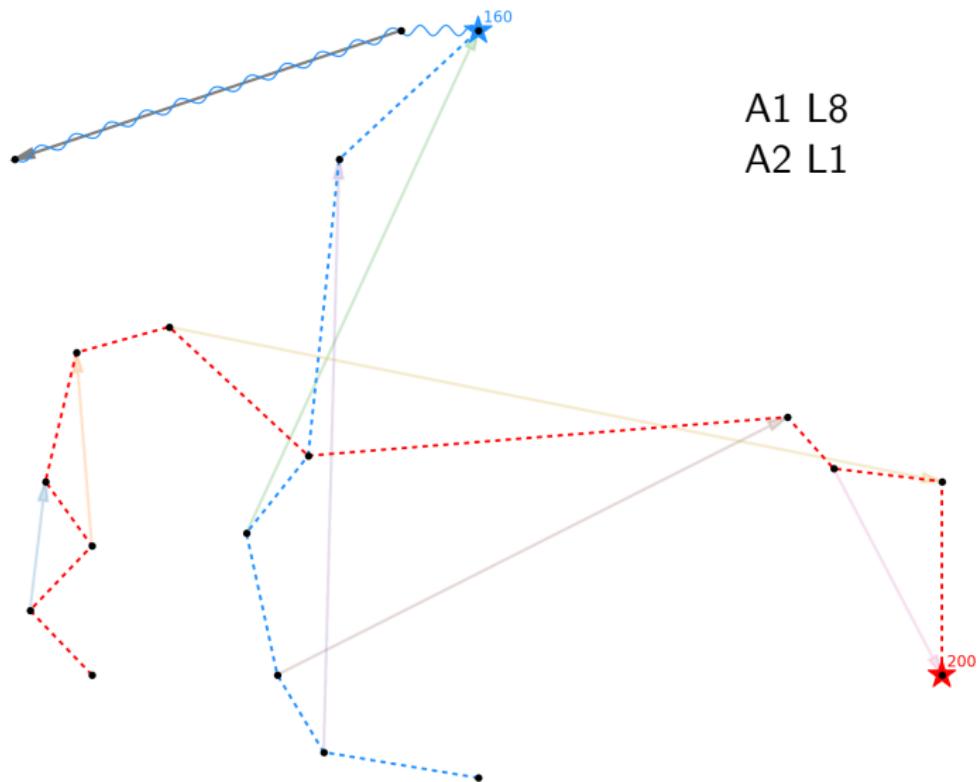
A1 L14

Description through practical example

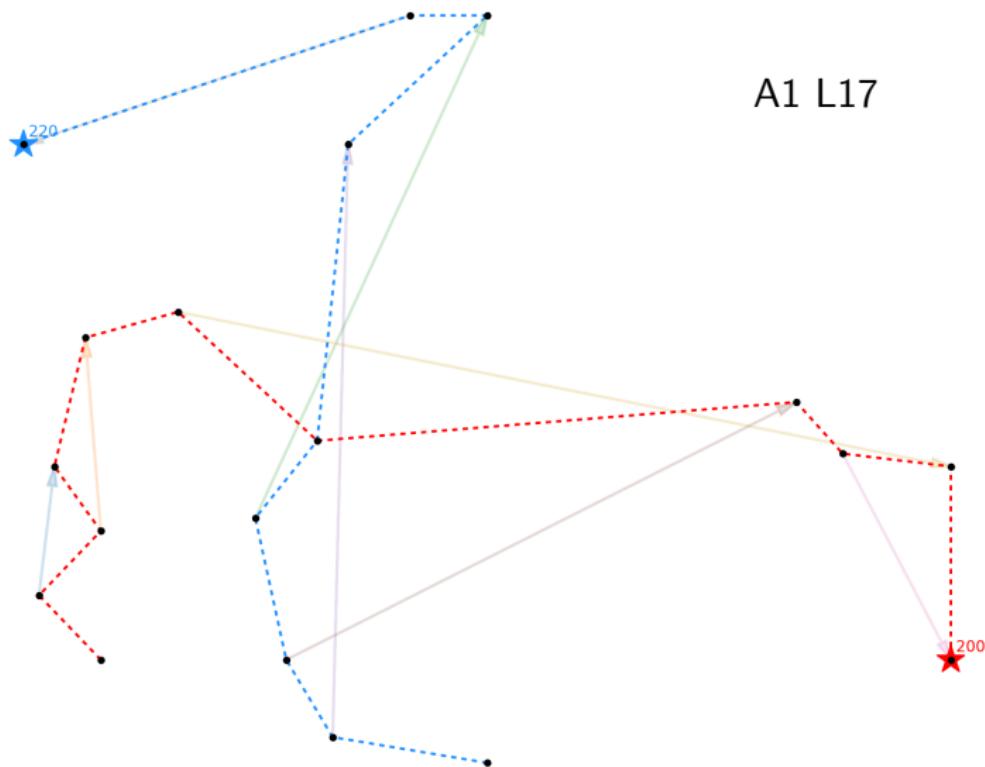


A1 L6

Description through practical example



Description through practical example



A1 L17

Results

Solution generation

Every instance was solved thrice:

- Without transfers

Solution generation

Every instance was solved thrice:

- Without transfers
- With transfers with the solution without transfer as initial solution

Solution generation

Every instance was solved thrice:

- Without transfers
- With transfers with the solution without transfer as initial solution
- With transfers without initial solution

Result summary

- The larger the instance, the lower the objective function value

Result summary

- The larger the instance, the lower the objective function value
- Compared to an individual vehicle scenario
 - Reduction of the fleet size by a third
 - Reduction of total distance driven by up to 25%

Result summary

- The larger the instance, the lower the objective function value
- Compared to an individual vehicle scenario
 - Reduction of the fleet size by a third
 - Reduction of total distance driven by up to 25%
- Solutions with transfers are typically 1 to 4% better than solutions without transfers

Result summary

- The larger the instance, the lower the objective function value
- Compared to an individual vehicle scenario
 - Reduction of the fleet size by a third
 - Reduction of total distance driven by up to 25%
- Solutions with transfers are typically 1 to 4% better than solutions without transfers
- Ride-sharing and transfers increase travel time by 8 to 16%

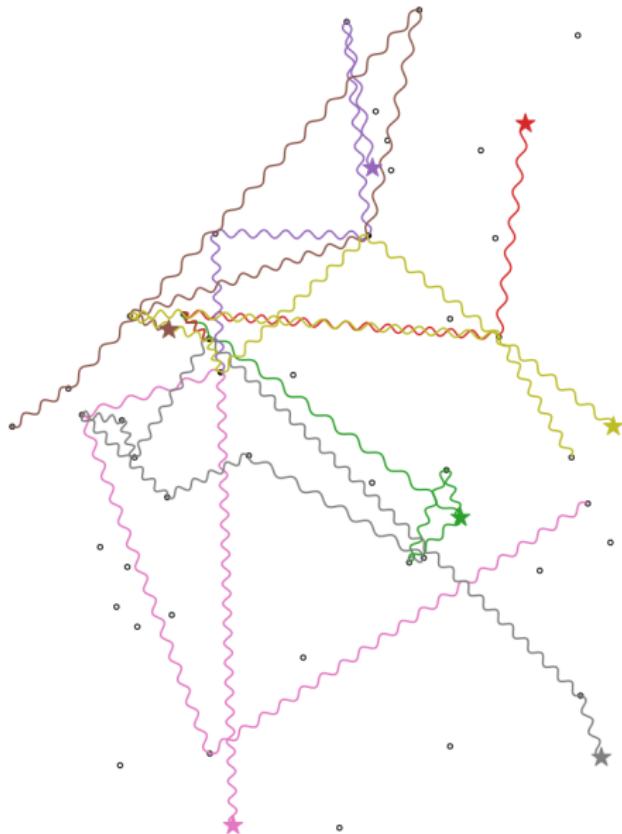
Result summary

- The larger the instance, the lower the objective function value
- Compared to an individual vehicle scenario
 - Reduction of the fleet size by a third
 - Reduction of total distance driven by up to 25%
- Solutions with transfers are typically 1 to 4% better than solutions without transfers
- Ride-sharing and transfers increase travel time by 8 to 16%
- Transfers are responsible for 1-3% of the travel time extension

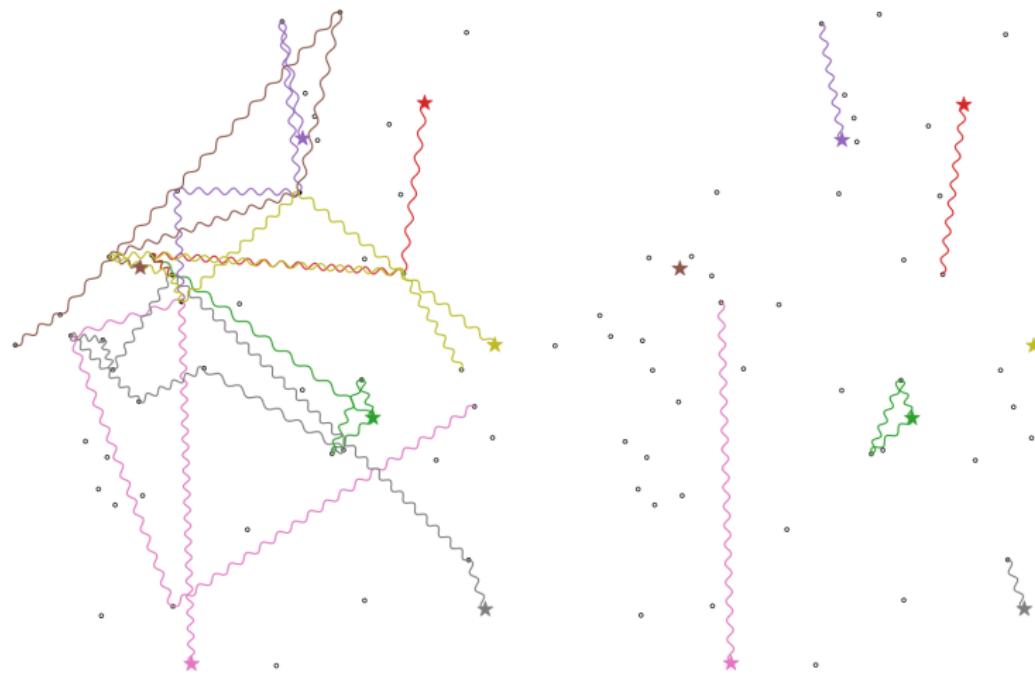
Result summary

- The larger the instance, the lower the objective function value
- Compared to an individual vehicle scenario
 - Reduction of the fleet size by a third
 - Reduction of total distance driven by up to 25%
- Solutions with transfers are typically 1 to 4% better than solutions without transfers
- Ride-sharing and transfers increase travel time by 8 to 16%
- Transfers are responsible for 1-3% of the travel time extension
- Transfers increase computation time by a factor of 2 to 3

Solution robustness



Solution robustness



Conclusion

Conclusion

- We developed a heuristic to solve the DARP-T

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy
- Compared to a scenario with individual vehicles, ride-sharing and transfers can significantly reduce the number of vehicles required and the total distance driven

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy
- Compared to a scenario with individual vehicles, ride-sharing and transfers can significantly reduce the number of vehicles required and the total distance driven
- Transfers further improve the solution

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy
- Compared to a scenario with individual vehicles, ride-sharing and transfers can significantly reduce the number of vehicles required and the total distance driven
- Transfers further improve the solution
- Transfers further increase time spent in a vehicle

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy
- Compared to a scenario with individual vehicles, ride-sharing and transfers can significantly reduce the number of vehicles required and the total distance driven
- Transfers further improve the solution
- Transfers further increase time spent in a vehicle
- Transfers increase computation time

Conclusion

- We developed a heuristic to solve the DARP-T
- Using a solution without transfer as initial guess is a successful strategy
- Compared to a scenario with individual vehicles, ride-sharing and transfers can significantly reduce the number of vehicles required and the total distance driven
- Transfers further improve the solution
- Transfers further increase time spent in a vehicle
- Transfers increase computation time
- Transfers make solution less robust

Thank you for your attention!
Any questions?

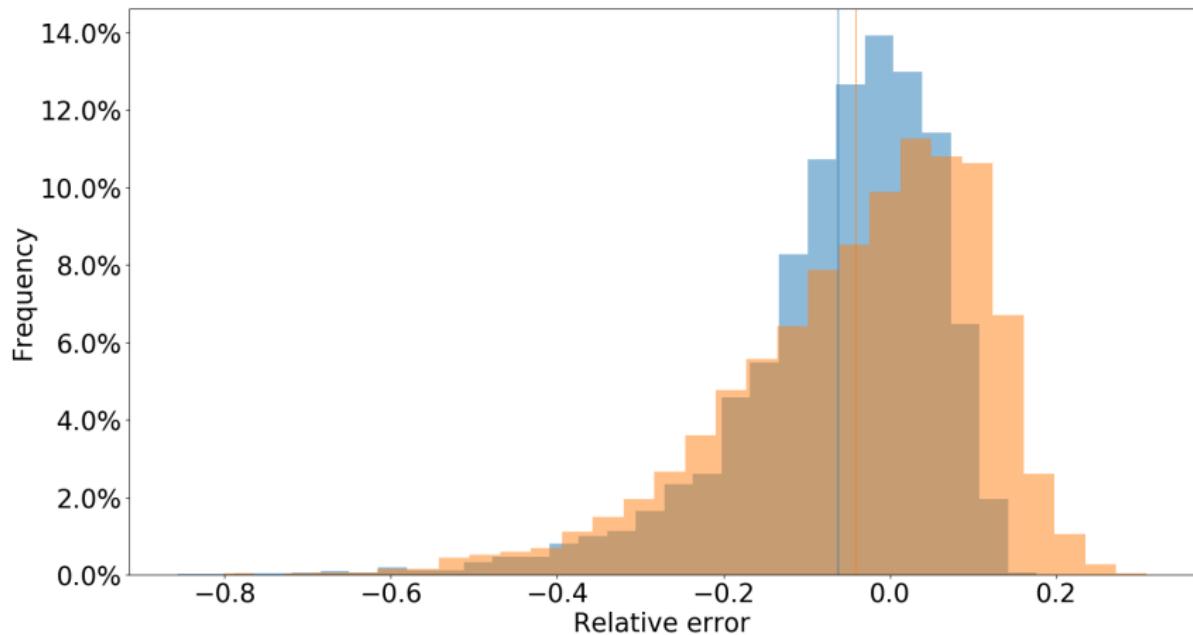
User inconvenience

Instance	Without delay (%)			Transfer-red (%)		Average delay (min)			Average relative delay (%)		
	nt	ntt	t	ntt	t	nt	ntt	t	nt	ntt	t
150_70	61.3	55.9	52.6	5.2	7.5	6.8	8.0	8.1	9.2	11.0	11.3
150_60	64.7	57.1	55.5	4.4	7.7	6.2	7.2	7.8	8.5	10.0	10.8
500_180	47.2	42.7	43.4	5.3	7.1	9.5	10.3	10.1	13.3	14.5	14.2
500_175	48.9	42.7	41.0	7.0	9.0	9.3	10.4	10.4	12.8	14.8	14.6
1000_360	44.2	39.7	38.4	5.4	8.6	9.9	10.8	10.9	14.1	15.4	15.6
1000_330	44.0	40.0	39.3	6.4	8.5	9.7	10.6	10.7	13.7	15.1	15.2

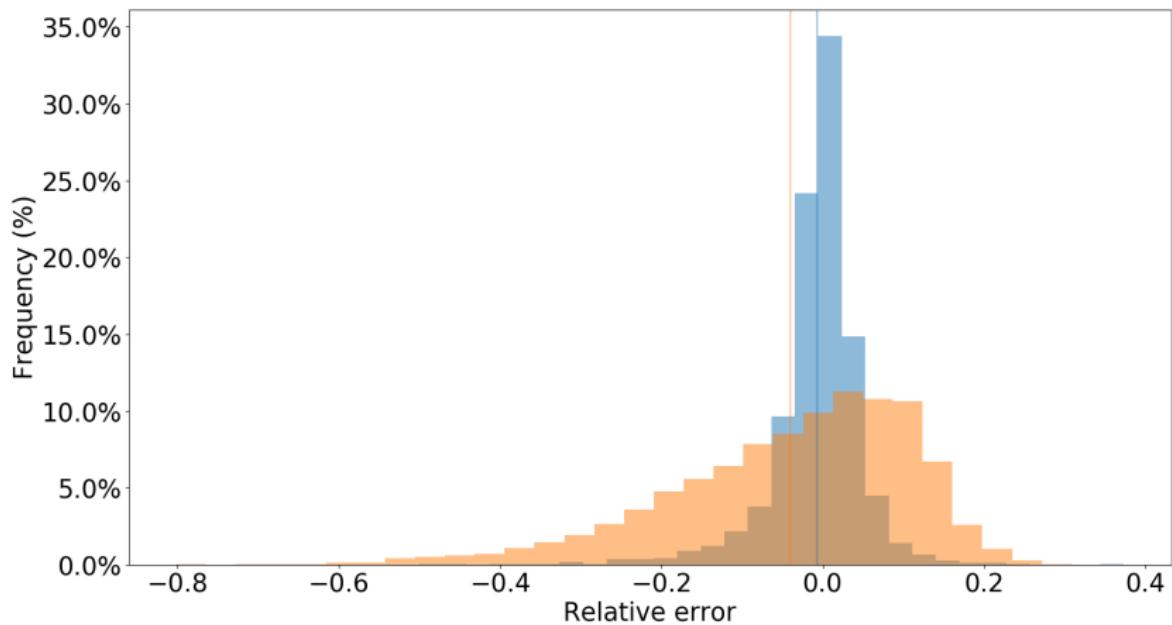
Performance

Instances	Objective function value					Number of transfers		Running time (min))		
	nt	ntt	Gap(%)	t	Gap(%)	ntt	t	nt	ntt	t
150_70	1.0725 (0.0)	1.0471 (0.0)	-2.3	1.0593 (0.0)	-1.1	6.8	10.0	60	83	170
150_60	1.5810 (5.6)	1.4573 (4.4)	-1.5	1.5580 (5.4)	/	6.4	10.6	69	80	153
500_180	0.8826 (0.0)	0.8545 (0.0)	-3.2	0.8769 (0.0)	-0.7	23.6	30.8	306	398	673
500_175	0.9507 (0.8)	0.9260 (0.8)	-2.6	0.9124 (0.6)	/	29.8	39.2	509	392	849
1000_360	0.8049 (0.0)	0.7801 (0.0)	-3.1	0.7871 (0.0)	-2.2	43.4	70.0	873	972	1838
1000_330	0.8043 (0.0)	0.7788 (0.0)	-3.2	0.7934 (0.0)	-1.3	50.6	69.2	1039	1067	1930

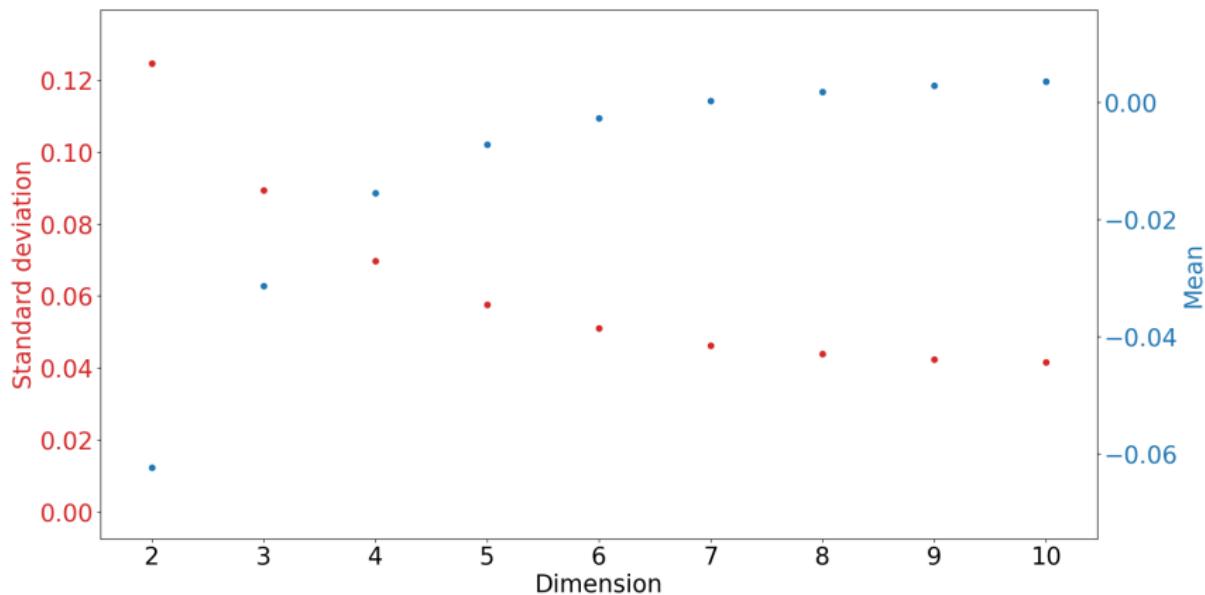
2D multi-dimensional scaling Comparison



2D vs 5D multi-dimensional scaling



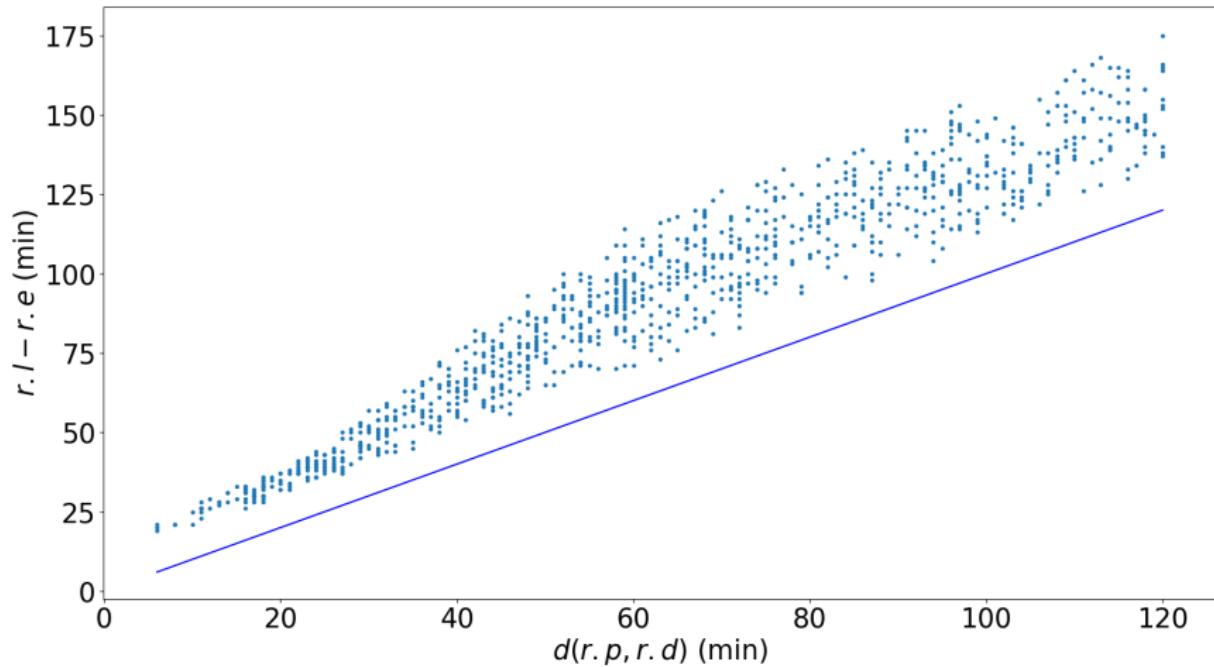
Standard deviation and mean versus dimension



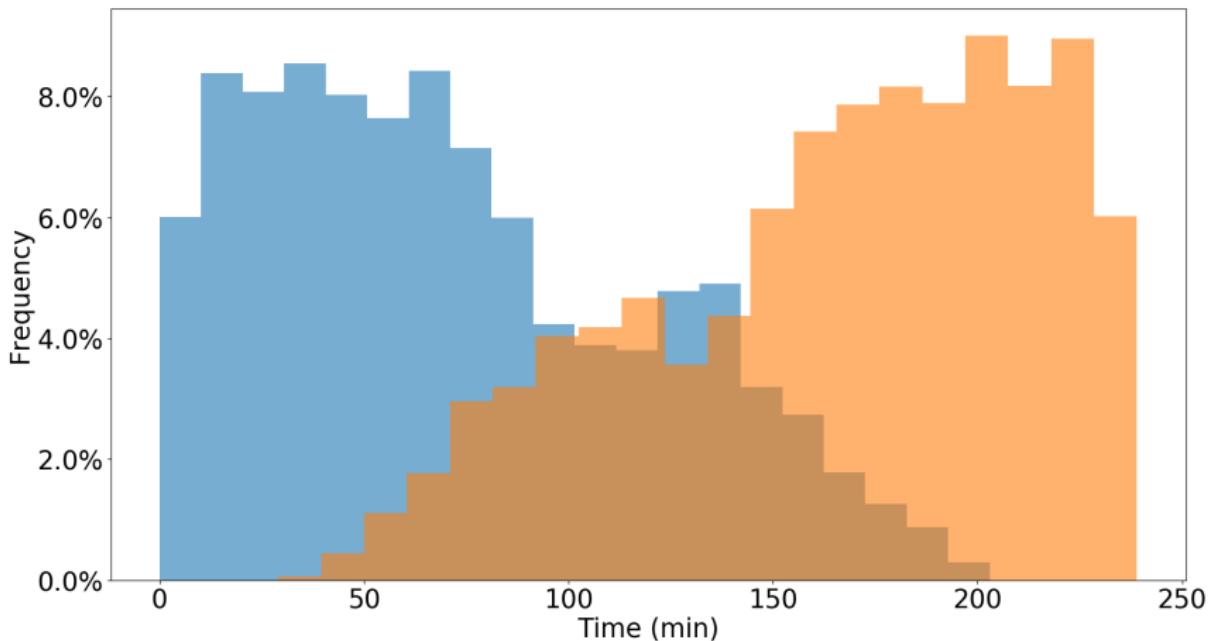
Open Source Routing Machine routes overlay



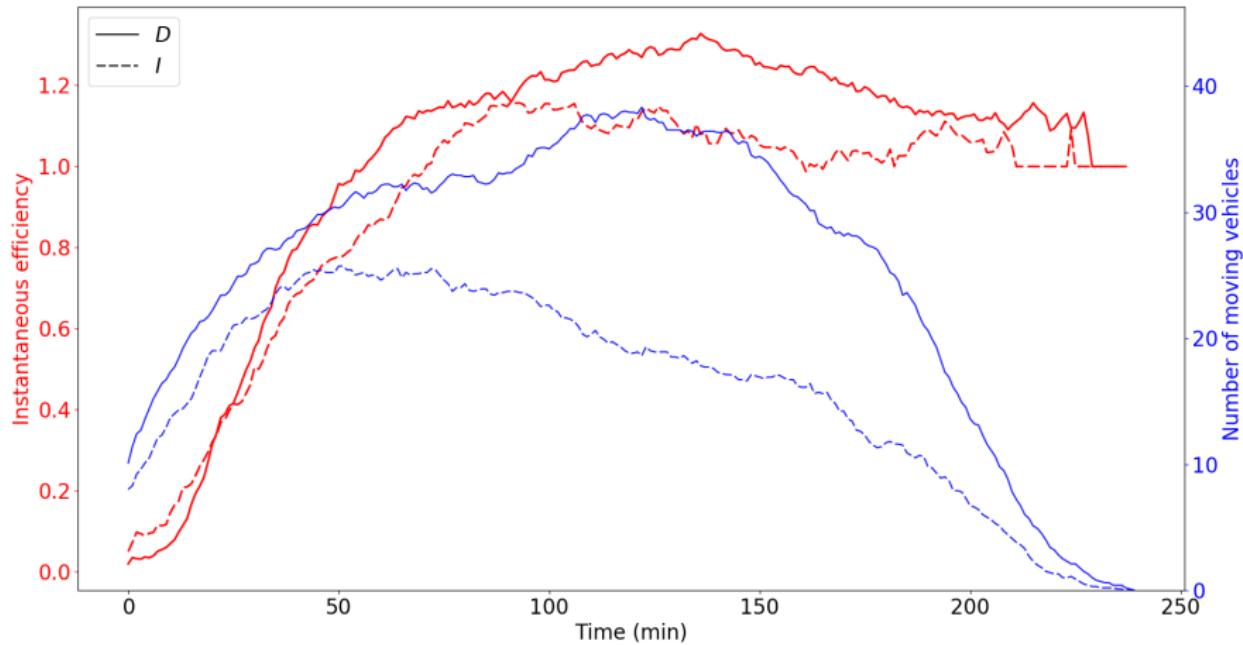
Time window vs travel time



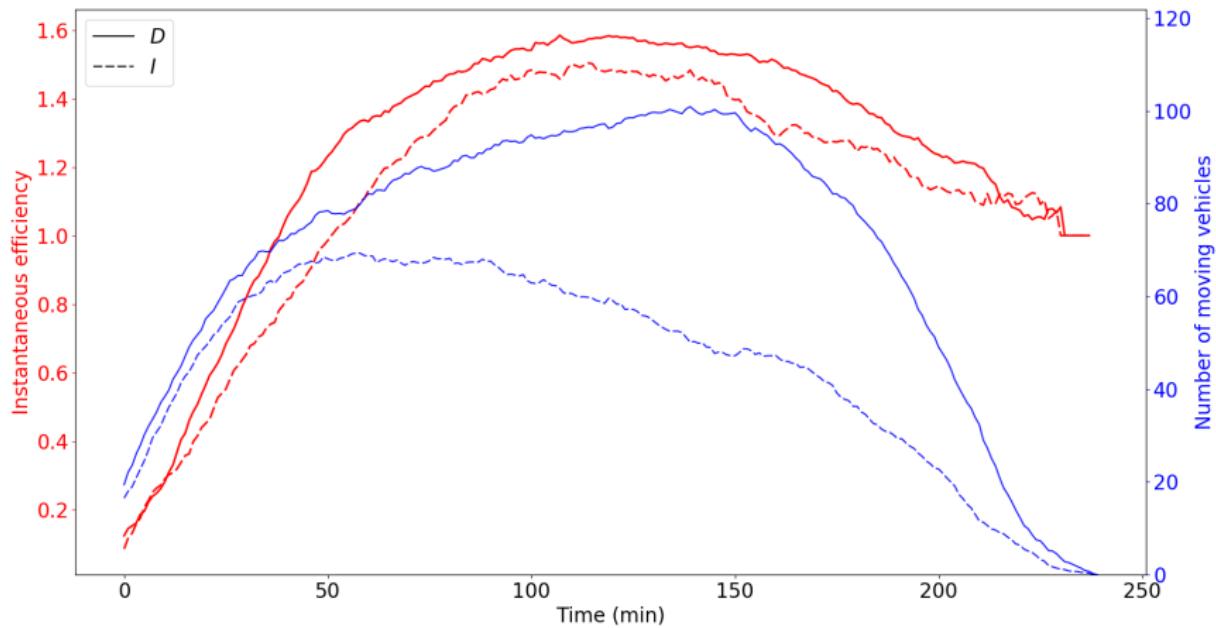
r.e and r.l distribution



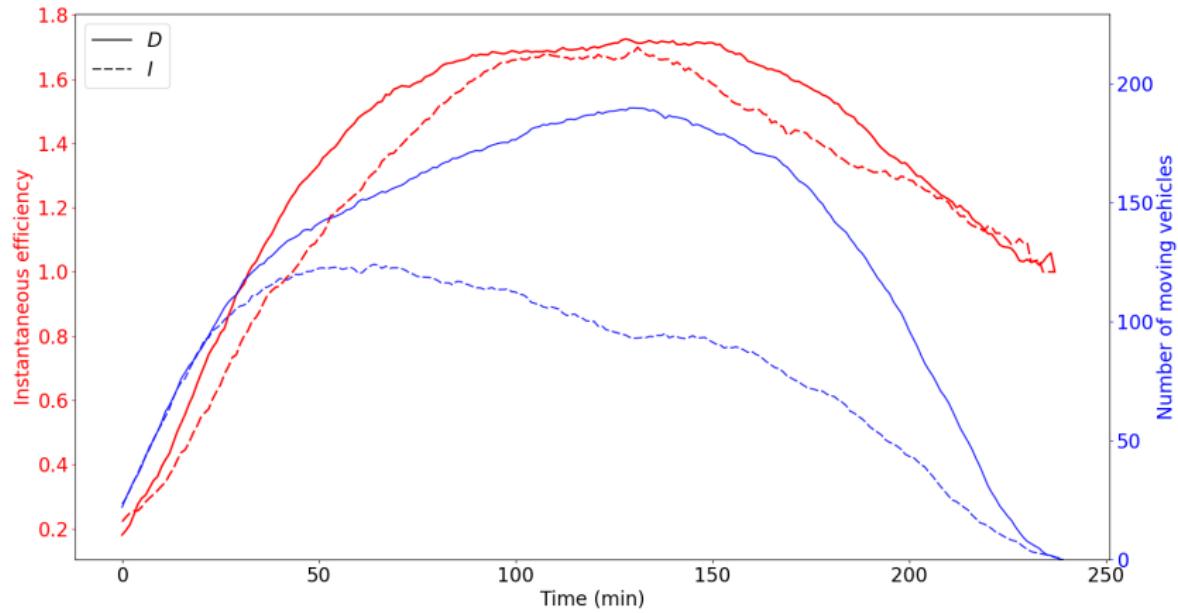
Instantaneous efficiency and number of moving vehicles vs time



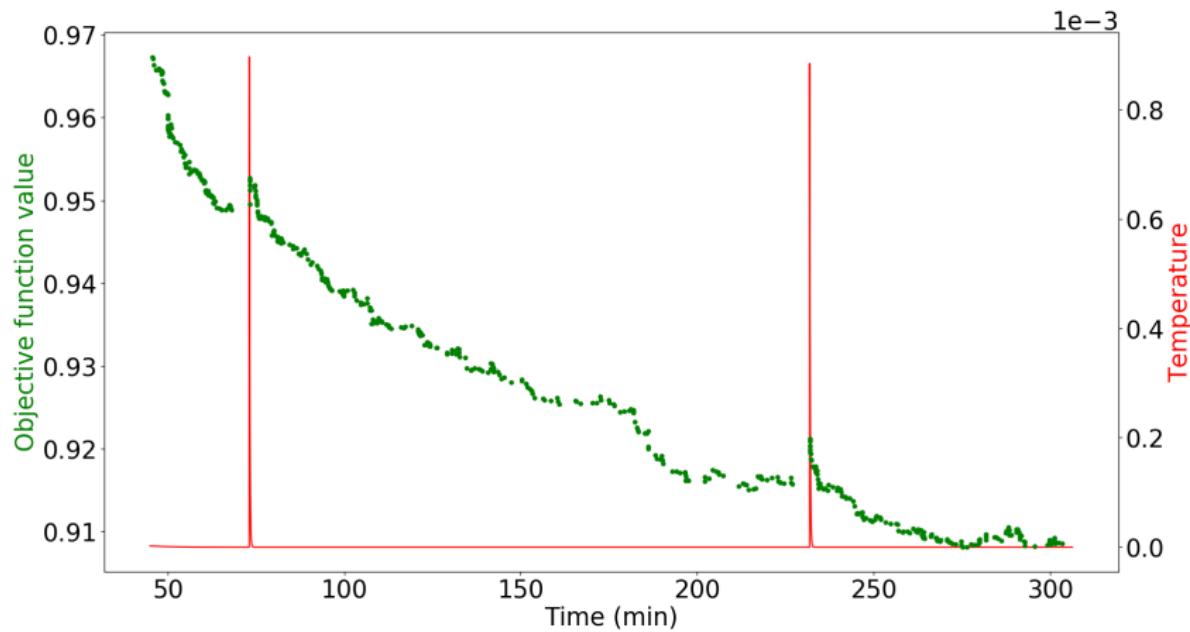
Instantaneous efficiency and number of moving vehicles vs time



Instantaneous efficiency and number of moving vehicles vs time



Example reheat



Runtime

