Stochastic Vehicle Routing Problems

Ola Jabali

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano

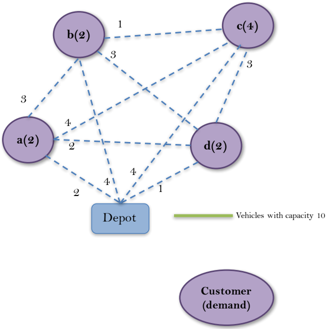
The VeRoLog PhD school 2018

June 2, 2018

Collaborators

- Fausto Errico, École de technologie supérieure de Montréal, Canada
- Guy Desaulniers École Polytechnique de Montréal, GERAD
- Michel Gendreau, École Polytechnique de Montréal and CIRRELT, Canada
- Florent Hernandez, Kronos Incorporated, Canada
- Ton de Kok, Eindhoven University of Technology, The Netherlands
- Gilbert Laporte, HEC Montréal and CIRRELT, Canada
- Walter Rei, Université du Québec à Montréal and CIRRELT, Canada
- Majid Salavati-Khoshghalb, Université de Montréal and CIRRELT, Canada
- Duygu Taş, MEF University, Turkey
- Tom Van Woensel, Eindhoven University of Technology, The Netherlands

VRP



The Vehicle Routing Problem (VRP)- Basic setting

- Given a set of customers and their associated demand
- Given a depot and a set of homogeneous capacitated vehicles
- Given the traveling time from one location to another

Objective: Design feasible routes that minimize total cost

The Vehicle Routing Problem (VRP)

- Almost 60 years old
- Some statistics

Keyword \Journal	EJOR	OR	TS	C&OR
"Vehicle Routing"	1180	72	127	715

SVRP

The Vehicle Routing Problem (VRP)- Basic setting

- Given set of customers and their associated demand
- Given a depot and a set of homogeneous capacitated vehicles
- Given the traveling time from one location to another

Objective: Design feasible routes that minimize total cost

What can be stochastic

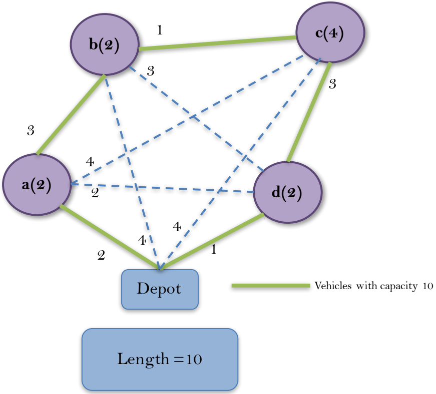
- Customer demand
- Travel or service time
- Presence of customers
- ...

The Stochastic Vehicle Routing Problem (SVRP)

- Almost 50 years old
- Some statistics

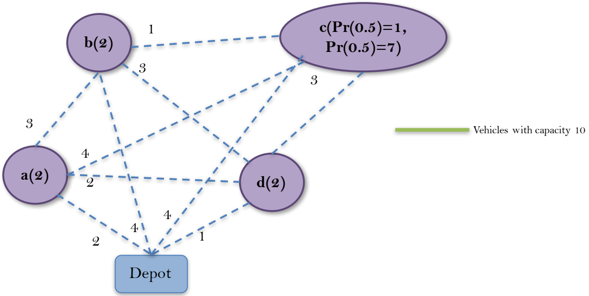
Keyword \Journal	EJOR	OR	TS	C&OR
“Vehicle Routing"		72	127	
+“Stochastic"		17	18	

Example: VRP



Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

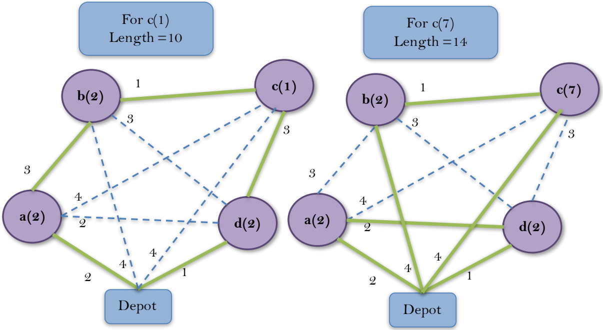
Example: VRP with stochastic demand



Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Example: VRP with stochastic demand

If we know the demand of customer c in advance

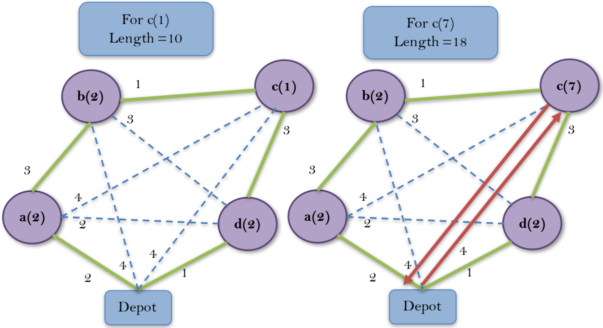


WS = 12.0 = expected length, if information is available in advance

Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Example: VRP with stochastic demand

If we don't know the demand of customer c in advance, how about applying the VRP solution

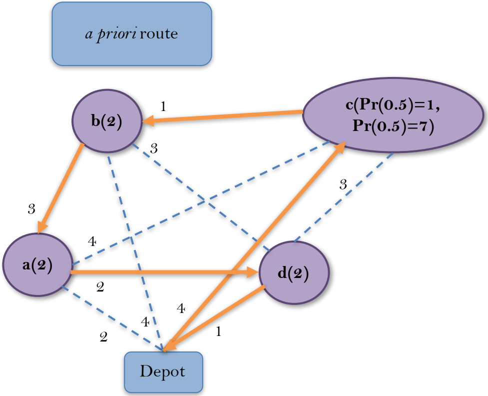


Expected effective length=EEF=14

Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Example: VRP with stochastic demand

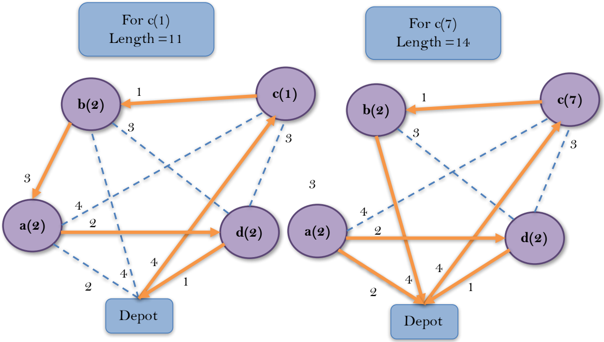
If we don't know the demand of customer c in advance, how about finding an *a priori* route according to some return trip rules



Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Example: VRP with stochastic demand

If we don't know the demand of customer c in advance, how about applying the the *a priori* route



RP = 12.5 = expected effective length, under recourse policy

Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Stochastic programming basic relationships

$$\text{WS} \leq \text{RP} \leq \text{EEV}$$

Relationships

- EVPI = Expected Value of Perfect Information = RP - WS
- VSS = Value of the Stochastic Solution = EEV - RP
- In the example: WS = 12, RP = 12.5, EEV = 14, EVPI = 0.5, VSS = 1.5

Adapted from: F. V. Louveaux, Stochastic Integer Programming, *https://www.stoprog.org/sites/default/files/tutorials/SP13/Louveaux.pdf (last accessed 1/6/2018)*

Main classes of stochastic VRPs

VRP with stochastic demands (VRPSD)

- A probability distribution is specified for the demand of each customer
- Demands are typically assumed to be independent

VRP with stochastic travel times (VRPSTT)

- The travel times required to move between vertices, as well as sometimes service times, are random variables

VRP with stochastic customers (VRPSC)

- Each customer has a given probability of requiring a visit

SVRP general reading

Surveys

- Oyola, J., Arntzen, H. and Woodruff, D.L., 2017. The stochastic vehicle routing problem, a literature review, part II: solution methods. *EURO Journal on Transportation and Logistics*, 6(4), pp.349-388.
- Oyola, J., Arntzen, H. and Woodruff, D.L., 2016. The stochastic vehicle routing problem, a literature review, part I: models *EURO Journal on Transportation and Logistics*, doi: 10.1007/s13676-016-0100-5.

Overview

- Gendreau, M., Jabali, O. and Rei, W., 2014. Chapter 8: Stochastic vehicle routing problems. *In Vehicle Routing: Problems, Methods, and Applications, Second Edition* (pp. 213-239). Society for Industrial and Applied Mathematics.
- Gendreau, M., Jabali, O. and Rei, W., 2016. 50th anniversary invited article –future research directions in stochastic vehicle routing. *Transportation Science*, 50(4), pp.1163-1173.

Basic Concepts in Stochastic Optimization

Dealing with uncertainty in optimization

Uncertainty

- Exists in many problems
- Often should be explicitly dealt with

Main tools for handling uncertainty

- Stochastic programming with recourse (1955)
- Dynamic programming (1958)
- Chance-constrained programming (1959)
- Robust optimization (more recently)

Information and decision-making

Key questions for any stochastic optimization problem

- When do the values taken by the uncertain parameters become known?
- What changes can I make in my plans on the basis of new obtained information ?

Interaction

Informational process

$\Downarrow$

decisional process

Decisional practices

- Determined through time
- Adjusted through time

Stochastic programming with recourse

Proposed by Dantzig and by Beale in 1955

- The key idea is to divide problems in different stages, between which information is revealed
- The simplest case is with only two stages. The second stage deals with **recourse actions**, which are undertaken to adapt plans to the realization of uncertainty

Basic reference

- J.R. Birge and F. Louveaux, Introduction to Stochastic Programming, 2nd edition, Springer, 2011

Dynamic programming

Proposed by Bellman in 1958

- A method developed to tackle effectively sequential decision problems.
- The solution method relies on a time decomposition of the problem according to stages. It exploits the so-called *Principle of Optimality*
- Good for problems with a limited number of possible states and actions

Basic reference

- D.P. Bertsekas, Dynamic Programming and Optimal Control, 3rd edition, Athena Scientific, 2005.

Chance-constrained programming

Proposed by Charnes and Cooper in 1959

- The key idea is to allow some constraints to be satisfied only with some probability

Basic reference

- Charnes, A., & Cooper, W. W. (1959). Chance-constrained programming. *Management science*, 6(1), 73-79.

Robust optimization

Proposed in the last decades

- Uncertainty is represented by the fact that the uncertain parameter vector must belong to a given polyhedral set
- Robust optimization looks in a minimax fashion for the solution that provides the best “worst case"

Basic references

- Ben-Tal, A., El Ghaoui, L. and Nemirovski, A., 2009. *Robust optimization*. Princeton University Press.
- Bertsimas, D., Brown, D.B. and Caramanis, C., 2011. Theory and applications of robust optimization. *SIAM review*, 53(3), pp.464-501.

SVRP Modeling Paradigms

Reoptimization

Overview

- Based on the implicit assumption that information is revealed over time as the vehicles perform their assigned routes
- Routes are created piece by piece based on the currently available information

Drawback

- Not always practical

Solution methods

- Relies on Dynamic programming and related approaches

A priori optimization

Overview

- A solution must be determined beforehand
- This solution is “confronted" with the realization of the stochastic parameters in a second step

Approaches

- Chance-constrained programming
- (Two-stage) stochastic programming with recourse
- Robust optimization
- Other ..

A priori optimization: Chance-constrained programming

Overview

- Probabilistic constraints are sometimes transformed into deterministic ones
- For example: In VRP with stochastic demands, $P(\text{total demand assigned to a route} \leq \text{capacity}) \geq 1 - \alpha$

Drawback

- This model completely ignores what happens when things do not “go well"

A priori optimization: Robust optimization

Overview

- Not used very much in stochastic VRP up to now, but papers have been appearing in the last few years for node and arc routing problems.

Drawback

- Solutions may be overly pessimistic

A priori optimization: Stochastic programming with recourse

Overview

- When is the information related to the uncertain parameters revealed?
- Recourse: what must be done to “adjust" the a priori solution to accommodate the revealed information

Drawback

- Typically produces costly solutions with respect to real-time optimization solutions. However, if recourse actions are correctly defined, it is likely to be closer to actual industrial practices

Solution methods

- Integer L-shaped
- Branch-cut-and-price
- Heuristics (including matheuristics)

What we will cover

	Stochastic programming with recourse	Chance Constraints	Robust Optimization	Reoptimization
VRPSD	✓		✓	✓
VRSTT	✓	✓		
VRPSC	✓			

Mainly exact methods

The Vehicle Routing Problem with Stochastic Demand

The Vehicle Routing Problem with Stochastic Demand

Based on

- Jabali, O., Rei, W., Gendreau, M. and Laporte, G., 2014. Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. *Discrete Applied Mathematics*, 177, pp.121-136.

The vehicle routing problem with stochastic demand

Examples

- A central bank that has to collect money on a daily basis from several locations
- Delivery of money to automatic teller machines
- Beer distribution
- Trash collection

$\Downarrow$

Tactical planning

Design a set of routes

$\Downarrow$

Operational planning

Insufficient residual capacity to meet the observed demand
$\Rightarrow$ *route failure*

The vehicle routing problem with stochastic demand

a priori optimization

First stage	find m routes that visit all clients once
	⇓
Second stage	execute the routes apply complete recourse actions

Setting

demands follow known distributions and are revealed when a vehicle arrives at a client's location

given a set of operational rules

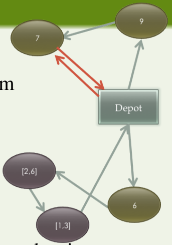
VRPSD

Objective: Min. routing cost + average recourse cost

Constraints: The total expected demand of each route should not exceed the capacity of a vehicle, each client is served by a single vehicle and each route starts and ends at the depot

Literature Review

Classical (complete) recourse



- Gendreau, Laporte and Séguin (1995):
  - First implementation of the L-Shaped algorithm on the problem
- Hjorring and Holt (1999):
  - 1-VRP with stochastic demands
  - Partial routes cuts
- Laporte, Louveaux and van Hamme (2002):
  - Improvement of the general lower bound for the m-VRP with stochastic demands
  - Generalization of the partial route cuts

Advantages

- Consistent routes
- Benchmark for other policies

VRPSD SP model with classical reccourse

Definitions

$G(V, E)$ is a complete undirected graph, $V = \{v_1, \ldots, v_n\}$
$E = \{(v_i, v_j): v_i, v_j \in V, i < j\}$
v_1 is the depot with m identical vehicles
D is the capacity of each vehicle
ξ_i is the demand of client i, $\xi_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and $\xi_i \in (0, D), i = 2, \ldots, n$
c_{ij} is the travel cost associated with edge $(v_i, v_j) \in E, i < j$.

Decision variables

$$x_{ij} = \{0, 1\} \quad \text{for } i, j > 1$$
$$x_{1j} = \{0, 1, 2\} \quad \text{for } j > 1$$

The considered case

Client demands are independent
Recourse rules $\Rightarrow$ return to depot and replenish when failure occurs
$\mathcal{Q}(x)$ is the expected recourse cost of solution x

Model

$$\text{(VRPSD)} \qquad \text{Minimize} \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x)$$

subject to

$$\begin{aligned}
\sum_{j=2}^{n} x_{1j} &= 2m, && \\
\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} &= 2 && (k = 2, \dots, n), \\
\sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mu_i}{D} \right\rceil & && (S \subset V \setminus \{v_1\}, 2 \leq |S| \leq n-2), \\
0 \leq x_{ij} &\leq 1 && (1 \leq i < j < n), \\
0 \leq x_{0j} &\leq 2 && (j = 2, \dots, n), \\
x &= (x_{ij}) && \text{integer.}
\end{aligned}$$

Recourse cost computation

Given solution x the recourse cost is decomposable by routes

$$\mathcal{Q}(x) \quad = \quad \sum_{k=1}^{m} \min\{\mathcal{Q}^{k,1}, \mathcal{Q}^{k,2}\}$$

$\mathcal{Q}^{k,\delta}$ is the expected cost of the recourse of route k for orientation δ
The computation of $\mathcal{Q}^{k,1}$ for route k defined by $(v_{i_1} = v_1, v_{i_2}, \dots, v_{i_{t+1}} = v_1)$ is given by

$$\mathcal{Q}^{k,1} \quad = \quad 2\sum_{j=2}^{t}\sum_{l=1}^{j-1} P\Big(\sum_{s=1}^{j-1}\xi_{i_s} \leq lD < \sum_{s=1}^{j}\xi_{i_s}\Big) c_{1 i_j}.$$

the probability that the l^{th} failure occurs at client v_{i_s}

Note

In most applications, the demand probability distributions adhere to the cumulative property, i.e.,

- the sum of two or more independent random variables with a distribution Ψ yields a random variable with distribution Ψ

Integer L-shaped algorithm

Laporte and Louveaux (1993)

- Solves integer stochastic programmes
- Extension of the L-shaped method proposed by Slyke and Wets for continuous stochastic programs
- Application of Benders decomposition to stochastic programming

Assumptions

- Assumption 1: $\mathcal{Q}(x)$ is computable
- Assumption 2: There exists a finite value L which is a general lower bound on the recourse function

Outline

- Follows a branch-and-cut solution strategy that solves the Current Problem (CP) at each node
- Approximates the recourse function $\mathcal{Q}(x)$ by a valid lower bound defined by θ

Integer L-shaped algorithm for the VRPSD

VRPSD

$$\text{Minimize} \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x)$$

subject to

$$\sum_{j=2}^{n} x_{1j} = 2m,$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad (k = 2, \dots, n),$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mu_i}{D} \right\rceil$$

$$(S \subset V \setminus \{v_1\},\ 2 \leq |S| \leq n-2),$$

$$0 \leq x_{ij} \leq 1 \qquad (1 \leq i < j < n),$$

$$0 \leq x_{0j} \leq 2 \qquad (j = 2, \dots, n),$$

$$x = (x_{ij}) \qquad \text{integer.}$$

Initial CP

$$\text{Minimize} \sum_{i<j} c_{ij} x_{ij} + \theta$$

subject to

$$\sum_{j=2}^{n} x_{1j} = 2m,$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad (k = 2, \dots, n),$$

$$0 \leq x_{ij} \leq 1 \qquad (1 \leq i < j < n),$$

$$0 \leq x_{0j} \leq 2 \qquad (j = 2, \dots, n).$$

Integer L-shaped algorithm for the VRPSD

Step 0 Set the iteration counter $\nu := 0$ and introduce the bounding constraint $\theta \geq L$ into the CP. Set the value of the best known solution $\bar{z} := +\infty$. The only pendent node is the initial CP.

Step 1 Select a pendent node from the list. If none exists stop.

Step 2 Set $\nu := \nu + 1$ and solve CP. Let (x^ν, θ^ν) be the obtained optimal solution.

Step 3 Check for violated subtour elimination and capacity constraints. At this stage, valid inequalities or lower bounding functionals may also be generated. If a violated constraint is found add it to the CP and return to Step 2. Otherwise, if $cx^\nu + \theta^\nu \geq \bar{z}$, fathom the current node and return to Step 1.

Step 4 If the solution is not integer, then branch on a fractional variable. Append corresponding subproblems to the list of pendent nodes and return to Step 1.

Step 5 Compute $\mathcal{Q}(x^\nu)$ and set $z^\nu := cx^\nu + Q(x^\nu)$. If $z^\nu < \bar{z}$ then $\bar{z} = z^\nu$.

Step 6 If $\Theta^\nu \geq Q(x^\nu)$, then fathom the current node and return to Step 1. Otherwise add an optimality cut defined as

$$\sum_{\substack{1<i<j \\ x_{ij}^\nu = 1}} x_{ij} \leq \sum_{1<i<j} x_{ij}^\nu - 1$$

and go to Step 2.

Integer L-shaped algorithm for the VRPSD

Main challenges

- Approximation of $\mathcal{Q}(x)$
  - Hard to obtain a good general lower bound $L \Rightarrow$ (Laporte et al., 2002)
  - Information provided by optimality cuts is very local
  - High number of LBF cuts need to be added
- Quality of the upper bound
  - No guarantees that a good set of routes will be obtained early in the solution process

Improvements

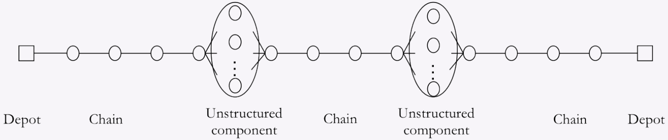
- Generalized definition of partial routes
- New families of LBFs
- Separation procedures for these LBFs

Computation of L (based on Laporte et al. (2002))

- Consider the m closest clients to the depot
- Estimate a lower bound on the cost entailed by a single failure in each of the m routes
- Distribute the total mean and variance of the m closest customers, such that the expected cost is minimized

General partial route

Example



Description of a general partial route

- A solution can be decomposed into several components anchored articulation verteces
- Components are either *unstructured components* whose vertex sets are called *Unstructured Vertex Sets* (UVSs), or *chains* whose vertex sets are called *Chain Vertex Sets* (CVSs)

General partial route

Definitions

- Let b denote the number of chains, and let $b-1$ denote the number of unstructured components in a partial route h. This partial route then consists of $2b-1$ components
- Let S_h^r denote the r^{th} ordered CVS, defined as $S_h^r = \{v_{1^{rh}}, \ldots, v_{l_{rh}^{rh}}\}$, where $v_{k^{rh}}$ is the k^{th} vertex in S_h^r and l_{rh} is the number of vertices in S_h^r. Let U_h^r denote the r^{th} UVS in h.
- We write $(v_i, v_j) \in S_h^r$ if v_i and v_j are consecutive in S_h^r, and we write l^{rh} instead of l_{rh}^{rh}.

$$\sum_{(v_i, v_j) \in S_h^r} x_{ij} = |S_h^r| - 1.$$

$$\sum_{v_i, v_j \in U_h^r} x_{ij} = |U_h^r| - 1.$$

For each $1 \leq r \leq b-1$,

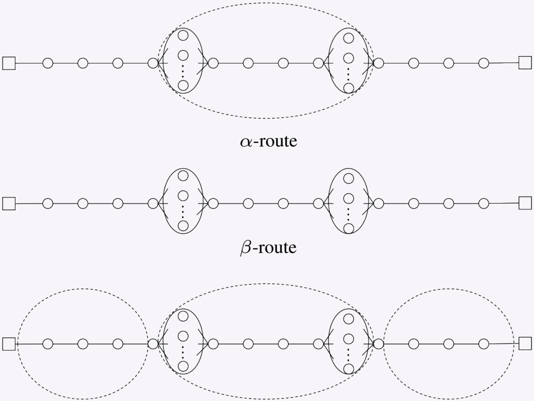
$$\sum_{v_j \in U_h^r} x_{l^{rh}, j} = 1.$$

Similarly, for each $2 \leq r \leq b$,

$$\sum_{v_j \in U_h^{r-1}} x_{1^{rh}, j} = 1.$$

General partial route

Three LBFs



α-route

β-route

γ-route

Lower bounding functionals

Lower bound on the cost of recourse

- P_h is a lower bound on the cost of recourse associated with general partial route h
- For each UVS, $r = 1, \ldots, b-1$, we create an artificial client v^r with demand

$$\xi_r = \sum_{v_i \in U_h^r} \xi_i \quad \text{and} \quad c_{1r} = \min_{v_j \in U_h^r} \{c_{1j}\}.$$

The partial route is then constructed as
$(v_0 = v_{1_{1h}}, \ldots, v_{l_{rh}}, v^1, v_{1_{2h}}, \ldots, v_{l_{2h}}, v^2, \ldots, v^{b-1}, v_{1_{bh}}, \ldots, v_{l_{bh}})$, and

$$P_h \quad = \quad \min\{\mathcal{Q}^{k,1}, \mathcal{Q}^{k,2}\}.$$

Lower bounding functionals

Lower bound on the cost of recourse

- To compute a lower bound P on the total cost of recourse, first define

$$R_h = (\cup_{r=1}^{b} S_h^r) \cup (\cup_{r=1}^{b-1} U_h^r).$$

- Assuming $f \leq m$ partial routes, let P_{r+1} be a lower bound on the cost of recourse for $m - f$ routes involving the client set $V \setminus \cup_{h=1}^{f} R_h$, with $P_{m+1} = 0$. Then the lower bound is

$$P = \sum_{h=1}^{f+1} P_h.$$

Lower bounding functionals

Valid inequality

$$\begin{aligned} W_h(x) &= \sum_{r=1}^{b} \sum_{\substack{(v_i,v_j)\in S_h^r \\ v_i \neq v_1}} 3x_{ij} + \sum_{(v_1,v_j)\in S_h^1} x_{1j} + \sum_{(v_1,v_j)\in S_h^b} x_{1j} + \sum_{r=1}^{b-1} \sum_{v_i,v_j \in U_h^r} 3x_{ij} \\ &+ \sum_{r=1}^{b-1} \sum_{\substack{v_j \in U_h^r \\ v_{l^r_h} \neq v_1}} 3x_{l^r_h,j} + \sum_{r=2}^{b} \sum_{\substack{v_j \in U_h^{r-1} \\ v_{1^r_h} \neq v_1}} 3x_{1^r_h,j} + \sum_{\substack{v_j \in U_h^1 \\ v_{l^1_h} = v_1}} x_{l^1_h,j} \\ &+ \sum_{\substack{v_j \in U_h^{b-1} \\ v_{1^b_h} = v_1 \\ v_{l^{b-1}_h} \neq v_1}} x_{1^b_h,j} - (3|R_h| - 5). \end{aligned}$$

Proposition

Let $\bar{x}$ be a solution satisfying the constraints of CP, then constraint

$$\Theta \geq L + (P - L)\left(\sum_{h=1}^{r} W_h(\bar{x}) - r + 1\right)$$

is a valid inequality for the (VRPSD).

Separation procedure for general partial routes

Main features

- Detect connections to the depot
- Generate a UVS by sequentially adding all adjacent fractional arcs
- Sequentially detect integer arcs
  - If an articulation vertex is encountered then decompose the UVS
  - If two segments are linked to a UVS then the UVS is expanded
- Generate partial route

Test problems

Laporte, Louveaux and Van hamme (2002)

- n vertices in the $[0, 100]^2$ square.
- Five rectangular obstacles in $[20, 80]^2$ were generated, each having a base of 4 and height of 25, covering 5% of the entire area.
- Let $\overline{f} = \sum_{i=2}^{n} \mathbf{E}(\xi_i)/mD$ define the filling coefficient

m	n	$\bar{f}$
2	60, 70, 80, 90	90%, 95%
3	40, 45	80%
3	50, 60, 70, 80	80%, 85%, 90%, 95%
4	40, 50	75%, 80%, 85%, 90%, 95%
4	45	80%, 85%, 90%, 95%
4	60	80%, 82%, 85%

- A total of 481 instances
- The separation procedure of the subtour elimination and capacity constraints was performed using the CVRPSEP package of Lysgaard et al. 2004
- Computation time of 10,000 seconds

Number of solved instances

m	n	LBF_α	LBF_β	LBF_γ	Number of instances
2	60	7	9	14	20
2	70	9	10	11	20
2	80	5	7	10	20
2	90	2	3	7	20
3	40	9	9	9	10
3	45	9	9	9	10
3	50	22	23	23	55
3	60	21	21	22	56
3	70	11	12	12	47
3	80	9	10	15	48
4	40	8	9	10	56
4	45	4	4	5	37
4	50	8	9	10	53
4	60	1	1	1	29
		125	136	158	481

Cumulative percentage of instances solved for several ranges of gaps

n	LBF_α	LBF_β	LBF_γ
$= 0\%$	25.99%	28.27%	32.85%
$\leq 1\%$	50.94%	53.01%	56.96%
$\leq 3\%$	80.87%	81.70%	79.63%
$\leq 5\%$	94.39%	94.80%	86.90%
$\leq 7\%$	98.13%	98.54%	88.98%

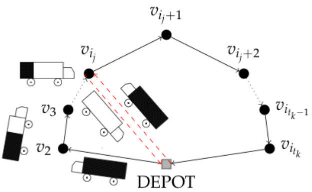
Other algorithms for the VRPSD with classical recourse

- Christiansen and Lysgaard (2007) developed branch-and-price algorithm
- Gauvin et al. (2014) developed a branch-cut-and-price algorithm

Other recourse policies for the VRPSD

Other recourse

- The classical recourse strategy for the VRPSD is not very smart
- Pairing strategy by Ak and Erera (2007) have suggested “pairing" routes to enhance operations and reduce the number of back and forth trips back to the depot
- Restocking rules by Yang, Mathur, and Ballou (2000): proposed a policy of *optimal restocking* to given routes governed by customer *thresholds*



Odysseus 2018 alert!

- MO2a: Alexandre Florio, Richard Hartl and Stefan Minner. *Branch- Cut-and-Price for the Vehicle Routing Problem with Stochastic Demands under Optimal Restocking*.
- TH3a: Majid Salavati-Khoshgalb, Ola Jabali, Walter Rei and Michel Gendreau. *An Exact Algorithm to Solve the Vehicle Routing Problem with Stochastic Demands under an Optimal Restocking Policy*.

Other recourse policies for the VRPSD

Rule-based recourse

- Each rule-based policy can be derived from a given operational convention
- It generates a set of thresholds associated with the customer visits that are scheduled along a given route
- These thresholds determine when the vehicle performing the route should preventively return to the depot
- Gives more control to the user and streamlines organizational policies

Other algorithms for the VRPSD with classical recourse

- Salavati-Khoshghalb M., Gendreau Michel, Jabali Ola, Rei Walter, “A rule-based recourse for the vehicle routing problem with stochastic demands", CIRRELT, CIRRELT-2017-36, 2017

Reoptimization optimization for the VRPSD: Relevant literature

Single vehicle case

- Considering a single vehicle, Secomandi (2000) proposes an approximate policy iteration procedure that estimates the cost-to-go through a parametric function.
- Considering a single vehicle, Secomandi (2000) develops a one-step rollout policy
- Novoa and Storer (2008) show that using a higher quality initial fixed route leads to improved policies.
- Bertazzia and Secomandi (2018) develop and approach to approximate the expected cost of a route when executing any rollout algorithm for VRPSD with restocking

Multi-vehicle case

- Hvattum et al. (2006, 2007) m-VRP with stochastic demand where customer orders are placed over a given planing horizon
- Goodson et al. (2013) propose rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits
- Goodson et al. (2015) develop restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits

Re-optimization for the VRPSD

- Goodson, J.C., Ohlmann, J.W. and Thomas, B.W., 2013. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research,* 61(1), pp.138-154.

Considered model

- The objective is maximizing the expected served demand
- Customer demand is served as much as possible when a vehicle arrives
- Unserved demand after an initial vehicle visit may be satisfied on subsequent visits by the same vehicle or by another.

Solution methods

- A family of rollout policies based on fixed routes to obtain dynamic solutions
- Rollout polices based on the notions of the pre- and post-decision state
- A dynamic decomposition scheme for large scale problems

Robust optimization for the VRPSD

- Sungur, I., Ordónez, F. and Dessouky, M., 2008. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5), pp.509-523.

Considered model

- Propose a robust optimization approach for the capacitated vehicle routing problem with demand uncertainty
- Assuming that uncertain parameters belong to a given bounded uncertainty set consisting of deviations around an expected demand value
- Miller-Tucker-Zemlin (MTZ) formulation
- The objective is finding an optimal a priori route that is feasible for every demand realization
- Derived a robust counterpart for the VRPSD which requires the solution of a single CVRP with modified data

Robust optimization for the VRPSD

- Gounaris, C.E., Wiesemann, W. and Floudas, C.A., 2013. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3), pp.677-693.

Considered model

- Propose a robust optimization approach for the capacitated vehicle routing problem with demand uncertainty
- Consider the generic case where the customer demands are supported on a (typically nonrectangular) polyhedron
- Robust optimization counterparts of several deterministic CVRP formulations are derived
- Robust rounded capacity inequalities are developed and separated efficiently

The Vehicle Routing Problem with Stochastic Travel Times

VRP with stochastic service or travel times

- The travel times required to move between vertices and/or service times are random variables
- Possibly the most interesting of all SVRP variants
- Reason: it is much more difficult than others, because delays “propagate" along a route
- Usual recourse entails paying penalties for soft time windows or overtime
- All solution approaches seem relevant, but present significant implementation challenges

VRPSTT: Relevant literature

- Laporte et al. (1992) introduce travel and service time uncertainty in a non-capacitated version of the VRP with a max route duration
- Kenyon and Morton (2003) study the VRP with stochastic travel and service times and no capacity constraints
- Lei et al. (2012) consider a capacitated vehicle routing problem with soft time windows, soft route-duration limits, and stochastic service and travel times. The recourse action is the payment of a penalty when the route-duration limit is violated
- Adulyasak and Jaillet (2014) focus on the VRP with capacity constraints, customer deadlines and stochastic travel times. Minimizing the sum of the probabilities of deadline violations

The vehicle routing with soft time windows and stochastic travel times

Based on

- D. Taş, M. Gendreau, N. Dellaert, T. Van Woensel, A.G. De Kok (2014). "Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach", *European Journal of Operational Research*, 236(3):789-99.

Considered model

- Transportation costs: total distance + number of vehicles + total expected overtime
- Service costs: early and late arrivals penalties

Problem Description

- $G = (N, A)$ where $N = \{0, 1, ..., n\}$ and $A = \{(i, j) \mid i, j \in N, i \neq j\}$
- Each customer has a known demand ($q_i \geq 0$), a fixed service duration ($s_i \geq 0$) a soft time window ($[l_i, u_i]$, $l_i \geq 0$, $u_i \geq 0$)
- No waiting!
- Along each arc (i, j)
  - a distance d_{ij}
  - a travel time T_{ij} (with a known probability distribution)
- Capacity of each vehicle $v \in V$, Q

Notation

x_{ijv}	:	Equal to 1 if vehicle v covers arc (i, j), 0 otherwise
$\mathbf{x}$	:	Vector of vehicle assignments and customer sequences in these vehicle routes, where $\mathbf{x} = \{x_{ijv} \mid i, j \in N, v \in V\}$
$D_{jv}(\mathbf{x})$	:	$Expected$ delay at node j when it is served by vehicle v
$E_{jv}(\mathbf{x})$	:	$Expected$ earliness at node j when it is served by vehicle v
$O_v(\mathbf{x})$	:	$Expected$ overtime of the driver working on route of vehicle v
c_d	:	Penalty cost paid for one unit of delay
c_e	:	Penalty cost paid for one unit of earliness
c_t	:	Cost paid for one unit of distance
c_o	:	Cost paid for one unit of overtime
c_f	:	Fixed cost paid for each vehicle used for servicing

Objective function

$$\min \quad \sum_{v \in V} \Bigg[\rho \frac{1}{C_1} \Bigg(c_d \sum_{j \in N} D_{jv}(\mathbf{x}) + c_e \sum_{j \in N} E_{jv}(\mathbf{x}) \Bigg) + (1-\rho) \frac{1}{C_2} \Bigg(c_t \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijv} + c_f \sum_{j \in N \setminus \{0\}} x_{0jv} + c_o O_v(\mathbf{x}) \Bigg) \Bigg] \quad (1)$$

Model Formulation

$$\min \quad \sum_{v \in V} \left[\rho \frac{1}{C_1} \left(c_d \sum_{j \in N} D_{jv}(\mathbf{x}) + c_e \sum_{j \in N} E_{jv}(\mathbf{x}) \right) \right.$$
$$\left. +(1-\rho) \frac{1}{C_2} \left(c_t \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijv} + c_f \sum_{j \in N \setminus \{0\}} x_{0jv} + c_o O_v(\mathbf{x}) \right) \right] \tag{2}$$

subject to

$$\sum_{j \in N} \sum_{v \in V} x_{ijv} = 1, \qquad i \in N \setminus \{0\}, \tag{3}$$

$$\sum_{i \in N} x_{ikv} - \sum_{j \in N} x_{kjv} = 0, \qquad k \in N \setminus \{0\}, v \in V, \tag{4}$$

$$\sum_{j \in N} x_{0jv} = 1, \qquad v \in V, \tag{5}$$

$$\sum_{i \in N} x_{i0v} = 1, \qquad v \in V, \tag{6}$$

$$\sum_{i \in N \setminus \{0\}} q_i \sum_{j \in N} x_{ijv} \leq Q, \qquad v \in V, \tag{7}$$

$$\sum_{i \in B} \sum_{j \in B} x_{ijv} \leq |B| - 1, \qquad B \subseteq N \setminus \{0\}, v \in V, \tag{8}$$

$$x_{ijv} \in \{0, 1\}, \qquad i \in N, j \in N, v \in V. \tag{9}$$

Properties of the Arrival Times

- Let the arrival time of vehicle v at node j be:

$$Y_{jv} = \sum_{(l,k) \in A_{jv}} T_{lk} \tag{10}$$

where A_{jv} represents the set of arcs which are covered by vehicle v until node j

- Assumption mean and variance of the arrival time at node j which is visited by vehicle v immediately after node i:

$$E[Y_{jv}] = E[Y_{iv}] + E[T_{ij}] \tag{11}$$

$$\text{Var}(Y_{jv}) = \text{Var}(Y_{iv}) + \text{Var}(T_{ij}) \tag{12}$$

- Random traversal time spent for one unit of distance is Gamma distributed (with shape parameter α and scale parameter λ)

- Uncertainty per km

- Calculate expected delay, earliness and overtime exactly

Calculations with Gamma Distribution

- T: Gamma distributed travel time spent for one unit of distance

$$f(t) = \frac{(e^{-t/\lambda})(t^{\alpha-1})}{\Gamma(\alpha)\lambda^{\alpha}} \tag{13}$$

$$F(\delta) = \text{Prob}\{t \leq \delta\} = \Gamma_{\alpha,\lambda}(\delta) = \int_0^{\delta} \frac{(e^{-z/\lambda})(z^{\alpha-1})}{\Gamma(\alpha)\lambda^{\alpha}} dz \tag{14}$$

- Mean and variance of T_{ij}:

$$E[T_{ij}] = \alpha\lambda d_{ij} \tag{15}$$

$$\text{Var}(T_{ij}) = \alpha\lambda^2 d_{ij} \tag{16}$$

- Shape and scale parameters of Y_{jv}:

$$\alpha_{jv} = \alpha \sum_{(l,k)\in A_{jv}} d_{lk} \tag{17}$$

$$\lambda_{jv} = \lambda \tag{18}$$

Calculations with Gamma Distribution

- Expected delay:

$$D_{jv}(\mathbf{x}) = \int_{u'_j}^{\infty} (z - u'_j) \frac{(e^{-z/\lambda_{jv}})(z^{\alpha_{jv}-1})}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}} dz,$$

$$= \int_{u'_j}^{\infty} \frac{(e^{-z/\lambda_{jv}})(z^{\alpha_{jv}})}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}} dz - u'_j \int_{u'_j}^{\infty} \frac{(e^{-z/\lambda_{jv}})(z^{\alpha_{jv}-1})}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}} dz,$$

$$= \alpha_{jv}\lambda_{jv}(1 - \Gamma_{\alpha_{jv}+1,\lambda_{jv}}(u'_j)) - u'_j(1 - \Gamma_{\alpha_{jv},\lambda_{jv}}(u'_j)), \qquad (19)$$

- where u'_j is the upper bound of the shifted time window at node j ($u'_j = u_j - s_{jv}$) and s_{jv} is the total time spent by the vehicle v for servicing the nodes before visiting node j.

Calculations with Gamma Distribution

- Expected earliness:

$$E_{jv}(\mathbf{x}) = \int_0^{l'_j} (l'_j - z)\frac{(e^{-z/\lambda_{jv}})(z)^{\alpha_{jv}-1}}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}}dz,$$

$$= l'_j \int_0^{l'_j} \frac{(e^{-z/\lambda_{jv}})(z)^{\alpha_{jv}-1}}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}}dz - \int_0^{l'_j} \frac{(e^{-z/\lambda_{jv}})(z)^{\alpha_{jv}}}{\Gamma(\alpha_{jv})(\lambda_{jv})^{\alpha_{jv}}}dz,$$

$$= l'_j \Gamma_{\alpha_{jv},\lambda_{jv}}(l'_j) - \alpha_{jv}\lambda_{jv}\Gamma_{\alpha_{jv}+1,\lambda_{jv}}(l'_j), \tag{20}$$

- where l'_j is the lower bound of the shifted time window at node j ($l'_j = l_j - s_{jv}$).

Calculations with Gamma Distribution

- Expected overtime:

$$
\begin{aligned}
O_v(\mathbf{x}) &= \int_{w'}^{\infty} (z - w') \frac{(e^{-z/\lambda_{0v}})(z)^{\alpha_{0v}-1}}{\Gamma(\alpha_{0v})(\lambda_{0v})^{\alpha_{0v}}} dz, \\
&= \int_{w'}^{\infty} \frac{(e^{-z/\lambda_{0v}})(z)^{\alpha_{0v}}}{\Gamma(\alpha_{0v})(\lambda_{0v})^{\alpha_{0v}}} dz - w' \int_{w'}^{\infty} \frac{(e^{-z/\lambda_{0v}})(z)^{\alpha_{0v}-1}}{\Gamma(\alpha_{0v})(\lambda_{0v})^{\alpha_{0v}}} dz, \\
&= \alpha_{0v}\lambda_{0v}(1 - \Gamma_{\alpha_{0v}+1,\lambda_{0v}}(w')) - w'(1 - \Gamma_{\alpha_{0v},\lambda_{0v}}(w')), \qquad (21)
\end{aligned}
$$

- where w' is the agreed labor shift time (w) minus the total service time spent by vehicle v for servicing at all nodes on its route (s_{0v}).

Column Generation

- Master problem: A set partitioning problem

$$\min \quad \sum_{p \in P} K_p y_p \tag{22}$$

$$\text{subject to} \sum_{p \in P} a_{ip} y_p = 1, \qquad i \in N \setminus \{0\}, \tag{23}$$

$$y_p \in \{0,1\}, \qquad p \in P. \tag{24}$$

- P: set of all feasible vehicle routes that start from the depot and end at the depot,
- K_p: total weighted cost of route p,
- a_{ip}: 1 if customer i is served by route p and 0, otherwise.

Column Generation

- Pricing subproblem: for each vehicle v, an Elementary Shortest Path Problem with Resource Constraints (ESPPRC)

$$\min \quad \overline{K}_p \tag{25}$$

$$\text{subject to} \quad (4) - (9). \tag{26}$$

- p: route of vehicle v,
- $\overline{K}_p$: reduced cost of route p.

$$\begin{aligned}\overline{K}_p =& K_p - \sum_{i \in N \setminus \{0\}} a_{ip} u_i \\ =& \rho \frac{1}{C_1} \left(c_d \sum_{j \in N} D_{jv}(\mathbf{x}) + c_e \sum_{j \in N} E_{jv}(\mathbf{x}) \right) \\ &+ (1-\rho) \frac{1}{C_2} \left(c_t \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijv} + c_f \sum_{j \in N \setminus \{0\}} x_{0jv} + c_o O_v(\mathbf{x}) \right) - \sum_{i \in N \setminus \{0\}} a_{ip} u_i,\end{aligned} \tag{27}$$

- $u_i, i \in N \setminus \{0\}$: dual price associated with the constraints (23).

ESPPRC

- Solve with the algorithm of Feillet et al. (2004)
  - Using node resources $\rightarrow$ unreachable nodes
- Each label on a node represents
  - A path from the depot to that node
  - Cost of the path and the consumption of the resources along the path
- Apply the state space augmentation technique of Boland et al. (2006), and Righini and Salani (2008)
  - Multiple visits are forbidden for the nodes in a given set S

ESPPRC

- A state $(W_p^1, ..., W_p^R, a_p^S, \mathbf{V}_p^S)$ is associated with each path p from depot to node i
  - $(W_p^1, ..., W_p^R)$: consumption of each of the R resources along the path p
  - a_p^S: number of nodes in S which are unreachable by path p
  - $\mathbf{V}_p^S$: vector of unreachable nodes in S
    - $V_p^b = 1$ if node $b \in S$ is unreachable by path p, 0 otherwise
- Each path p is represented by a label, $(L_p, \overline{K}_p)$
  - $L_p = (W_p^1, ..., W_p^R, a_p^S, \mathbf{V}_p^S)$
  - $\overline{K}_p$: reduced cost of path p
  - $\overline{K}_p$ is calculated with respect to the optimal starting time of path p from the depot

ESPPRC

- Let p and p^* be two distinct paths from the depot to node i
- Starting from the depot at the optimal departure time
- Arriving at node i at different times (different expected arrival times)
- For the dominance relation
  - Adjust the starting time of one path (path p)
  - Arrive at node i at the same time as the other path (path p^*)
- Dominance relation:

Definition

If p and p^* are two distinct paths from the depot to node i with labels $(L_p, \overline{K}_p)$ and $(L_{p^*}, \overline{K}_{p^*})$, respectively, then path p dominates path p^* if and only if $W_p^r \leq W_{p^*}^r$ for $r = 1, \ldots, R$, $a_p^S \leq a_{p^*}^S$, $V_p^b \leq V_{p^*}^b$ for all $b \in S$, $\overline{K}_p \leq \overline{K}_{p^*}$, $\overline{K}_{p_{p^*}} \leq \overline{K}_{p^*}$ and $(L_p, \overline{K}_p) \neq (L_{p^*}, \overline{K}_{p^*})$.

- Non-dominated label $\rightarrow$ efficient path

ESPPRC

- Apply accelerating methods in ESPPRC
- Intermediate Column Pool (ICP)
  - Keep some of efficient and some of dominated paths in ICP
  - First search ICP
  - If search fails, solve the ESPPRC
  - Check the size at each iteration $\rightarrow$ cleaning
- Stopping the ESPPRC prematurely
  - Number of efficient elementary paths with negative reduced costs

Algorithms

Algorithm with state space augmentation technique to solve the ESPPRC

$S \leftarrow \emptyset$
$S' \leftarrow \emptyset$
repeat
  $S \leftarrow S \cup S'$
  $S' \leftarrow \emptyset$
  $\Pi_0 = \{(0,0,\mathbf{0},0)\}$
  forall the $i \in N \setminus \{0\}$ **do**
    $\Pi_i \leftarrow \emptyset$
  end
  $I = \{0\}$
  repeat
    Choose $i \in I$
    forall the $(i,j) \in A$ **do**
      $H_{ij} \leftarrow \emptyset$
      forall the $\pi_p = (W_p^1, a_p^S, \mathbf{V}_p^S, \overline{K}_p) \in \Pi_i$ **do**
        if $(j \notin S)$ or $(j \in S$ and $V_p^j = 0)$ **then**
          if Extend$(i, \pi_p, j) \neq$ FALSE **then**
            $H_{ij} \leftarrow H_{ij} \cup \{$Extend$(i, \pi_p, j)\}$
          end
        end
      end
      if $j \in N \setminus \{0\}$ **then**
        $\Pi_j \leftarrow EFF(\Pi_j \cup H_{ij})$
        if Π_j has changed and $j \notin I$ **then**
          $I \leftarrow I \cup \{j\}$
        end
      end
    end
    $I \leftarrow I \setminus \{i\}$
  until $I = \emptyset$;
  $\Pi_0 \leftarrow EFF(\Pi_0)$
  if There is at least one elementary path on the depot with negative reduced cost **then**
    Send such paths to the RLPMP
  else
    if The minimum reduced cost is negative **then**
      Select the customer with the highest multiplicity in the solution with the minimum reduced cost
      $S' \leftarrow \{$selected customer$\}$
    end
  end
until $S' = \emptyset$;

Extend (i, π_p, j)

if $W_p^1 + w_{ij}^1 > Q$ **then**
  return FALSE
else
  compute $W_{p'}^1$ and $\overline{K}_{p'}$
  $a_{p'}^S \leftarrow a_p^S$
  $\mathbf{V}_{p'}^S \leftarrow \mathbf{V}_p^S$
  if $j \in S$ **then**
    $a_{p'}^S \leftarrow a_{p'}^S + 1$
    $V_{p'}^j \leftarrow 1$
  end
  foreach $b \in S$ and $(j,b) \in A$ such that $W_{p'}^1 + w_{jb}^1 > Q$ **do**
    $a_{p'}^S \leftarrow a_{p'}^S + 1$
    $V_{p'}^b \leftarrow 1$
  end
  return $\pi_{p'} = (L_{p'}, \overline{K}_{p'})$
end

Service Cost Component

- Total service cost → optimal starting time of that path from the depot
- Continuous function of its corresponding vehicle's departure time from the depot
- Further prove that it is convex:

Proposition

For all routes, the total service cost is a convex function of the corresponding vehicle's departure time from the depot.

- Calculate the optimal departure times from the depot → Golden Section Search method

Branching

- Applied strategy: branching on arcs
- Homogeneous fleet of vehicles $\rightarrow$ sum of flows, $f_{ij} = \sum_{v \in V} x_{ijv}$
- When $\sum_{v \in V} x_{ijv} = 1$, force arc (i, j) into the solution
- When $\sum_{v \in V} x_{ijv} = 0$, exclude arc (i, j) from the solution
- Several fractional flow variables $\rightarrow$ choose the arc (i, j) on which f_{ij} is the closest one to the midpoint (0.5)

Branching

- Column pool $\rightarrow$ all distinctive columns obtained at the root node
- Two separate methods, Breadth-First (BF) and Depth-First (DF)
- BF method:
  - Starting value for the UB $\rightarrow$ solve an integer programming at the root node
  - Number of customers is medium or large $\rightarrow$ huge amount of time or computer memory
- DF method:
  - Starting value for the UB $\rightarrow$ IFS generated at the root node
  - Select child node with the minimum LB value to proceed into the next level

Computational Results - Data sets

- Performed on Solomon's problem instances
- Three types of geographic distribution:
  - Clustered (C)
  - Random (R)
  - Randomly Clustered (RC)
- Two types of instances with respect to the time windows:
  - Tight time windows (C1, R1 and RC1)
  - Large time windows (C2, R2 and RC2)
- One depot and 100 customers

Computational Results - Preliminary Tests & Parameters

- According to preliminary tests conducted for the column generation procedure:
  - Capacity of all vehicles $\rightarrow$ 50
  - Number of columns in ICP $\geq$ 150 $\rightarrow$ clean the pool
  - Columns that have been kept $\geq$ 15 iterations $\rightarrow$ remove from the ICP
  - Number of efficient elementary paths with negative reduced costs on depot $\geq$ 10 $\rightarrow$ stop the ESPPRC
- $\rho = 0.5$, $C_1 = 1.00$, $C_2 = 1.00$, and CV = 1.00 ($\alpha = 1$, γ=1)
- Two stopping criteria for our solution procedure:
  - Gap between the best LB and the best UB $\leq$ 0.005 (0.5%) $\rightarrow$ terminate
  - Limit for the total CPU time $\rightarrow$ 3 hours
- Implementation: Visual C++, IBM ILOG CPLEX 12.2 on an Intel Core Duo with 2.93 GHz and 4 GB of RAM

Computational Results

Results of problem instances in RC set with 20 customers with DF method

Instance	RootLB	RootUB	BestLB	BestUB	CPU	Gap%	Tree
RC101	2184.44	2422.11	2186.59	2196.93	725.8	0.47	12
RC102	2178.20	2412.26	2178.61	2189.38	289.6	0.49	10
RC103	2174.45	2223.97	2175.57	2186.42	941.1	0.50	15
RC104	2173.48	2220.13	2175.60	2186.26	2004.7	0.49	19
RC105	2179.18	2231.68	2180.55	2191.32	1035.9	0.49	15
RC106	2175.17	2225.12	2178.97	2187.59	1971.5	0.40	16
RC107	2172.33	2220.75	2174.47	2184.52	1508.4	0.46	16
RC108	2172.11	2215.73	2174.30	2184.04	818.5	0.45	18
RC201	2216.92	2315.64	2217.82	2219.63	43.1	0.08	4
RC202	2188.57	2274.88	2188.57	2199.02	42.8	0.48	5
RC203	2176.18	2264.96	2177.85	2187.52	493.9	0.44	15
RC204	2175.16	2242.22	2176.82	2186.32	426.2	0.44	15
RC205	2193.60	2278.38	2193.86	2202.13	44.3	0.38	6
RC206	2191.51	2271.94	2192.26	2202.62	37.9	0.47	5
RC207	2182.12	2236.98	2183.35	2193.96	753.2	0.49	17
RC208	2171.98	2215.05	2174.09	2183.96	1564.3	0.45	17

Computational Results

Results of problem instances in RC set with 25 customers with DF method

Instance	RootLB	RootUB	BestLB	BestUB	CPU	Gap%	Tree
RC101	2663.82	2895.13	2664.57	2687.43	10800.0	0.86	34
RC102	2656.18	2885.91	2656.18	2678.37	10800.0	0.84	25
RC103	2652.06	2903.37	2652.06	2677.50	10800.1	0.96	29
RC104	2651.93	2902.25	2651.93	2670.48	10800.6	0.70	31
RC105	2657.93	2886.28	2657.93	2681.08	10800.0	0.87	25
RC106	2651.72	2913.33	2651.84	2679.39	10800.3	1.04	29
RC107	2648.38	2908.49	2648.42	2674.13	10800.2	0.97	36
RC108	2648.18	2901.36	2648.18	2677.44	10800.0	1.10	37
RC201	2708.91	3050.71	2709.69	2721.21	142.3	0.43	8
RC202	2683.89	2994.67	2683.89	2689.62	105.7	0.21	5
RC203	2662.52	2929.48	2662.52	2674.46	3019.5	0.45	19
RC204	2660.69	2929.48	2660.69	2673.72	549.1	0.49	16
RC205	2686.93	2983.00	2686.93	2699.54	379.7	0.47	18
RC206	2684.94	2989.09	2686.64	2693.69	121.4	0.26	8
RC207	2657.66	2937.62	2657.66	2683.87	10800.0	0.99	28
RC208	2648.18	2902.54	2648.18	2677.90	10800.2	1.12	40

Computational Results

- Problem instances with 100 customers
- Limit for total CPU is set to 8 hours
- Applied strategy $\rightarrow$ DF method

  Average results of problem instances in C, R and RC sets with 100 customers obtained by DF method with 8 hour CPU limit

Set	Method	Avg. Gap%
C1-100	DF	7.26
C2-100	DF	22.45
R1-100	DF	4.01
RC1-100	DF	2.82

Other VRPSTT aspects: hard time windows

Based on

- F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, (2015). "A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times", *European Journal of Operational Research*, 249(1):55-66.

The VRPTW-ST

A two-stage stochastic programming with recourse model

- Stochastic service times: discrete with finite support (and mutually independent)
- Hard time windows
- No demands, nor vehicle capacity

The VRPTW-ST (1)

A priori plan:

A set of vehicle routes such that:

- Each route starts and ends at the depot
- All the customers are assigned to exactly one route

Operationally-Feasible (op-feasible) a priori plan

Given the service time realizations, an a-priori plan is said op-feasible if : Service at customers (including the evaluation) starts within the given time windows

- Vehicles may arrive before the beginning of a time window.
- Late service time is not allowed

If the a priori plan is NOT op-feasible:

- A recourse action will be taken
  - Some customer (or service) is skipped
  - A penalty π is paid

The VRPTW-ST (2)

The VRPTW-ST consists of:

- Selecting an a priori plan such that:
- The total expected costs are minimized
- The cost accounts for recourse actions and combines:
  - Expected routing costs (includes possible route shortcuts)
  - Expected penalty costs

Two practically reasonable conditions are imposed

- The probability that each route is op-feasible must be greater than or equal to a given reliability threshold $\alpha \in (0, 1)$
- No route requires more than one recourse action

The VRPTW-ST: recourse actions

Skip-Current Recourse: (C)

- Once at customer, the actual service time is evaluated.
- If such service time is infeasible with the next client's TW:
  - The service at the current client is skipped
  - A penalty π is payed

Skip-Next Recourse: (N)

- Once at customer, the actual service time is evaluated.
- such service time is infeasible with the next client's TW:
  - The visit and service at the next customer client is skipped
  - A penalty π is payed

Branch-cut-and-price algorithm

- 87/116 instances solved to optimality for recourse C
- 61/116 instances solved to optimality for recourse N
- Instances with up to 50 customers were solved for both recourse policies

VRPSTT: a chance-constrained model

Based on

- Errico, F., Desaulniers, G., Gendreau, M., Rei, W. and Rousseau, L.M., 2016. The vehicle routing problem with hard time windows and stochastic service times. *EURO Journal on Transportation and Logistics*, to appear.

Considered model

- Stochastic service times (discrete customer service-time probability distribution with finite support)
- Hard time windows
- No demands, nor vehicle capacity
- VRPTW-ST
- Motivation: Dispatching of technicians or repairmen

Innovation

- Use chance-constrained stochastic model and developing an exact method

Notation and assumptions

- A directed graph $G = (V, A)$, where
  - $V = \{0, 1, \ldots, n\}$ is the node set
    - 0 represents a depot
    - $V_c = \{1, \ldots, n\}$ the customer set
  - $A = \{(i, j) \mid i, j \in V\}$ is the arc set
- A non-negative travel cost c_{ij} and travel time t_{ij} are associated with each arc (i, j) in A.
- A hard time window $[a_i, b_i]$, $i \in V_c$
- A stochastic service time s_i, $i \in V_c$
- Service time probability functions are supposed to be known and
  - Discrete with finite support
  - Mutually independent

The VRPTW-ST with Chance Constraint

Definition of a Successful Route

Given a service time realization, a route is said successful if :

- Route starts and ends in node 0;
- Service at customers starts within the given time windows.
  - Vehicles may arrive before the beginning of a time window
  - Late service time is not allowed

The VRPTW-ST finds a set of cost minimizing routes such that:

- Each route start and end in node 0
- The routes induce a proper partition of all customers
- The global probability that the planned route is successful is higher than a given reliability threshold $0 < \alpha < 1$;

Formulation

- $\mathcal{R}$: set of all possible routes.
- $a_{ir} = 1$ parameter if route r visits customer i and 0 otherwise
- c_r the cost associated with route r
- $x_r = 1$ binary variable if route r is chosen, 0 otherwise

Formulation:

$$\min \sum_{r \in \mathcal{R}} c_r x_r \tag{28}$$

$$s.t. \sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \qquad \forall i \in V_c \tag{29}$$

$$\Pr\{\text{All routes are successful}\} \geq \alpha \tag{30}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}, \tag{31}$$

Linearization

Proposition

Let $\mathcal{R}'$ denote a set of routes inducing a proper partition of the customers set V_c. Given any two routes $r_1, r_2 \in \mathcal{R}'$, the success probability of r_1 is independent from the success probability of r_2.

This can be used to linearize constraint (30):

$$\Pr\{\text{ All routes are successful}\} \geq \alpha$$

$$\prod_{r \in \mathcal{R}: x_r = 1} \Pr\{\text{ Route } r \text{ is successful }\} \geq \alpha,$$

$$\sum_{r \in \mathcal{R}} x_r \ln(\Pr\{\text{ Route } r \text{ is successful }\}) \geq ln(\alpha)$$

$$\sum_{r \in \mathcal{R}} \beta_r x_r \leq \beta,$$

where

$$\beta_r := -\ln(\Pr\{\text{ Route } r \text{ is successful }\})$$
$$\beta := -ln(\alpha)$$

Computing the route success probability (1)

Observations:

- Consider a route $r = (v_0, \ldots, v_q, v_{q+1})$ where v_0 and v_{q+1} are 0
- Consider $\bar{t}_{v_i}$ the random variable for the service starting time at customer v_i
- r is successful $\Leftrightarrow a_{v_i} \leq \bar{t}_{v_i} \leq b_{v_i}$, for all customers in r
- To compute the route success probability we need the probability distributions of $\bar{t}_{v_i}$
- $\bar{t}_{v_i}$ are sums of independent random variables
    - Their distribution can be computed by convolution
    - Under certain hypothesis, convolutions have nice properties (closed forms, etc)
    - **Not in this case: Time windows truncate/modify the distributions**

Computing the route success probability (2)

- Starting service times $\bar{t}_{v_i}$ are linked to arrival times t_{v_i}:

$$\bar{t}_{v_i} = \begin{cases} a_{v_i} & t_{v_i} < a_{v_i} \\ t_{v_i} & a_{v_i} \leq t_{v_i} \leq b_{v_i} \end{cases}$$

- For the corresponding probability mass functions $m^t_{v_i}$ and $\bar{m}^t_{v_i}$

$$\bar{m}^t_{v_i}(z) = \begin{cases} 0 & z < a_{v_i}\,, \\ \sum_{l \leq a_{v_i}} m^t_{v_i}(l) & z = a_{v_i}\,, \\ m^t_{v_i}(z) & a_{v_i} < z \leq b_{v_i}\,, \\ 0 & z > b_{v_i}\,. \end{cases}$$

- Observe that for a given v_i:

$$\Pr\{r \text{ is successful up to } v_i\} = \sum_{z \in \mathcal{N}} \bar{m}^t_{v_i}(z)$$

Computing the route success probability (3)

$$t_{v_i} = \bar{t}_{v_i - 1} + s_{v_i - 1} + t_{v_i - 1, v_i}$$

Iterative procedure:

- for $(i = 1, \ldots, q-1)$ do
  - a) Truncation Step: Starting from $m^t_{v_i}$ obtain $\bar{m}^t_{v_i}$
  - b) Convolution Step: Compute $m^t_{v_{i+1}}(z) = \sum_{k \in \mathcal{N}} m^s_{v_i}(k) \bar{m}^t_{v_i}(z - t_{v_i, v_{i+1}} - k), \ \forall z \in \mathcal{N}$
- Compute: $\Pr\{r \text{ is successful}\} = \sum_{z \in \mathcal{N}} \bar{m}^t_{v_q}(z)$

Critical point: algorithmic complexity depends on

- The quality of the time discretization
- The customer time windows widths
- The amplitude of the distribution supports

A Branch-and-Price-and-Cut Algorithm

- Method based on implicit enumeration
- Linear relaxation are solved by column generation
- If violated inequalities are found, new cuts are added and the process is iterated
- Integrality recovered by branching

Column generation

- Restricted Master Problem: limited number of columns are considered
- Subproblem: identify columns with negative reduced cost
- In VRP contexts: Elementary Shortest Path Problem with Resource Constraints (ESPPRC)
- Usually solved by labeling algorithms (dynamic programming)

ESPPRC required major modifications for the VRPTW-ST

Classic ESPPRC

Labeling algorithm minimizing the route reduced cost

- Origin/destination graph
  - Nodes $\rightarrow$ clients, arcs $\rightarrow$ vehicle movements
  - Resource windows are associated with nodes (time windows, etc.)
  - Costs and resource consumption are associated with arcs (time, capacity consumption, etc.)
- Partial routes are iteratively extended
- Labels associated with node implicitly represent partial route
- Typically for classic VRPTW: $E = (C, T, L, V^1, \ldots, V^n)$
- Labels are extended according to extension functions: e.g., $T_j = T_i + t_{ij}$
- Dominance rules are very important to eliminate suboptimal labels. E^1 dominates E^2 if $E^1 \leq E^2$, i.e.,:
  - $C^1 \leq C^2$
  - $T^1 \leq T^2$
  - $L^1 \leq L^2$
  - $V_i^1 \leq V_i^2$, for all customers $i \in N$

Shortest path with probabilistic resource consumption

- Find route minimizing the reduced cost: $\bar{c}_r = c_r - \sum_{i \in V_c} a_{ir}\gamma_i + \beta_r \delta$
  - γ_i dual variables associated with set partitioning constraints
  - δ dual variable associated with chance constraint
  - Reminder: $\beta_r = -\ln(\Pr\{r \text{ is successful }\})$
- Issues:
  - The consumption of the time resource is probabilistic
  - We have a probabilistic constraint on the route success probability
- Possible answer: substitute Time resource with Route success probability
- Problem: the extension of the route success probability requires the truncated arrival time probability distribution
  - More label components are needed
- For VRPTW-ST: $E_i = (C_i, \bar{M}_i^t(a_i), \ldots, \bar{M}_i^t(b_i), V_i^1, \ldots, V_i^n)$
  - $\bar{M}_i^t(z) := \sum_{l \leq z} \bar{m}_i^t(l)$

Extension functions: reduced cost

Proposition: Reduced cost decomposition

The reduced cost of a route $r = (v_0, \dots, v_q, v_{q+1})$, can be expressed as

$$\bar{c}_r = \sum_{i=1}^{q+1} \bar{c}_{v_{i-1}, v_i},$$

where

$$\begin{aligned}
\bar{c}_{v_{i-1}, v_i} &:= c_{v_{i-1}, v_i} - \gamma_{v_i} + \delta p_{v_{i-1}, v_i}, \; i = 1, \dots, q \\
\bar{c}_{v_q, v_{q+1}} &:= c_{v_q, 0}. \\
p_{v_{i-1}, v_i} &:= -\ln(\bar{M}^t_{v_i}(b_{v_i}) / \bar{M}^t_{v_{i-1}}(b_{v_{i-1}})),
\end{aligned}$$

- Extension function: $C_j = C_i + \bar{c}_{ij}$
- The non-decreasing property does not hold $\Rightarrow$ more difficult dominance properties

Other extension functions

- Components $\bar{M}^t(a), \ldots, \bar{M}^t(b)$
  - Derived from the previous algorithm to compute the route success probability

$$\bar{M}_j^t(z_j) = \sum_{k \in \mathcal{N}} m_i^s(k) \bar{M}_i^t(z_j - t_{ij} - k)$$

  for all $z_j \in [a_j, b_j]$, where $m_i^s(\cdot)$ is the service time probability mass function

- Components $V_1, \ldots, V_n$
  - Similar to Feillet (2004)

Dominance for the VRPTW-ST

Definition (Dominance)

Consider partial routes r_i^1, r_i^2 ending in a generic node i. E_i^1 dominates E_i^2 if:

- Any feasible extension e of r_i^2 ending at a given node j is also feasible for r_i^1
- For any such extension e, $C_j^1 \leq C_j^2$

Proposition (Dominance rule for the VRPTW-ST)

If r^1 and r^2 are such that

- $c_i^1 - \sum_{h \in \mathcal{N}(r^1)} \gamma_h \leq c_i^2 - \sum_{h \in \mathcal{N}(r^2)} \gamma_h$,
- $V_i^{1h} \leq V_i^{2h}$ for all $h \in V_c$,
- $\bar{M}_i^{1t}(z_i) \geq \bar{M}_i^{2t}(z_i)$, for all $z_i \in [a_i, b_i]$,

and at least one of the above inequalities is strict, then r^1 dominates r^2.

Implementation details and accelerating strategies

- Initial columns: feasible solution given by dedicated trips $0 - i - 0$ for each customer i
- Decremental state space (Boland et al. 2006, Righini and Salani 2008)
- ng-path relaxation (Baldacci et al. 2011).
- Heuristic dynamic programming:
  - Temporarily elimination of arcs with high values of $c_{ij} - \gamma_j$
  - Aggressive dominance rules:
    - Consider and gradually extend subsets of the visit components $V_1, \ldots, V_n$
    - Consider and gradually extend subsets of cumulative distribution components $\bar{M}^t(a), \ldots, \bar{M}^t(b)$
- Heuristic column generator: Multi-start Tabu search (Desaulniers et al. 2008):

Cutting planes and branching strategies

- Cutting planes: Subset-row inequalities (Jepsen et al. 2008):

$$\sum_{r \in \mathcal{R}} \left\lfloor \frac{1}{k} \sum_{i \in S} a_{ir} \right\rfloor x_r \leq \left\lfloor \frac{|S|}{k} \right\rfloor, \ \forall S \subseteq V_c, \ 2 \leq k \leq |S|.$$

  - As Jepsen et al. we only consider $|S| = 3$ and $k = 2$ (easier to find):

$$\sum_{r \in \mathcal{R}_S} x_r \leq 1, \ \ \forall S \in V_c \ : |S| = 3,$$

    where $\mathcal{R}_S$ is the subset of paths visiting at least two customers in S.

- Branching strategies:
  - Number of vehicles
  - On arc-flow variables:

$$X_{ij} = \sum_{r \in \mathcal{R}} b_{ijr} x_r, \ \forall (i,j) \in A,$$

Instance set

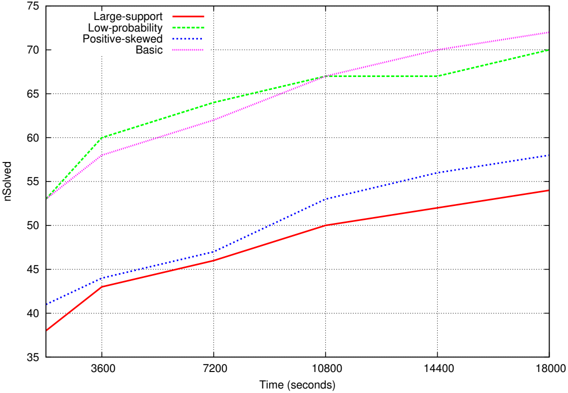
The time horizon was discretized in intervals of 0.1 minutes
Four families of Instances derived from the VRPTW database of Solomon (1987):

- 1) Basic:
    - Symmetric triangular distributions
    - Median corresponding to original service time values: 100 for R and RC, 900 for C
    - Support: $[80, 120]$ for R and RC, $[700, 1100]$ for C
    - Minimum success probability: $\alpha = 95\%$
- 2) Low-probability: Similar to Basic case, but the minimum success probability is $\alpha = 85\%$
- 3) Large-support: Similar to Basic case, but larger support: $[50, 150]$ for R and RC, $[450, 1350]$ for C
- 4) Positive-skewed: Similar to Large-support case, but different median values: 70 for R and RC, 630 for C

- Capacity and demand are disregarded
- Number of customers: 25 and 50 for R1, RC1, C1; 25 for R2, RC2, C2. (85 X 4 = 340 Total)
- Max CPU time: 5h on Intel i7-2600 3.40GHz, 16G RAM

Performance on benchmark instances (1)

Number of optimally solved instances (over 85):



- Instance families with larger support are more difficult
- Approx. 80% of the instances are solved within the first hour

Deterministic VS Stochastic Model

Deterministic (Median values) VS Stochastic (Large-support)

class	PercCostDAvg	PercSuccDAvg	count
1	-6.8	-44.9	39
2	-0.1	-5.1	15
	-5.0	54	

- General tendency: modest cost decrease $\Longleftrightarrow$ consistent decrease of success probability
- Some differences:
    - Family 1 : $-6.8 \Longleftrightarrow -44.9\%$
    - Family 2 : $0.1 \Longleftrightarrow 5\%$
- Stochastic model is convenient

VRP with stochastic customers

- Each customer has a given probability of requiring a visit
- Problem grounded in the pioneering work of Jaillet (1985) on the Probabilistic Traveling Salesman Problem (PTSP)
- At first sight, the VRPSC is of no interest under the re-optimization approach
- Recourse: skip absent customers
- Gendreau et al. (1995,1996) developed a priori based exact and heuristic algorithms for the VRPSC
- Bent, Pascal and Van Hentenryck (2004) study the VRPTW where some customers are known at planning time while others are dynamic

SVRP

Challenges

- We have been simplifying both the probability distribution functions and the recourse policies (to obtain manageable models)
- Mimicking companies' reactions to uncertainty is challenging
- The informational processes assumed in current SVRP models do not necessarily reflect the ones available in practice
- Failures on a route are solely dealt with by the vehicle performing the route
- In most models a unique stochastic dimension is handled
- ....

Expressing stochasticity

- Utilizing existing large amounts of data to express uncertain parameters
- Using a mix of distributions to depict a stochastic feature
- Allow a more staggered view with respect to the reliability of the estimates
- ..

Perspectives on SVRP

VRPSD

- Developing more realistic recourse polices
- Recourses involving collaboration between vehicles, i.e., more global forms of recourse
- Demand information may be transmitted before arriving at a customer location
- Correlated demands

VRPSTT

- Handling correlated travel times
- Accounting for the underlying road network
- More reliable estimates in the vicinity of the current location

Odysseus 2018 alert!

- TH3a: Fausto Errico, Guy Desaulniers, Andrea Lodi and Borzou Rostami. *Exact and approximate solution methods for the vehicle routing problem with stochastic and correlated travel times.*

Perspectives on SVRP

VRPSC

- More accurate modeling of customer presence
- Not present customers may not necessarily be skipped

Other stochastic aspects

- Energy consumption stochasticity
- Energy estimation reliability
- Cost-related stochasticity

Odysseus 2018 alert!

- TH3a: Samuel Pelletier and Fan E. *The electric vehicle routing problem with energy consumption uncertainty*

Odysseus 2018 recommendations for SVPR

- Session MO2a: Stochastic Vehicle Routing 1 (Green Room)
- Session MO4c: Stochastic Programming (White Room)
- Session TU1a: Robust Vehicle Routing (Green Room)
- Session WE1a: Stochastic Vehicle Routing 2 (Green Room)
- Session TH3a: Stochastic Vehicle Routing 3 (Green Room)
- Session FR3a: Dynamic Vehicle Routing (Green Room)

Thank you for your attention!