Stochastic Vehicle Routing: an Overview and some Recent Advances

Michel Gendreau
CIRRELT and MAGI
École Polytechnique de Montréal

VIII ALIO/EURO Workshop on Applied Combinatorial Optimization

Montevideo – December 8-10, 2014

Outline

1. Introduction
2. Basic Concepts in Stochastic Optimization
3. Modeling Paradigms
4. Problems with Stochastic Demands
5. Problems with Stochastic Customers
6. Problems with Stochastic Service or Travel Times
7. Conclusion and perspectives

Acknowledgements

- Fausto Errico (CIRRELT and École de technologie supérieure)
- Guy Desaulniers (GERAD and École Polytechnique)
- Walter Rei (CIRRELT and UQÀM)
- Louis-Martin Rousseau (CIRRELT and École Polytechnique)

Introduction

Vehicle Routing Problems

- Introduced by Dantzig and Ramser in 1959
- One of the most studied problem in the area of logistics
- The basic problem involves delivering given quantities of some product to a given set of customers using a fleet of vehicles with limited capacities.
- The objective is to determine a set of minimum-cost routes to satisfy customer demands.

Vehicle Routing Problems

Many variants involving different constraints or parameters:

- Introduction of travel and service times with route duration or time window constraints
- Multiple depots
- Multiple types of vehicles
- …

What is Stochastic Vehicle Routing?

Basically, any vehicle routing problem in which one or several of the parameters are not deterministic:

- Demands
- Travel or service times
- Presence of customers
- …

Main classes of stochastic VRPs

- VRP with stochastic demands (VRPSD)
  - A probability distribution is specified for the demand of each customer.
  - One usually assumes that demands are independent (this may not always be very realistic...).
- VRP with stochastic customers (VRPSC)
  - Each customer has a given probability of requiring a visit.
- VRP with stochastic travel times (VRPSTT)
  - The travel times required to move between vertices, as well as sometimes service times, are random variables.

Basic Concepts in Stochastic Optimization

Dealing with uncertainty in optimization

- Very early in the development of operations research, some top contributors realized that :
  - In many problems there is very significant uncertainty in key parameters;
  - This uncertainty must be dealt with explicitly.
- This led to the development of :
  - Stochastic programming with recourse (1955)
  - Dynamic programming (1958)
  - Chance-constrained programming (1959)
  - Robust optimization (more recently)

Information and decision-making

In any stochastic optimization problem, a key issue is:

- How do the revelation of information on the uncertain parameters and decision-making (optimization) interact?
  - When do the values taken by the uncertain parameters become known?
  - What changes can I (must I) make in my plans on the basis of new information that I obtain?

Stochastic programming with recourse

- Proposed separately 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, 2^{nd} 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 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.

  E.g., in VRP with stochastic demands,

  Pr{total demand assigned to route $r \le capacity$ } $\ge 1-\alpha$

Robust optimization

- Here, uncertainty is represented by the fact that the uncertain parameter vector must belong to a given polyhedral set (without any probability defined)
  - E.g., in VRP with stochastic demands,

    having set upper and lower bounds for each demand, together with an upper bound on total demand.
- Robust optimization looks in a minimax fashion for the solution that provides the best “worst case”.

Modelling paradigms

Real-time optimization

Also called re-optimization

- Based on the implicit assumption that information is revealed over time as the vehicles perform their assigned routes.
- Relies on Dynamic programming and related approaches (Secomandi et al.)
- Routes are created piece by piece on the basis on the information currently available.
- Not always practical (e.g., recurrent situations)

A priori optimization

- A solution must be determined beforehand; this solution is “confronted” to the realization of the stochastic parameters in a second step.
- Approaches:
  - Chance-constrained programming
  - (Two-stage) stochastic programming with recourse
  - Robust optimization
  - [“Ad hoc” approaches]

Chance-constrained programming

- Probabilistic constraints can sometimes be transformed into deterministic ones (e.g., in in VRP with stochastic demands, when one imposes that

  Pr{total demand assigned to route *r* ≤ *cap.* } ≥ 1-α,

  if customer demands are independent and Poisson).

- This model completely ignores what happens when things do not “turn out correctly”.

Robust optimization

- Not used very much in stochastic VRP up to now.
- Model may be overly pessimistic.

Stochastic programming with recourse

- **Recourse** is a key concept in a priori optimization
  - What must be done to "adjust" the a priori solution to the values observed for the stochastic parameters!
  - Another key issue is deciding when information on the uncertain parameters is provided to decision-makers.
- Solution methods:
  - Integer L-shaped (Laporte and Louveaux)
  - Column generation (Branch & Price)
  - Heuristics (including metaheuristics)
- Probably closer to actual industrial practices, if recourse actions are correctly defined!

VRP with stochastic demands

VRP with stochastic demands (VRPSD)

- A probability distribution is specified for the demand of each customer.
- One usually assumes that demands are independent
  (this may not always be very realistic...).
- Probably, the most extensively studied SVRP:
  - Under the reoptimization approach (Secomandi)
  - Under the a priori approach (several authors) using both the chance-constrained and the recourse models.

VRP with stochastic demands

- Classical recourse strategy:
  - Return to depot to restore vehicle capacity
  - Does not always seem very appropriate or “intelligent”

- Other recourse strategies are possible, however, and often closer to actual industrial practices.
  - Fixed threshold policies
  - Variable threshold policies
  - Preventive restocking (Yang, Ballou, Mathur, 2000)
  - Pairing routes (Erera et al., 2009)

VRP with stochastic demands

- Approximate solutions can be obtained fairly easily using metaheuristics
(e.g., Tabu Search, as in Gendreau et al., 1996).

- Computing effectively the value of the recourse function still remains a challenge.

VRP with stochastic customers

VRP with stochastic customers (VPRSC)

- 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 reoptimization approach.

VRP with stochastic customers (VPRSC)

- Recourse action:
  - “Skip” absent customers
- Has been extensively studied by Gendreau, Laporte and Séguin in the 1990’s:
  - Exact and heuristic solution approaches
- Can also be used to model the Consistent VRP (working paper with Ola Jabali and Walter Rei).

VRP with stochastic service or travel times

VRP with stochastic service or travel times

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

VRP with stochastic service times: a chance-constrained formulation

Following material from

- F. Errico, G. Desaulniers, M. Gendreau, W. Rei, L.-M. Rousseau. The Vehicle Routing Problem with hard time windows and stochastic service times.

  Forthcoming (hopefully…!)

Presentation Outline

Presentation Outline

1. The VRPTW-ST
2. Chance-constrained model
3. Two-stage stochastic model
4. Conclusions

Context

We consider a VRP with

- Stochastic service times
- Hard time windows
- No demands, nor vehicle capacity
- VRPTW-ST

Context

We consider a VRP with

- Stochastic service times
- Hard time windows
- No demands, nor vehicle capacity
- VRPTW-ST

Several applications :

- Dispatching of technicians or repairmen :
  - Perform specific services at the customers
  - Details of the service to perform are unknown beforehand
- Energy production planning :
  - Several power plants are connected in a network
  - Maintenance operations (implying outage) must planned in specific hard time windows (technicians are not available otherwise)
  - Duration of the operations is unknown beforehand

Related literature

- Several papers on VRP/m-TSP with stochastic travel times and customer deadlines or soft time windows (see Adulyasak and Jaillet, 2014)
- TSP with hard time windows and stochastic travel times in Jula et al. (2006), Chang et al. (2009)
  - Heuristic methods
- VRP with stochastic travel times, demand uncertainty and customer deadlines in Lee et al. (2012)
  - Robust optimization approach
- TSP with customers deadlines and stochastic customers in Campbell and Thomas (2008)

Related literature

- Several papers on VRP/m-TSP with stochastic travel times and customer deadlines or soft time windows (see Adulyasak and Jaillet, 2014)
- TSP with hard time windows and stochastic travel times in Jula et al. (2006), Chang et al. (2009)
    - Heuristic methods
- VRP with stochastic travel times, demand uncertainty and customer deadlines in Lee et al. (2012)
    - Robust optimization approach
- TSP with customers deadlines and stochastic customers in Campbell and Thomas (2008)
- **With respect to previous works, we aim to**
    - 1. Use chance-constrained stochastic model
      2. Use two-stage stochastic programming with recourse
    - Develop an exact solution method

Presentation Outline

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.

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 $[a_i, b_i]$, $i \in V_c$

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, $i \in V_c$.

Notation and assumptions

- A directed graph $G = (V, A)$, where
  - $V = \{0, 1, \dots, n\}$ is the node set
    - 0 represents a depot
    - $V_c = \{1, \dots, n\}$ 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 (Successful Route)

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

(i) Route starts and ends in node 0 ;

(ii) 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 with Chance Constraint

Definition (**Successful Route**)

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

(i) Route starts and ends in node 0 ;

(ii) 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 route such that :

1. Routes start and end in node 0 ;
2. Routes induce a proper partition of all customers
3. The global probability that the route plan is Successful is higher than a given reliability threshold $0 < \alpha < 1$;
4. The travel cost is minimized.

Formulation

- $\mathcal{R}$: set of all possible routes.
- $a_{ir} = 1$ prameter 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

- $\mathcal{R}$: set of all possible routes.
- $a_{ir} = 1$ prameter 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 :

$$\min \sum_{r \in \mathcal{R}} c_r x_r \tag{1}$$

$$s.t. \sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \qquad \forall i \in V_c \tag{2}$$

$$\Pr\{\text{All routes are Successful}\} \geq \alpha \tag{3}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}, \tag{4}$$

Linearization

Mutually independent service time $\Rightarrow$

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.

Linearization

Mutually independent service time $\Rightarrow$

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 (3) :

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

Linearization

Mutually independent service time $\Rightarrow$

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 (3) :

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

Linearization

Mutually independent service time $\Rightarrow$

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 (3) :

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

Linearization

Mutually independent service time $\Rightarrow$

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 (3) :

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

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 $\bar{t}_{v_i}$

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 hour case** : Time windows truncate/modify the distributions
- $\Rightarrow$ We actually need to carry out computations

Computing the route success probability (2)

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

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

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 $\bar{m}^t_{v_i}$

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

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)

Simple iterative procedure :

1. 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 $m^t_{v_{i+1}}(z) = (\bar{m}^t_{v_i} * m^s_{v_i})(z - t_{v_i, v_{i+1}}), \; \forall z \in \mathcal{N}$
2. Compute : $\Pr\{r \text{ is successful}\} = \sum_{z \in \mathcal{N}} \bar{m}^t_{v_q}(z)$

Computing the route success probability (3)

Simple iterative procedure :

1. 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 $m^t_{v_{i+1}}(z) = (\bar{m}^t_{v_i} * m^s_{v_i})(z - t_{v_i, v_{i+1}}), \ \forall z \in \mathcal{N}$
2. 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 precess is iterated.
- Integrality recovered by branching

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 precess 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)

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 precess 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

Label setting 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 route are iteratively extended

Classic ESPPRC

Label setting algorithm minimizing the route reduced cost

- Origin/destination graph
  - Nodes → clients, arcs → vehicle movements
  - Resource windows are associated with nodes (time windows, etc)
  - Costs and resource consumption are associated with arcs (time, capacity consumption, etc)
- Partial route are iteratively extended
- Labels associated with node implicitly represent partial route
- Typically for classic VRPTW : $E = (C, T, L, V^1, \ldots, V^n)$

Classic ESPPRC

Label setting 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 route 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 : eg. $T_j = T_i + t_{ij}$

Classic ESPPRC

Label setting 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 route 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 : eg. $T_j = T_i + t_{ij}$
- 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
  - Remind $\beta_r = -\ln(\Pr\{r \text{ is successful }\})$

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
  - Remind $\beta_r = -\ln(\Pr\{r \text{ is successful }\})$
- **Issues** :
  1. The consumption of the time resource is probabilistic
  2. We have a probabilistic constraint on the route success probability
- Possible answer : substitute Time resource with Route success probability.

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
    - Remind $\beta_r = -\ln(\Pr\{r \text{ is successful }\})$
- **Issues** :
    1. The consumption of the time resource is probabilistic
    2. 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

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
  - Remind $\beta_r = -\ln(\Pr\{r \text{ is successful }\})$
- **Issues** :
  1. The consumption of the time resource is probabilistic
  2. 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 : $E_i = (C_i, \bar{M}_i^t(a_i), \ldots, \bar{M}_i^t(b_i), V_i^1, \ldots, V_i^n)$
  - $\bar{M}_i^t(z) := \sum_{t \leq z} \bar{m}_i^t(l)$

Extension functions : Reduced Cost

Proposition : Reduced cost decomposition

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

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

where

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

Extension functions : Reduced Cost

Proposition : Reduced cost decomposition

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

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

where

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

- 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

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 :

i) Any feasible extension e of r_i^2 ending at a given node j is also feasible for r_i^1

ii) For any such extension e, $C_j^1 \leq C_j^2$

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 :

i) Any feasible extension e of r_i^2 ending at a given node j is also feasible for r_i^1

ii) 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

(i) $c_i^1 - \sum_{h \in \mathcal{N}(r^1)} \gamma_h \leq c_i^2 - \sum_{h \in \mathcal{N}(r^2)} \gamma_h$,

(ii) $V_i^{1h} \leq V_i^{2h}$ for all $h \in V_c$,

(iii) $\bar{M}_i^{1t}(z_i) \geq \bar{M}_i^{2t}(z_i)$, 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

1. Initial columns : feasible solution given by dedicated trips $0 - i - 0$ for each customer i

Implementation details and accelerating strategies

1. Initial columns : feasible solution given by dedicated trips $0 - i - 0$ for each customer i
2. Decremental state space (Boland et al. 2006, Righini and Salani 2008)

Implementation details and accelerating strategies

1. Initial columns : feasible solution given by dedicated trips $0 - i - 0$ for each customer i
2. Decremental state space (Boland et al. 2006, Righini and Salani 2008)
3. *ng*-path relaxation (Baldacci et al. 2011).

Implementation details and accelerating strategies

1. Initial columns : feasible solution given by dedicated trips $0 - i - 0$ for each customer i
2. Decremental state space (Boland et al. 2006, Righini and Salani 2008)
3. *ng*-path relaxation (Baldacci et al. 2011).
4. Heuristic dynamic programming :
   - Temporarily elimination of arcs with high values of $c_{ij} - \gamma_j$

Implementation details and accelerating strategies

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

Implementation details and accelerating strategies

1. Initial columns : feasible solution given by dedicated trips $0 - i - 0$ for each customer i
2. Decremental state space (Boland et al. 2006, Righini and Salani 2008)
3. *ng*-path relaxation (Baldacci et al. 2011).
4. Heuristic dynamic programming :
   - Temporarily elimination of arcs with high values of $c_{ij} - \gamma_j$
   - Aggressive dominance rules :
     1. Consider and gradually extend subsets of the visit components $V_1, \ldots, V_n$
     2. Consider and gradually extend subsets of cumulative distribution components $\bar{M}^t(a), \ldots, \bar{M}^t(b)$
5. Heuristic column generator : Multi-start Tabu search (Desaulniers et al. 2008) :
   - Applied to columns in the current basis
   - Moves : insertion/deletion of individual customers
   - Adapted for the VRPTW-ST : search space restricted to solution feasible w.r.t. the worst case scenario

Cutting planes and branching strategies

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

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

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

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

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

Cutting planes and branching strategies

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

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

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

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

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

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

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

Experiments

Two experimental campaigns :

1. Test the algorithm on benchmark instances
2. Evaluate stochastic model behavior and compare with deterministic models

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original values : 100 for R and RC, 900 for C
     - Support : $[80, 120]$ for R and RC, $[700, 1100]$ for C.
     - Minimum success probability : $\alpha = 95\%$

Fausto Errico fausto.errico@cirrelt.ca

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original 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 $\alpha = 85\%$

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original 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

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original 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

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original 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

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original 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)

Instance set

Instances derived from the VRPTW database of Solomon (1987) :

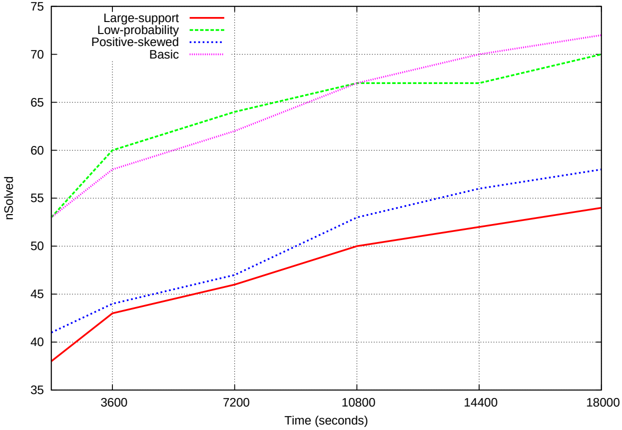
- We build 4 instance families :
  1. **Basic** :
     - Symmetric triangular distributions
     - Median corresponding to original values : 100 for R and RC, 900 for C
     - Support : $[80, 120]$ for R and RC, $[700, 1100]$ for C.
     - Minimum success probability : $\alpha = 95\%$
  2. **Low-probability** :
     - 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

Algorithmic features

- We performed a large amount of tests to evaluate several parameters and accelerating techniques
- Most important remarks :
  - *ng*-paths : best neighborhood cardinality : 10 ;
  - Heuristic dominance is very helpful
  - Heuristic column generation (tabu search) improves about 20% of running times
  - Subset-row inequalities are very effective

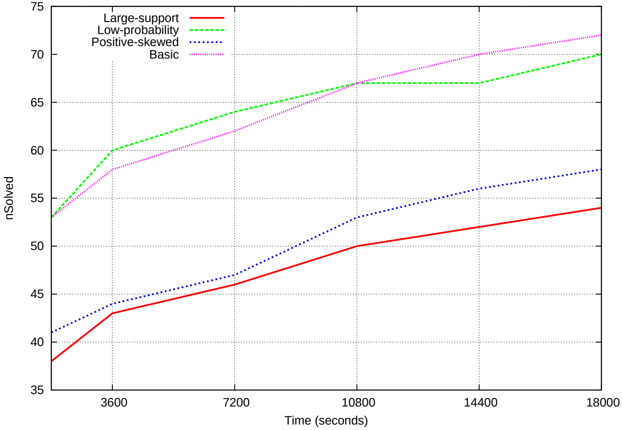
Performance on benchmark instances (1)

Number of optimally solved instances (over 85) :



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

Performance on benchmark instances (2)

Basic instance family

class	fam	dim	CostAvg	nVeh	SuccProbAvg	TimeAvg	total	count
1	C	25	207.8	3.2	98.3	2633.1	9	9
1	R	25	464.8	5.0	99.4	13.8	12	12
1	RC	25	351.4	3.3	99.3	41.7	8	8
1		25	353.8	4.0	99.0	834.4	29	29
1	C	50	390.6	5.6	97.4	8763.7	9	5
1	R	50	790.8	8.5	97.7	1123.0	12	11
1	RC	50	756.4	6.6	98.6	3008.9	8	7
1		50	693.3	7.3	97.9	3358.0	29	23
1			504.0	5.4	98.5	1950.6	58	52
2	C	25	214.6	2.0	99.8	3515.6	8	7
2	R	25	387.5	2.9	99.4	2580.4	11	10
2	RC	25	331.7	3.0	100.0	2594.9	8	5
2		25	319.8	2.6	99.7	2881.3	27	22
			449.2	4.6	98.9	2227.3	85	74

Performance on benchmark instances (2)

Basic instance family

class	fam	dim	CostAvg	nVeh	SuccProbAvg	TimeAvg	total	count
1	C	25	207.8	3.2	98.3	2633.1	9	9
1	R	25	464.8	5.0	99.4	13.8	12	12
1	RC	25	351.4	3.3	99.3	41.7	8	8
1		25	353.8	4.0	99.0	834.4	29	29
1	C	50	390.6	5.6	97.4	8763.7	9	5
1	R	50	790.8	8.5	97.7	1123.0	12	11
1	RC	50	756.4	6.6	98.6	3008.9	8	7
1		50	693.3	7.3	97.9	3358.0	29	23
1			504.0	5.4	98.5	1950.6	58	52
2	C	25	214.6	2.0	99.8	3515.6	8	7
2	R	25	387.5	2.9	99.4	2580.4	11	10
2	RC	25	331.7	3.0	100.0	2594.9	8	5
2		25	319.8	2.6	99.7	2881.3	27	22
			449.2	4.6	98.9	2227.3	85	74

- Efficiency of Stochastic B&P have common tendencies with Deterministic B&P :
  - Complexity increase with number of customers
  - Family C is more difficult than R and RC
  - Class 2 is mode difficult that 1 (Larger time windows)
  - Higher number of customer per routes corresponds to difficult problems

Deterministic VS Stochastic Model (1)

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

class	PercCostDAvg	PercVehDAvg	PercSuccDAvg	count
1	-6.8	-56.4	-44.91242	39
2	-0.1	0.0	-5.12083	15
	-5.0	-40.7	-33.85920	54

Deterministic VS Stochastic Model (1)

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

class	PercCostDAvg	PercVehDAvg	PercSuccDAvg	count
1	-6.8	-56.4	-44.91242	39
2	-0.1	0.0	-5.12083	15
	-5.0	-40.7	-33.85920	54

- General tendency : modest cost decrease $\Longleftrightarrow$ consistent decrease of success probability ($-5.0 \Longleftrightarrow -33.9\%$)
- Some differences :
  - Family 1 : $-6.8 \Longleftrightarrow -44.9\%$
  - Family 2 : $0.1 \Longleftrightarrow 5\%$
- Stochastic model is convenient

Deterministic VS Stochastic Model (2)

Deterministic (**Worst-case values**) VS Stochastic (**Large-support**)

class	PercCostDAvg	PercVehDAvg	PercSuccDAvg	count
1	9.6	74.4	2.93339	39
2	1.7	0.0	0.06861	15
	7.4	53.7	2.13762	54

Deterministic VS Stochastic Model (2)

Deterministic (**Worst-case values**) VS Stochastic (**Large-support**)

class	PercCostDAvg	PercVehDAvg	PercSuccDAvg	count
1	9.6	74.4	2.93339	39
2	1.7	0.0	0.06861	15
	7.4	53.7	2.13762	54

- General tendency : relevant cost increase $\Longleftrightarrow$ small increase of success probability ($+7.4 \Longleftrightarrow +2.1\%$)
- Some differences :
  - Family 1 : $+9.6 \Longleftrightarrow +2.9\%$
  - Family 1 : $+1.7 \Longleftrightarrow +0.07\%$
- Stochastic model is still convenient

Conclusions and perspectives

Conclusion and perspectives

- Stochastic vehicle routing is a rich and promising research area.
- Much work remains to be done in the area of recourse definition.
- SVRP models and solution techniques may also be useful for tackling problems that are not really stochastic, but which exhibit similar structures
- Up to now, very little work on problems with stochastic travel and service times, while one may argue that travel or service times are uncertain in most routing problems!
- Correlation between uncertain parameters is possibly a major stumbling block in many application areas, but almost no one seems to work on ways to deal with it.