A Scenario Based Single Vehicle Routing Problem with Stochastic Demands: Flow Models

Marcus Poggi.*, Georges Spyrides, Thibaut Vidal
Departamento de Informática, PUC-Rio, Brazil
{ poggi, gspyrides, vidalt}@inf.puc-rio.br
* and Accenture

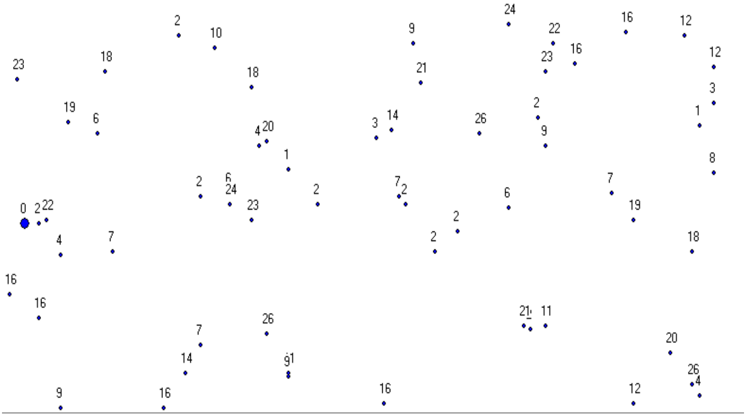
November 22, 2019

Capacitated Vehicle Routing Problem (CVRP)

Given

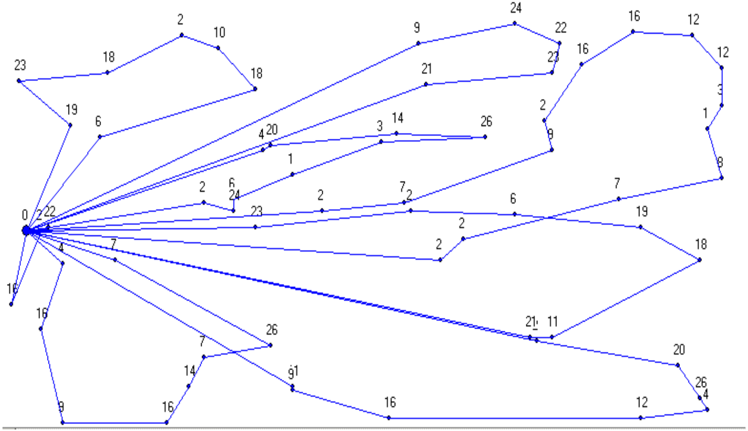
- An undirected graph G=(V,E) V= {0,1,...,n}
  - Vertex 0 represents a depot and remaining vertices represent clients
  - Edge lengths are denoted by c(e)
- Client demands are q(1),...,q(n)
- K vehicles with capacity C
- Determine routes for each vehicle satisfying the following constraints:
  - (i) each route starts and ends at the depot,
  - (ii) each client is visited by exactly one vehicle
  - (iii) the total demand of clients visited in a route is at most C
- The objective is to minimize the total route length.

Example - Instance A-n62-k8



Vehicle Capacity: 100

Optimal Solution A-n62-k8



Optimal Solution Value: 1288

Routing Problems with Stochastic Demands

- The common version assumes the demand at a client is only revealed at arrival: Pick-up problem
- The objective is to minimize the expected total distance traveled.
- Distances are deterministic (no probabilistic duration here)
- A **failure** along a route occurs when the client demand uncovered does not fit in the vehicle
- Different ways to manage **failures** (recourse) lead to different VRPSD versions
- Different ways to model the demand uncertainty lead to different approaches

Routing Problems with Stochastic Demands

- The common version assumes the demand at a client is only revealed at arrival: Pick-up problem
- The objective is to minimize the expected total distance traveled.
- Distances are deterministic (no probabilistic duration here)
- A **failure** along a route occurs when the client demand uncovered does not fit in the vehicle
- Different ways to manage **failures** (recourse) lead to different VRPSD versions
- Different ways to model the demand uncertainty lead to different approaches

Routing Problems with Stochastic Demands

- The common version assumes the demand at a client is only revealed at arrival: Pick-up problem
- The objective is to minimize the expected total distance traveled.
- Distances are deterministic (no probabilistic duration here)
- A **failure** along a route occurs when the client demand uncovered does not fit in the vehicle
- Different ways to manage **failures** (recourse) lead to different VRPSD versions
- Different ways to model the demand uncertainty lead to different approaches

Stochastic Demands: Some Literature

- Tillman, The Multiterminal delivery Problem with Stochastic Demands, TS, 1969
- Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks Dror, Laporte, Trudeau, TS, 1989 (MDP)
- A vehicle routing problem with stochastic demand, Bertsimas, OR, 1992.

Stochastic Demands: Managing Failures

- Recourse Action on client loading failure:
  - Preventive: Goes back to unload whenever there is a high probability of failure on the next client
  - Non-Preventive: occurs a failure on current client
    - Goes to the depot and returns to current client (popular!)
    - Goes to the depot and exchanges next, 2, 3, ... clients in an optimal way
    - Goes to the depot and reshuffles all remaining clients optimally
  - Chance constraints: guarantees a low probability of failure
  - Penalty for not serving a client
  - Optimal Re-stocking Policy: Given a fixed sequence of visits and demand distributions(i.i.) optimal re-stocking can be determined: for return to the depot and back to current client.

Stochastic Demands: Managing Failures

- Recourse Action on client loading failure:
  - Preventive: Goes back to unload whenever there is a high probability of failure on the next client
  - Non-Preventive: occurs a failure on current client
    - Goes to the depot and returns to current client (popular!)
    - Goes to the depot and exchanges next, 2, 3, ... clients in an optimal way
    - Goes to the depot and reshuffles all remaining clients optimally
  - Chance constraints: guarantees a low probability of failure
  - Penalty for not serving a client
  - Optimal Re-stocking Policy: Given a fixed sequence of visits and demand distributions(i.i.) optimal re-stocking can be determined: for return to the depot and back to current client.

Stochastic Demands: More literature

- Louveaux, Salazar Gonzalez, 2018), arbitrary discrete distribution, preventive, small demands, L-Shaped
- Dinh, Fukasawa, Luedtke, 2017, Chance constraints
- Florio, Hartl, Minner, 2019, BCP (Probabilistic Duration), Optimal Re-stocking Policy (Dynamic Programming + BCP), non-preventive,
- Salavati, Gendreau, Jabali, Rei, 2019, Optimal Re-stocking Policy, Preventive and Non-preventive, L-Shaped

Single Vehicle Routing with Stochastic Demands

- Zhang et al., T. Sci., 2014: Approx Dyn. Prog.
- Florio, Hartl, Minner, EJOR, 2018, MDP, Hamiltonian Policy

Routing Problems with Stochastic Demands

- What applied VRPSD problems are out there?
- What experiment would supply evidence that the Stochastic approach should be chosen?
  - Take a (representative) history of working (similar) days demands, and simulate the proposed routing strategy
- How much would it help should there be long routes?
- How much uncertainty would have significant impact?
  - A first guess: Either some demands vary a lot, or there are positive correlations
  - Is the i.i.d. hypothesis adequate?
- There many transportation companies with steady pick-ups
- Many clients have significant uncertainty on the demand
- Mostly they execute the same routes every day

Routing Problems with Stochastic Demands

- What applied VRPSD problems are out there?
- What experiment would supply evidence that the Stochastic approach should be chosen?
  - Take a (representative) history of working (similar) days demands, and simulate the proposed routing strategy
- How much would it help should there be long routes?
- How much uncertainty would have significant impact?
  - A first guess: Either some demands vary a lot, or there are positive correlations
  - Is the i.i.d. hypothesis adequate?
- There many transportation companies with steady pick-ups
- Many clients have significant uncertainty on the demand
- Mostly they execute the same routes every day

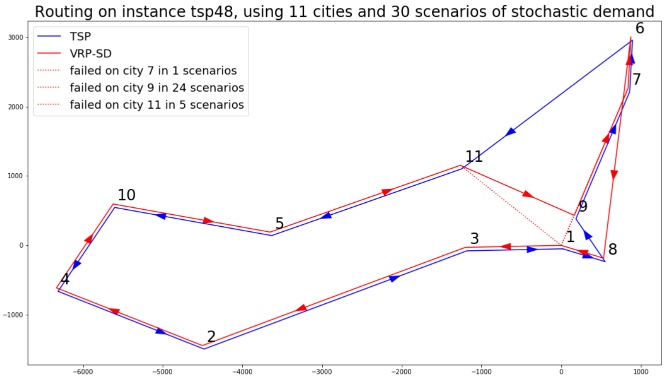
Routing Problems with Stochastic Demands

- What applied VRPSD problems are out there?
- What experiment would supply evidence that the Stochastic approach should be chosen?
  - Take a (representative) history of working (similar) days demands, and simulate the proposed routing strategy
- How much would it help should there be long routes?
- How much uncertainty would have significant impact?
  - A first guess: Either some demands vary a lot, or there are positive correlations
  - Is the i.i.d. hypothesis adequate?
- There many transportation companies with steady pick-ups
- Many clients have significant uncertainty on the demand
- Mostly they execute the same routes every day

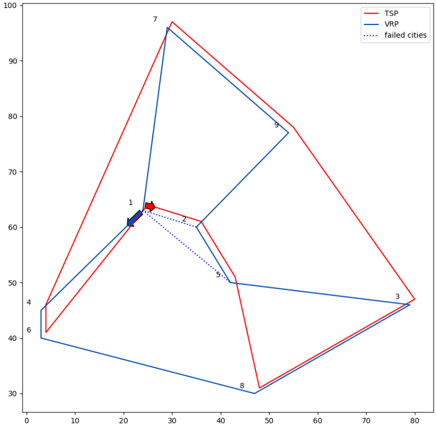
Routing Problems with Stochastic Demands

- What applied VRPSD problems are out there?
- What experiment would supply evidence that the Stochastic approach should be chosen?
  - Take a (representative) history of working (similar) days demands, and simulate the proposed routing strategy
- How much would it help should there be long routes?
- How much uncertainty would have significant impact?
  - A first guess: Either some demands vary a lot, or there are positive correlations
  - Is the i.i.d. hypothesis adequate?
- There many transportation companies with steady pick-ups
- Many clients have significant uncertainty on the demand
- Mostly they execute the same routes every day

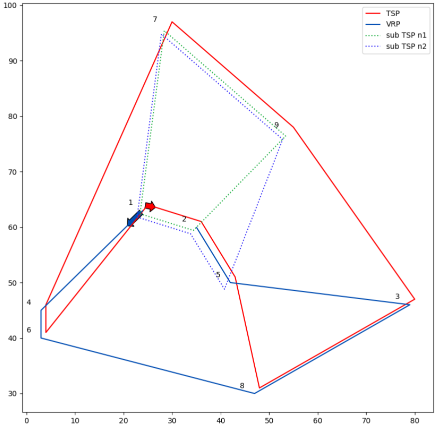
Go to Depot and Back Recourse Policy



Go to Depot and Back Recourse Policy



Reshuffle Recourse Policy



The VRPSDS Problem

Scenario approach for Single Vehicle Routing

Given:

- A vehicle with capacity Q, travel costs c_{ij}, and demands d_{vs};
- Assume the sum of the demands fall in (Q,2Q) with high probability
- In a given demand scenario a recourse must be taken on the failure client;
- Recourse function to manage *failure*:
  - Go to the depot unload and return to current client
  - Go to the depot and follow the optimal route for remaining clients

Determine:

- Sequence of client visits that minimizes the expected total traveling cost.

Optimal sequence is such that, for the scenarios considered, the expect overhead due to exceeding vehicle capacity is minimized

The VRPSDS Problem

Scenario approach OR Periodic VRP

- Periodic VRP: demands are known for the next S days
- Scenario approach: a sample of client demands from several "same days" are available

Formulation Based on the Directed TSP Flow formulation

Let:

- y_{ij}: (i, j) is used in the first level tour
- x_{ijk}: (i, j) is traversed by the flow from vertex 1 to vertex k

Minimizing the Expected Total Distance Traveled

Single Vehicle TSP with Recourse Cost $\alpha(y, x)$

$$\min \quad \sum_{i=1}^{n} \sum_{j=2}^{n+1} c_{ij} y_{i,j} + \alpha(y, x)$$

TSP-Flow Constraints

Minimizing the Expected Total Distance Traveled

Single Vehicle TSP with Recourse Cost $\alpha(y,x)$

TSP-Flow Constraints

$$\text{(out d)} \quad \sum_{i=1, i\neq j}^{n} y_{ij} = 1 \qquad \forall j = 2, n+1$$

$$\text{(in d)} \quad \sum_{j=2, j\neq i}^{n+1} y_{ij} = 1 \qquad \forall i = 1, n$$

$$\text{(flow out)} \quad \sum_{j=2}^{n} x_{1jk} = 1 \qquad \forall k = 2, n+1$$

$$\text{(flow cons)} \quad \sum_{j=2}^{n+1} x_{ijk} - \sum_{j=1}^{n} x_{jik} = \{-1 \;\; i = k; \;\; 0 \;\; i \neq k\} \qquad \forall i = 1, n \quad \forall k = 1, n$$

$$\text{(couple)} \quad x_{ijk} - y_{ij} \leq 0 \qquad \forall i = 1, n; \quad j = 2, n+1; \quad k = 2, n+1$$

$$y_{ij} \in \{0, 1\}$$

$$x_{ijk} \in \{0, 1\}$$

Recourse: Return to the Depot and Back to the current client

$$\min \quad \sum_{i=1}^{n} \sum_{j=2}^{n+1} c_{ij} y_{i,j} + \frac{1}{nscen} \sum_{s=1}^{nscen} \sum_{k=2}^{n} (c_{k(n+1)} + c_{1k}).pay_{ks}$$

TSP-Flow Constraints

Scenario Constraints

Scenario Variables

Let:

- o_i: contains the position of vertex i on the first level route
- w_{is}: indicates client i is served after capacity failure in scenario s.
- p^+_{ks}: the failure occurs after client k in the route
- p^-_{ks}: if it is positive the failure occurs before client k in the route
- s_{ks}: one if only p^+_{ks} can be positive, if it is zero, only p^-_{ks} can be positive
- pay_{ks}: indicates fail on client k in scenario s: go to the depot and return
- $unpay_{ijs}$: indicates the reshuffle recourse will not use edge (i, j) in s

Scenario Constraints

$$
\begin{array}{llr}
\text{(serve)} & \sum_{i=1}^{n} \sum_{j=2, j \neq i}^{n} d_{js} x_{ijk} - Q.w_{ks} \leq Q & \forall k = 2, \ldots, n \\
\text{(serve comp)} & \sum_{i=1}^{n} \sum_{j=2, j \neq i}^{n} d_{js} x_{ijk} - Q.w_{ks} \geq 0 & \forall k = 2, \ldots, n \\
\text{(ord rout)} & \sum_{i=1}^{n} \sum_{j=2, j \neq i}^{n} x_{ijk} - o_k = 0 & \forall k = 2, \ldots, n \\
\text{(prec ik)} & w_{is} - w_{ks} \leq 1 - x_{ijk} & \forall i > 1, i \neq k, k = 2, \ldots, n \\
\text{(prec jk)} & w_{js} - w_{ks} \leq 1 - x_{ijk} & \forall j > 1, j \neq k, k = 2, \ldots, n \\
\text{(2stage cl)} & \sum_{k=2}^{n} d_{ks} w_{ks} \leq Q & \forall s \\
\text{(notserv)} & \sum_{k=2}^{n} w_{ks} - nns_s = 0 & \forall s \\
\text{(last serv)} & o_k + nns_s + p_{ks}^{+} - p_{ks}^{-} = n & \forall k = 2, \ldots, n \quad \forall s \\
\text{(pos sign)} & p_{ks}^{+} - n.s_{ks} \leq 0 & \forall k = 2, \ldots, n \quad \forall s \\
\text{(neg sign)} & p_{ks}^{-} + n.s_{ks} \leq n & \forall k = 2, \ldots, n \quad \forall s \\
\text{(set pay)} & pay_{ks} + p_{ks}^{+} + p_{ks}^{-} \geq 1 & \forall k = 2, \ldots, n \quad \forall s
\end{array}
$$

Reshuffle Recourse

Return to the depot and perform optimal route on clients not served

$$\min \quad \sum_{i=1}^{n}\sum_{j=2}^{n+1} c_{ij} y_{i,j} + \frac{1}{nscen}\sum_{s=1}^{nscen}(\sum_{k=2}^{n} c_{k(n+1)}.pay_{ks} - \sum_{i=1}^{n}\sum_{j=2}^{n+1} c_{ij}.unpay_{ijs} + \sum_{i=1}^{n}\sum_{j=2}^{n+1} c_{ij}.g_{ijs})$$

TSP-Flow Constraints

Scenario Constraints

TSP Scenario

TSP Scenario

$$\sum_{i=1}^{n} g_{ijs} - w_{js} = 0 \qquad \forall j = 2, \ldots, n+1$$

$$\sum_{j=2}^{n+1} g_{ijs} - w_{is} = 0 \qquad \forall i = 1, \ldots, n$$

$$\sum_{j=2}^{n} f_{1jks} - w_{ks} = 0 \qquad \forall k = 2, \ldots, n+1$$

$$\sum_{j=2}^{n+1} f_{ijks} - \sum_{j=1}^{n} f_{jiks} = \{-w_{ks} \;\; i = k; \;\; 0 \;\; i \neq k\} \qquad \forall i = 1, \ldots, n \quad \forall k = 1, \ldots, n$$

$$unpay_{ijs} \leq w_{is}$$

$$unpay_{ijs} \leq w_{js}$$

$$unpay_{ijs} \geq w_{is} + w_{js} - 1$$

$$unpay_{ijs} \in \{0, 1\}$$

$$g_{ijs} \in \{0, 1\}$$

$$f_{ijks} \in \{0, 1\}$$

Combinatorial Benders Cuts

Let y^+ be a solution for (TSP constraints)
Let $zr_\alpha(y^+)$ be the expected recourse cost associated with y^+
Then:

$$\alpha \geq zr_\alpha(y^+).(1 - n + \sum_{(i,j)|y^+_{ij}=1} y_{ij})$$

is the associated Benders Combinatorial (or Optimality) cut.

Prefix Combinatorial Benders Cuts

Let y^p be the incidence vector of edges corresponding to a **prefix** of a route
Let p be the number of edges in the path starting at the depot
Let $zr^s_\alpha(y^p)$ be the recourse cost associated with y^p in scenario s (Depot and back or Reshuffle)
Let also α_s be the recourse cost associated to scenario s
If in scenario s we have $\sum_{(i,j)|y^p_{ij}=1} d_{js} > Q$; Then:

$$\alpha_s \geq zr^s_\alpha(y^p).(1 - p + \sum_{(i,j)|y^p_{ij}=1} y_{ij})$$

is the associated Benders Combinatorial (or Optimality) cut for scenario s.

Naive 1: Benders Decomposition - Two Stage

- Initialization: set $i = 1$, a lower bound $\underline{z} = -\infty$, and an upper bound $\overline{z} = +\infty$
  - S1 Solve current first stage determining trial solution y^i and let $\underline{z} = z(y^i)$
  - S2 Solve the second stage determining $zr_\alpha(y^i)$
  - S3 Set $\overline{z} = c.y^i + zr_\alpha(y^i)$
  - S4 Add the Benders Combinatorial (or Optimality) Cut to the first stage.
  - S5 IF $\overline{z} - \underline{z} < \epsilon$. STOP.
    Otherwise set $i = i + 1$, goto S1

Algorithms tested

- Naive Benders with Combinatorial Cuts
- Proposed model as is.

Instances

- Based on TSPLIB instances. Euclidean instances are hard.
- Demands:
  - Mean μ_i is generated as UNIF[50, 80] for each client
  - Client i in scenario s has demand q_{is} from UNIF$[0.7\mu_i, 1.3\mu_i]$

We also experiment with more uneven demands.

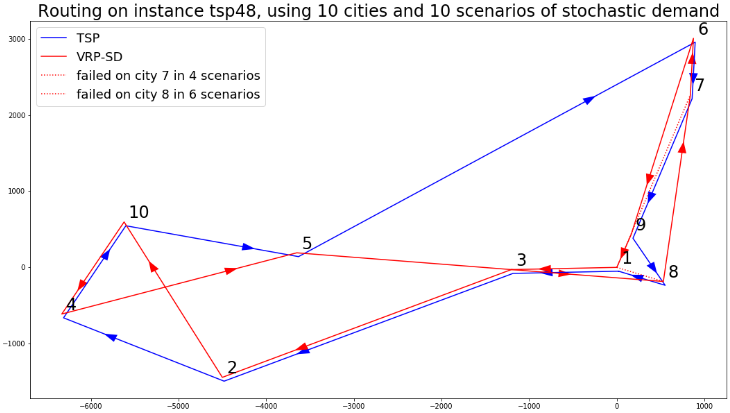
Test-Bed

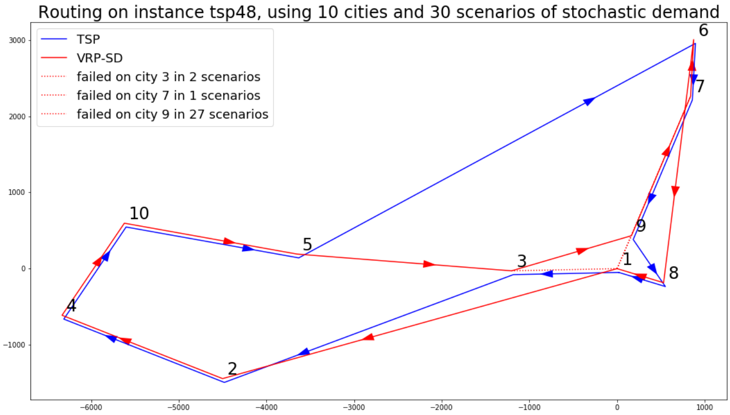
- All computational experiments were executed on the cloud, on VMs with the following characteristics:
  - Quad core intel Xeon processor @2.6 Ghz
  - 26 Gb of RAM
  - Windows Server 2012 - 64bit
- Code developed using Julia 1.1 + JuMP modeling language.
- Mathematical programming solver: Gurobi

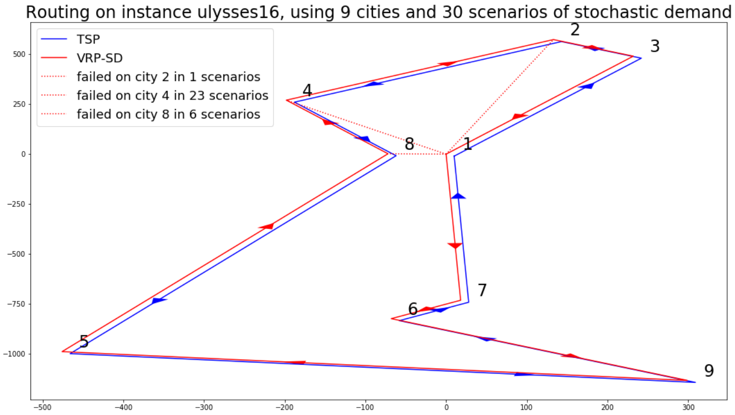
Results for Recourse Policy 1: Go to the depot and back

- TSP with combinatorial Benders can solve until 12 in a few minutes
- The model also solves for 12, takes longer
- Integrality gap between 5% and 15%
- Sensitive to instances (very easy and very hard)
- Large recourse cost implies harder instances
- Largest instance solved has 15 clients

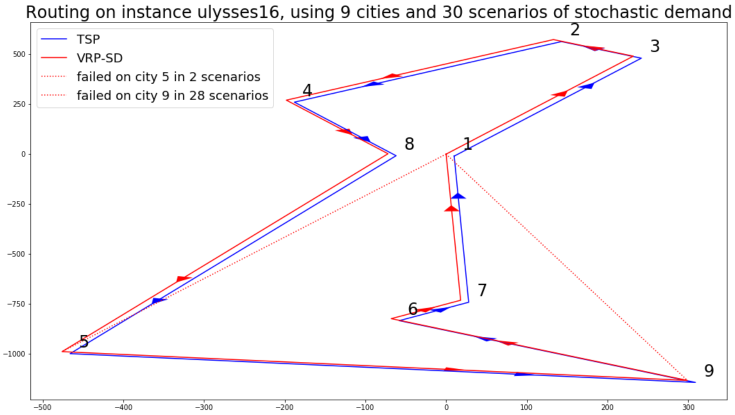
We present next some optimal solutions


Routing on instance tsp48, using 10 cities and 10 scenarios of stochastic demand
TSP
VRP-SD
failed on city 7 in 4 scenarios
failed on city 8 in 6 scenarios
1
2
3
4
5
6
7
8
9
10
3000
2000
1000
0
−1000
−6000
−5000
−4000
−3000
−2000
−1000
0
1000

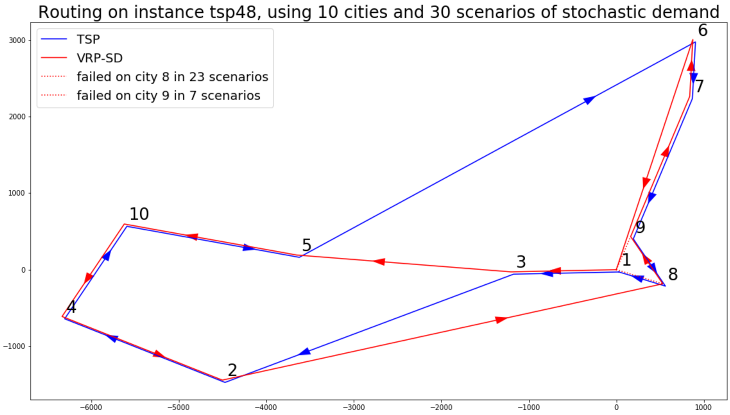

Routing on instance tsp48, using 10 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 3 in 2 scenarios
failed on city 7 in 1 scenarios
failed on city 9 in 27 scenarios
3000
2000
1000
0
−1000
−6000
−5000
−4000
−3000
−2000
−1000
0
1000
10
5
3
1
9
6
7
8
4
2


Routing on instance ulysses16, using 9 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 2 in 1 scenarios
failed on city 4 in 23 scenarios
failed on city 8 in 6 scenarios
1
2
3
4
5
6
7
8
9
500
250
0
−250
−500
−750
−1000
−500
−400
−300
−200
−100
0
100
200
300

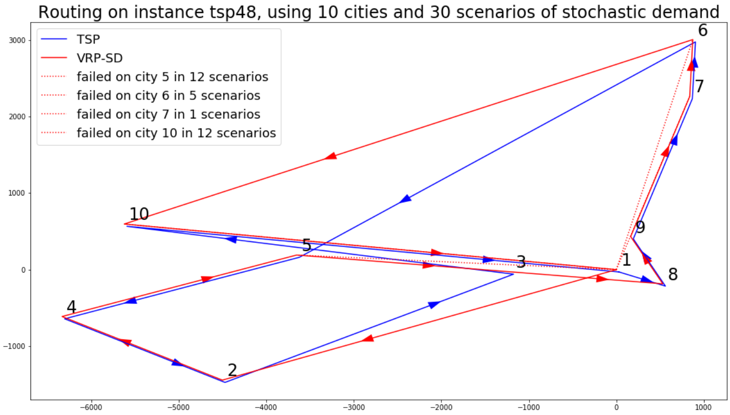
	2	3	4	5	6	7	8	9
0	92.0	46.0	79.0	47.0	45.0	63.0	66.0	49.0
1	66.0	80.0	60.0	46.0	68.0	49.0	59.0	62.0
2	68.0	51.0	70.0	60.0	75.0	56.0	49.0	77.0
3	79.0	47.0	64.0	50.0	58.0	70.0	47.0	50.0
4	90.0	45.0	64.0	45.0	44.0	48.0	58.0	58.0
5	70.0	65.0	83.0	64.0	61.0	59.0	49.0	64.0
6	55.0	55.0	68.0	47.0	60.0	62.0	84.0	63.0
7	80.0	69.0	60.0	79.0	44.0	77.0	52.0	74.0
8	81.0	70.0	59.0	48.0	65.0	63.0	62.0	54.0
9	92.0	61.0	92.0	66.0	42.0	54.0	82.0	83.0
10	67.0	61.0	94.0	78.0	66.0	83.0	66.0	54.0
11	83.0	61.0	92.0	63.0	75.0	48.0	47.0	56.0
12	84.0	62.0	69.0	51.0	74.0	56.0	60.0	79.0
13	80.0	69.0	69.0	50.0	64.0	71.0	68.0	70.0
14	91.0	75.0	65.0	49.0	56.0	76.0	52.0	59.0
15	81.0	71.0	74.0	80.0	45.0	76.0	47.0	67.0
16	59.0	63.0	86.0	71.0	60.0	66.0	65.0	87.0
17	63.0	56.0	101.0	45.0	42.0	48.0	65.0	77.0
18	85.0	59.0	64.0	63.0	73.0	48.0	78.0	83.0
19	66.0	57.0	99.0	64.0	61.0	75.0	66.0	66.0
20	59.0	48.0	71.0	47.0	42.0	52.0	74.0	70.0
21	79.0	74.0	59.0	66.0	69.0	65.0	81.0	63.0
22	69.0	53.0	57.0	59.0	72.0	53.0	72.0	71.0
23	58.0	72.0	77.0	79.0	42.0	68.0	62.0	52.0
24	57.0	45.0	86.0	62.0	75.0	78.0	54.0	47.0
25	85.0	57.0	92.0	52.0	60.0	64.0	69.0	84.0
26	52.0	48.0	61.0	57.0	68.0	80.0	80.0	77.0
27	78.0	67.0	82.0	83.0	65.0	76.0	75.0	55.0
28	54.0	66.0	91.0	55.0	70.0	58.0	66.0	60.0
29	71.0	48.0	90.0	68.0	60.0	78.0	64.0	56.0


Routing on instance ulysses16, using 9 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 5 in 2 scenarios
failed on city 9 in 28 scenarios
1
2
3
4
5
6
7
8
9
500
250
0
−250
−500
−750
−1000
−500
−400
−300
−200
−100
0
100
200
300

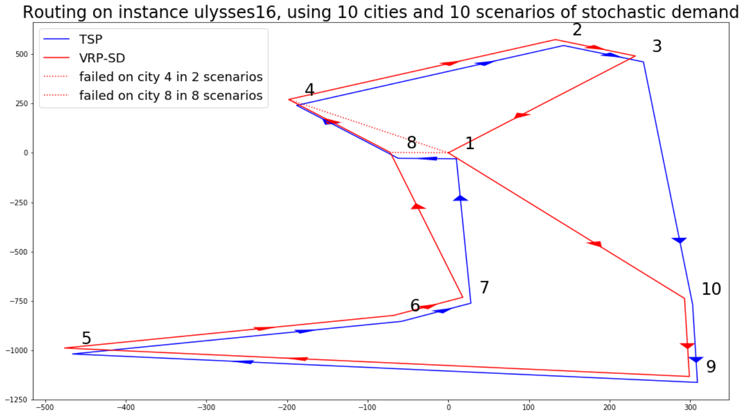
	2	3	4	5	6	7	8	9
0	24.0	23.0	31.0	57.0	44.0	29.0	26.0	70.0
1	34.0	36.0	28.0	77.0	48.0	48.0	28.0	70.0
2	39.0	31.0	30.0	71.0	29.0	44.0	16.0	65.0
3	24.0	27.0	26.0	66.0	37.0	52.0	21.0	88.0
4	33.0	33.0	23.0	63.0	45.0	47.0	20.0	97.0
5	24.0	30.0	18.0	65.0	36.0	31.0	21.0	100.0
6	41.0	40.0	30.0	88.0	35.0	38.0	22.0	76.0
7	26.0	39.0	30.0	77.0	46.0	50.0	24.0	99.0
8	30.0	30.0	22.0	57.0	44.0	32.0	24.0	59.0
9	26.0	23.0	24.0	74.0	40.0	43.0	25.0	80.0
10	30.0	33.0	28.0	66.0	48.0	47.0	19.0	67.0
11	36.0	31.0	29.0	76.0	31.0	33.0	27.0	87.0
12	30.0	30.0	30.0	52.0	32.0	50.0	25.0	100.0
13	40.0	27.0	30.0	89.0	29.0	43.0	25.0	65.0
14	40.0	23.0	30.0	65.0	42.0	28.0	17.0	70.0
15	38.0	38.0	29.0	55.0	39.0	34.0	28.0	66.0
16	28.0	32.0	22.0	57.0	42.0	45.0	25.0	84.0
17	32.0	23.0	27.0	58.0	42.0	32.0	19.0	76.0
18	27.0	26.0	24.0	60.0	42.0	39.0	18.0	99.0
19	31.0	22.0	31.0	58.0	37.0	36.0	27.0	88.0
20	25.0	34.0	24.0	51.0	35.0	37.0	24.0	89.0
21	28.0	25.0	28.0	81.0	44.0	35.0	24.0	60.0
22	39.0	28.0	21.0	63.0	37.0	30.0	26.0	64.0
23	41.0	30.0	20.0	76.0	44.0	51.0	24.0	97.0
24	26.0	33.0	27.0	58.0	44.0	46.0	22.0	65.0
25	33.0	31.0	22.0	65.0	28.0	43.0	24.0	66.0
26	30.0	28.0	18.0	70.0	26.0	40.0	25.0	72.0
27	23.0	30.0	18.0	75.0	40.0	48.0	17.0	61.0
28	34.0	26.0	29.0	70.0	30.0	52.0	28.0	58.0
29	27.0	27.0	19.0	76.0	35.0	42.0	19.0	58.0

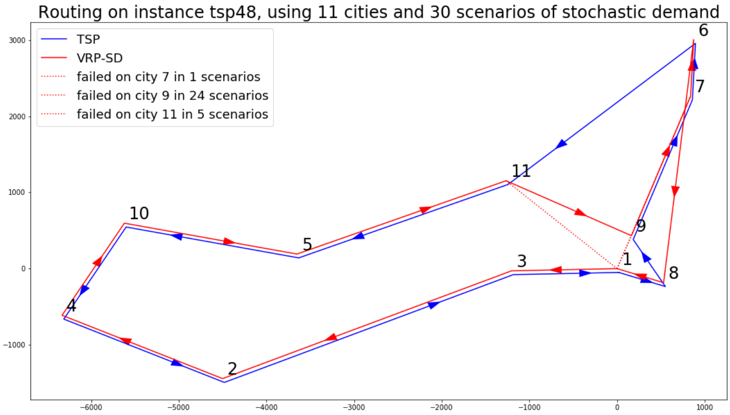

Routing on instance tsp48, using 10 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 8 in 23 scenarios
failed on city 9 in 7 scenarios
3000
2000
1000
0
−1000
−6000
−5000
−4000
−3000
−2000
−1000
0
1000
1
2
3
4
5
6
7
8
9
10

	2	3	4	5	6	7	8	9	10
0	89.0	67.0	49.0	72.0	76.0	79.0	66.0	48.0	64.0
1	61.0	56.0	46.0	91.0	74.0	61.0	71.0	50.0	50.0
2	58.0	78.0	54.0	76.0	82.0	61.0	73.0	54.0	46.0
3	81.0	56.0	56.0	89.0	47.0	73.0	70.0	68.0	63.0
4	71.0	54.0	52.0	61.0	82.0	61.0	50.0	61.0	53.0
5	54.0	50.0	56.0	82.0	61.0	72.0	66.0	64.0	60.0
6	52.0	55.0	42.0	62.0	51.0	48.0	71.0	69.0	52.0
7	53.0	45.0	62.0	73.0	80.0	59.0	66.0	53.0	52.0
8	79.0	76.0	46.0	78.0	73.0	74.0	59.0	63.0	48.0
9	62.0	70.0	43.0	54.0	66.0	56.0	55.0	62.0	64.0
10	68.0	59.0	56.0	77.0	74.0	67.0	86.0	61.0	36.0
11	70.0	62.0	36.0	82.0	84.0	61.0	54.0	66.0	44.0
12	67.0	58.0	57.0	85.0	53.0	65.0	53.0	70.0	62.0
13	56.0	58.0	65.0	64.0	83.0	55.0	57.0	52.0	61.0
14	80.0	43.0	40.0	50.0	73.0	72.0	58.0	66.0	53.0
15	59.0	50.0	55.0	58.0	64.0	66.0	62.0	52.0	45.0
16	65.0	71.0	50.0	90.0	63.0	66.0	73.0	54.0	45.0
17	90.0	64.0	46.0	61.0	77.0	75.0	62.0	46.0	48.0
18	67.0	52.0	43.0	81.0	61.0	65.0	59.0	44.0	59.0
19	78.0	68.0	53.0	68.0	74.0	76.0	51.0	51.0	37.0
20	52.0	75.0	49.0	86.0	74.0	63.0	57.0	75.0	52.0
21	74.0	78.0	63.0	64.0	46.0	56.0	62.0	68.0	37.0
22	54.0	71.0	66.0	68.0	83.0	77.0	50.0	52.0	63.0
23	69.0	62.0	43.0	55.0	56.0	68.0	65.0	48.0	38.0
24	59.0	58.0	53.0	67.0	55.0	44.0	86.0	42.0	60.0
25	59.0	58.0	40.0	55.0	77.0	59.0	57.0	55.0	45.0
26	74.0	45.0	63.0	52.0	55.0	74.0	81.0	75.0	64.0
27	83.0	60.0	52.0	51.0	63.0	49.0	69.0	45.0	41.0
28	76.0	70.0	39.0	56.0	66.0	78.0	55.0	45.0	36.0
29	63.0	78.0	52.0	53.0	49.0	58.0	71.0	70.0	53.0


Routing on instance tsp48, using 10 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 5 in 12 scenarios
failed on city 6 in 5 scenarios
failed on city 7 in 1 scenarios
failed on city 10 in 12 scenarios
1
2
3
4
5
6
7
8
9
10
3000
2000
1000
0
−1000
−6000
−5000
−4000
−3000
−2000
−1000
0
1000

	2	3	4	5	6	7	8	9	10
0	39.0	26.0	55.0	31.0	39.0	35.0	17.0	25.0	71.0
1	39.0	20.0	92.0	32.0	29.0	34.0	20.0	27.0	73.0
2	32.0	19.0	81.0	33.0	39.0	34.0	20.0	31.0	59.0
3	48.0	22.0	78.0	39.0	31.0	24.0	23.0	28.0	52.0
4	51.0	30.0	88.0	29.0	29.0	27.0	24.0	17.0	70.0
5	39.0	24.0	87.0	27.0	32.0	36.0	19.0	23.0	67.0
6	55.0	20.0	85.0	35.0	30.0	25.0	23.0	20.0	60.0
7	36.0	20.0	60.0	30.0	29.0	27.0	21.0	18.0	57.0
8	38.0	28.0	64.0	22.0	35.0	35.0	14.0	30.0	75.0
9	50.0	25.0	83.0	22.0	30.0	32.0	17.0	18.0	60.0
10	57.0	32.0	54.0	35.0	35.0	23.0	16.0	26.0	76.0
11	38.0	23.0	81.0	21.0	30.0	36.0	14.0	19.0	67.0
12	43.0	20.0	63.0	31.0	24.0	29.0	16.0	18.0	63.0
13	53.0	20.0	80.0	33.0	22.0	39.0	18.0	19.0	75.0
14	49.0	18.0	57.0	36.0	25.0	34.0	19.0	23.0	44.0
15	53.0	18.0	77.0	21.0	40.0	26.0	25.0	27.0	53.0
16	35.0	26.0	85.0	26.0	33.0	32.0	26.0	17.0	52.0
17	50.0	32.0	59.0	28.0	26.0	33.0	17.0	19.0	55.0
18	53.0	29.0	67.0	24.0	39.0	35.0	21.0	26.0	65.0
19	37.0	19.0	89.0	22.0	25.0	41.0	20.0	29.0	61.0
20	55.0	24.0	57.0	27.0	38.0	23.0	23.0	17.0	67.0
21	37.0	31.0	65.0	37.0	26.0	33.0	19.0	21.0	72.0
22	34.0	28.0	62.0	29.0	22.0	38.0	15.0	28.0	74.0
23	51.0	18.0	64.0	35.0	37.0	30.0	17.0	29.0	61.0
24	32.0	28.0	59.0	22.0	37.0	33.0	16.0	18.0	74.0
25	45.0	31.0	80.0	33.0	23.0	35.0	18.0	19.0	61.0
26	49.0	20.0	81.0	33.0	23.0	34.0	15.0	19.0	73.0
27	47.0	18.0	71.0	22.0	32.0	35.0	17.0	22.0	48.0
28	32.0	19.0	90.0	22.0	39.0	23.0	20.0	26.0	74.0
29	48.0	19.0	56.0	38.0	24.0	27.0	22.0	26.0	54.0


Routing on instance ulysses16, using 10 cities and 10 scenarios of stochastic demand
TSP
VRP-SD
failed on city 4 in 2 scenarios
failed on city 8 in 8 scenarios
1
2
3
4
5
6
7
8
9
10
500
250
0
−250
−500
−750
−1000
−1250
−500
−400
−300
−200
−100
0
100
200
300


Routing on instance tsp48, using 11 cities and 30 scenarios of stochastic demand
TSP
VRP-SD
failed on city 7 in 1 scenarios
failed on city 9 in 24 scenarios
failed on city 11 in 5 scenarios
1
2
3
4
5
6
7
8
9
10
11
3000
2000
1000
0
−1000
−6000
−5000
−4000
−3000
−2000
−1000
0
1000

Conclusions

- A first look at the problems shows difficulty starts for routes over 13 clients
- This means it is close to enumeration (matches the harder CVRP instances for BCP)
- Good initial upper bounds help significantly as for any enumeration algorithm

Conclusions

- A first look at the problems shows difficulty starts for routes over 13 clients
- This means it is close to enumeration (matches the harder CVRP instances for BCP)
- Good initial upper bounds help significantly as for any enumeration algorithm

Some Potentially Non-Investigated Questions:

- Dynamic Programming Approach for the Single Vehicle VRPSDS (Fukasawa?)
- For iid demands: DP for optimal next 2, 3, 4, ... clients
- Scenario based model for optimal next 2, 3, 4, ... clients

Some Potentially Non-Investigated Questions:

- Dynamic Programming Approach for the Single Vehicle VRPSDS (Fukasawa?)
- For iid demands: DP for optimal next 2, 3, 4, ... clients
- Scenario based model for optimal next 2, 3, 4, ... clients

Future work

- Study lifting of classical Benders cuts (generating route is known)
- Collect and try real Stochastic Demands for VRP's

Thank you!