A route relaxation based on the spatial aggregation of nodes for the generalized vehicle routing problem

François Lamothe [1,4,5] **Claudio Contardo** [2, 5]
Matthieu Gruson [1,4] Rafael Martinelli [3]

[1]ESG UQÀM [2]Concordia University [3]PUC-Rio
[4]CIRRELT [5]GERAD

May 18, 2023

Agenda

The Generalized VRP

- Input:
  - A set N of n nodes, partitioned into k clusters
  - A depot node
  - A demand $d_i > 0$ for each cluster
  - A vehicle capacity Q
  - A fleet of (un)limited size
  - Traveling costs c_{ij}
- Output
  - A set of routes respecting the vehicle capacities
  - Each cluster is visited once (at one node)
  - Minimum total traveling cost

The Generalized VRP

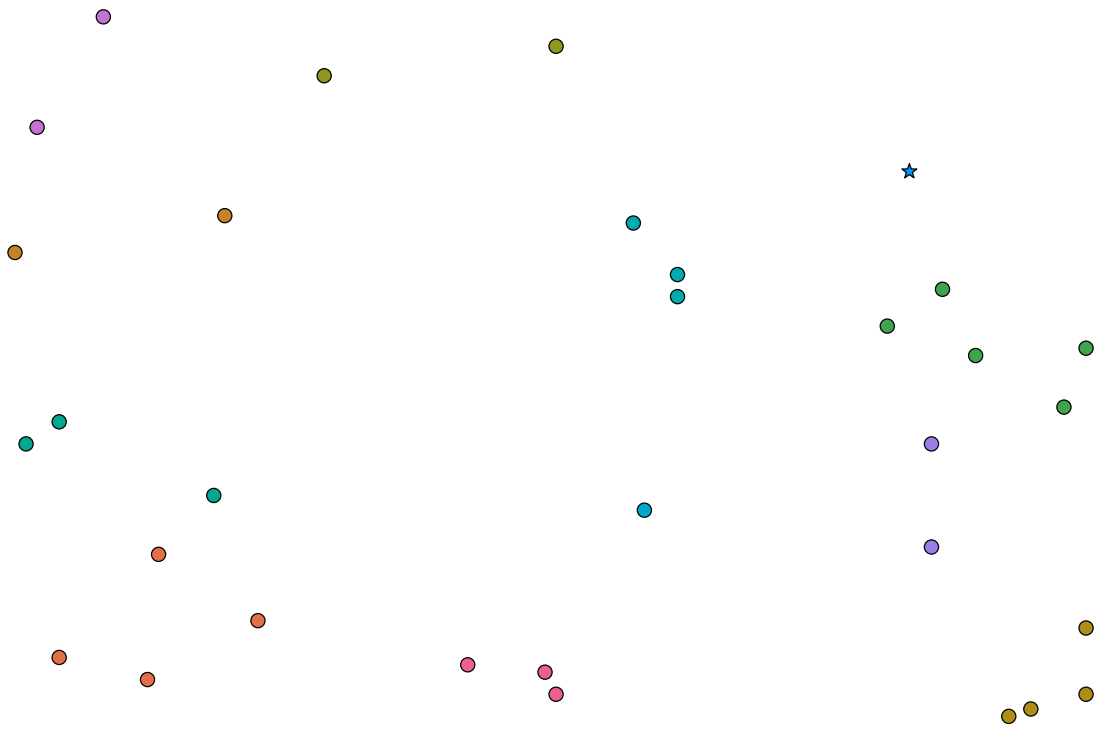


Figure: Instance A-n32-k5-C11-V2, $z^* = 386$

The Generalized VRP

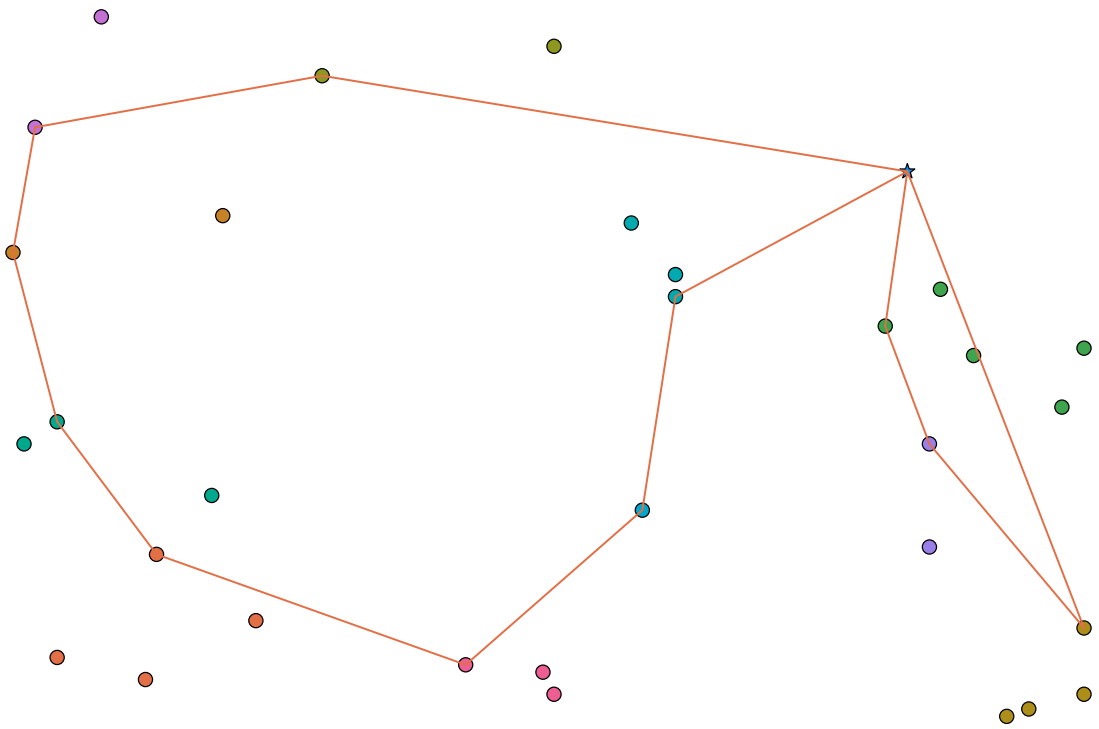


Figure: Instance A-n32-k5-C11-V2, $z^* = 386$

A very thorough (?) literature review

- Compact formulations, B&C
  - Bektas + Erdogan + Ropke 2011 (TS)
  - Ha + Bostel + Langevin + Rousseau 214 (C&OR)
- Problem reductions (to CARP most notably)
  - Ghiani + Improta 2000 (EJOR)
- CG, B&P
  - Martinelli + Pecin + Poggi 2014 (EJOR)
  - Reihaneh + Ghoniem 2018 (JORS)
  - Freitas + P. Silva + Uchoa 2023 (C&OR)

Mathematical formulation

$$\min \quad \sum_{l \in \Omega} c_l \theta_l$$

subject to

$$\sum_{l \in \Omega} a_{kl} \theta_l = 1 \qquad k \in \mathcal{C} \qquad (\alpha_k)$$

$$\theta \geq 0 \text{ and integer.}$$

Pricing subproblem

- A resource constrained SPP (elementary, ng, etc...)
- Solved through a labeling algorithm
- Label L: terminal node $v(L)$, load $q(L)$, reduced cost $\overline{c(L)}$, memory $\pi(L)$

Pricing subproblem

- Traditional dominance rule between two labels L_1, L_2

$$\overline{c(L_1)} \leq \overline{c(L_2)}$$
$$v(L_1) = v(L_2)$$
$$q(L_1) \leq q(L_2)$$
$$\pi(L_1) \subseteq \pi(L_2)$$

- The terminal condition $v(L_1) = v(L_2)$ seems too restrictive, especially if the number of nodes is much larger than the number of clusters

A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other

A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other
- More specifically, we consider *supernodes* such that every pair of nodes in the same *supernode* is at a distance $\leq \delta$

A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other
- More specifically, we consider *supernodes* such that every pair of nodes in the same *supernode* is at a distance $\leq \delta$
- We perform pricing on a reduced graph, with $m + 1$ nodes instead of $n + 1$

A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other
- More specifically, we consider *supernodes* such that every pair of nodes in the same *supernode* is at a distance $\leq \delta$
- We perform pricing on a reduced graph, with $m+1$ nodes instead of $n+1$
- The traveling cost between two supernodes is defined as $c_{st} = \min\{c_{ij} : i \in s, j \in t\}$

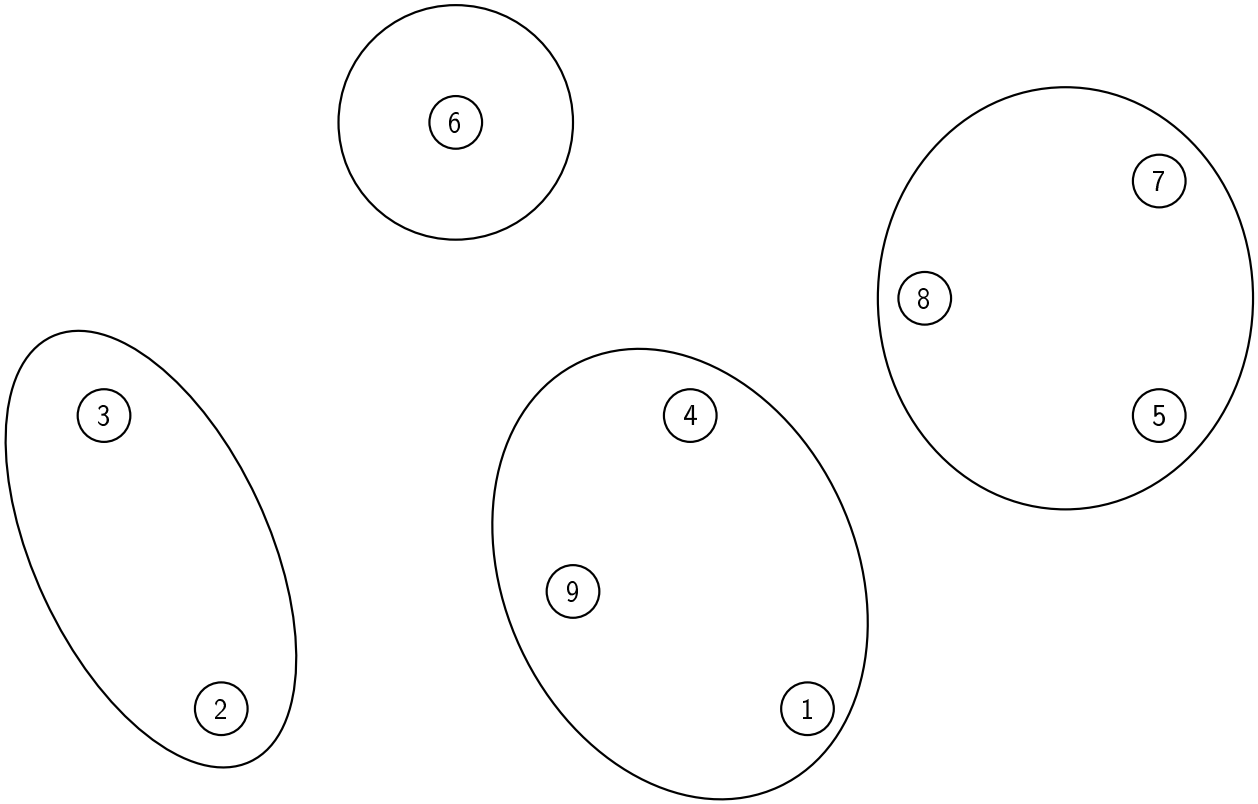
A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other
- More specifically, we consider *supernodes* such that every pair of nodes in the same *supernode* is at a distance $\leq \delta$
- We perform pricing on a reduced graph, with $m+1$ nodes instead of $n+1$
- The traveling cost between two supernodes is defined as $c_{st} = \min\{c_{ij} : i \in s, j \in t\}$
- This can be seen as *jumping* within a supernode: entering through a node and leaving from another

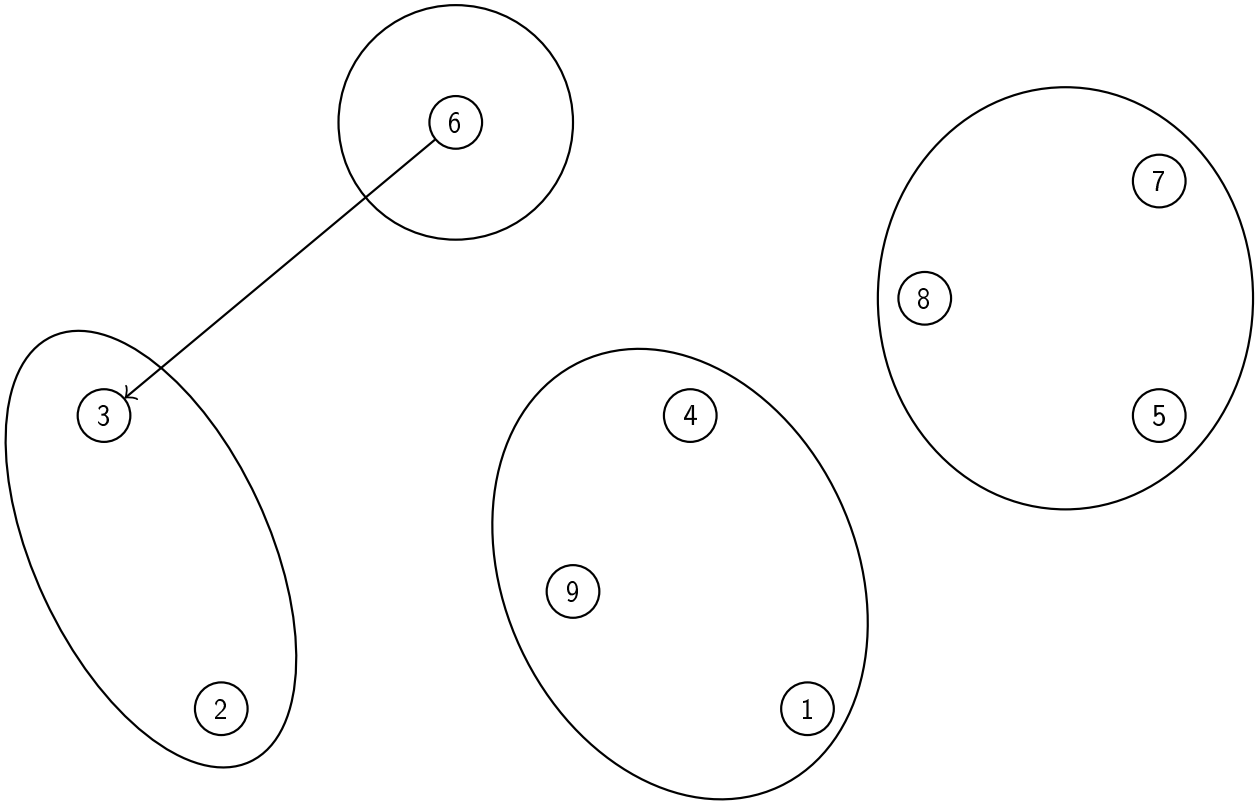
A new route relaxation

- We consider m *supernodes* (ideally with $m \ll n$), composed of nodes that are close from each other
- More specifically, we consider *supernodes* such that every pair of nodes in the same *supernode* is at a distance $\leq \delta$
- We perform pricing on a reduced graph, with $m+1$ nodes instead of $n+1$
- The traveling cost between two supernodes is defined as $c_{st} = \min\{c_{ij} : i \in s, j \in t\}$
- This can be seen as *jumping* within a supernode: entering through a node and leaving from another
- Terminal node $v(L)$ is now a supernode, not a disaggregated node

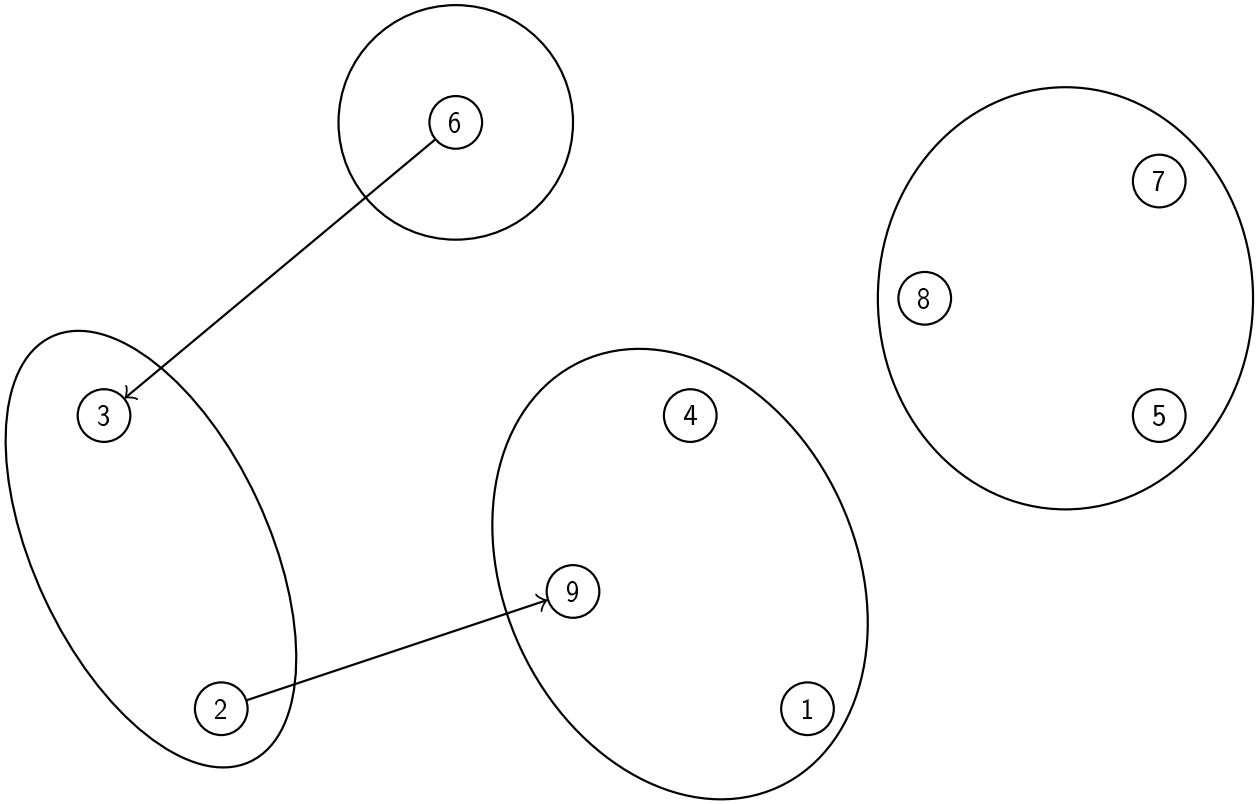
A new route relaxation



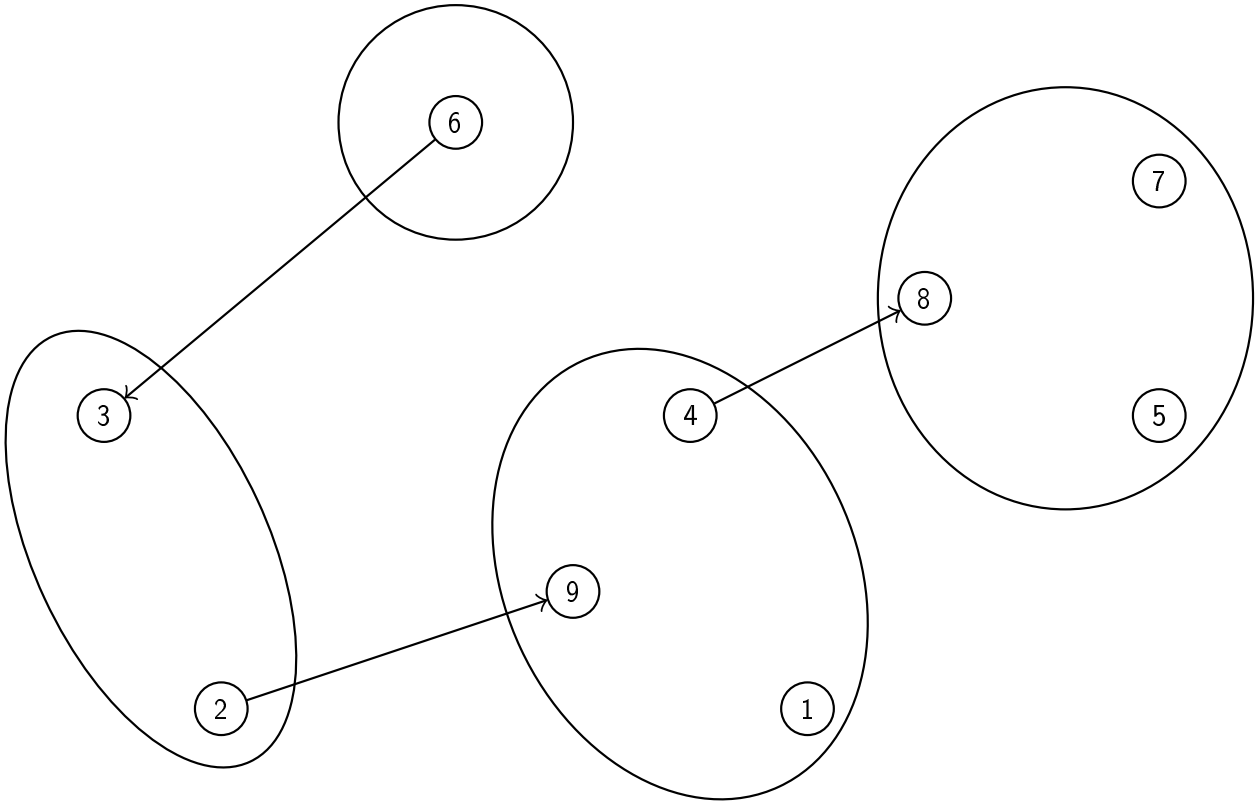
A new route relaxation



A new route relaxation



A new route relaxation



A new route relaxation

- New dominance rule

$$\overline{c(L_1)} \leq \overline{c(L_2)}$$
$$v(L_1) = v(L_2)$$
$$q(L_1) \leq q(L_2)$$
$$\pi(L_1) \subseteq \pi(L_2)$$

- In practice, what we do is to aggregate nodes that are close to each other into a single node

The Generalized VRP

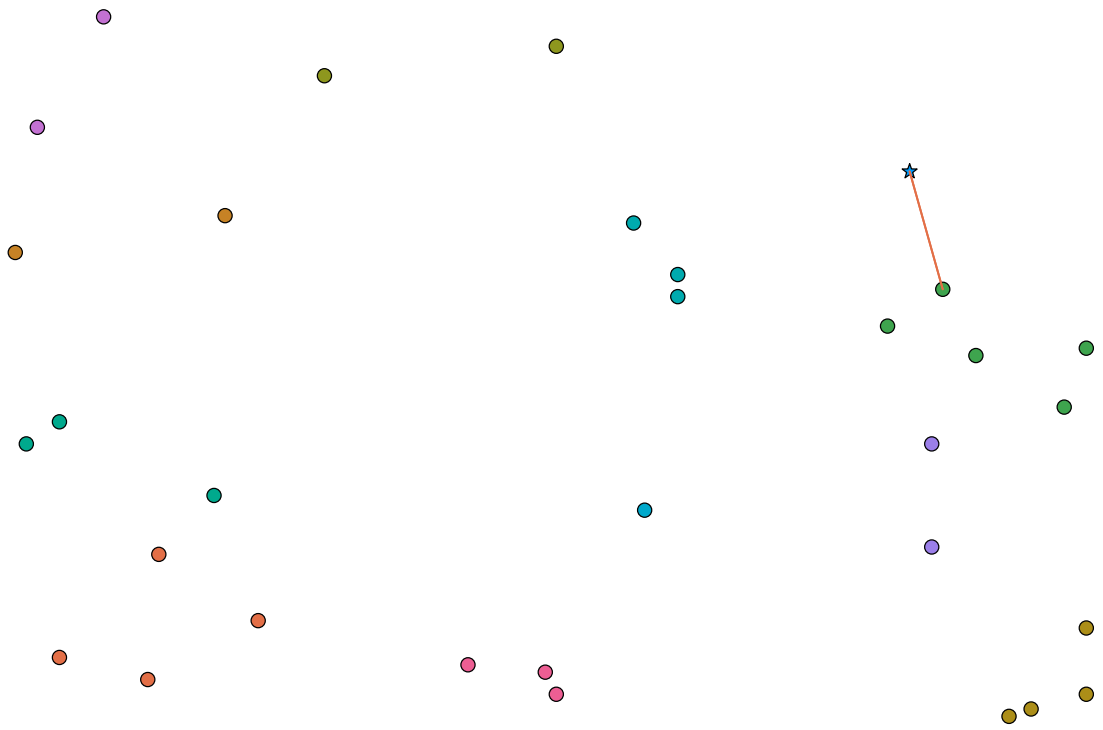


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

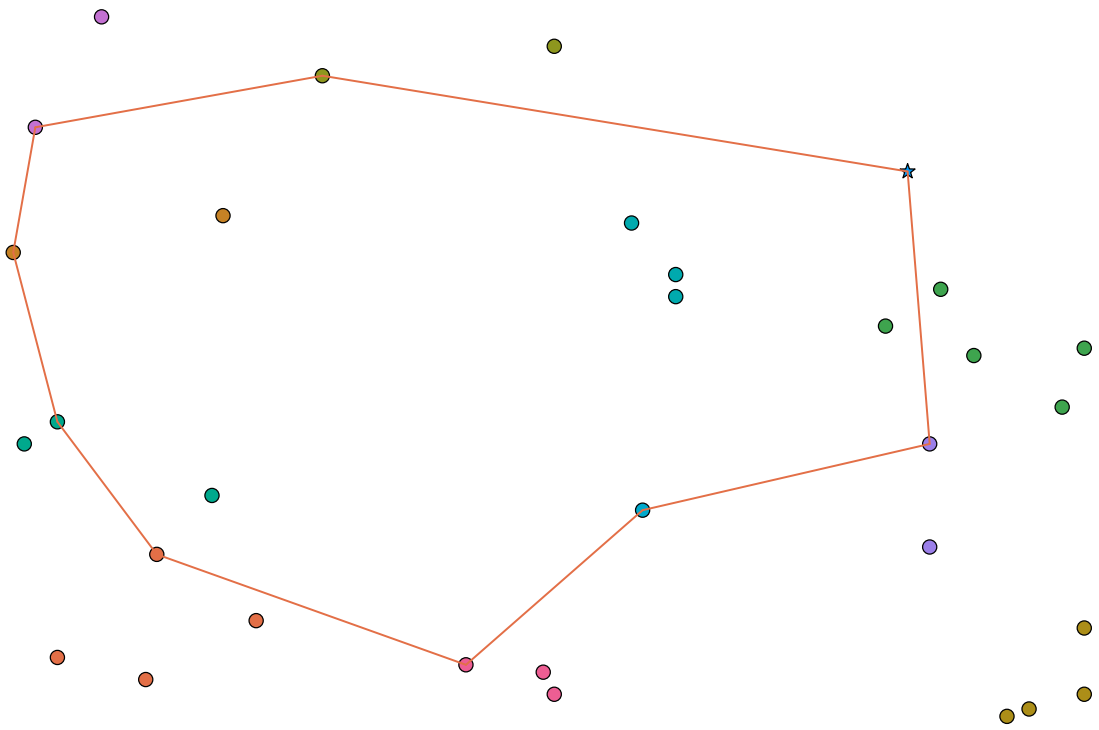


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

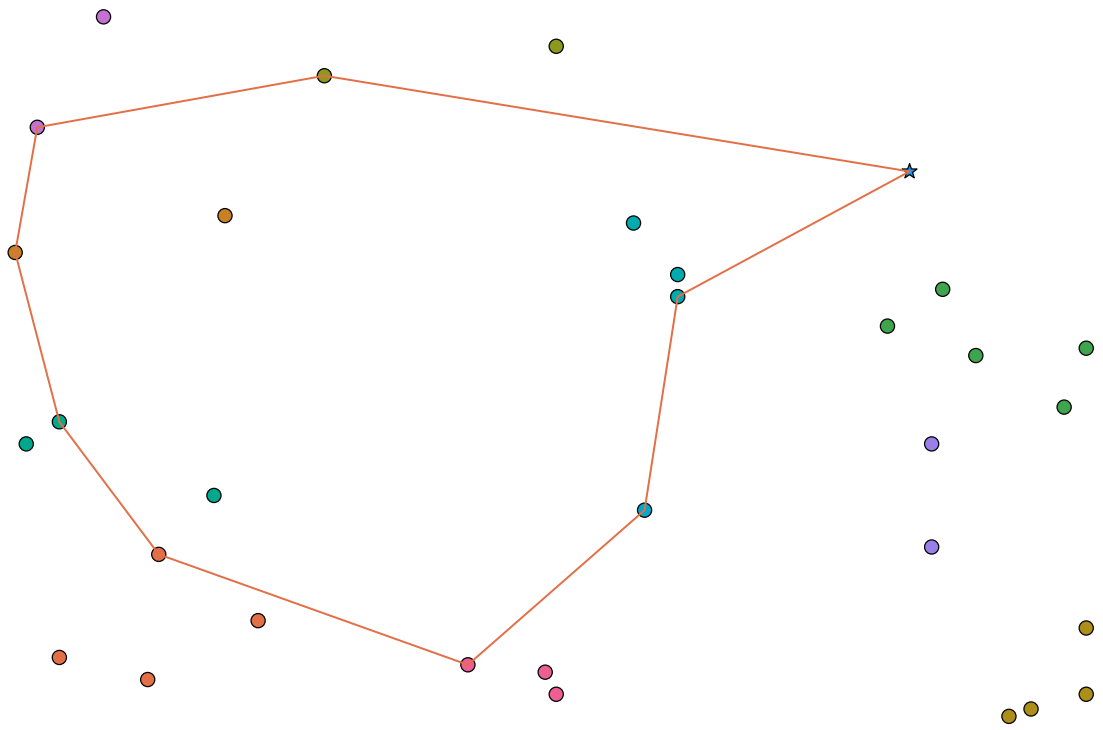


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

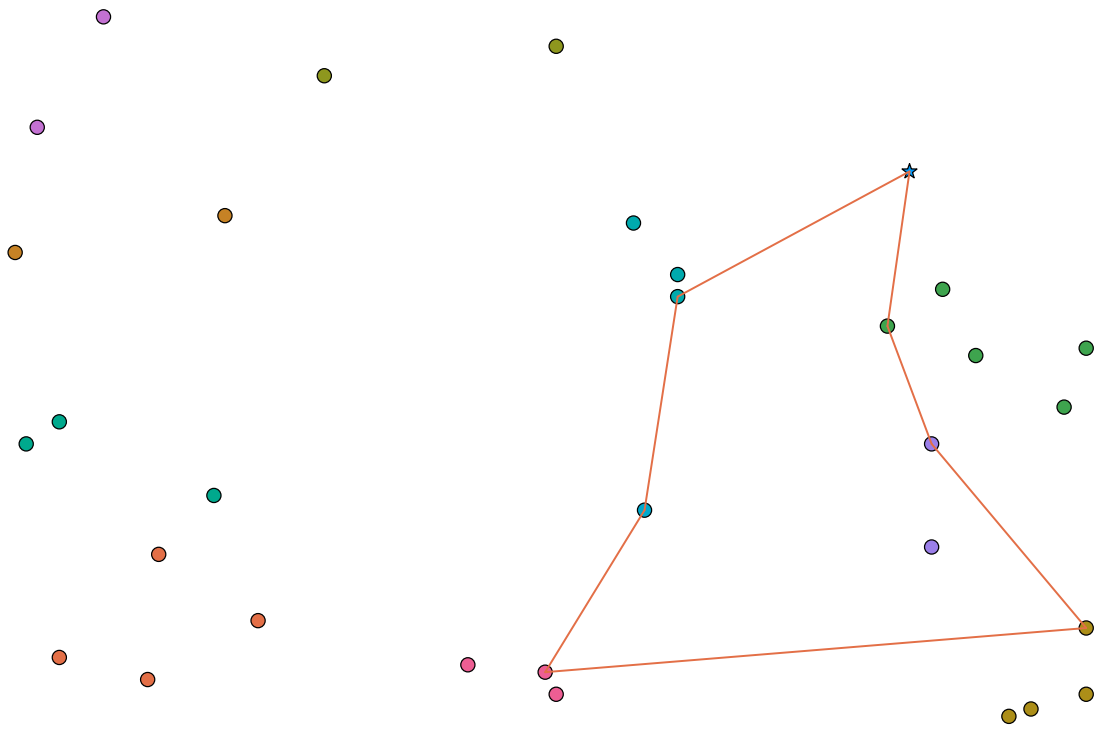


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

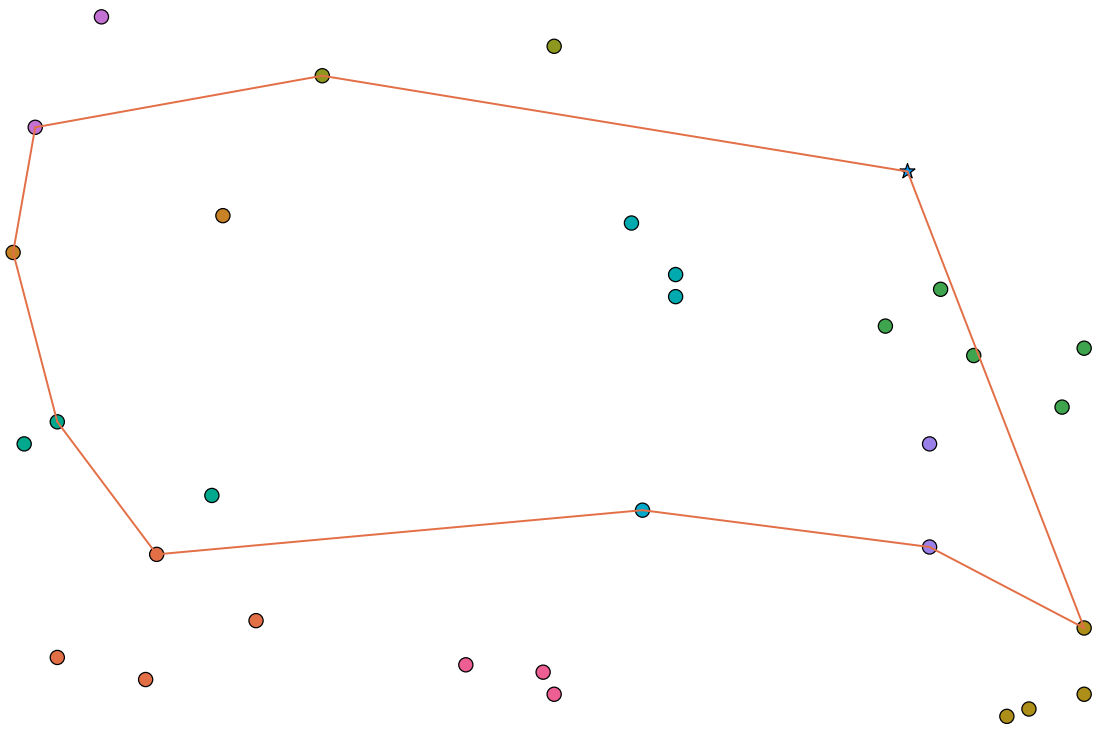


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

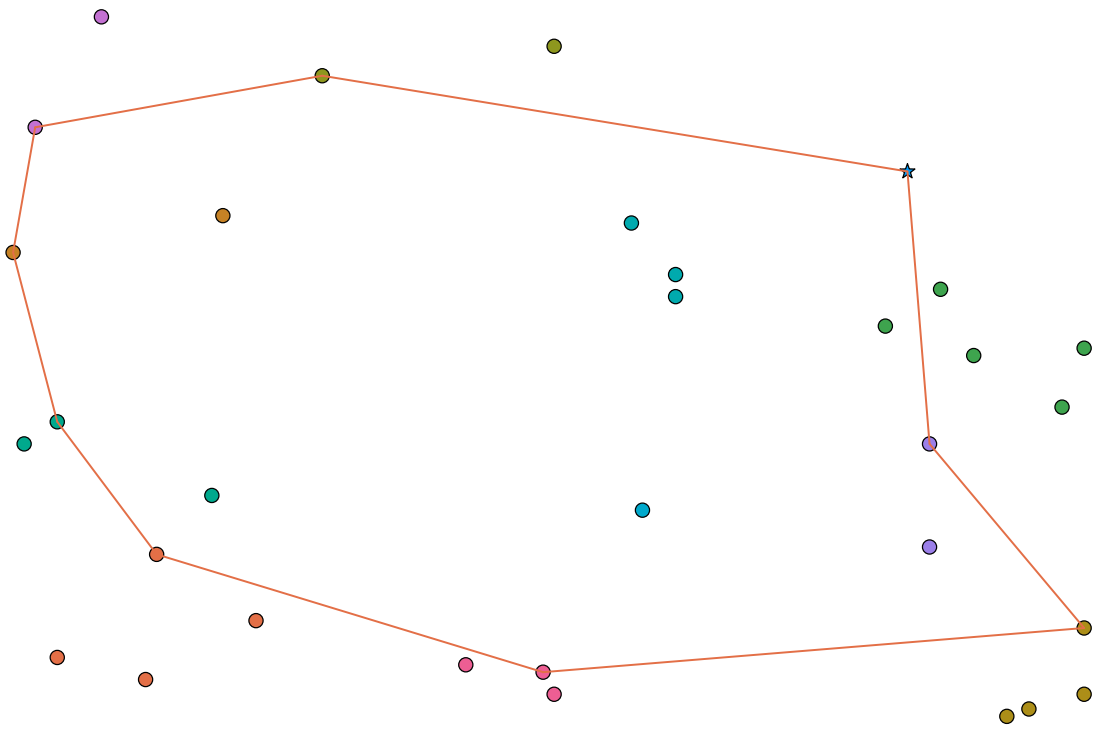


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

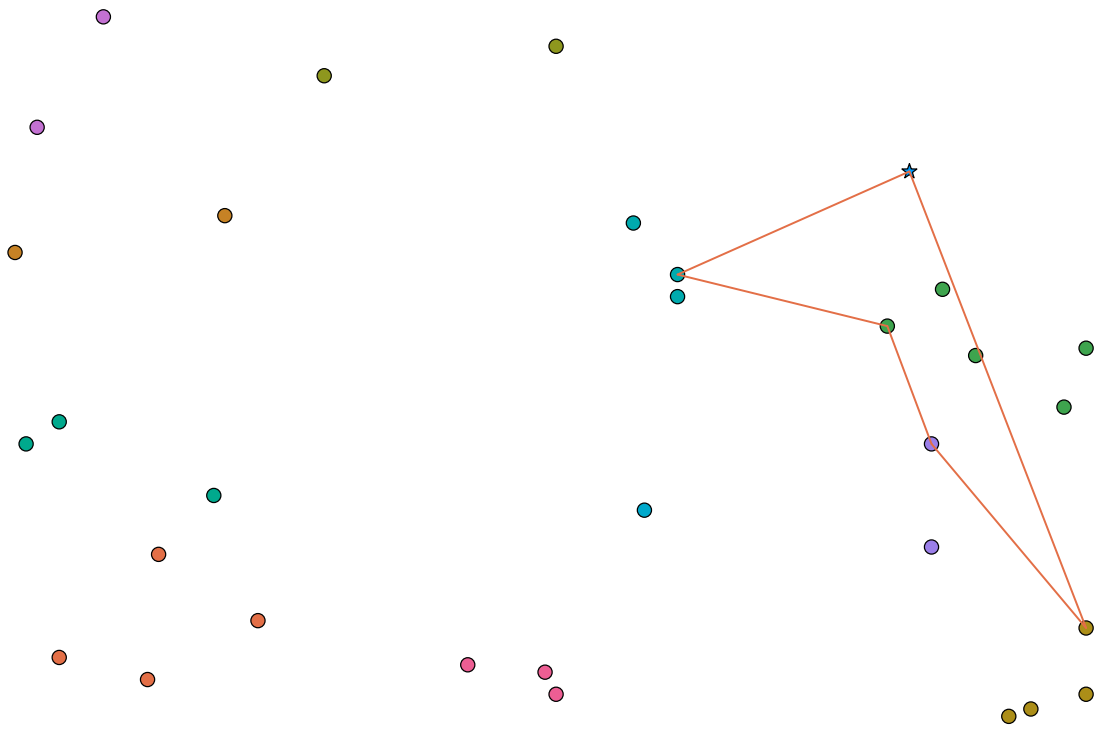


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

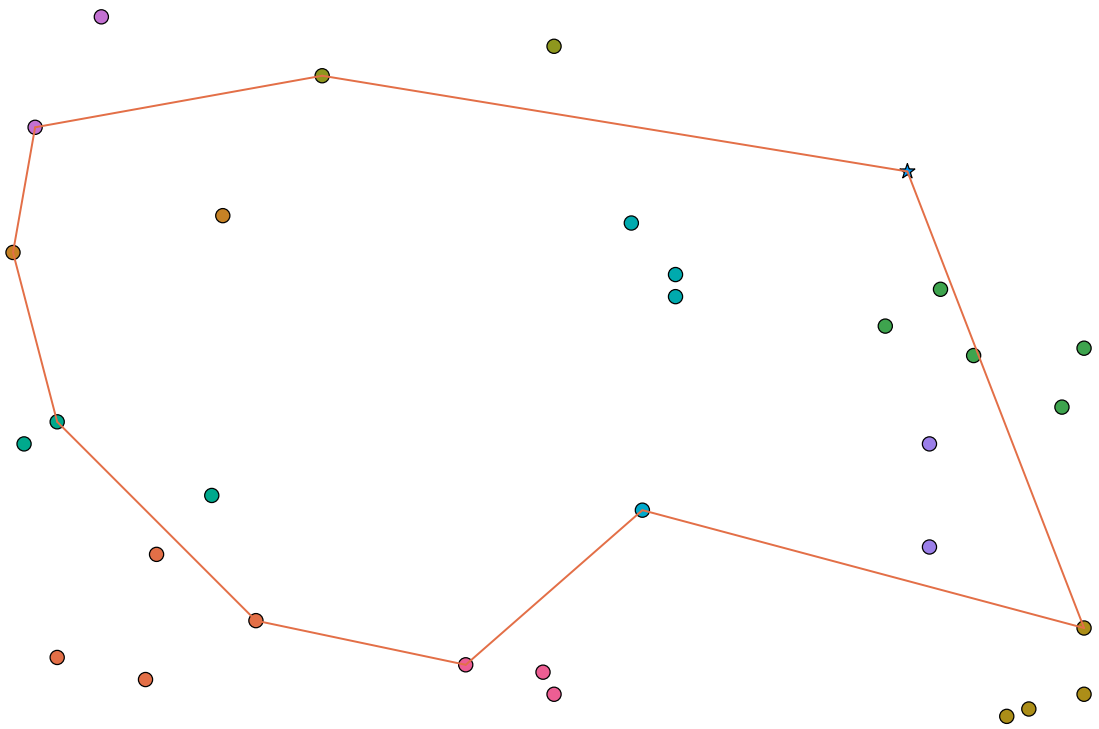


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

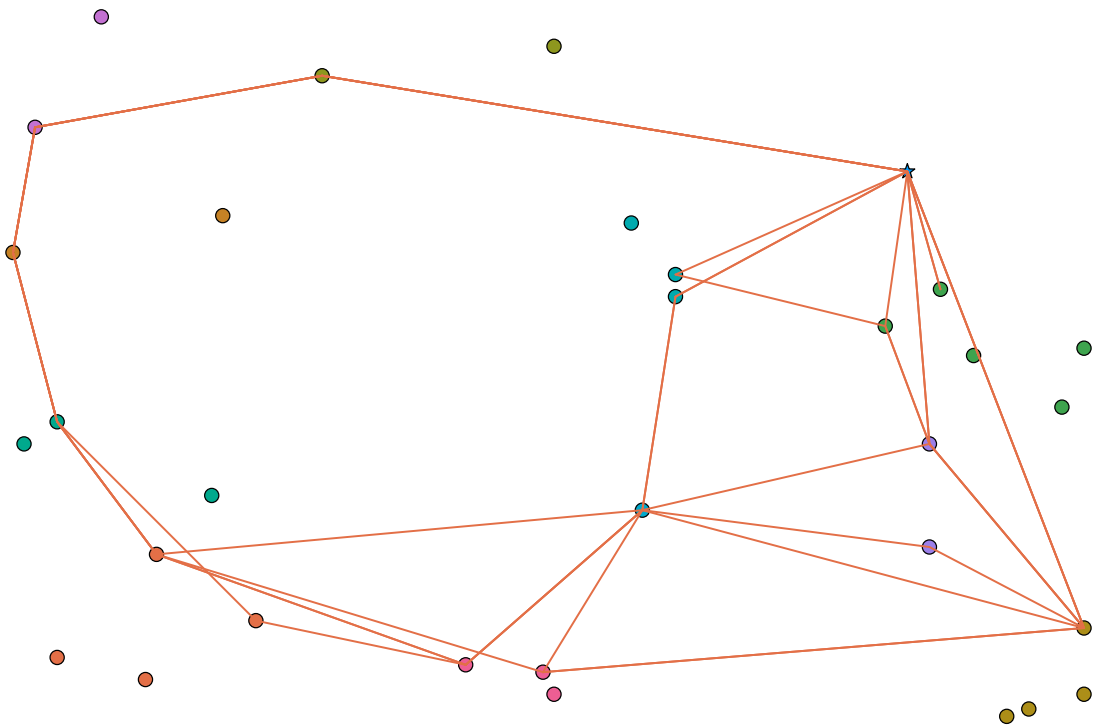


Figure: Instance A-n32-k5-C11-V2, fractional solution with feasible routes

The Generalized VRP

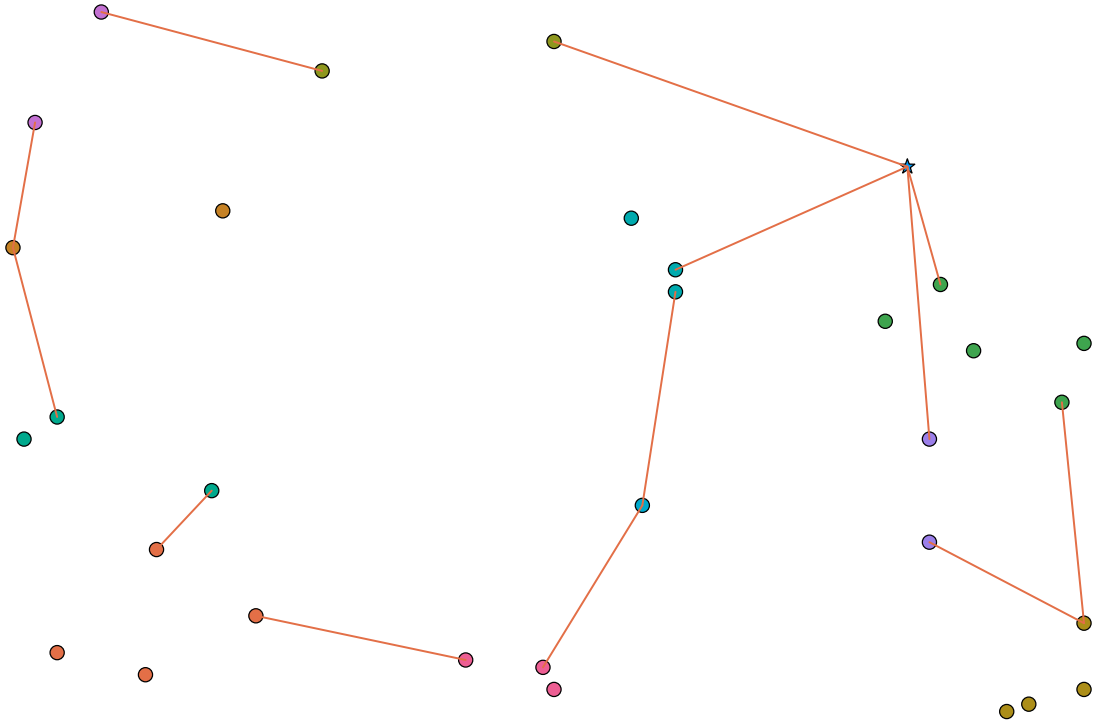


Figure: Instance A-n32-k5-C11-V2, integer solution with *very* relaxed routes, $z^* = 306$

The Generalized VRP

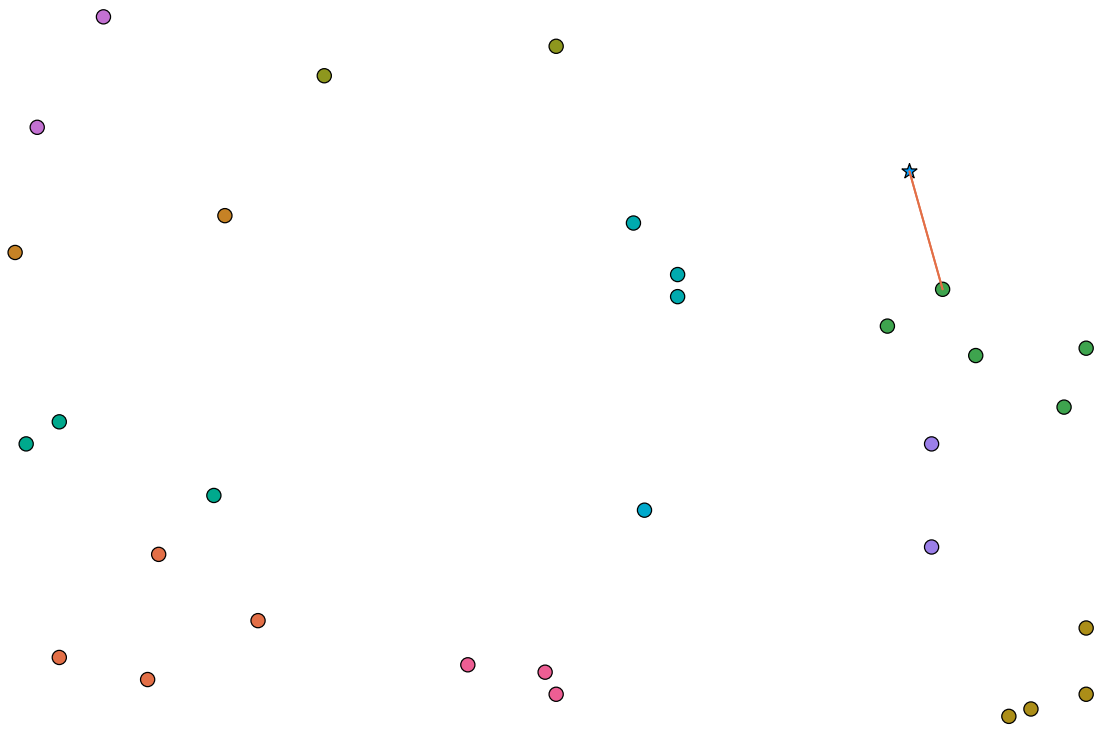


Figure: Instance A-n32-k5-C11-V2, fractional solution with relaxed routes

The Generalized VRP

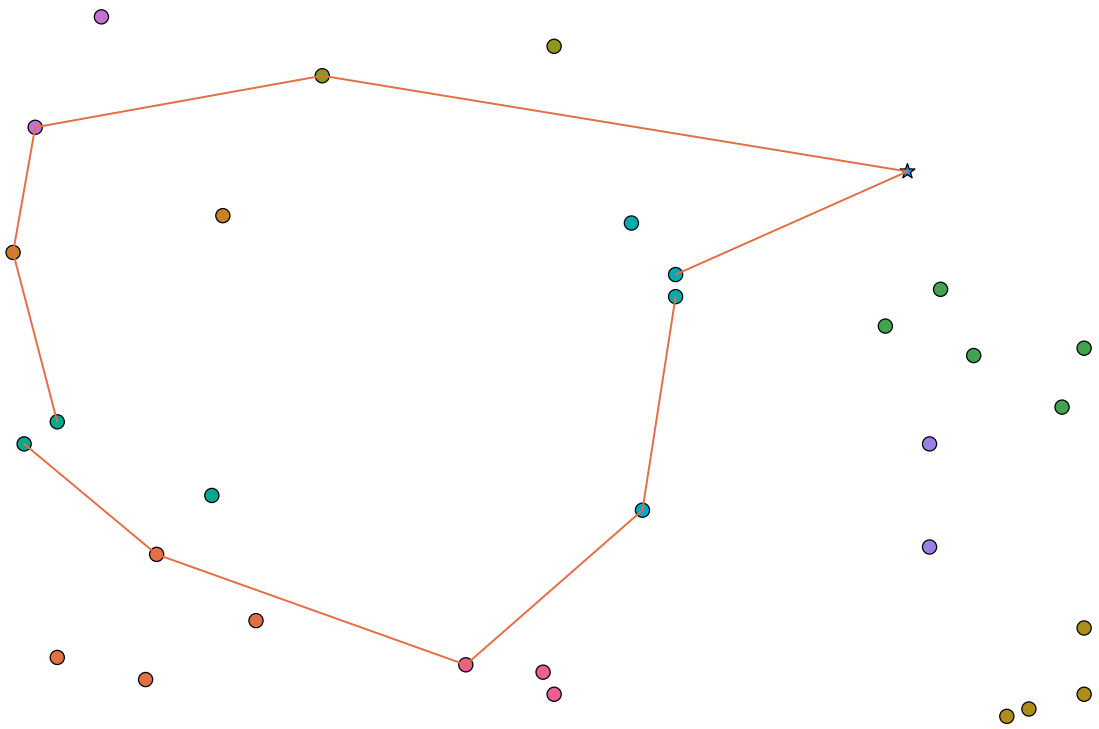


Figure: Instance A-n32-k5-C11-V2, fractional solution with relaxed routes

The Generalized VRP

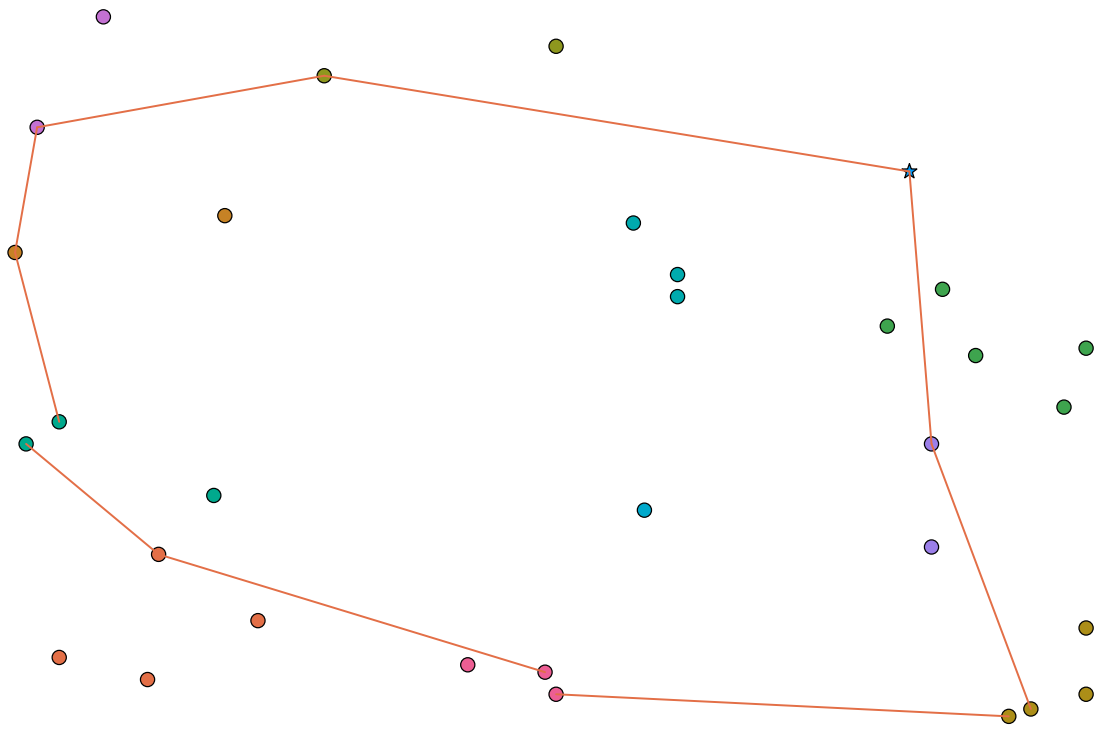


Figure: Instance A-n32-k5-C11-V2, fractional solution with relaxed routes

The Generalized VRP

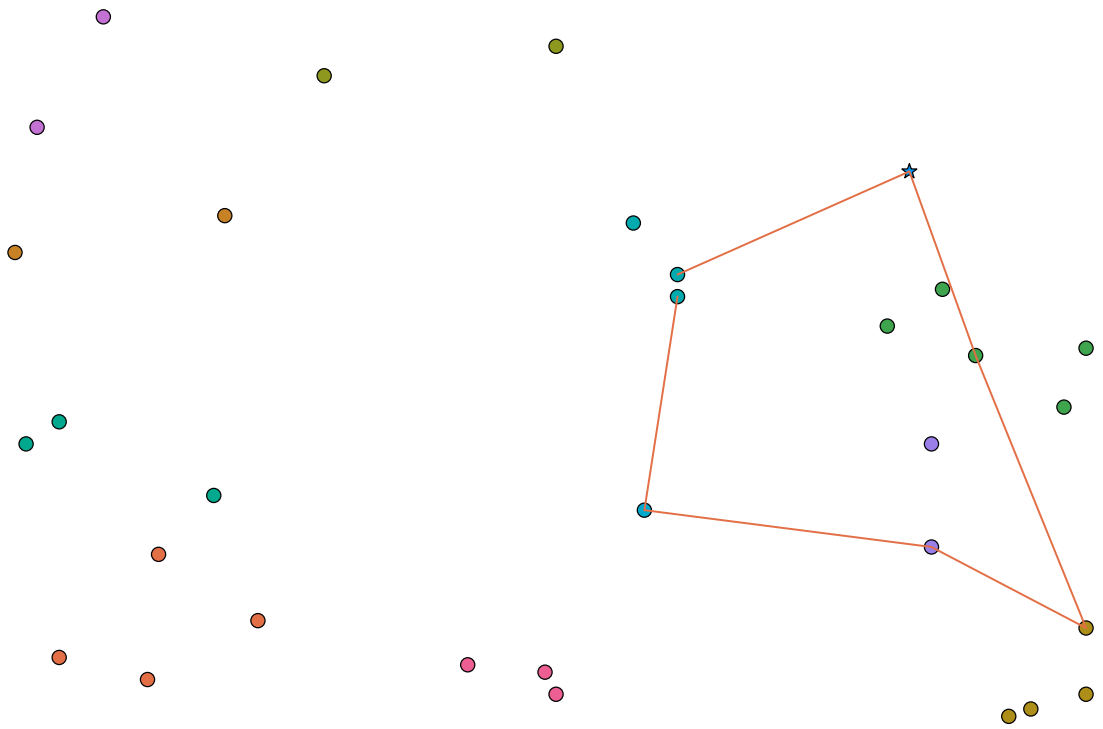


Figure: Instance A-n32-k5-C11-V2, fractional solution with relaxed routes

The Generalized VRP

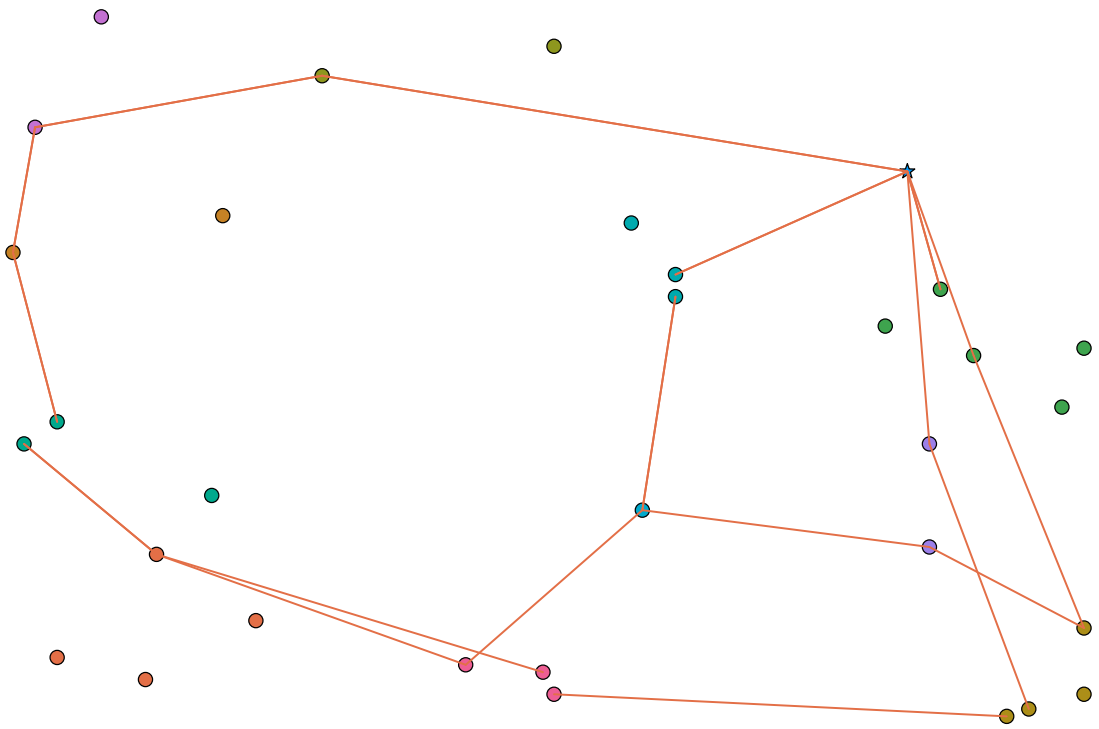


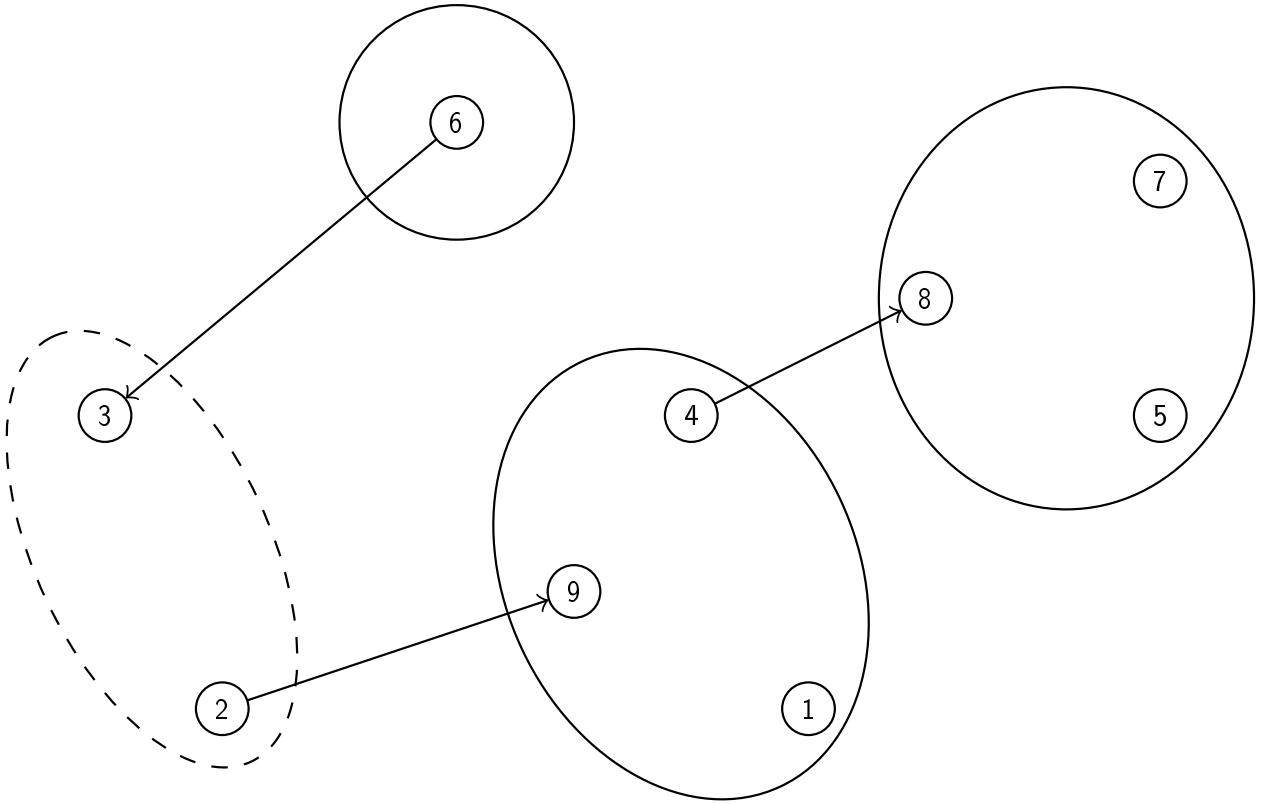
Figure: Instance A-n32-k5-C11-V2, fractional solution with relaxed routes

Implementation

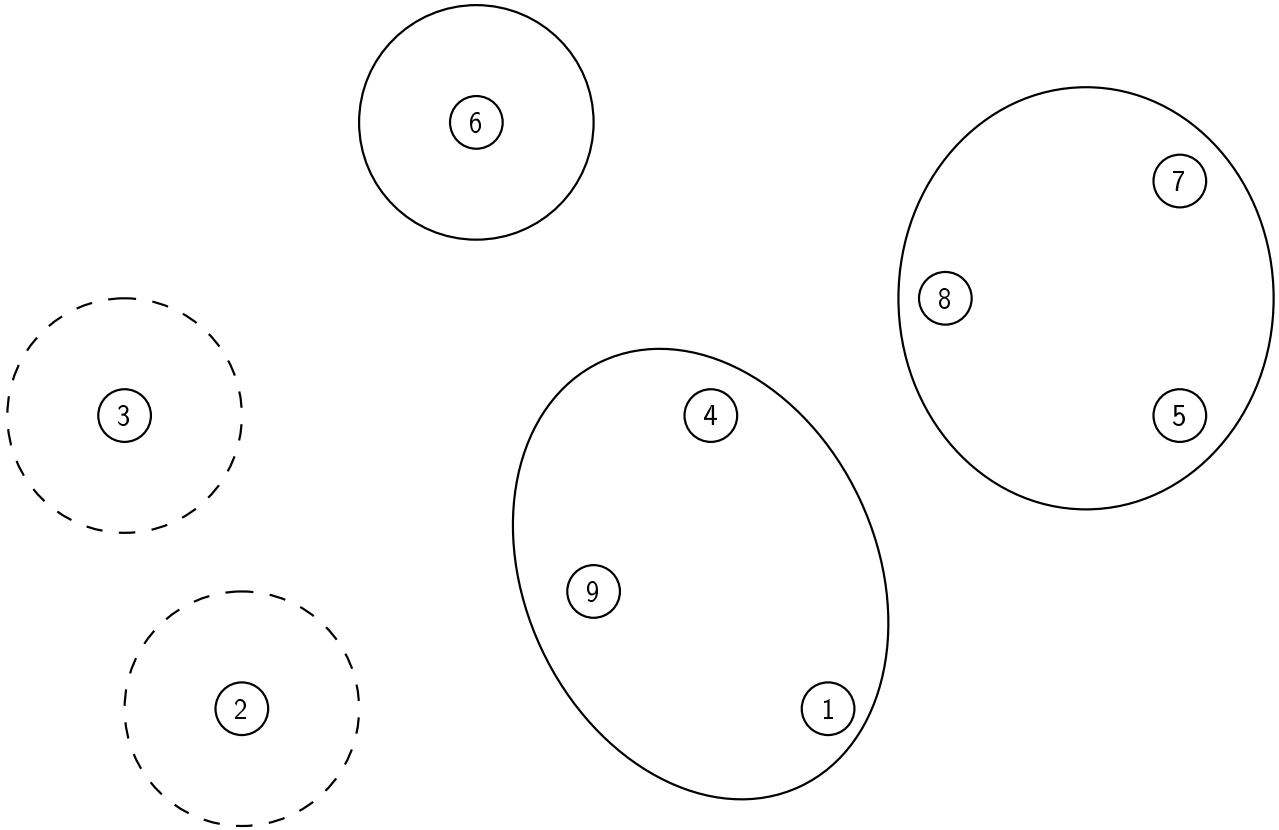
We are currently testing several possible implementations of this:

- A static aggregation of nodes in superclusters
- A dynamic aggregation (using DSSR)
  - With target $\delta = 0$ (feasible routes)
  - Or $\delta > 0$ (infeasible routes)

Decremental state-space relaxation



Decremental state-space relaxation



Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)
- Other branching strategies are possible:

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)
- Other branching strategies are possible:
  - Number of vehicles

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)
- Other branching strategies are possible:
  - Number of vehicles
  - Arcs between clusters

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)
- Other branching strategies are possible:
  - Number of vehicles
  - Arcs between clusters
  - Arcs between nodes

Some remarks

- The proposed approach serves at achieving a dual bound (it is a relaxation!)
- Therefore, it may produce infeasible primal solutions (disconnected), as in the example before
- Luckily, connectivity can be achieved through branching (by imposing that routes pass through a specific node in a cluster)
- Other branching strategies are possible:
  - Number of vehicles
  - Arcs between clusters
  - Arcs between nodes
  - Arcs between supernodes

Valid inequalities

- Our approach copes well with some strong cutting planes :-)
  - Rounded capacity cuts
  - Subset-row cuts
- However, *same vertex inequalities* (Bektas + Erdogan + Ropke 2011) become non-robust in our approach :-(

Experimental setup

- Julia code (v1.8)
- PUC-Rio HPC lab computers (didn't ask Rafa for the specs!)
- LPs soved using Gurobi 10
- MIP search: best bound
- Branching strategy: most fractional (1. # of veqs; 2. total flow crossing a supernode; 3. arc between clusters; 4. arc between supernodes)
- All cuts activated (RCCs, 3-SRCs)
- Our approach is initialized with $\delta = +\infty$ (one supernode per cluster), and generate connected paths only through DSSR

Partial results

Instance	Baseline			AggDssr			SpeedUp		
	CPU	LB	N	CPU	LB	m	%CPU	%LB	%m
Golden-1-C17-N241	356.37	1245.5	241	205.34	1245.5	184	1.74	1	1.31
Golden-1-C18-N241	560.34	1260.25	241	265.52	1247	194	2.11	0.98	1.24
Golden-1-C19-N241	673.66	1294.5	241	443.99	1285	190	1.52	0.99	1.27
Golden-1-C21-N241	1019.07	1335	241	491.72	1335	184	2.07	1	1.31
Golden-1-C22-N241	626.52	1363	241	905.65	1349.25	204	0.69	0.98	1.18
Golden-1-C25-N241	2459.49	1431.2	241	1446.06	1430	203	1.7	0.99	1.19
Golden-1-C27-N241	2251.72	1432	241	1244.55	1432	201	1.81	1	1.2
Golden-1-C31-N241	1363.08	1503	241	922.63	1503	201	1.48	1	1.2
Golden-1-C35-N241	3976.06	1570.71	241	4851.46	1570.71	239	0.82	1	1.01
Golden-2-C22-N321	1979.31	1679	321	578.54	1679	227	3.42	1	1.41
Golden-2-C23-N321	1616.33	1703	321	707.99	1703	244	2.28	1	1.32
Golden-2-C25-N321	2151.96	1806	321	841.65	1806	242	2.56	1	1.33
Golden-2-C27-N321	4985.62	1826	321	1120.09	1826	238	4.45	1	1.35
Golden-2-C30-N321	7200.92	-1E15	321	3200.3	1881.67	266	>2.25	–	1.21
Golden-2-C33-N321	7200.94	-1E15	321	4944.97	2000.21	280	>1.46	–	1.15
Golden-3-C27-N401	6788.38	2232	401	1399.33	2232	290	4.85	1	1.38
Golden-3-C29-N401	7200.98	0	401	5834.28	2260	318	>1.23	–	1.26
Average	3083		288	1730		230	>2.14	~ 1	1.25

Table: Caption

Thanks for your attention!