MTH8414

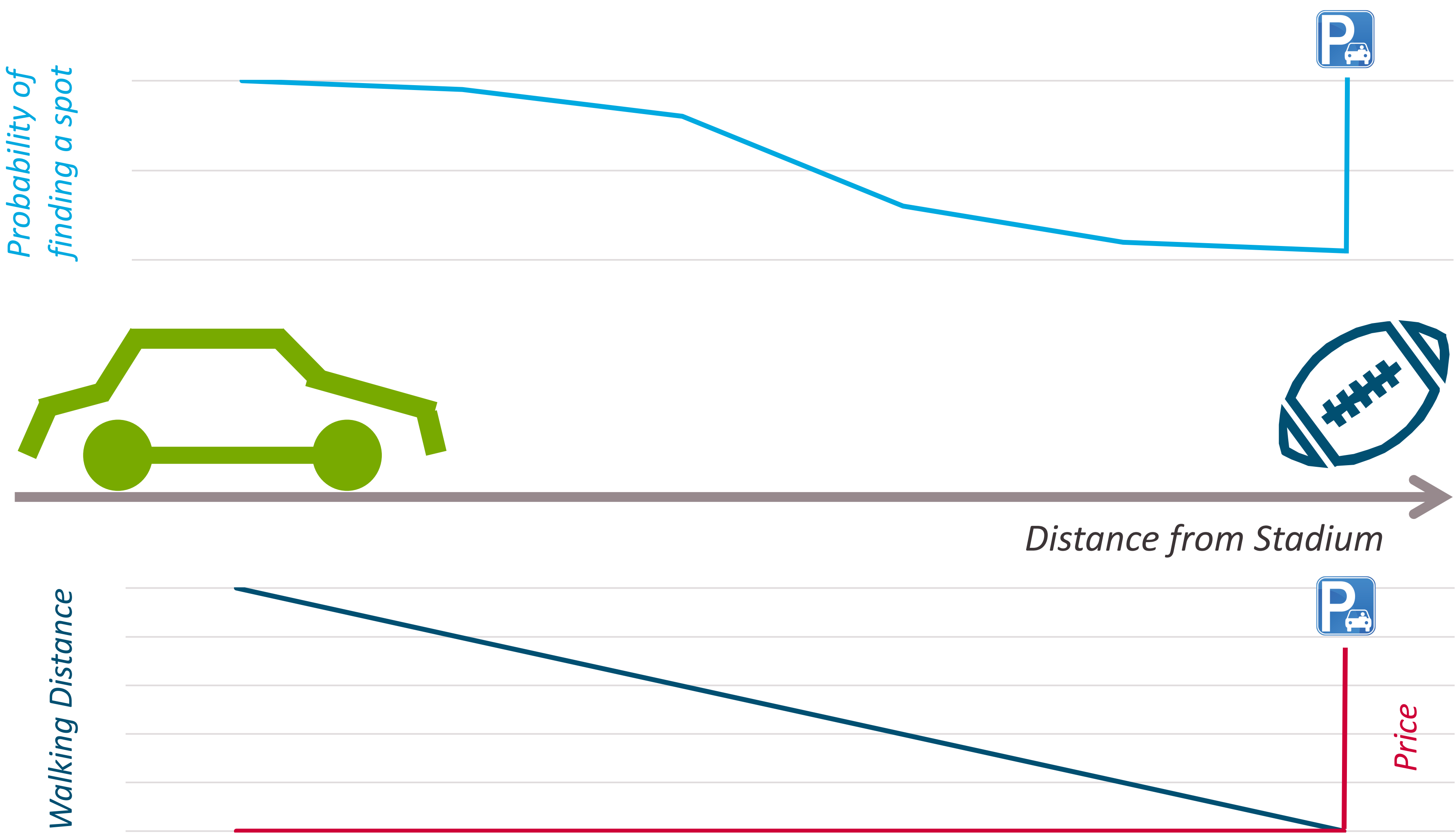
Planning Under Uncertainty

Merci à Gabrielle Gauthier Melançon, JDA Labs.

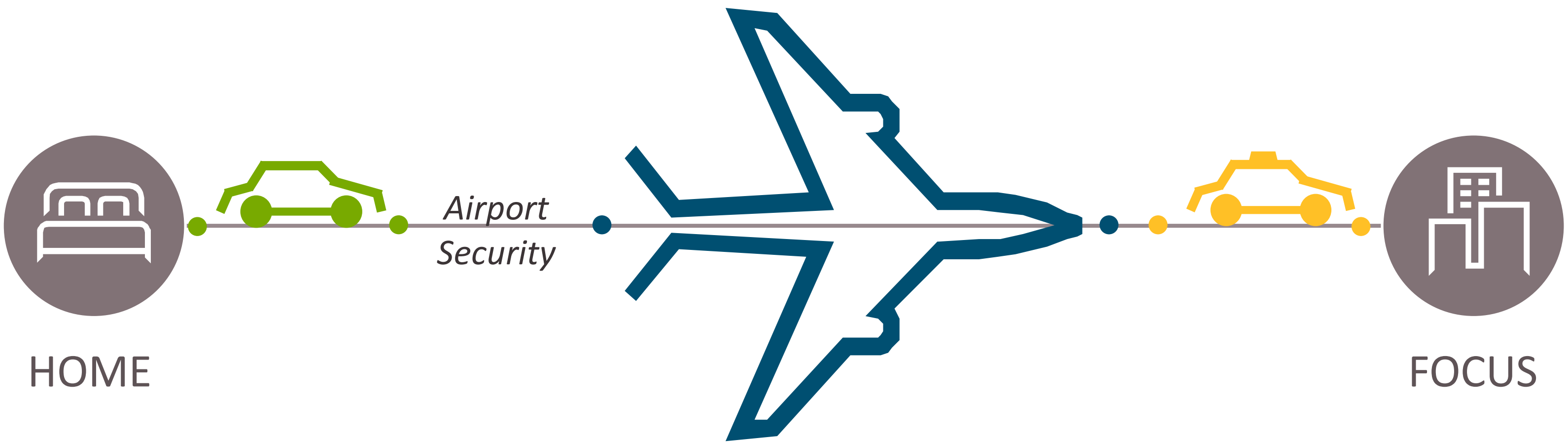
Agenda

- UNCERTAINTY
- APPROACHES
- APPLICATION

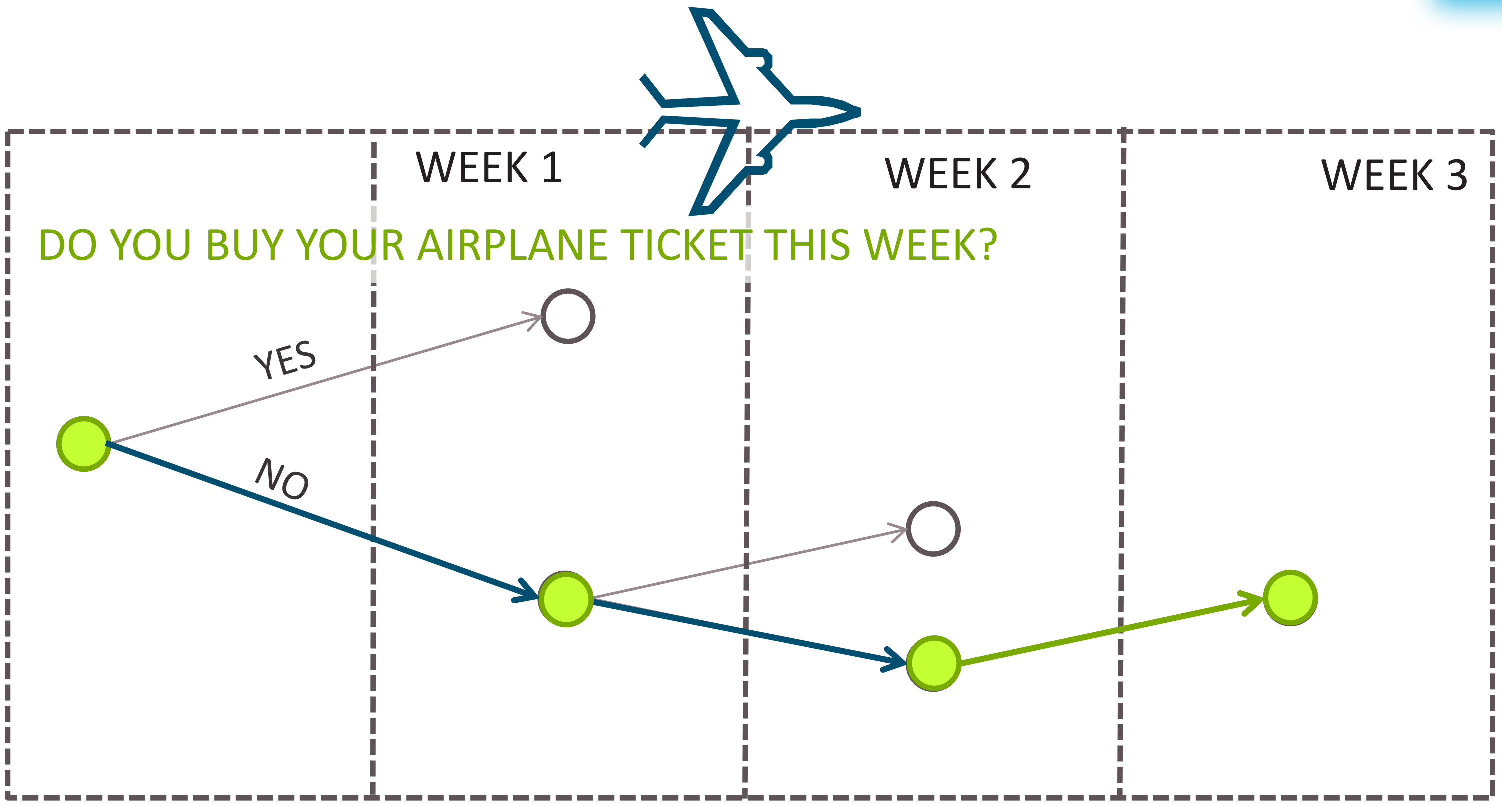
Finding a Parking Before a Game

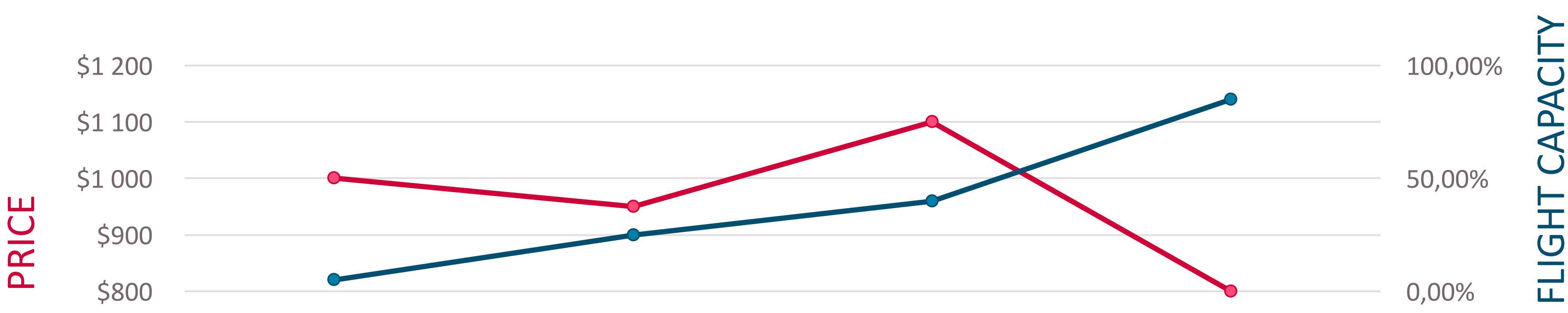


A Journey to conference

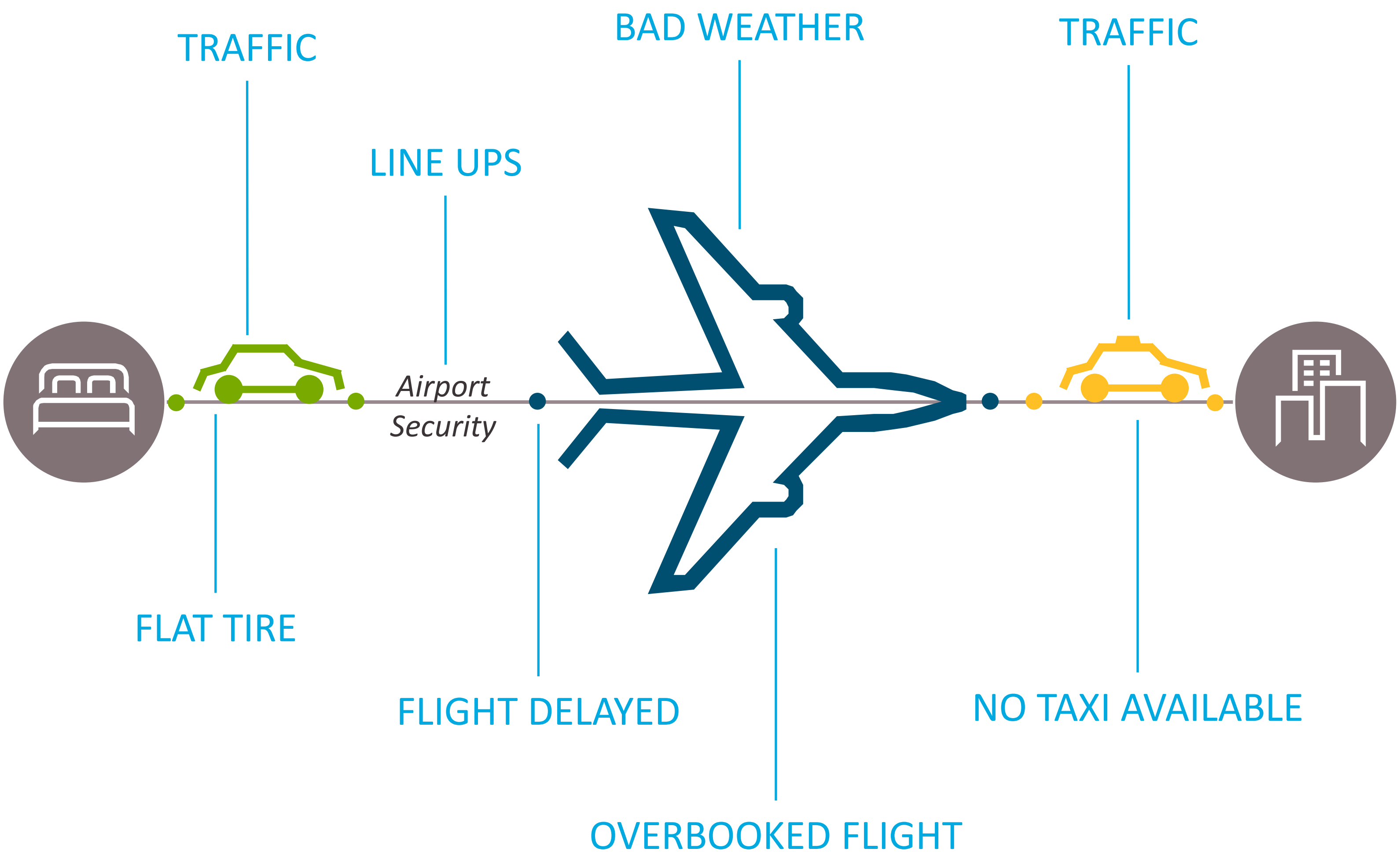


A Journey to a conference

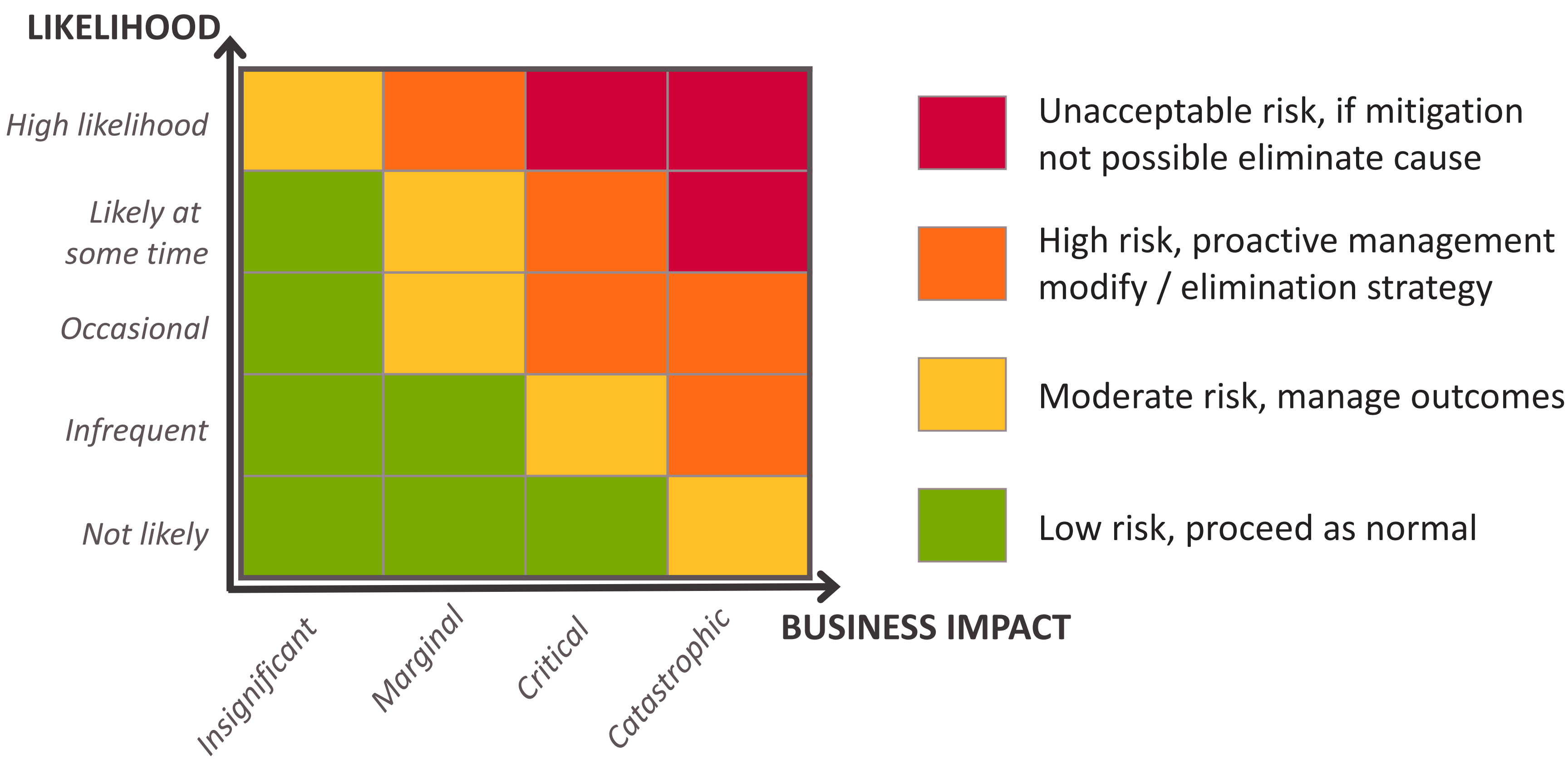




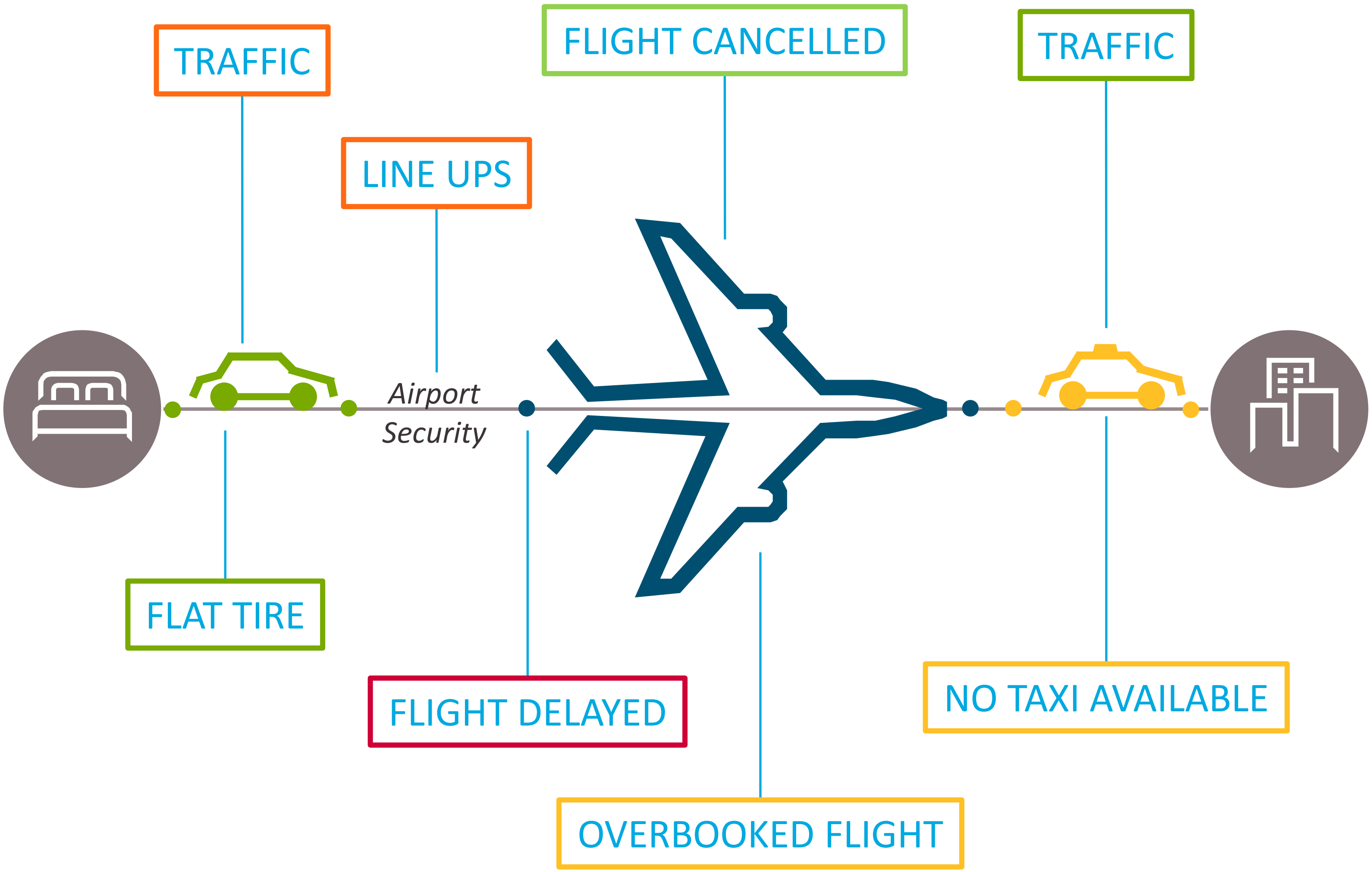
A Journey to a conference



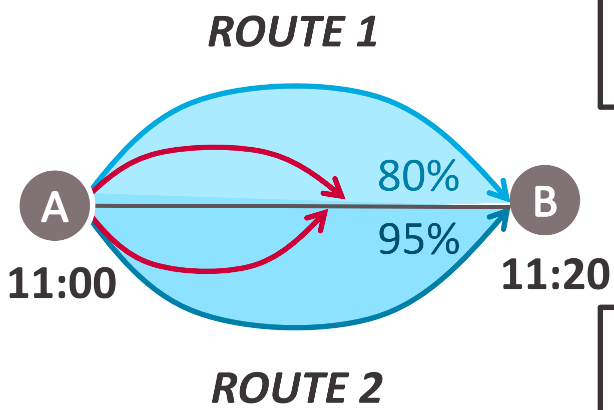
Analysis Framework for Proactive Risk Mitigation*



A Journey to a conference

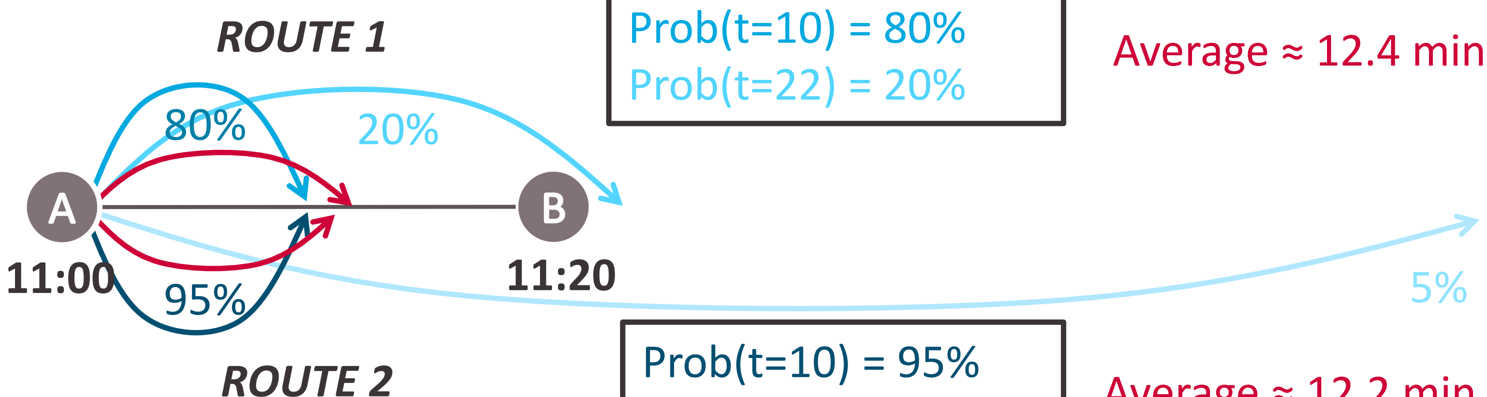


Challenges of Uncertainty



Average ≈ 12.4 min
Prob(t ≤ 20) = 80%

Average ≈ 12.2 min
Prob(t ≤ 20) = 95%



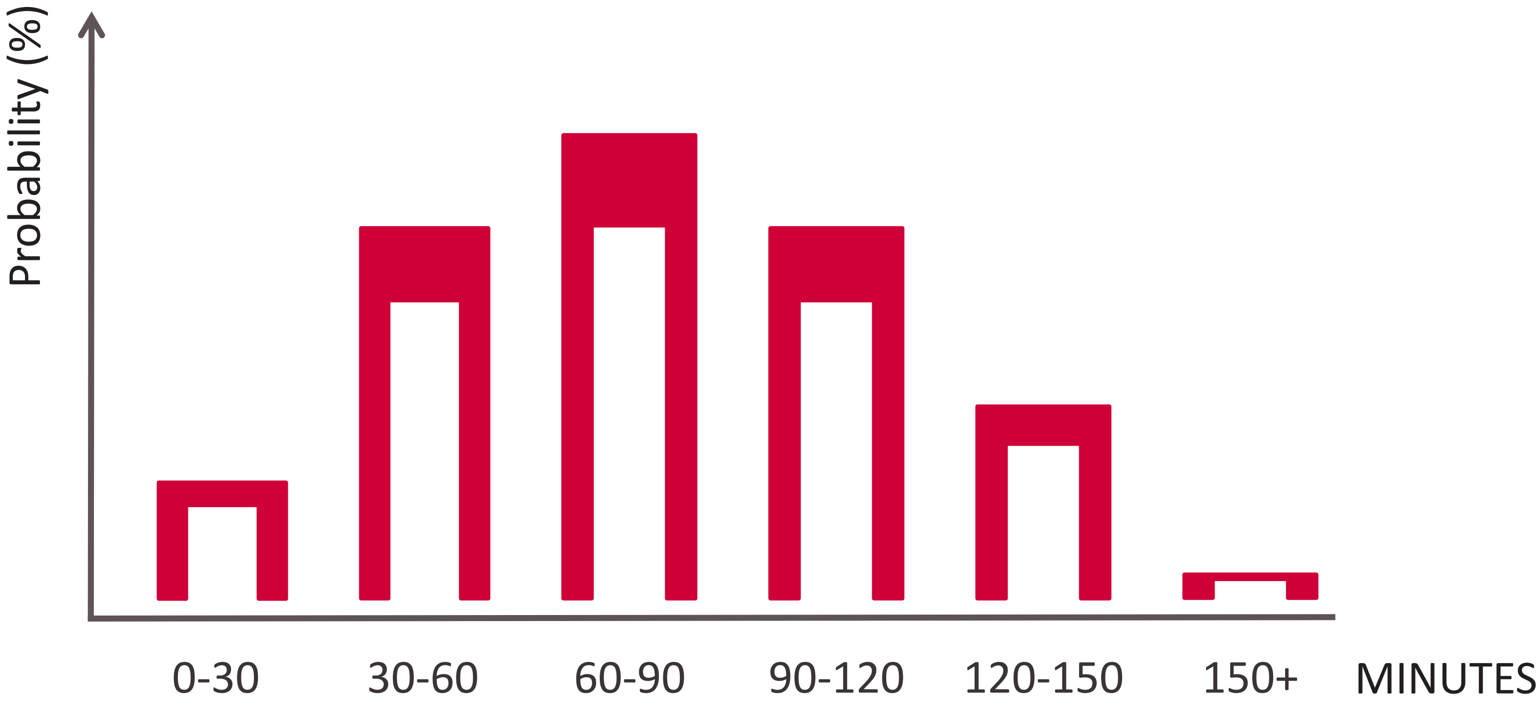
Prob(t=10) = 80%
Prob(t=22) = 20%

Average ≈ 12.4 min

Prob(t=10) = 95%
Prob(t=54) = 5%

Average ≈ 12.2 min

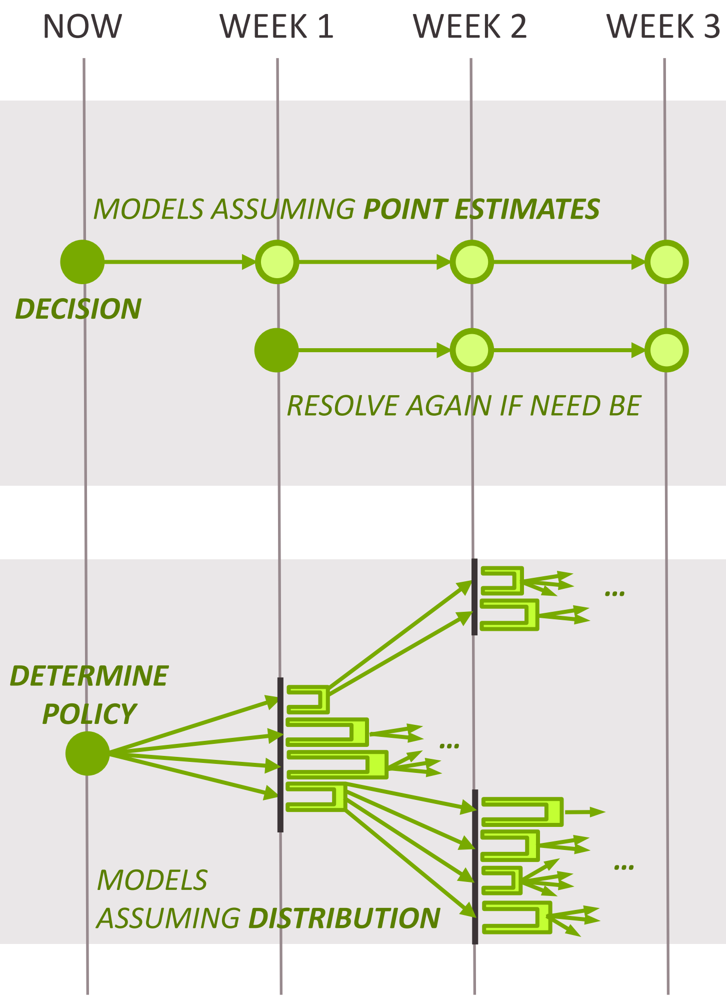
Graphical Probability Distribution



TIME TO CLEAR AIRPORT SECURITY

Decision Stages under Uncertainty

HERE-AND-NOW



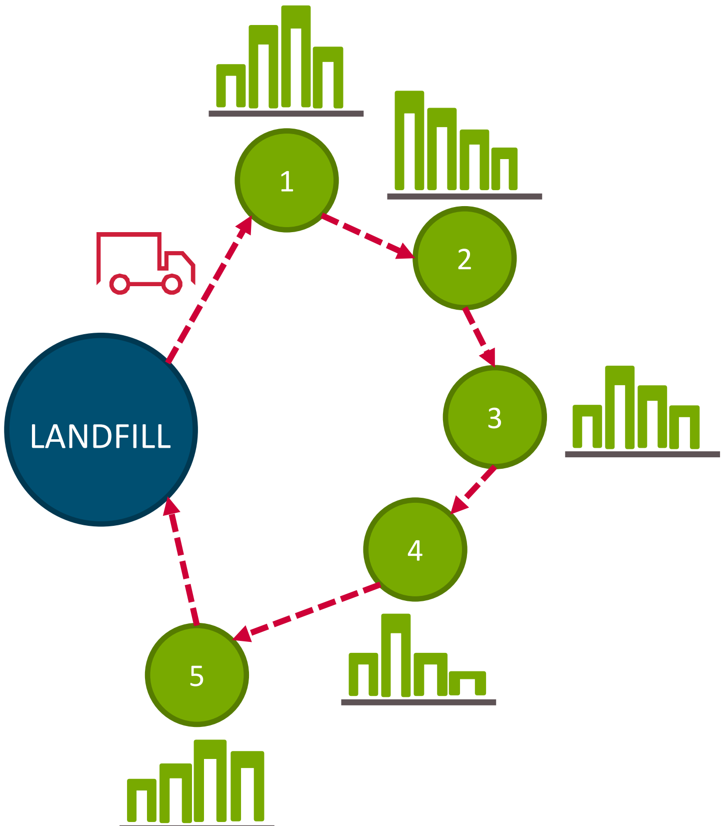
Before the realization of uncertainty, we take a « here-and-now » decision, by trying to guess the uncertainty

- Ex: Use averages, min or max values

WAIT-AND-SEE

Make a decision after the realization of uncertainty

Vehicle Routing Example

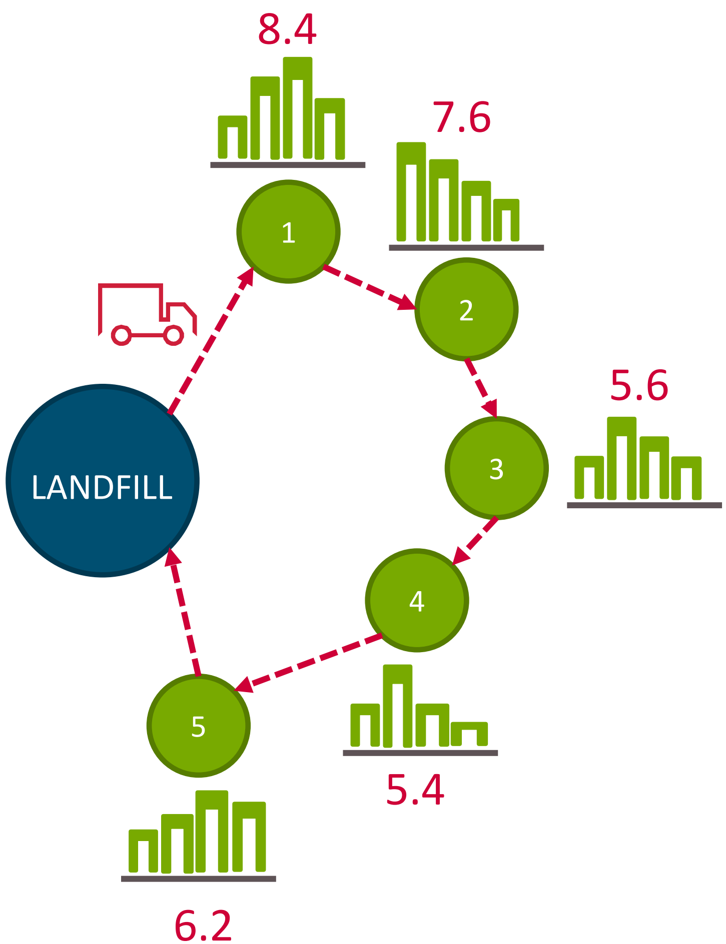


GOAL: MINIMIZE THE TOTAL TRAVEL DISTANCE

Unlimited fleet of vehicles of ***maximum capacity C***

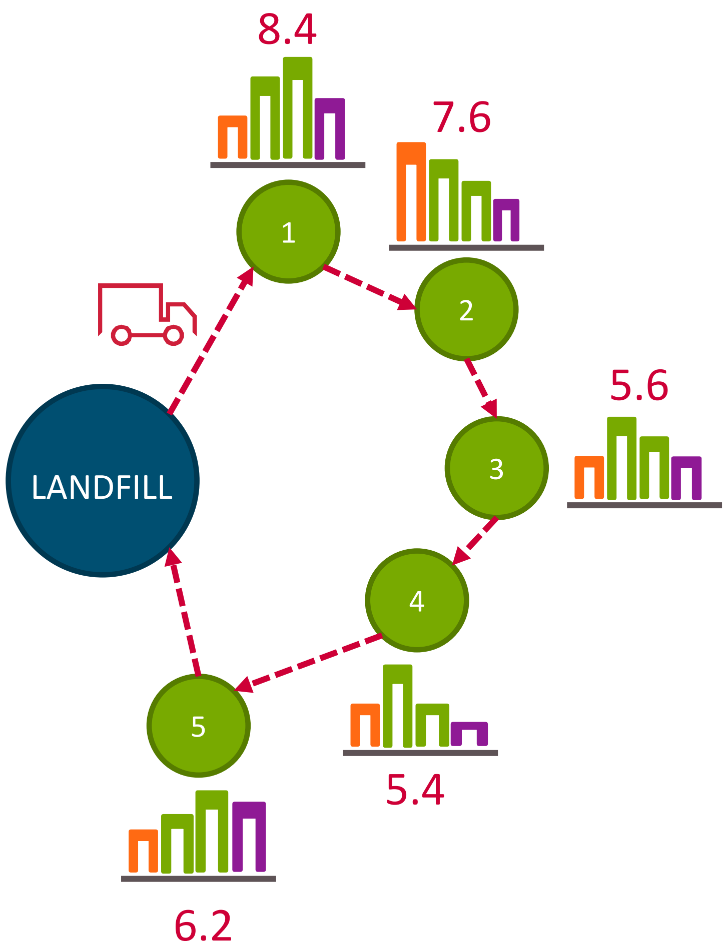
Independent random demands at n customers

Deterministic Approaches



- Before the realization of uncertainty, we take a « HERE-AND-NOW » decision, by trying to guess the uncertainty.
  - Use the average value (unbiased)
  - Use the minimum or maximum value (optimistic or pessimistic)
- Solve as a deterministic model.

Deterministic Approaches



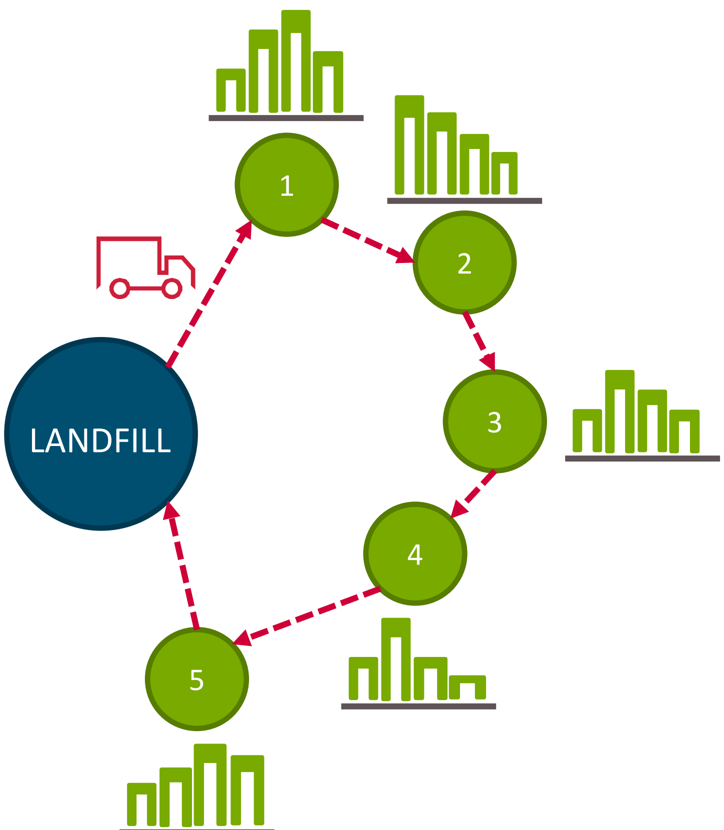
- Before the realization of uncertainty, we take a « HERE-AND-NOW » decision, by trying to guess the uncertainty.
  - Use the average value (unbiased)
  - Use the minimum or maximum value (optimistic or pessimistic)
- Solve as a deterministic model.

+ Pros
  + Easy and don't increase the model size.
  + Don't need a lot of information on the uncertainty.
- Cons
  - May over simplify the uncertainty.
  - The plan may be infeasible if the uncertainty is not as you planned.

Stochastic Approaches

- CHANCE CONSTRAINED OPTIMIZATION
- STOCHASTIC OPTIMIZATION
- ROBUST OPTIMIZATION

Uncertainty Impact on Plan?



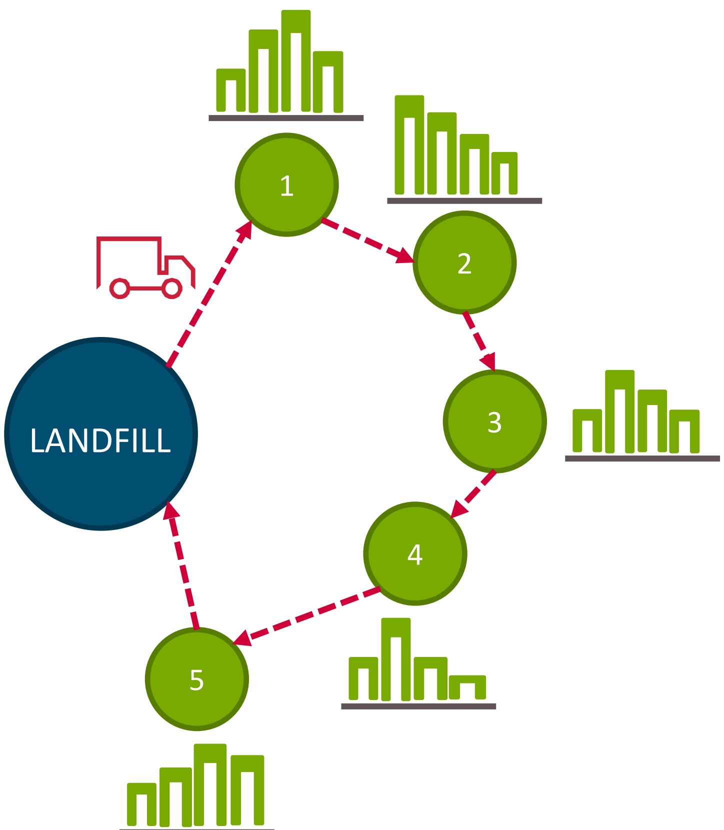
GOAL: MINIMIZE THE TOTAL TRAVEL DISTANCE

CONSTRAINT IMPACTED BY UNCERTAINTY:

MAXIMUM CAPACITY Q

1. Probability (exceed Q) = 0%
2. Probability (exceed Q) = 25%
3. Probability (exceed Q) = 45%
4. Probability (exceed Q) = 80%
5. Probability (exceed Q) = 100%

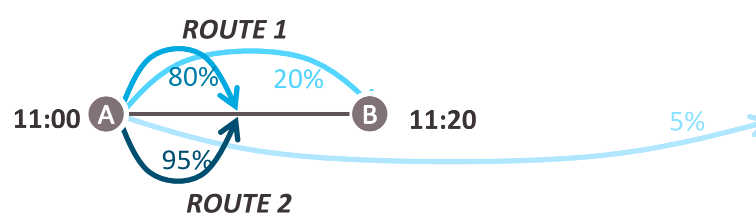
Chance Constrained Optimization



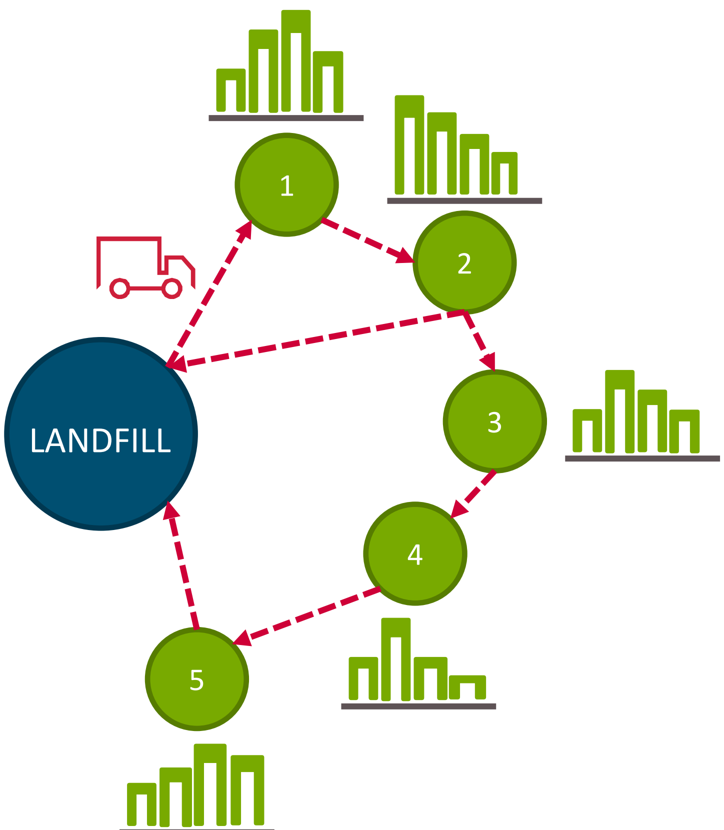
- Change Constrained Optimization enables to produce a plan that will be feasible with a certain probability.
  - In this case, we could aim for routes that will not exceed the capacity of the truck in at least 75% of uncertainty scenarios. (Maximum 25% of failure)

- Failure probability evaluated analytically or through simulation
- Known as Value-at-Risk in finance (VaR)

+ Pros
  + You can specify your level of risk.
- Cons
  - Neither says what to do in case of failure nor quantify the magnitude of the impact



Chance Constrained Optimization



GOAL: MINIMIZE THE TOTAL TRAVEL DISTANCE

CONSTRAINT IMPACTED BY UNCERTAINTY:

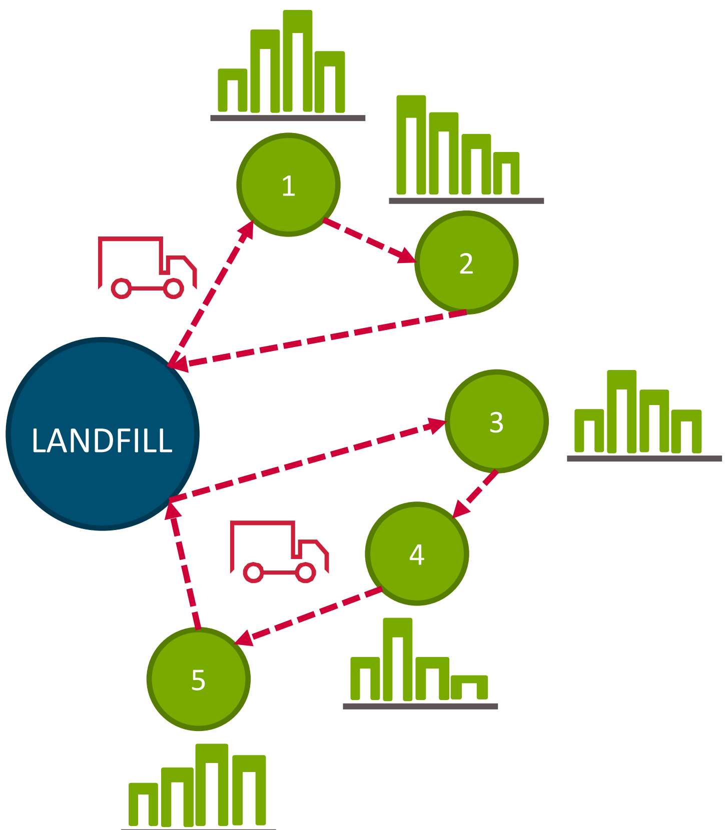
MAXIMUM CAPACITY Q

1. Probability (exceed Q) = 0%
2. Probability (exceed Q) = 25%

TRIP BACK

3. Probability (exceed Q) = 45% — THRESHOLD HIGHER THAN 25%
4. Probability (exceed Q) = 80%
5. Probability(exceed Q) = 100%

Chance Constrained Optimization



GOAL: MINIMIZE THE TOTAL TRAVEL DISTANCE

CONSTRAINT IMPACTED BY UNCERTAINTY:

MAXIMUM CAPACITY Q

1 Probability (exceed Q) = 0%

2 Probability (exceed Q) = 25%

TRIP BACK

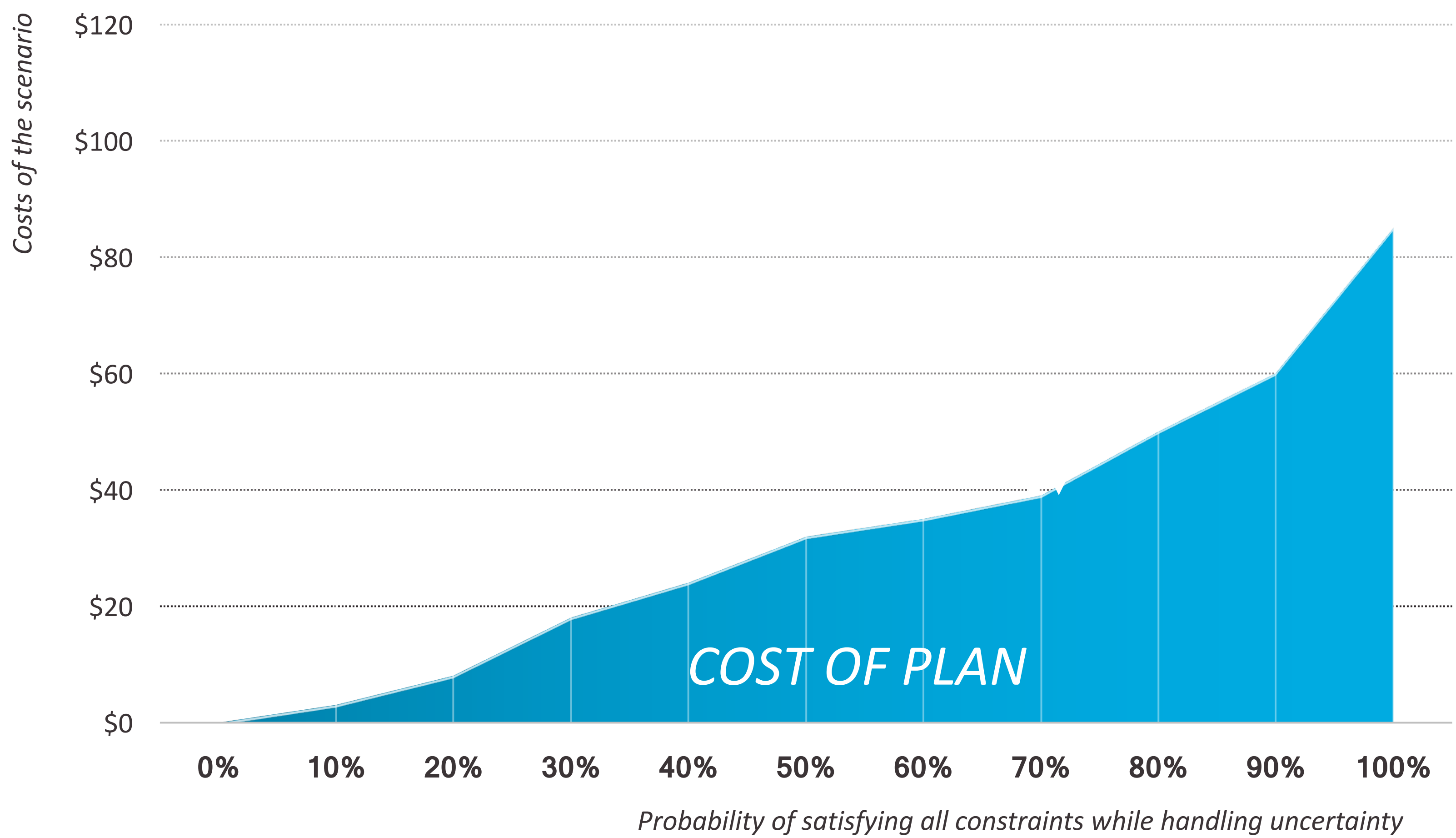
3 Probability (exceed Q) = 0%

4 Probability (exceed Q) = 8%

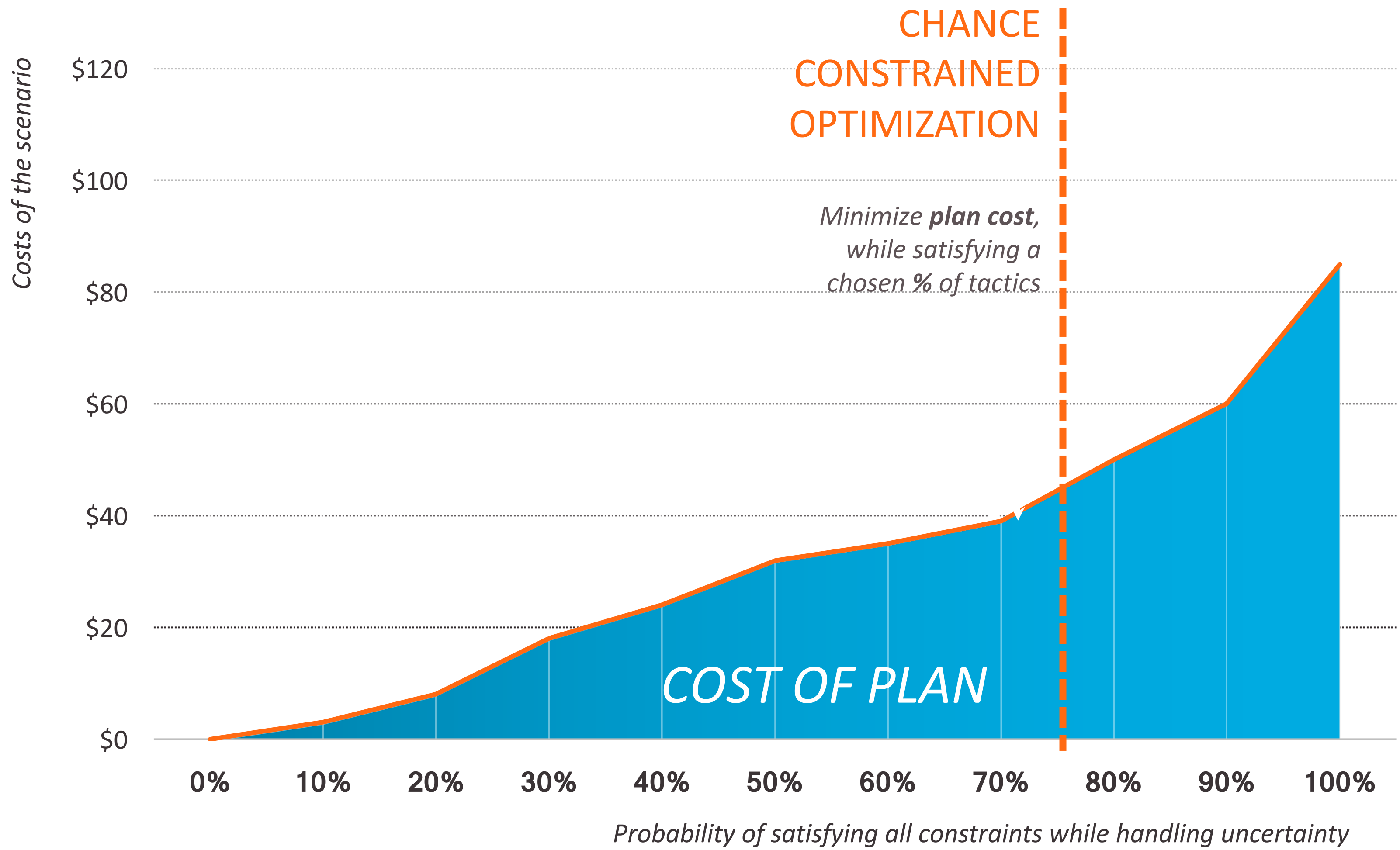
5 Probability(exceed Q) = 24%

TRIP BACK

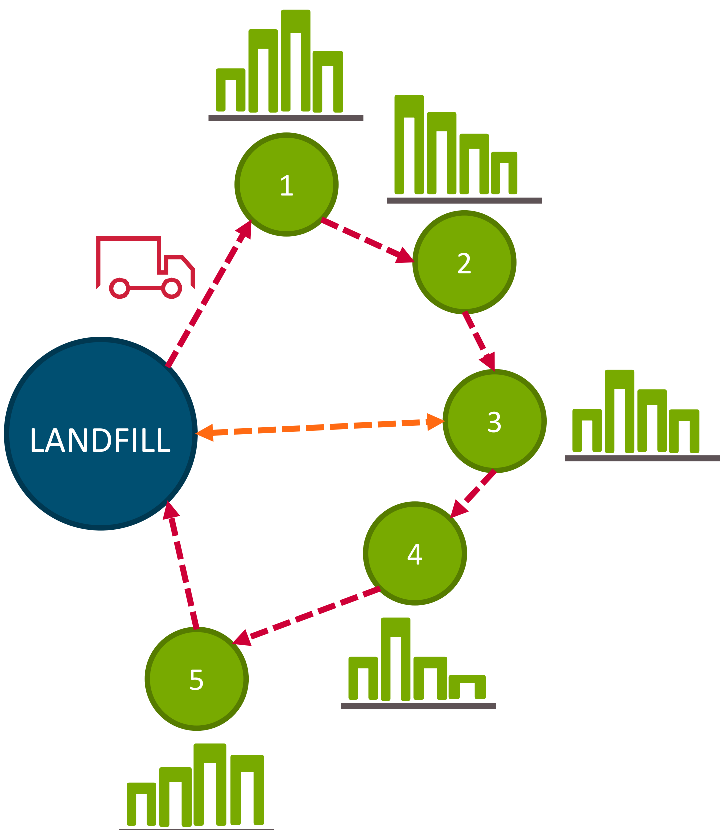
Chance Constrained Optimization



Chance Constrained Optimization



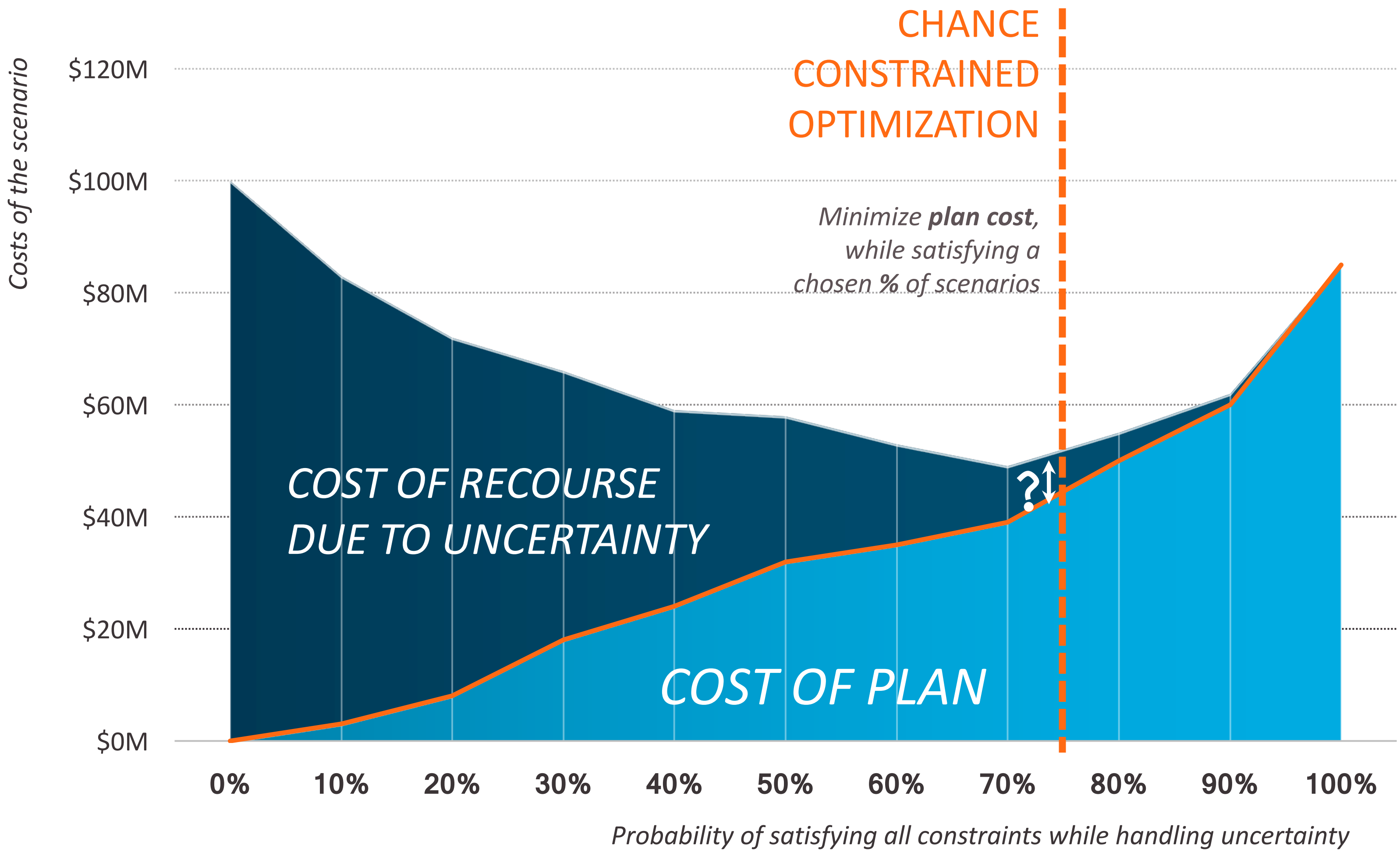
But What Happens When the Plan Fails?



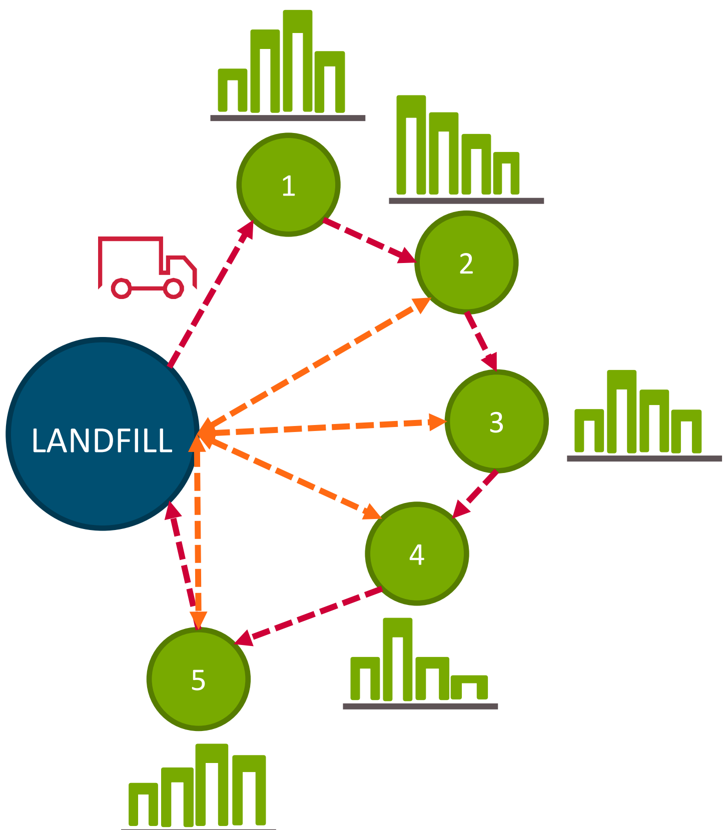
- A recourse action is a wait-and-see decision you can take when you observe the uncertainty.
  - In this case, a trip back and forth to the landfill can be an option if the uncertainty scenario exceeds the capacity.
  - In other cases, recourse actions can be to pay extra hours, short the demand, etc.

- In Chance Constrained Optimization, the recourse plan is not minimized. The plan doesn't tell you what to do and how much it will cost in case of failure.

How to Model Uncertainty?



Stochastic Programming



GOAL: MINIMIZE THE TOTAL TRAVEL DISTANCE

CONSTRAINT IMPACTED BY UNCERTAINTY:

MAXIMUM CAPACITY Q

1 Probability (exceed Q) = 0%

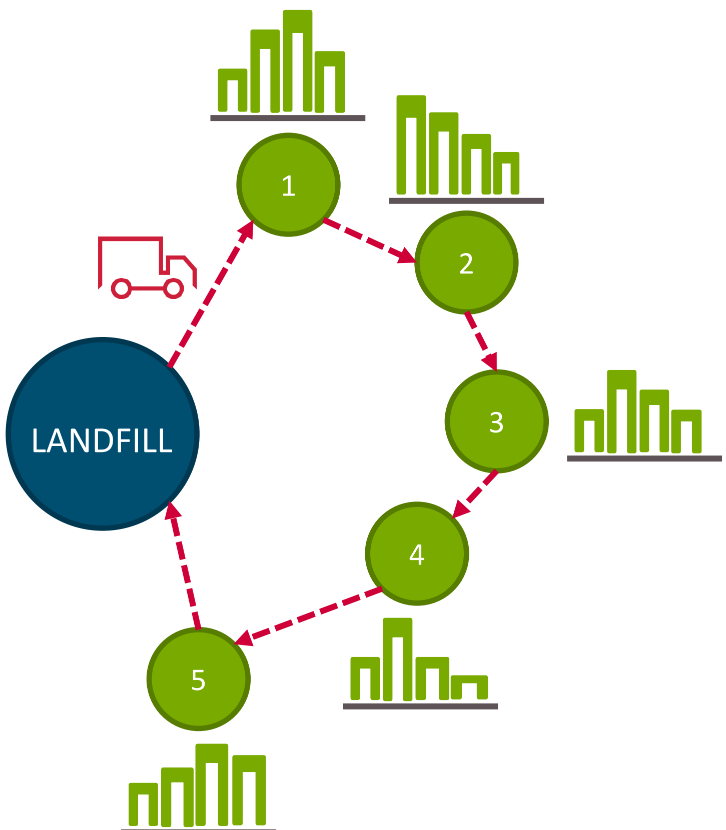
2 Probability (exceed Q) = 25% — *IF FAILURE: RECOURSE 46km*

3 Probability (exceed Q) = 45% — *IF FAILURE: RECOURSE 30km*

4 Probability (exceed Q) = 80% — *IF FAILURE: RECOURSE 12km*

5 Probability(exceed Q) = 100% — *IF FAILURE: RECOURSE 8km*

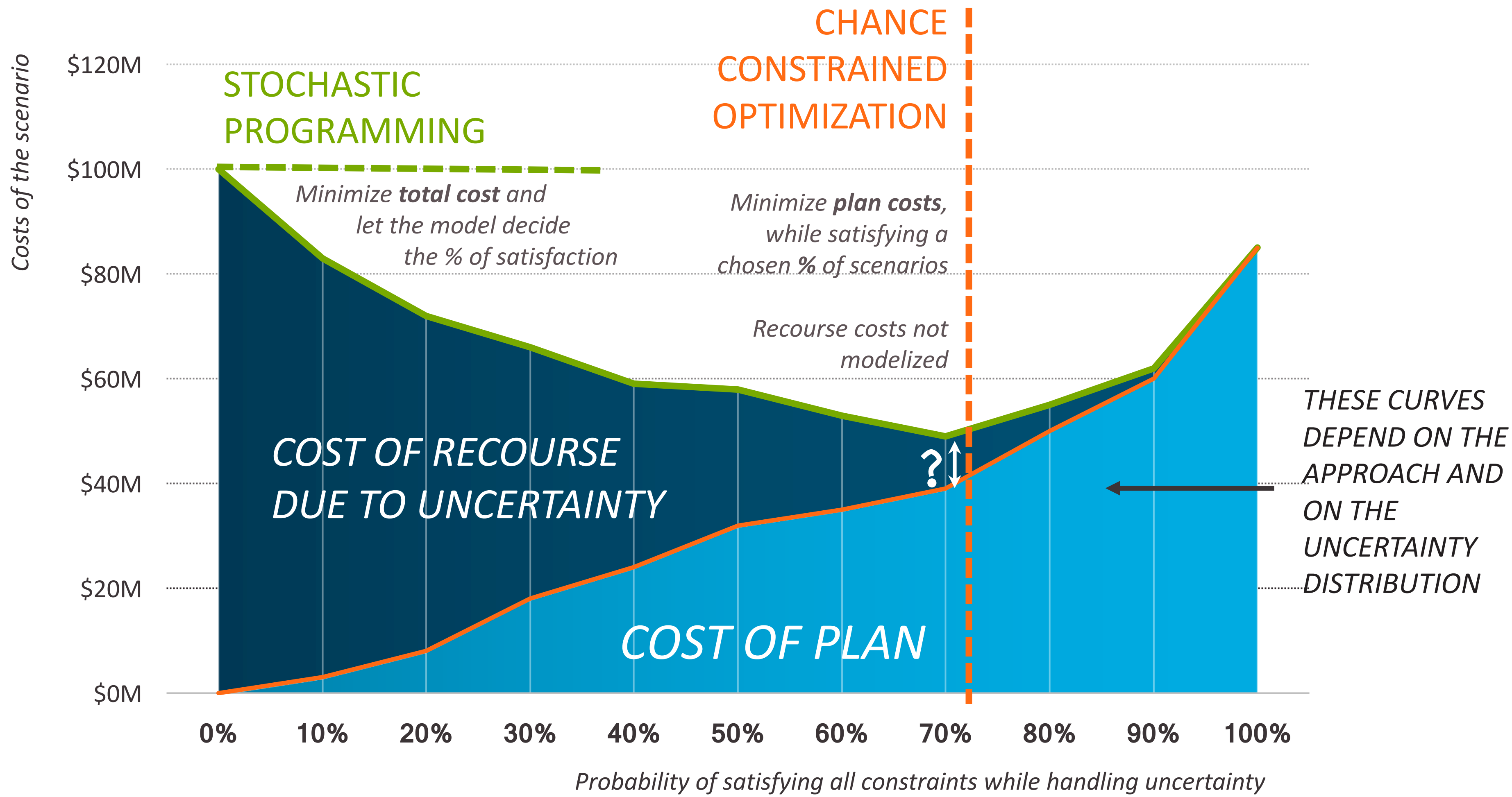
Stochastic Programming



- Optimize the plan cost and the recourse cost at the same time
- Often relies on scenarios generation (sampling) and aims to optimize the expected costs
- Received much attention since 1990s
- Some variants, e.g. Markov decision processes (MDPs), binary scenario tree, etc.

+ Pros
  + The recourse plan cost is handled
- Cons
  - Tricky to compute the recourse cost in some cases

How to Model Uncertainty?



What Do We Know About Uncertainty?

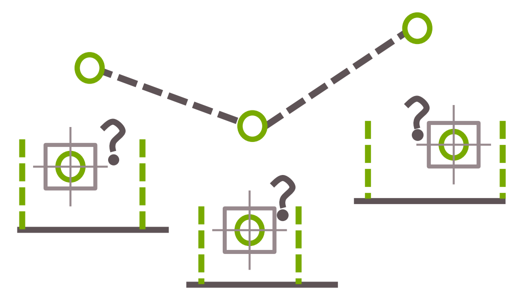
ÉCOLE POLYTECHNIQUE MONTRÉAL

UNCERTAINTY

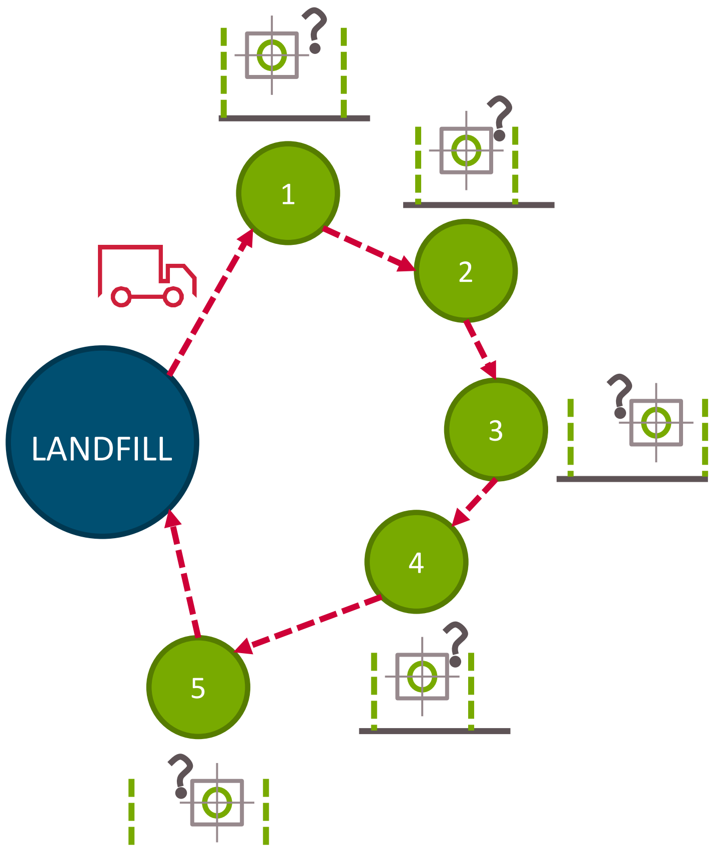
- Definition: The probabilistic model is known, but the realizations of the random variables are unknown (Ellsberg, 1961)

AMBIGUITY (UNCERTAIN UNCERTAINTY)

- Definition: The probability model itself is unknown



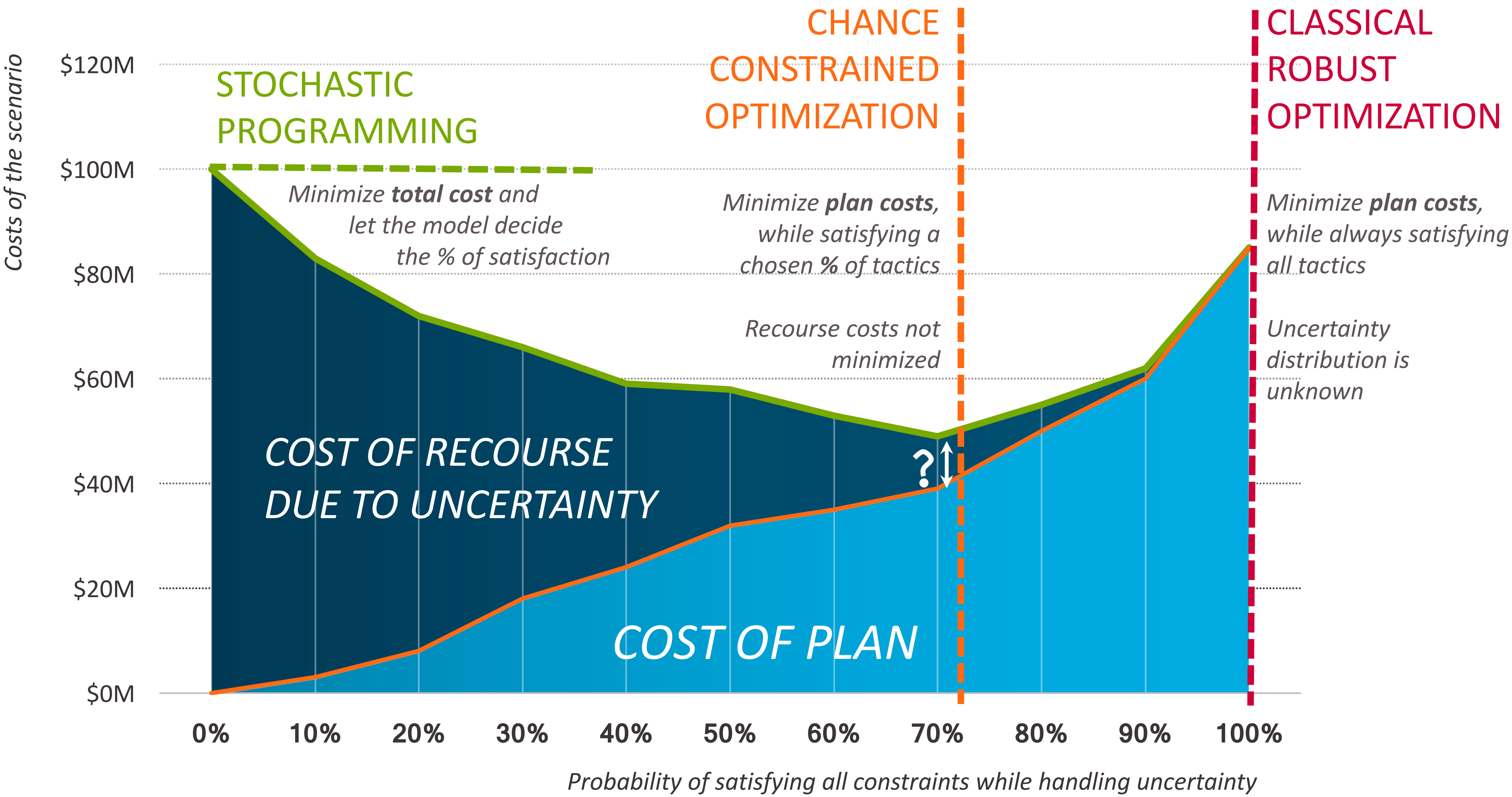
Robust Optimization



- When the uncertainty distribution is unknown, robust optimization will ensure that the model can still produce feasible plans.
- Classical robust optimization aims to ensure worst possible outcome

+ Pros
  + Doesn’t need a lot of information on uncertainty
  + Scalability
- Cons
  - Could still be conservative in some case
  - Works when plan adaptation involves only quantities (like fulfillment quantities).

How to Model Uncertainty?



Application

> INTEGRATED PRODUCTION-DISTRIBUTION PLANNING

> VEHICLE ROUTING WITH STOCHASTIC DEMAND

Integrated Production Distribution Planning

Production → Inventory → Distribution → Inventory → Demand

BATCH:
50 units/batch

COST:
$100/batch

CAPACITY:
200 units/truck

COST:
$500/truck

INVENTORY:
$1/unit

STOCK-OUT:
$20/unit

SALVAGE COST:
$10/unit

UNCERTAINTY IN DEMAND: ± 40%

DECISIONS:

- ✓ Nb of production batches each week
- ✓ Nb of committed trucks each week for each customer
- ✓ Replenishment quantity each week for each customer

Integrated Production Distribution Planning

BATCH:
50 units/batch

COST:
$100/batch

CAPACITY:
200 units/truck

COST:
$500/truck

INVENTORY:
$1/unit

STOCK-OUT:
$20/unit

SALVAGE COST:
$10/unit

UNCERTAINTY IN DEMAND: ± 40%

TESTED APPROACHES:

- Deterministic Model – Using the average demand
- Deterministic Model – Using the maximum demand
- Stochastic Optimization

Integrated Production Distribution Planning

DETERMINISTIC APPROACHES:

- Deterministic Model – Using the average demand

- Deterministic Model – Using the maximum demand

Integrated Production Distribution Planning

STOCHASTIC APPROACH

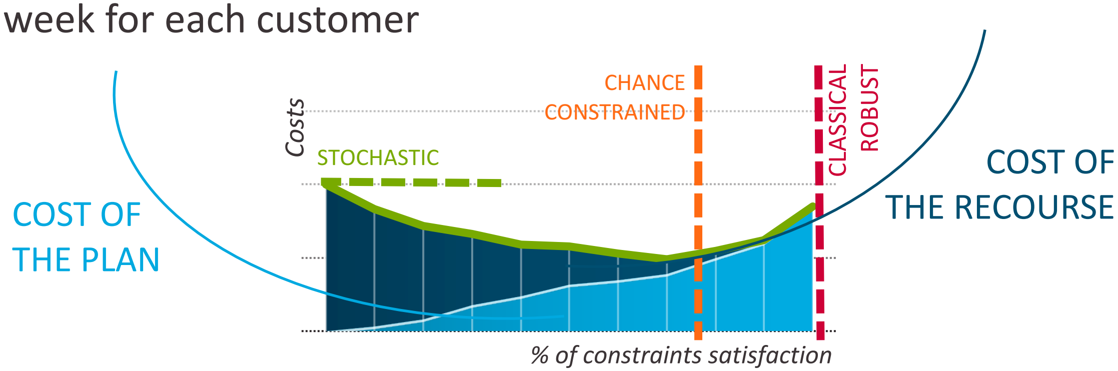
DECISIONS:

HERE-AND-NOW

- ✓ Nb of production batches each week
- ✓ Nb of committed trucks each week for each customer

WAIT-AND-SEE (Recourse action)

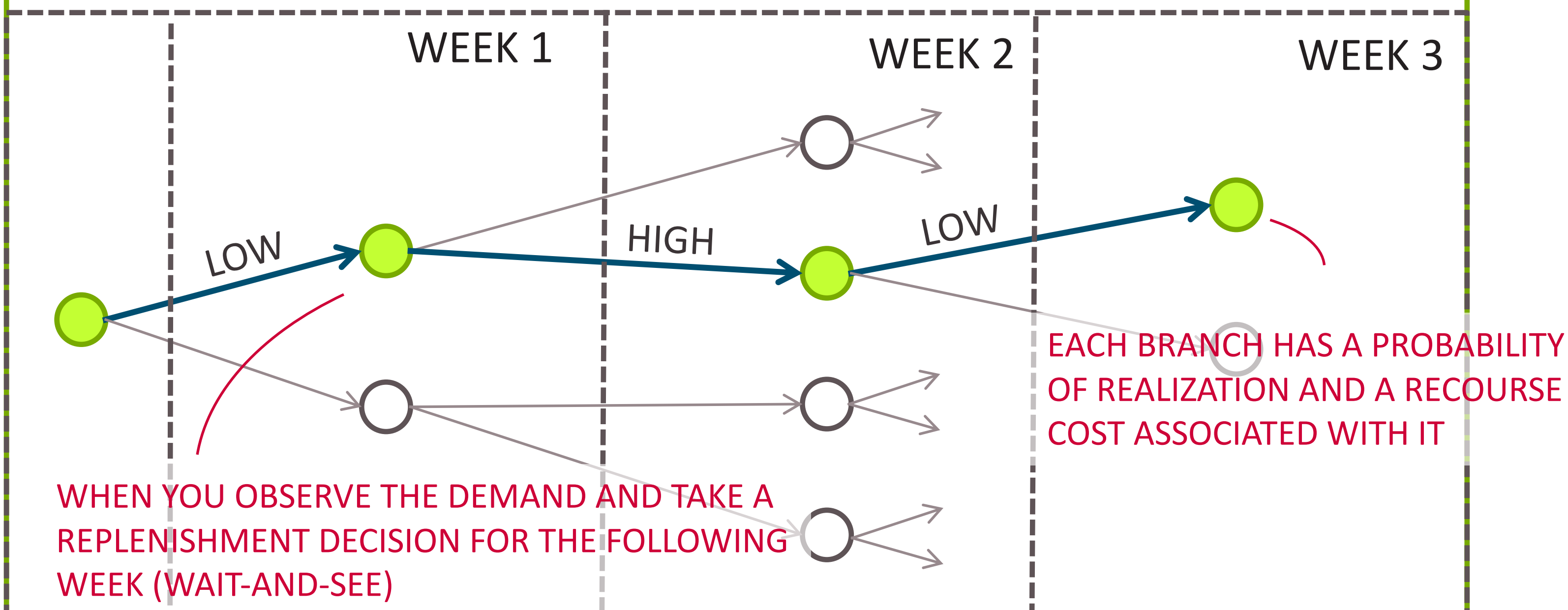
- ✓ Replenishment quantity each week for each customer



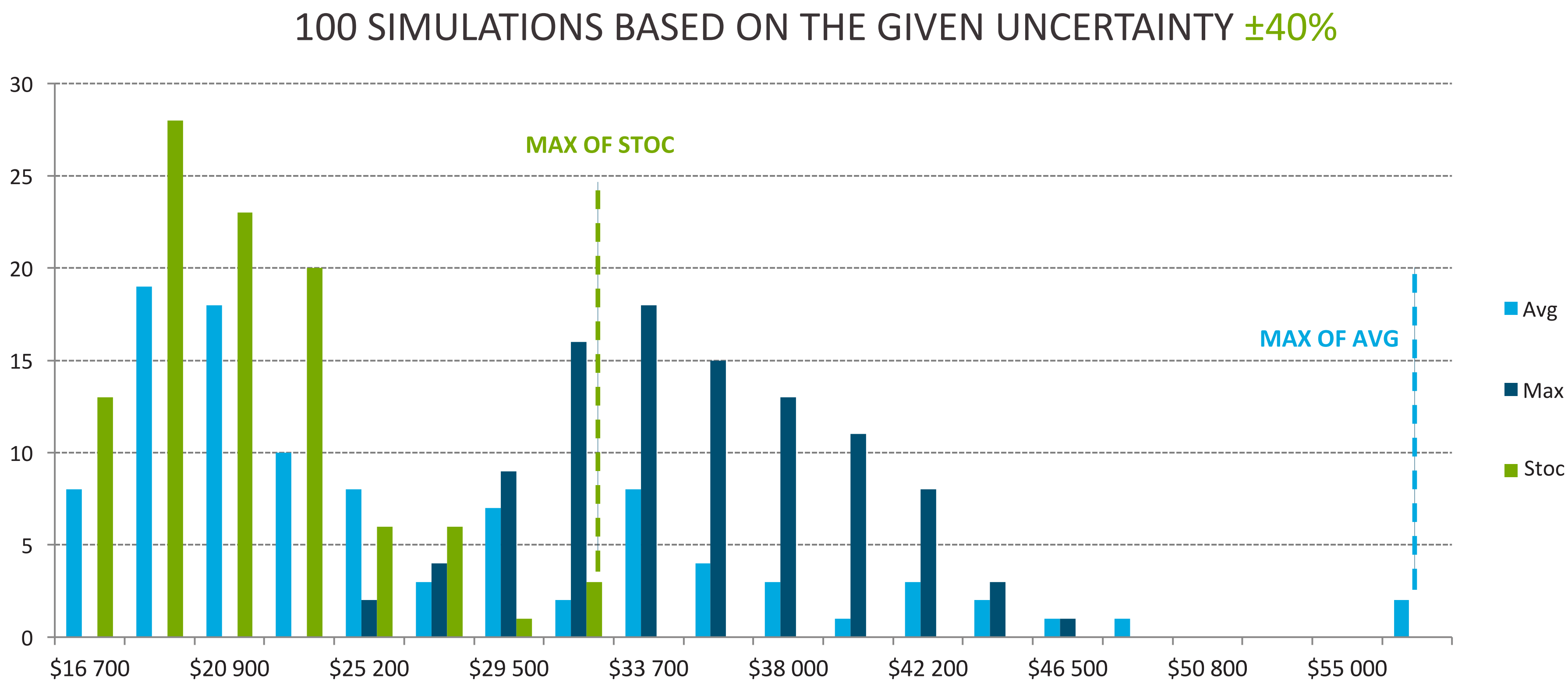
Integrated Production Distribution Planning

STOCHASTIC APPROACHES

– Stochastic Optimization



Integrated Production Distribution Planning



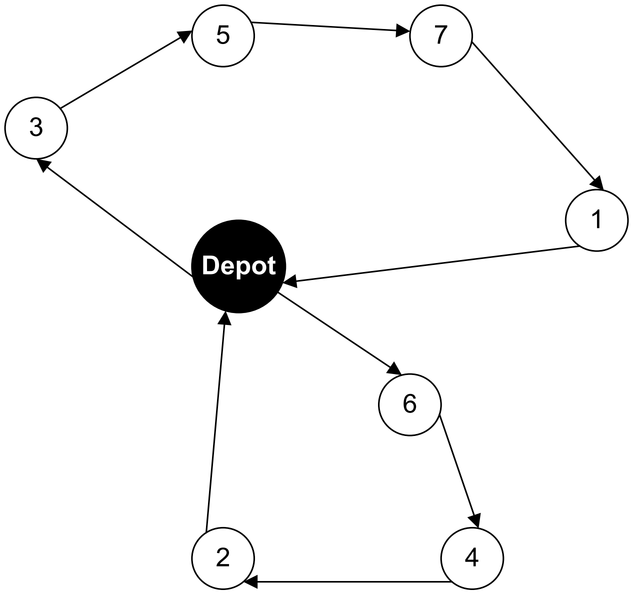
	AVG	MAX	STOC
Total cost (Avg)	$26,364	$35,301	$21,255
Total cost (Max)	$58,120	$46,503	$31,388
Standard Dev	$9,130	$4,635	$3,304
%Diff (Avg)	**24%**	**65%**	
%Diff (Max)	**85%**	**48%**	

Agenda

- The vehicle routing problem with stochastic demands and duration constraints (VRPSDDC)
  - Chance constraint programming formulation
  - Stochastic programming with recourse formulation
- GRASP + HC
  - General structure
  - Components
- Computational experiments
  - VRPSD
  - VRPSDDC
- Conclusions and perspectives

The VRP with stochastic demands

Definition: classical setting



- n customers $\{1, \ldots, v, \ldots, n\}$
- **Independent** random demands ξ_v ($\bar{\xi}_v \leq Q$)
- Unlimited fleet of vehicles with fixed and limited capacity Q
- Maximum expected load for each vehicle (i.e., $\sum_{v \in r} E[\xi_v] \leq Q$, where *r* is the route)
- Select a minimal-duration set of routes to service the demands of every customer

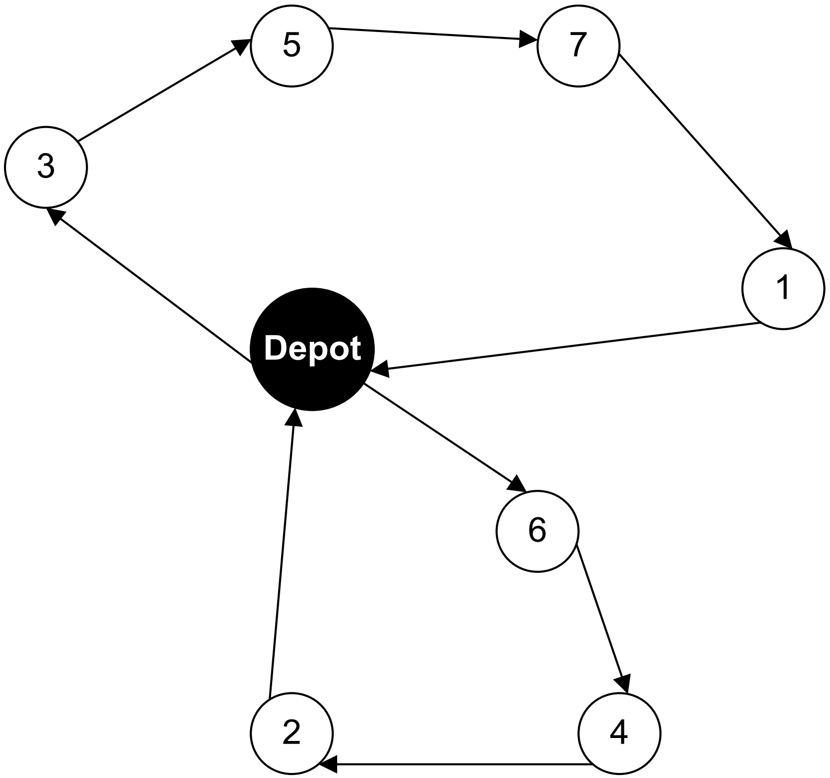
Laporte et al. (2002), Chrystiansen and Lysgaard (2007), Gendreau and Rei (2010), Goodson (2012)

The VRPSD

Modeling: two-stage stochastic programming

The VRPSD

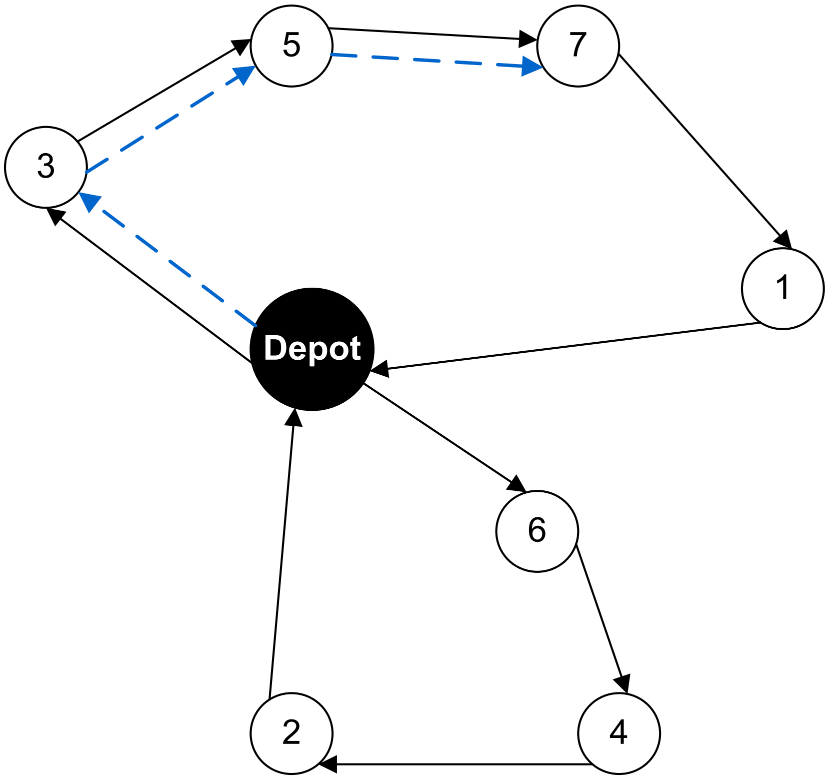
Modeling: two-stage stochastic programming



- Stage I: plan a set of routes $\mathcal{R}$

The VRPSD

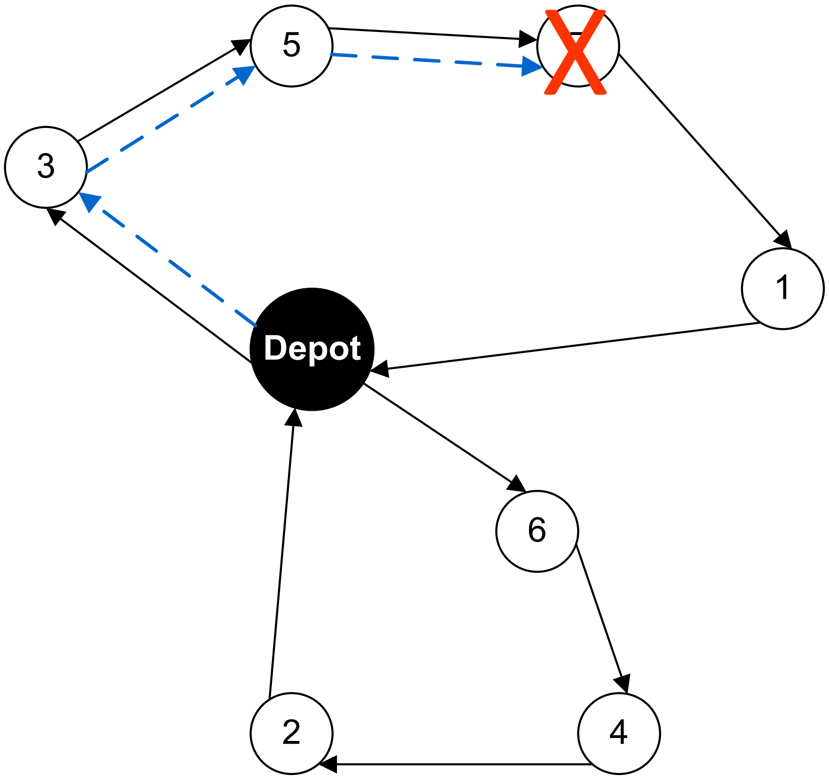
Modeling: two-stage stochastic programming



- Stage I: plan a set of routes $\mathcal{R}$
- Stage II: execute the planned routes

The VRPSD

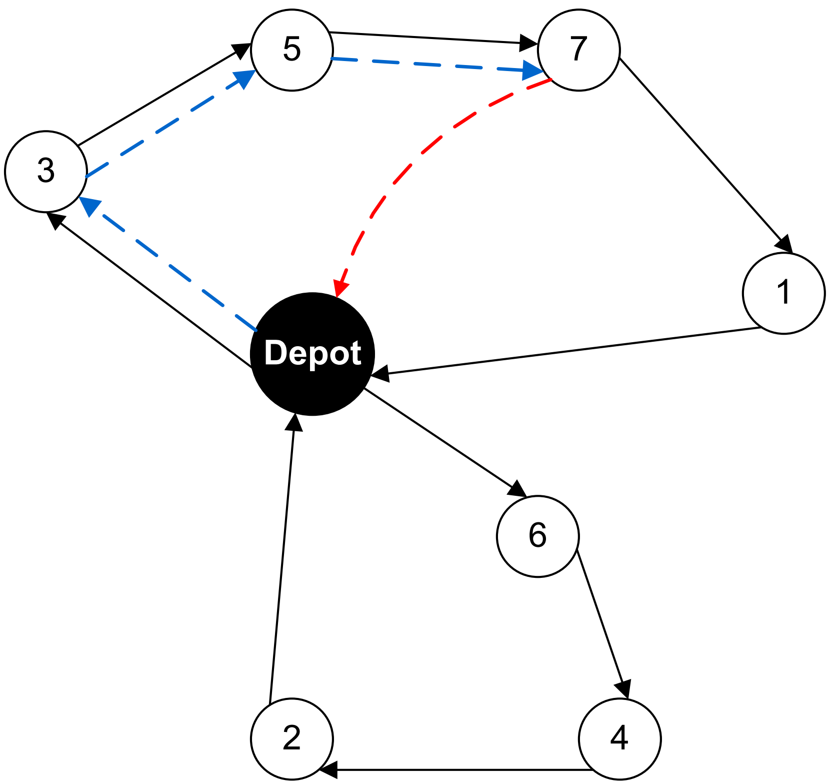
Modeling: two-stage stochastic programming



- Stage I: plan a set of routes $\mathcal{R}$
- Stage II: execute the planned routes
  - Route failure: the load exceeds Q

The VRPSD

Modeling: two-stage stochastic programming

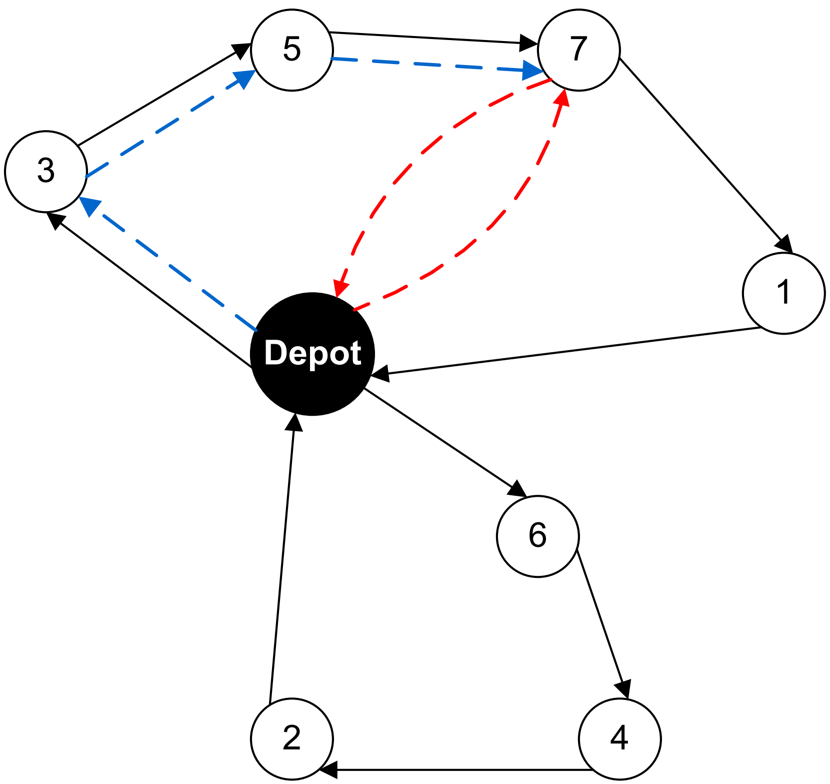


- Stage I: plan a set of routes $\mathcal{R}$
- Stage II: execute the planned routes
  - Route failure: the load exceeds Q
  - Recourse action: trip back to the depot

The VRP with stochastic demands

The VRPSD

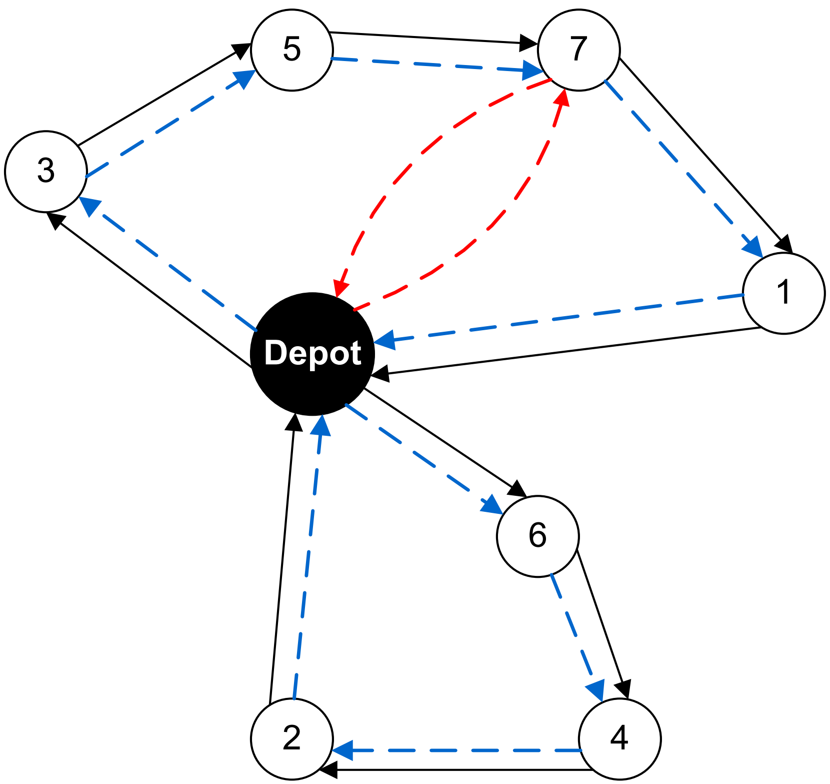
Modeling: two-stage stochastic programming



- Stage I: plan a set of routes $\mathcal{R}$
- Stage II: execute the planned routes
    - Route failure: the load exceeds Q
    - Recourse action: trip back to the depot

The VRPSD

Modeling: two-stage stochastic programming

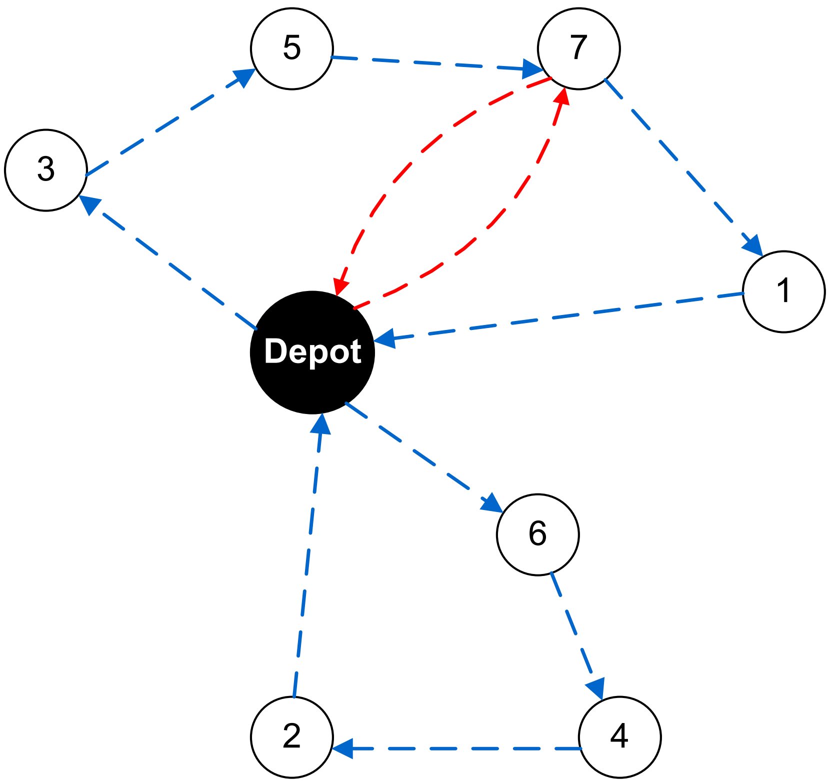


- Stage I: plan a set of routes $\mathcal{R}$
- Stage II: execute the planned routes
  - Route failure: the load exceeds Q
  - Recourse action: trip back to the depot
  - Resume route as planned

See also: alternative recourse actions

The VRPSD

Modeling: two-stage stochastic programming



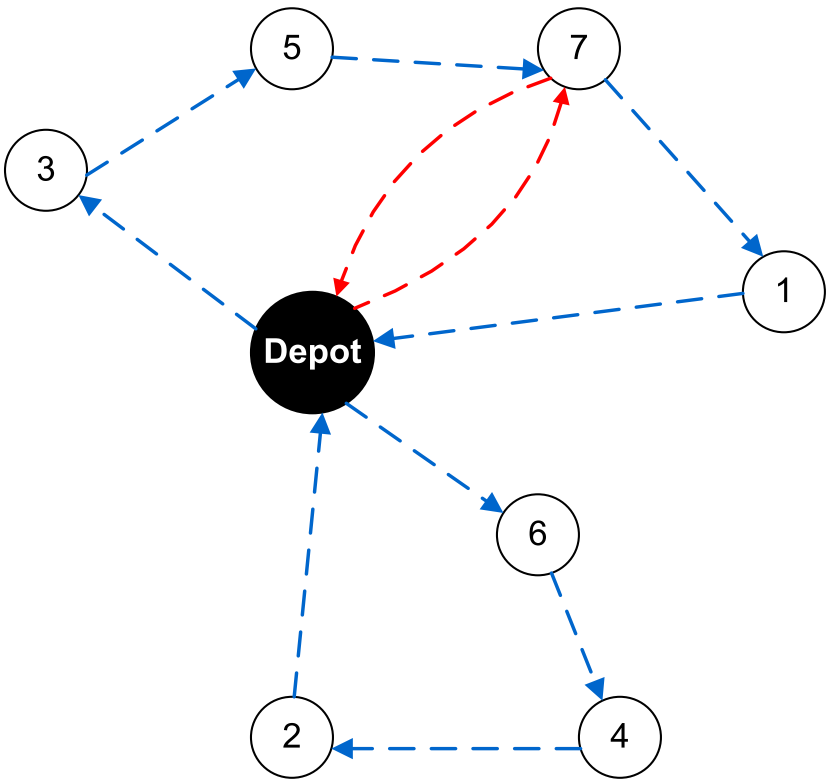
Total traveled distance of the transportation plan:

$$C(\mathcal{R}) = \sum_{r \in \mathcal{R}} C_r$$

Total distance of route r

The VRPSD

Modeling: two-stage stochastic programming



Total traveled distance of the transportation plan:

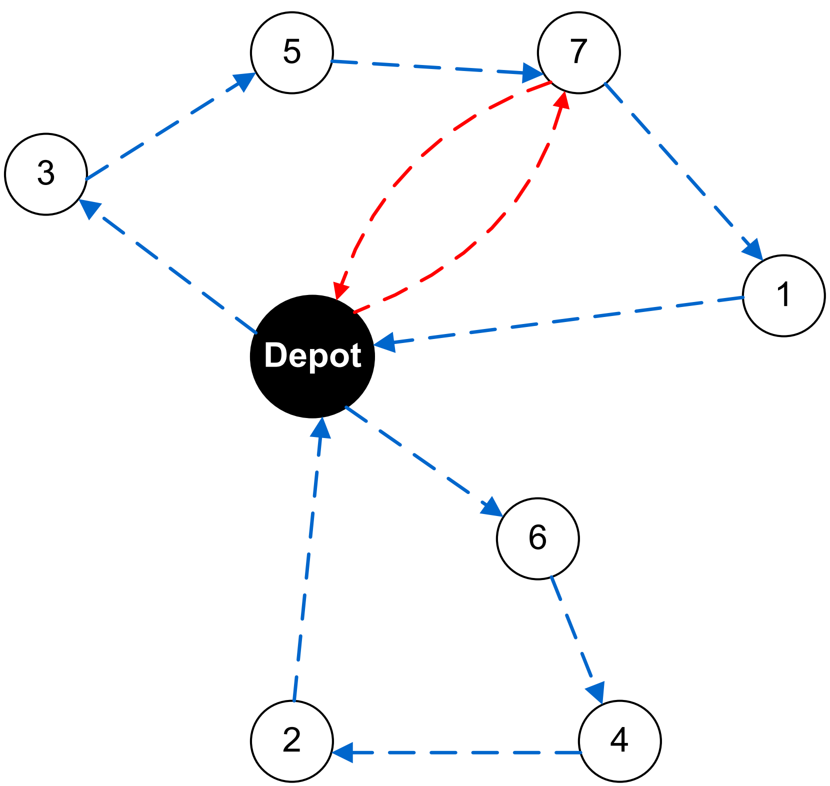
$$C(\mathcal{R}) = \sum_{r \in \mathcal{R}} C_r = \sum_{r \in \mathcal{R}} l_r + \sum_{r \in \mathcal{R}} G_r\left(\vec{\xi}\right)$$

Planned distance

Distance due to route failures

The VRPSD

Modeling: two-stage stochastic programming



Total traveled distance of the transportation plan:

$$C(\mathcal{R}) = \sum_{r \in \mathcal{R}} C_r$$

$$= \sum_{r \in \mathcal{R}} l_r + \sum_{r \in \mathcal{R}} G_r\left(\vec{\xi}\right)$$

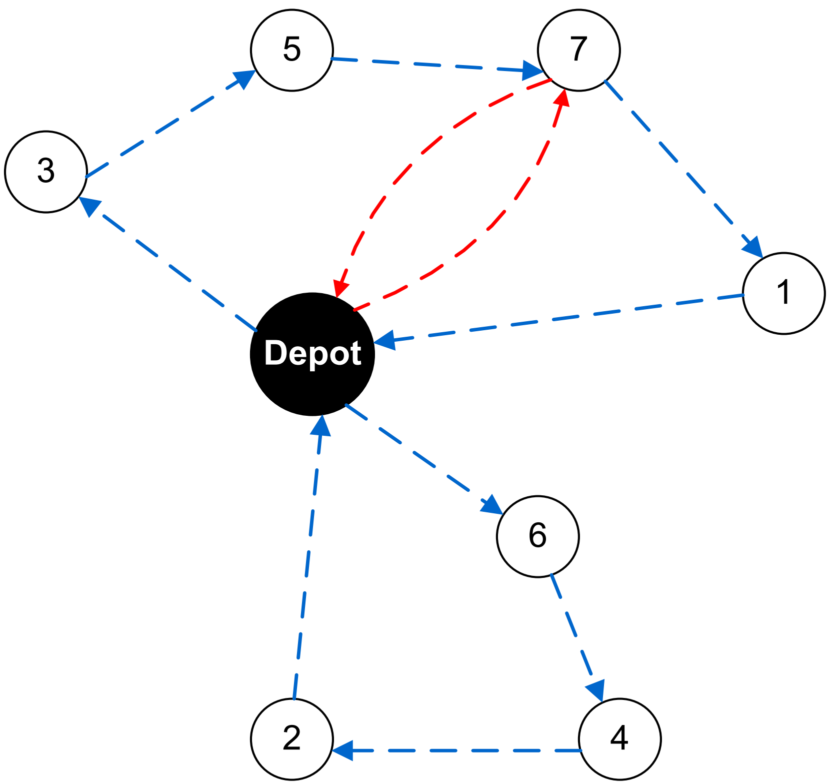
Random variable

- Mean: $\mu_{C(\mathcal{R})}$
- Standard deviation: $\sigma_{C(\mathcal{R})}$

The VRP with stochastic demands

The VRPSD

Modeling: two-stage stochastic programming



Objective: minimize

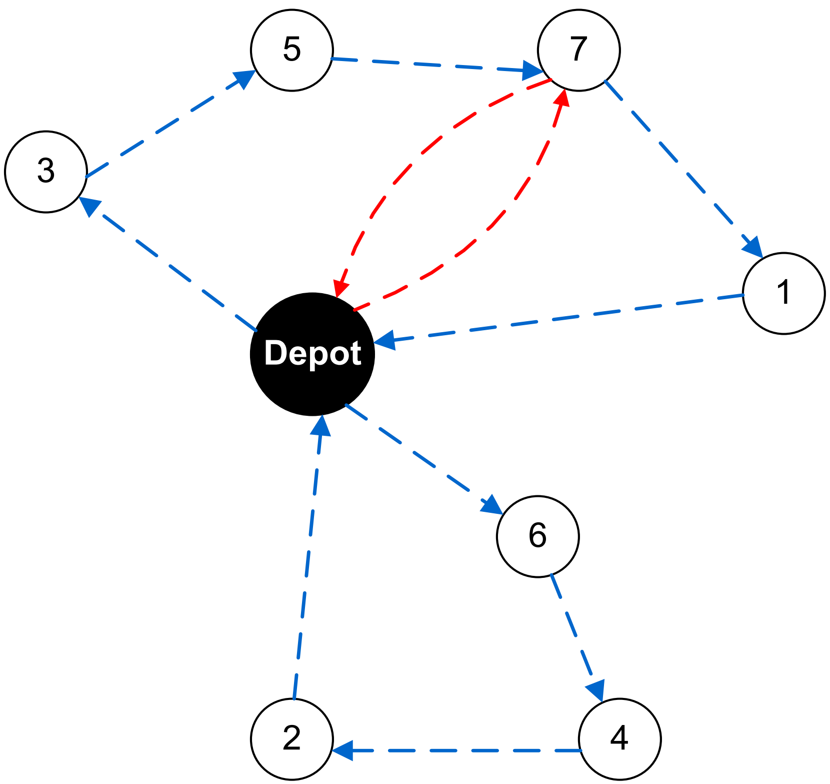
$$E\left[C(\mathcal{R})\right] = \sum_{r\in\mathcal{R}} l_r + \sum_{r\in\mathcal{R}} E\left[G_r\left(\vec{\xi}\right)\right]$$

$$E\left[G_r(\vec{\xi})\right] = \sum_{i\in r} 2 \times d_{v_i,0} \times Pr(v_i)$$

Probability that a route failure occurs while visiting node **(Dror et al. 1989, Laporte et al. 2002)**

The VRPSD

Modeling: two-stage stochastic programming



Objective: minimize

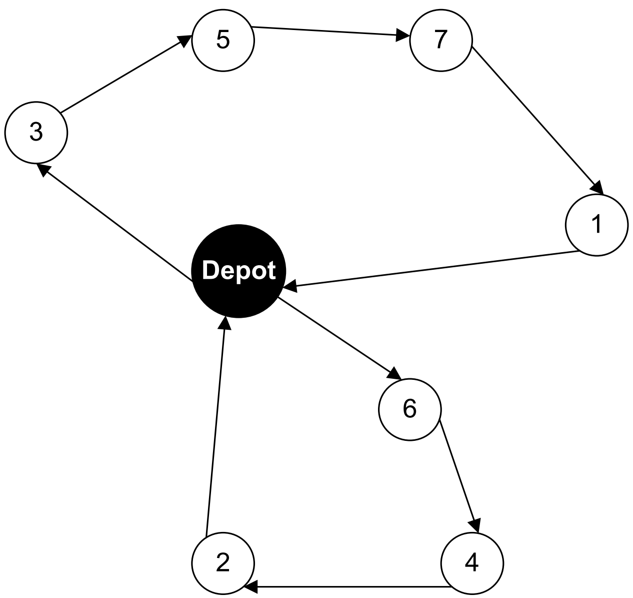
$$E\left[C(\mathcal{R})\right] = \sum_{r \in \mathcal{R}} l_r + \sum_{r \in \mathcal{R}} E\left[G_r\left(\vec{\xi}\right)\right]$$

Subject to:

$$\sum_{i \in r} E\left[\xi_i\right] \leq Q \quad \forall\ r \ \in \mathcal{R}$$

The VRPSD and duration constraints

Definition



- n customers $\{1, \ldots, v, \ldots, n\}$
- **Independent** random demands ξ_v ($\bar{\xi}_v \leq Q$)
- Unlimited fleet of vehicles with fixed and limited capacity Q
- Maximum expected load for each vehicle (i.e., $\sum_{v \in r} E[\xi_v] \leq Q$, where *r* is the route)
- Select a minimal-duration set of routes to service the demands of every customer
- **Maximum route duration *L***

See also: other extensions

Dealing with duration constraints on the VRPSD

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Challenge: the duration of a route is a random variable which value is only known when the vehicle returns to the depot

Dealing with duration constraints on the VRPSD

ÉCOLE POLYTECHNIQUE MONTRÉAL

> Challenge: the duration of a route is a random variable which value is only known when the vehicle returns to the depot

> What is in the literature:

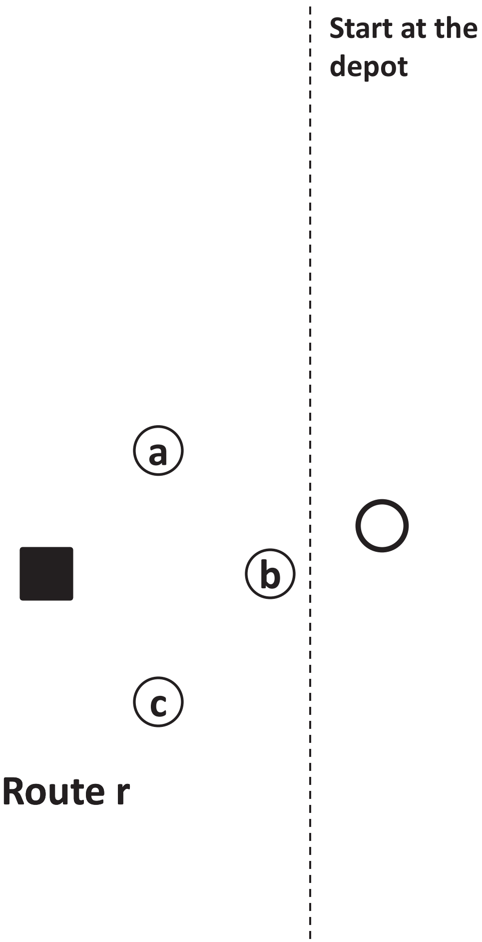
- Set a duration constraint over the expected duration of a route
  (Yang et al. 2000, Mendoza et al. 2010, 2011)
- Penalize duration excess on a second objective function
  (Tan et al. 2007)
- Set a hard constraint on the maximum duration of each route
  (Erera et al. 2012)

Dealing with duration constraints on the VRPSD

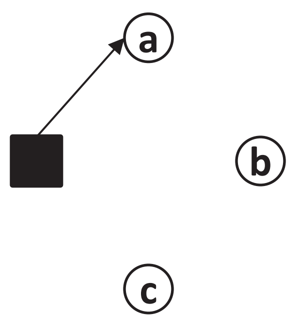
ÉCOLE POLYTECHNIQUE MONTRÉAL

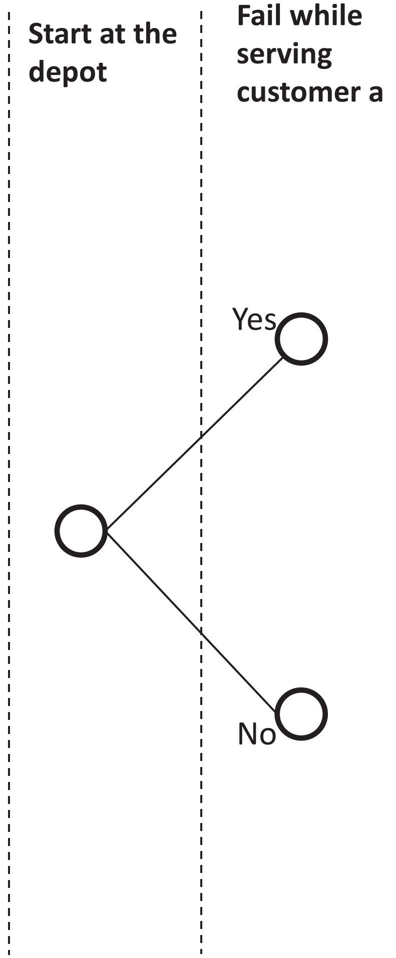
- Challenge: the duration of a route is a random variable which value is only known when the vehicle returns to the depot
- What is in the literature:
  - Set a duration constraint over the expected duration of a route (Yang et al. 2000, Mendoza et al. 2010, 2011)
  - Penalize duration excess on a second objective function (Tan et al. 2007)
  - Set a hard constraint on the maximum duration of each route (Erera et al. 2012)
- Two alternative approaches:
  - Chance constraint (CC)
  - Stochastic programming with recourse (DR)

Definition: route duration profile

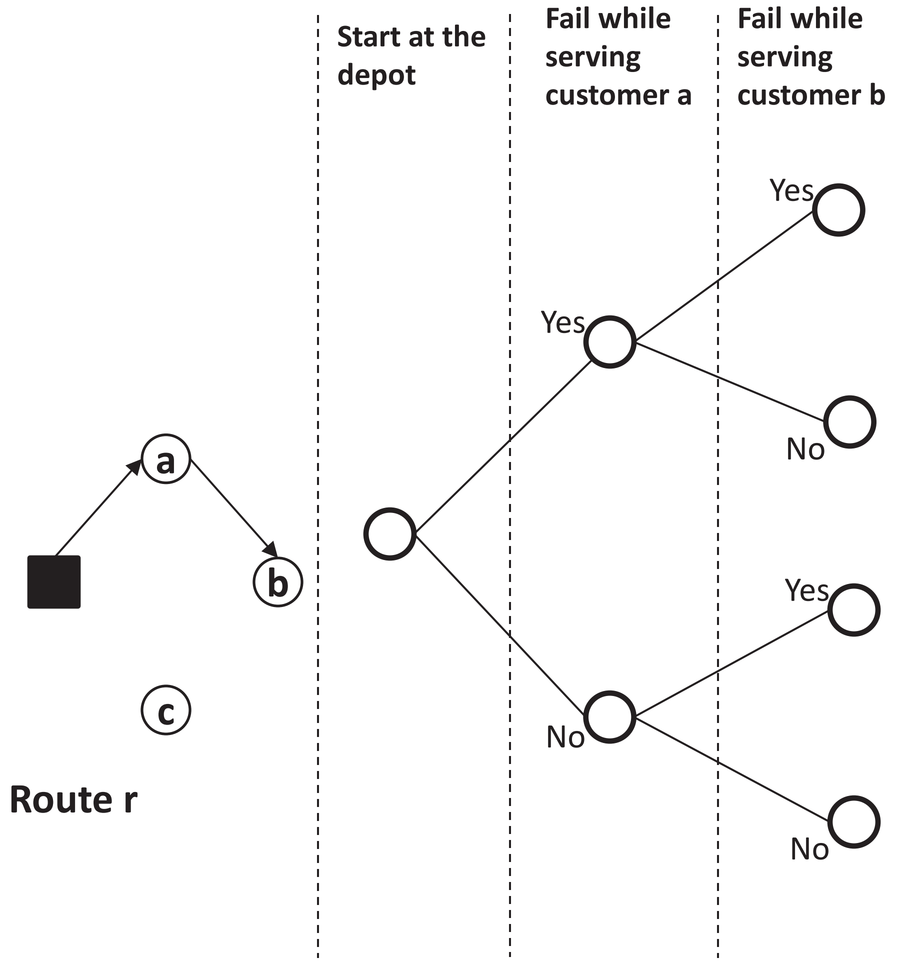


Definition: route duration profile

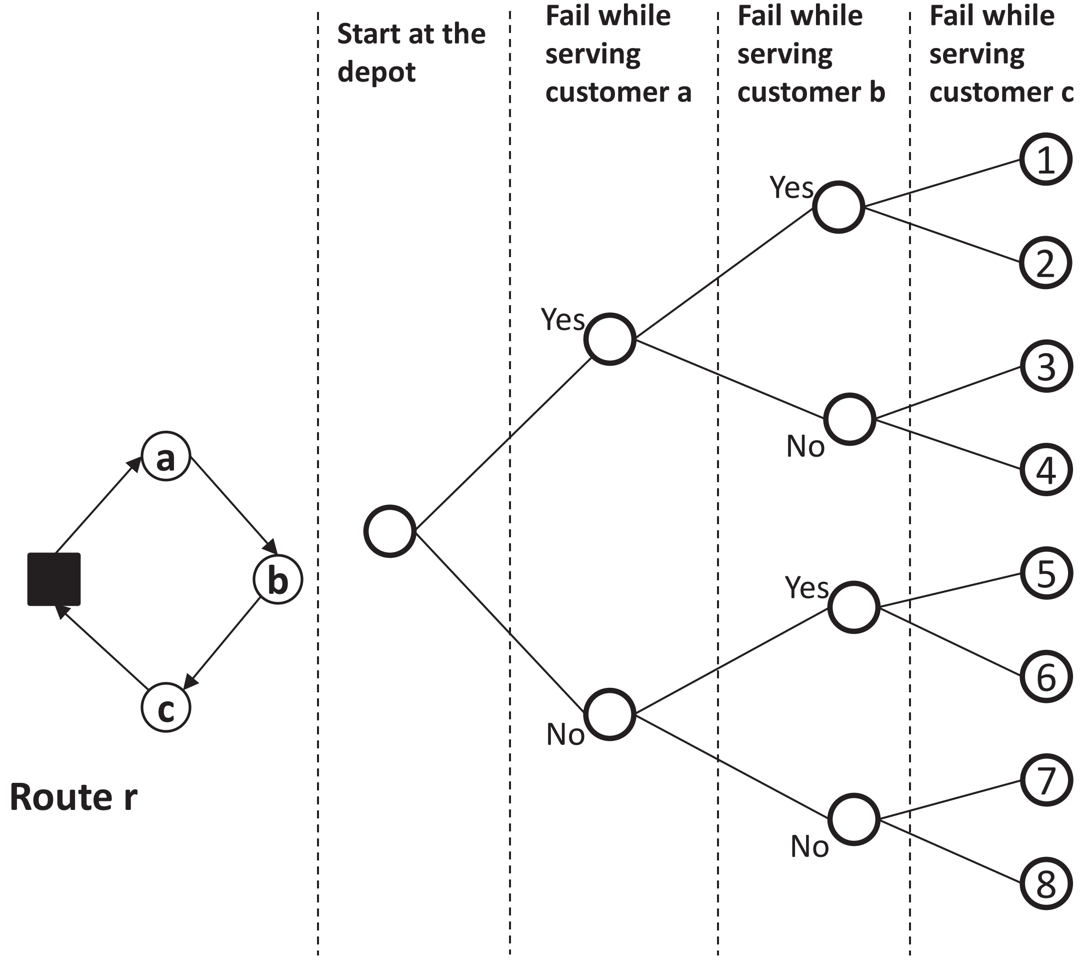




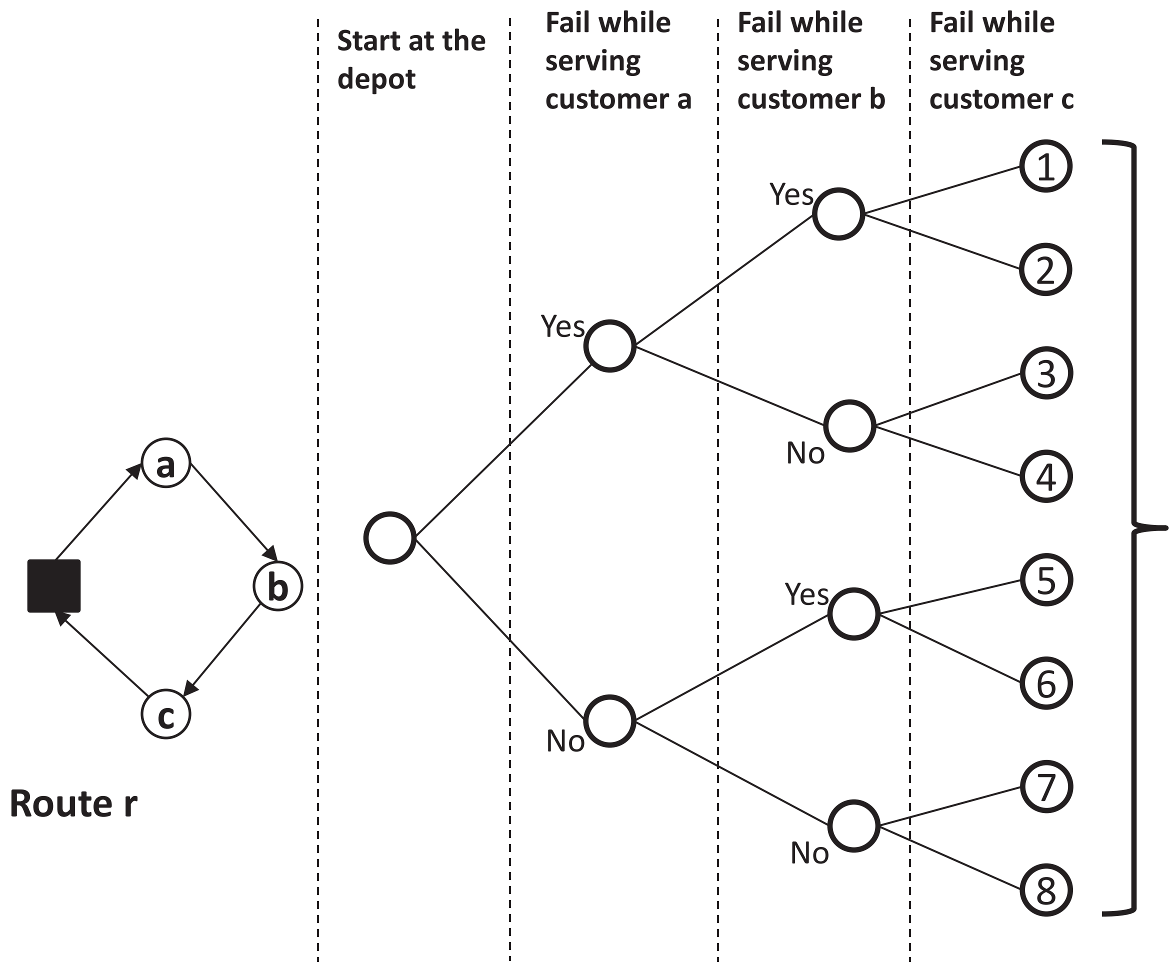
Definition: route duration profile



Definition: route duration profile

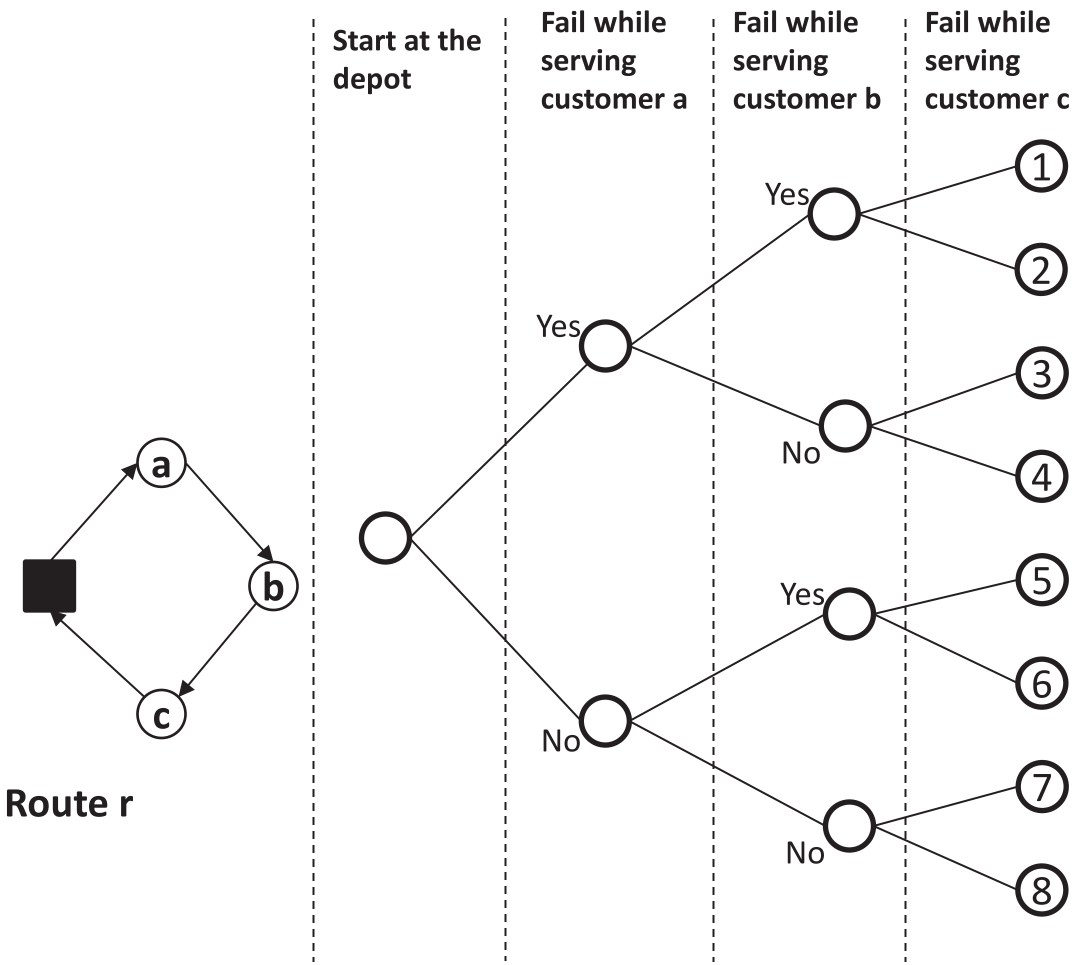


Definition: route duration profile



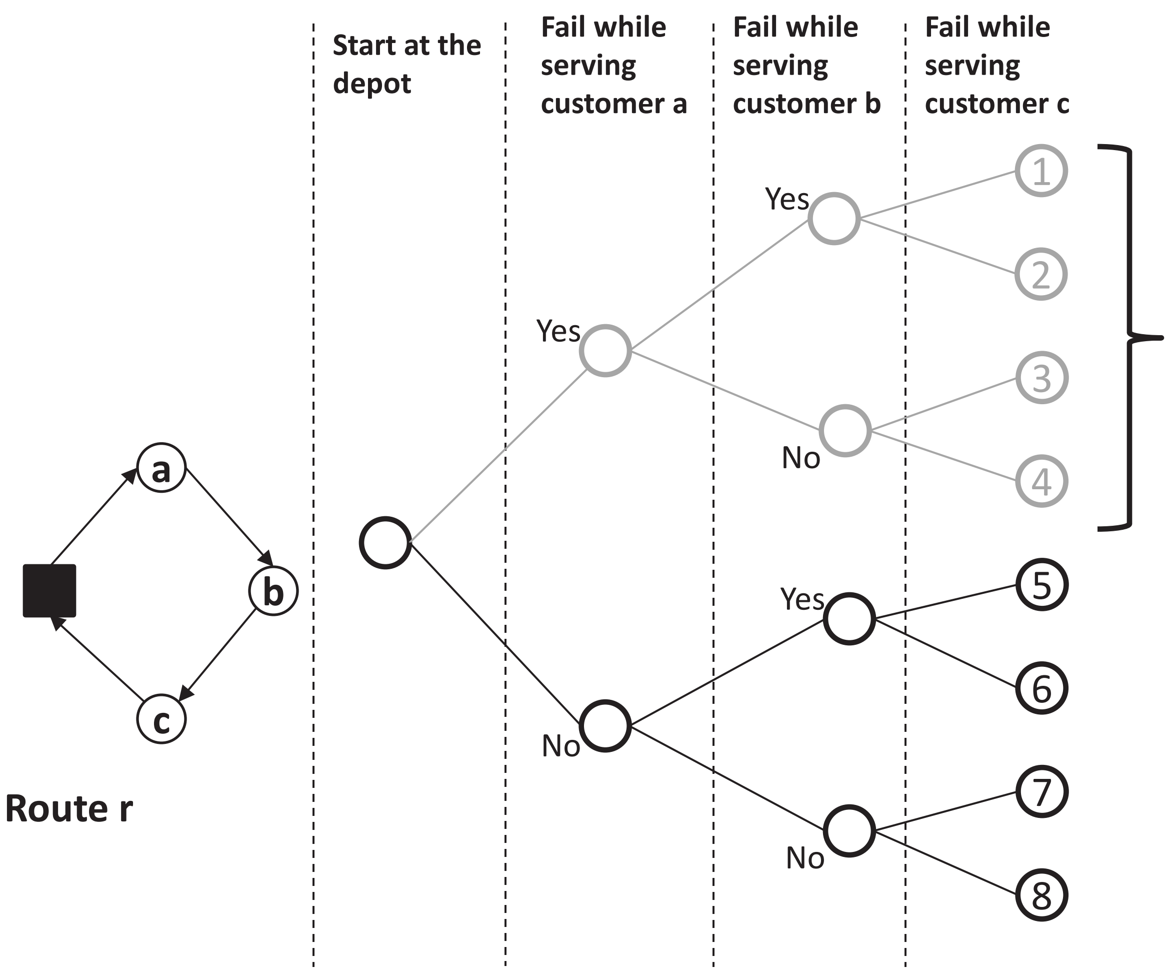
$\mathcal{P}_r$: Set of duration profiles of route r

Definition: route duration profile



- Each node in the last level of the tree represents a possible output for the duration of route r (duration profile)
- Let $\mathcal{P}_r$ denote the set of all possible length profiles of r and let $C(p) | p \in \mathcal{P}_r$ be the length of profile p.
- Knowing the probability of having a failure, due to the capacity constraint, while servicing customer in position *i* of the route (i.e., $\Pr(i)$) we can easily compute the probability of observing a given profile (i.e., $\Pr(p) \mid p \in \mathcal{P}_r$)

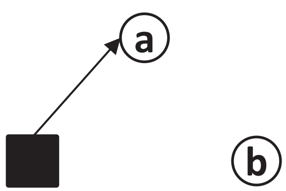
Definition: route duration profile



Elimination of the upper branch because $\bar{\xi}_v \leq Q$

Route duration profile

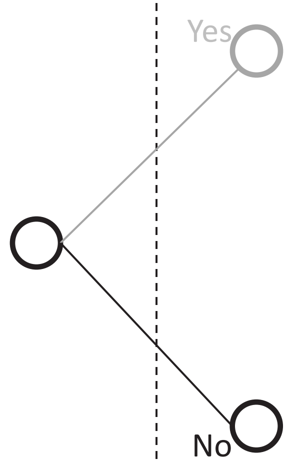
Example



Route r

Start at the depot

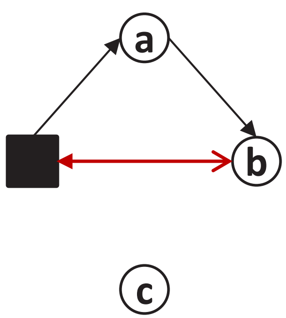
Fail while serving customer a



$$\Pr(p) = 1 - Pr(a)$$
$$C(p) = d_{(0,a)}$$

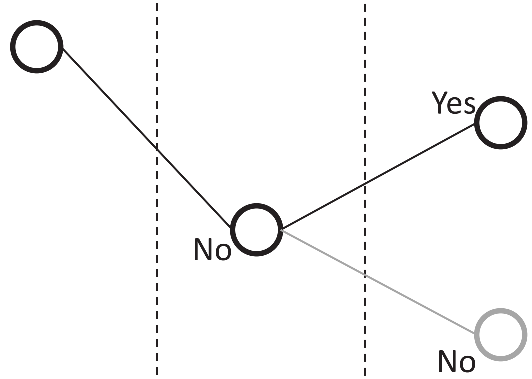
Route duration profile

Example



Route r

Start at the depot | **Fail while serving customer a** | **Fail while serving customer b**

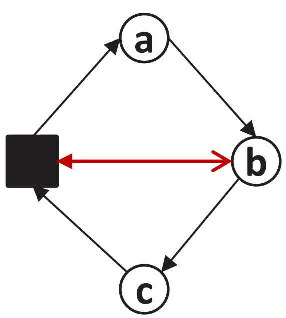


$$Pr(p) = (1 - Pr(a))\,Pr(b)$$

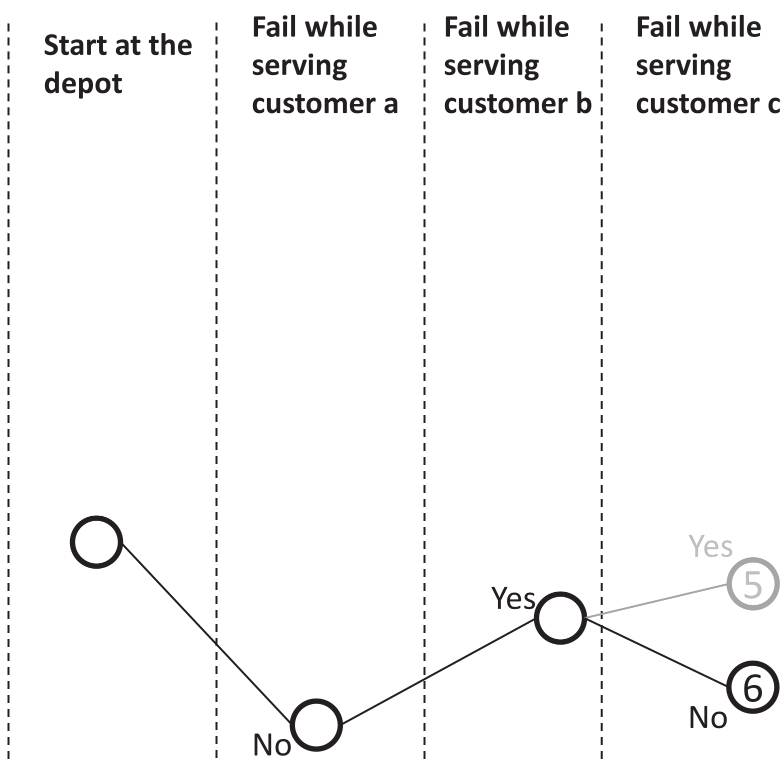
$$C(p) = d_{(0,a)} + d_{(a,b)} + 2d_{(b,0)}$$

Route duration profile

Example



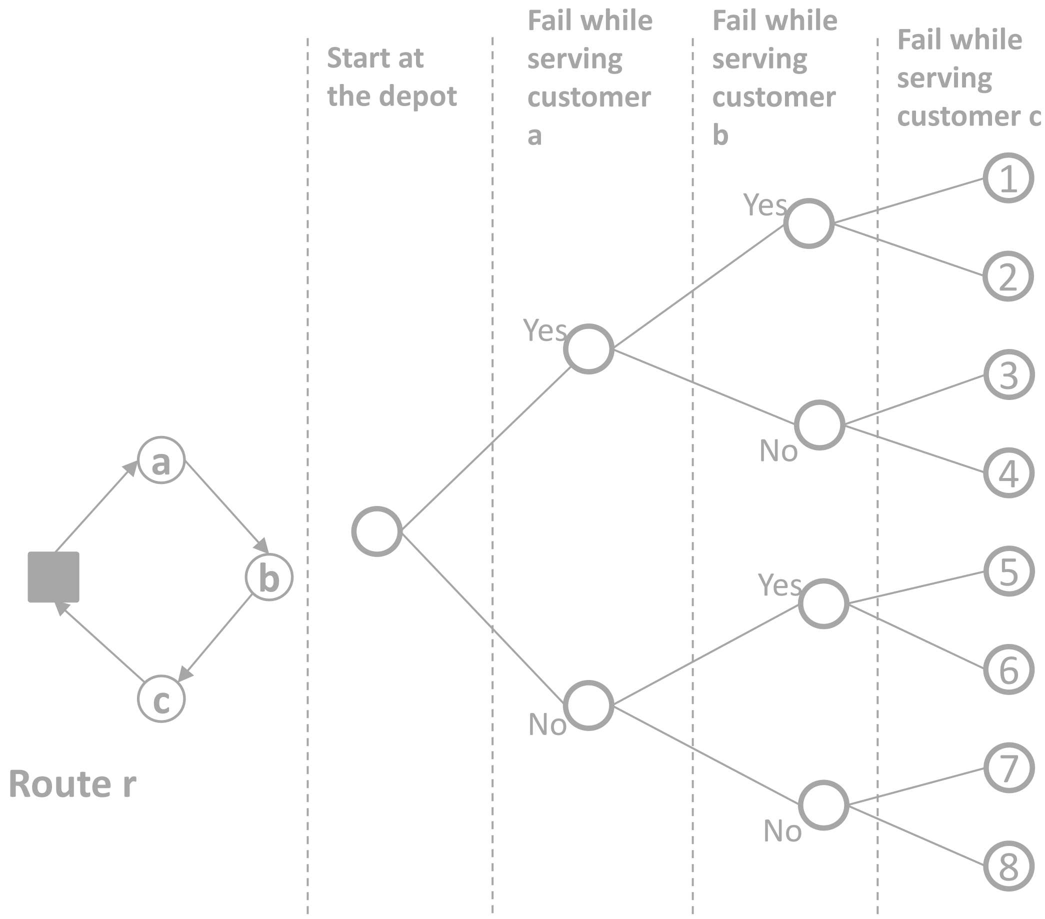
Route r



$$\Pr(p) = (1 - Pr(a))\, Pr(b)(1 - Pr(c))$$

$$C(p) = d_{(0,a)} + d_{(a,b)} + 2d_{(b,0)} + d_{(b,c)} + d_{(c,0)}$$

Chance constrained (CC)



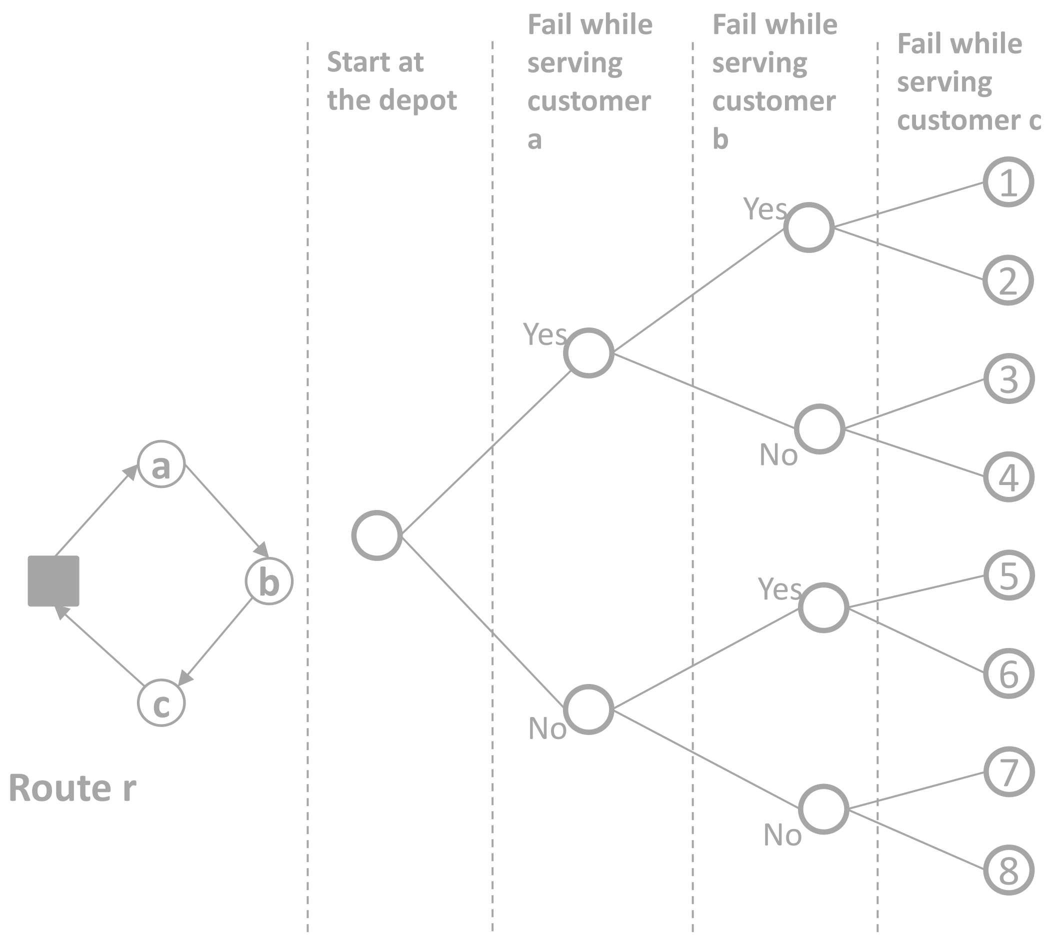
Probabilistic constraint:

$$\Pr(C_r \leq L) \geq 1 - \beta, \forall r \in \mathcal{R}$$

Calculated using the set $\mathcal{P}_r$

$$\Pr(C_r \leq L) = \sum_{p \in \mathcal{P}_r | C(p) \leq L} \Pr(p)$$

Stochastic programming with recourse



Solution cost includes the recourse cost for violating the duration constraint (i.e., overtime):

$$E[C] = \sum_{r \in \mathcal{R}} \left[E[C_r] + \sum_{p \in \mathcal{P}_r | C(p) \geq L} \Pr(p) \times \phi(C(p) - L) \right]$$

Three $\phi(\cdot)$ functions

- Linear
- Quadratic
- Piece-wise linear

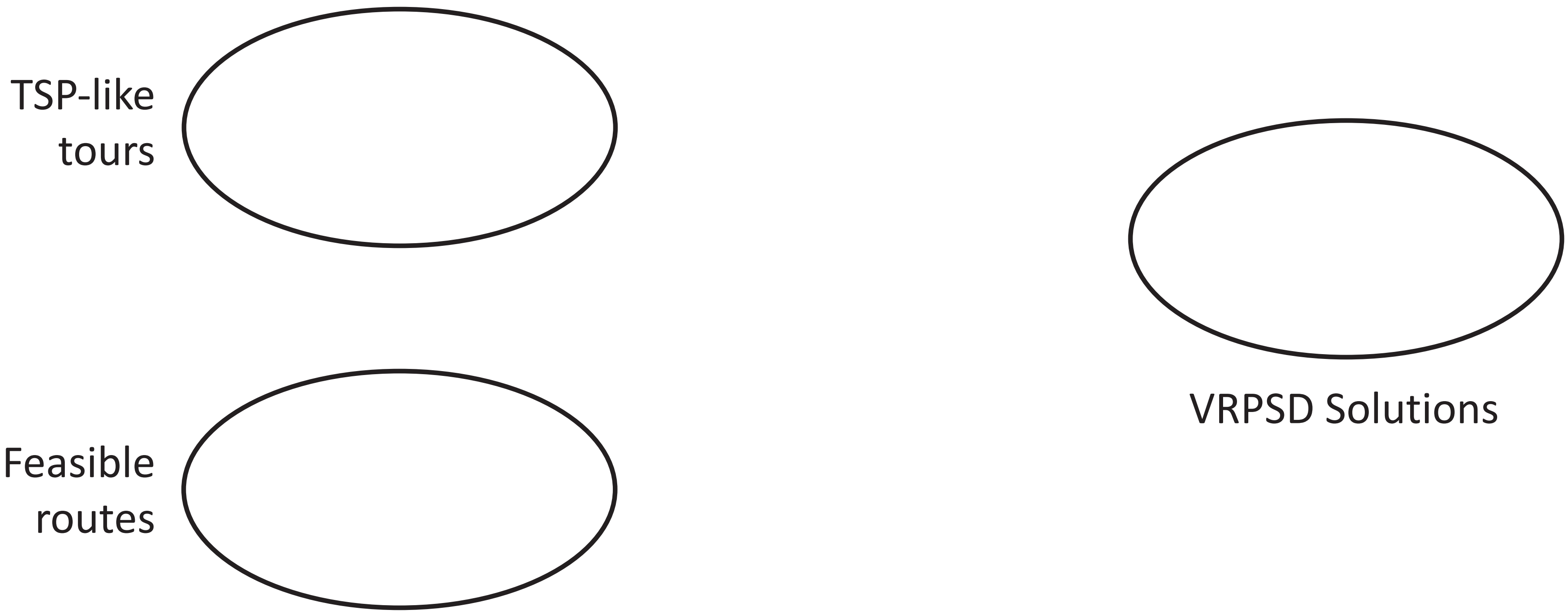
Agenda

ÉCOLE POLYTECHNIQUE MONTRÉAL

> The vehicle routing problem with stochastic demands and duration constraints (VRPSDDC)
  - Chance constraint programming formulation
  - Stochastic programming with recourse formulation
> GRASP + HC
  - General structure
  - Components
> Computational experiments
  - VRPSD
  - VRPSDDC
> Conclusions and perspectives

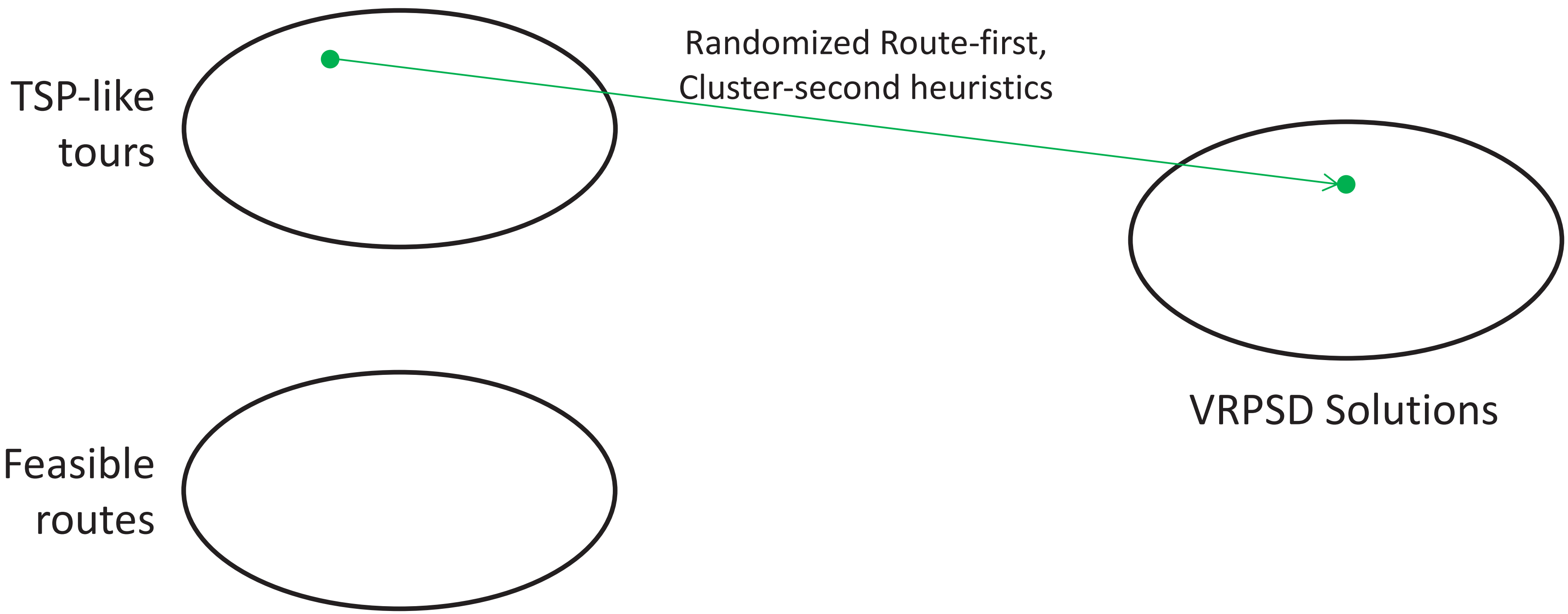
GRASP + HC: general structure

GRASP Iteration: generate start solution + local search



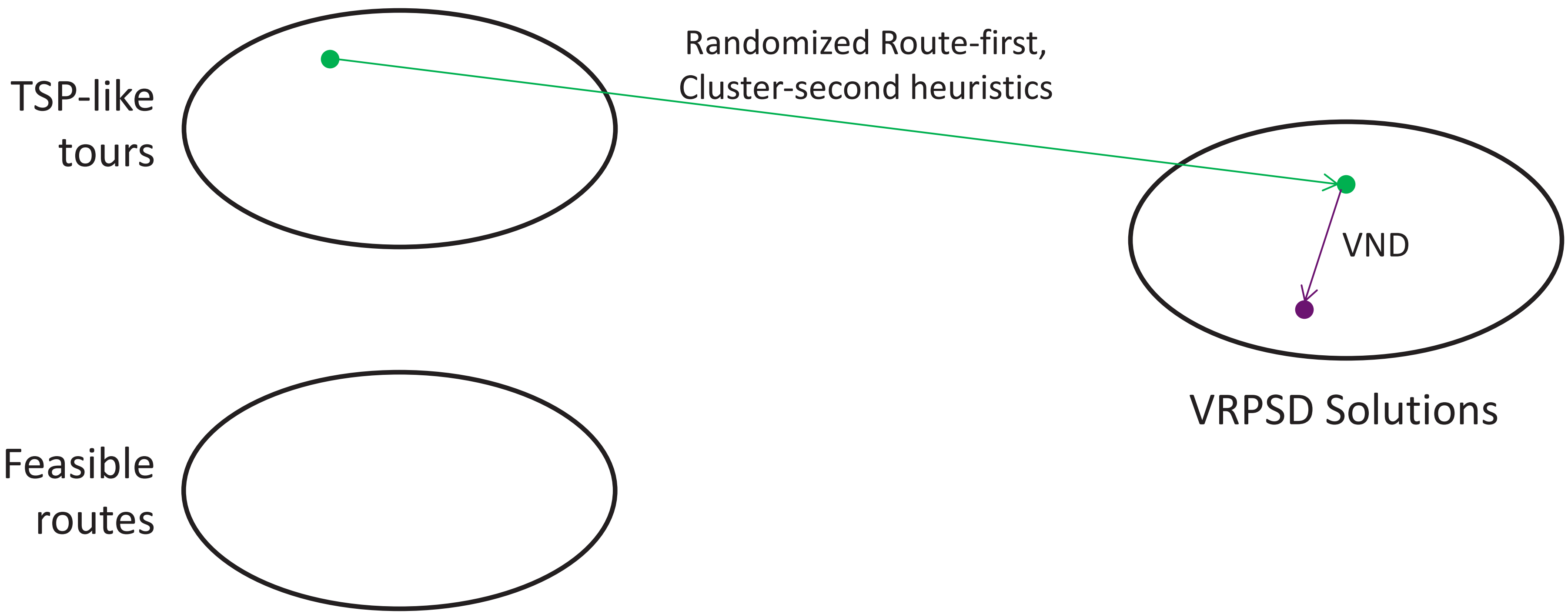
GRASP + HC: general structure

GRASP Iteration: generate start solution + local search



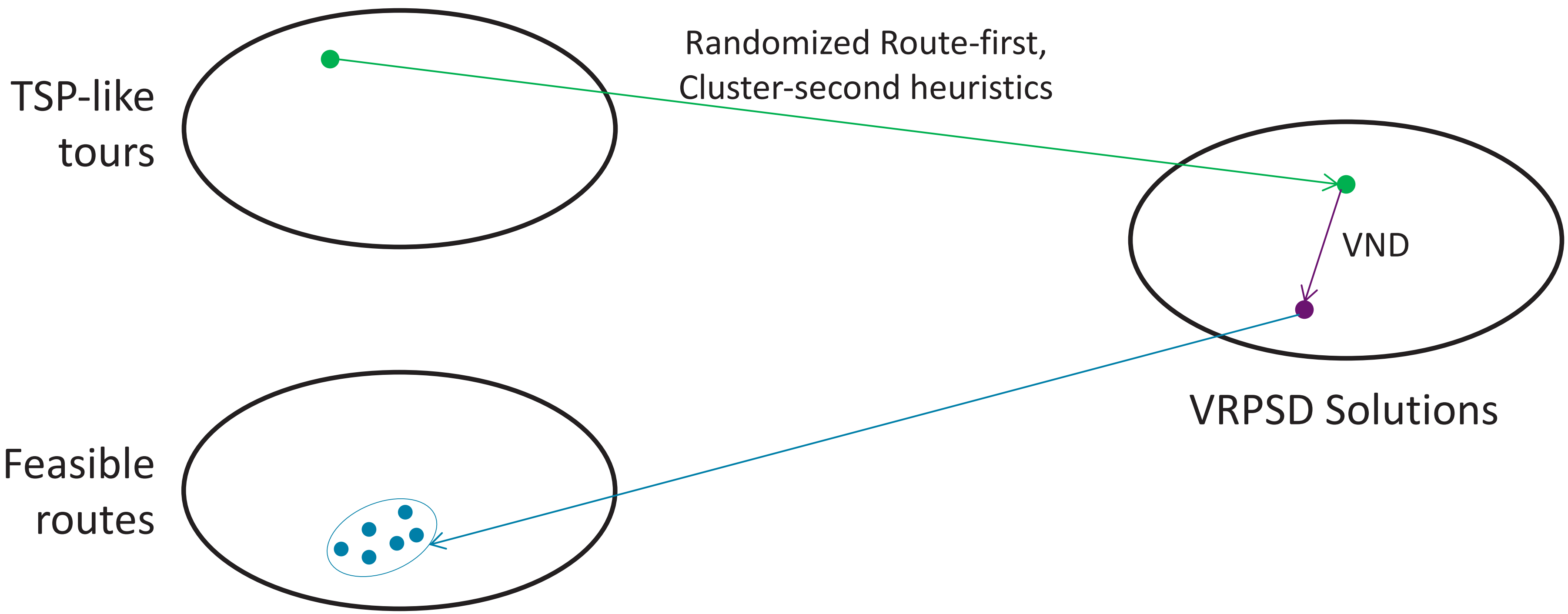
GRASP + HC: general structure

GRASP Iteration: generate start solution + local search



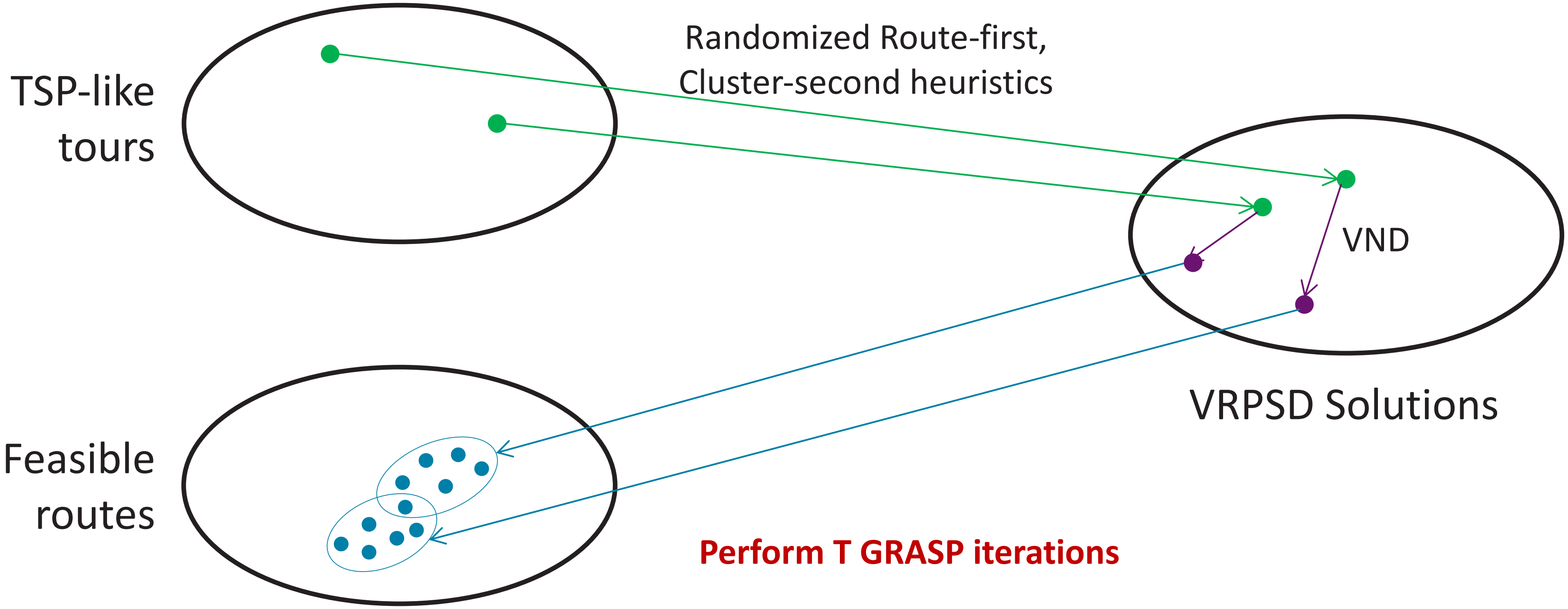
GRASP + HC: general structure

GRASP Iteration: generate start solution + local search + route storing



GRASP + HC: general structure

GRASP Iteration: generate start solution + local search + route storing

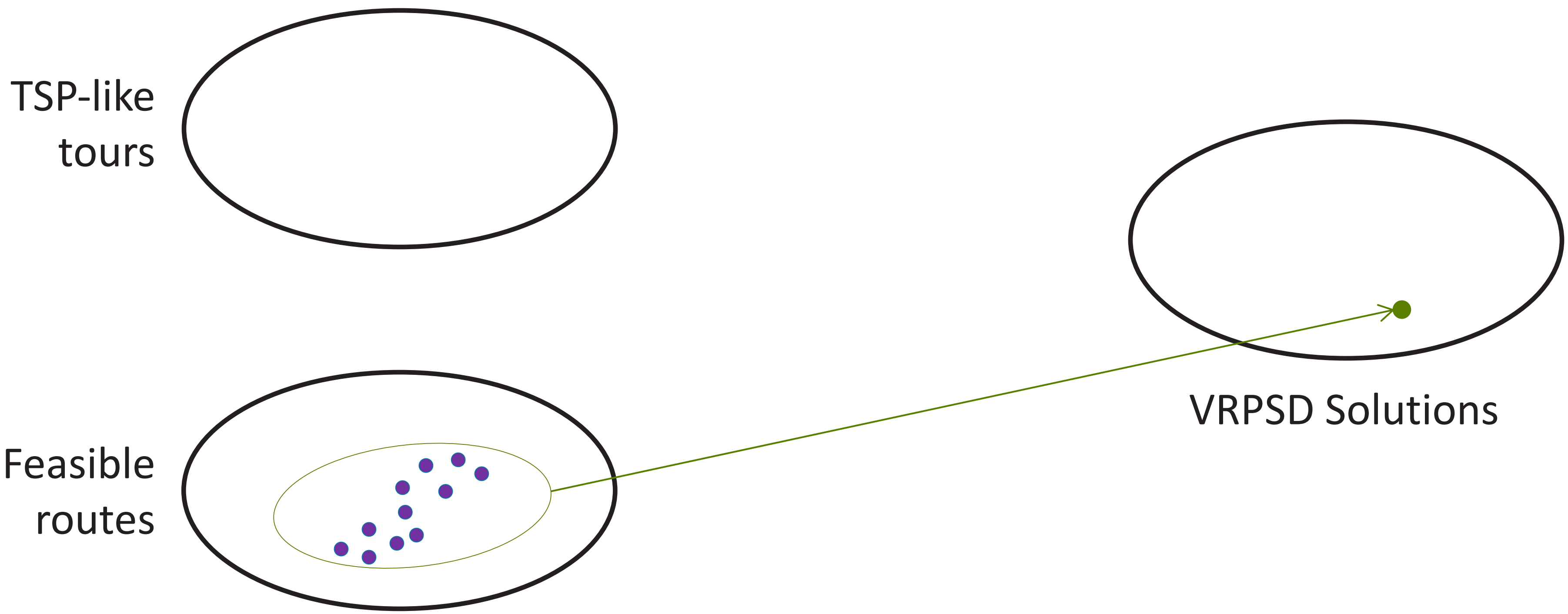


Perform T GRASP iterations

GRASP + HC: general structure

GRASP Iteration: generate start solution + local search + route storing

HC: solve a set-partitioning formulation over the set of stored routes Ω



GRASP + HC: building blocs

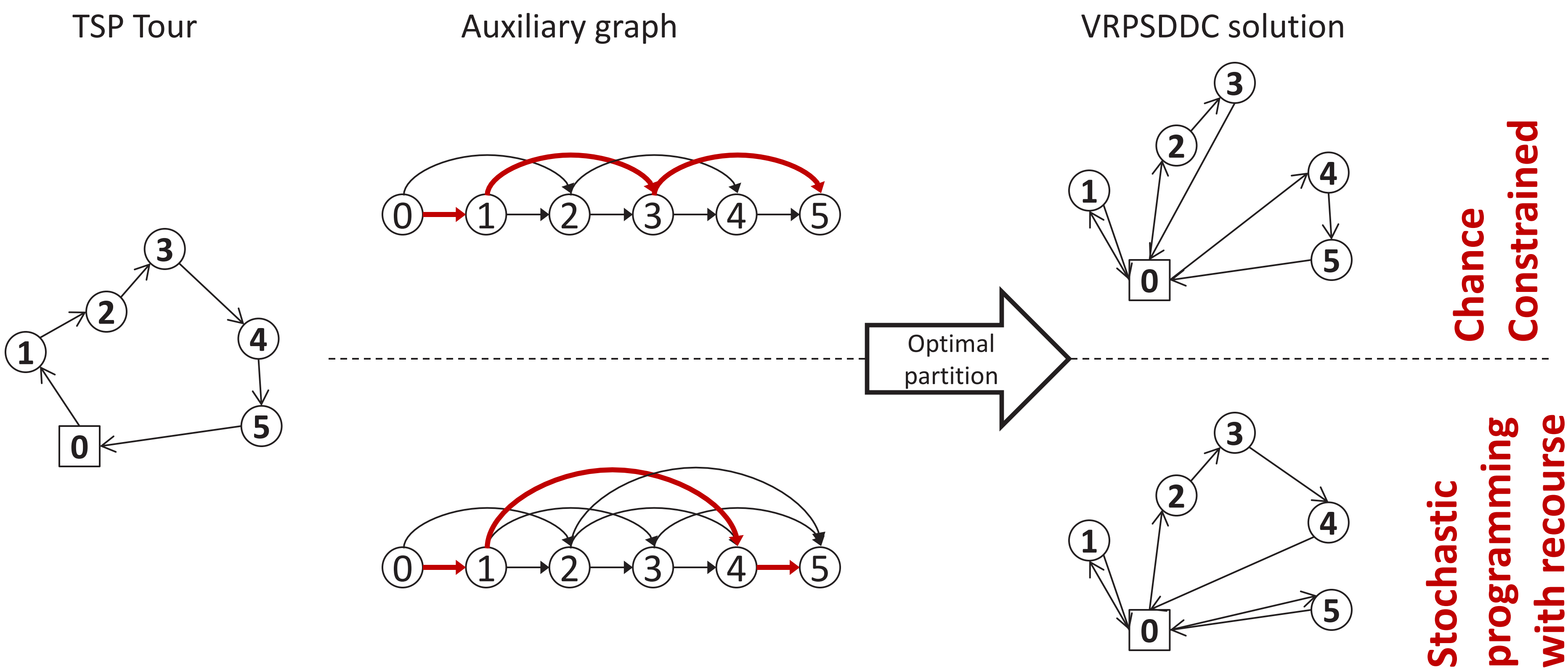
Randomized route first-cluster second heuristics: route-first

- Randomized Nearest Neighbor (RNN)
- Randomized Nearest Insertion (RNI)
- Randomized Farthest Insertion (RFI)
- Randomized Best Insertion (RBI)

GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

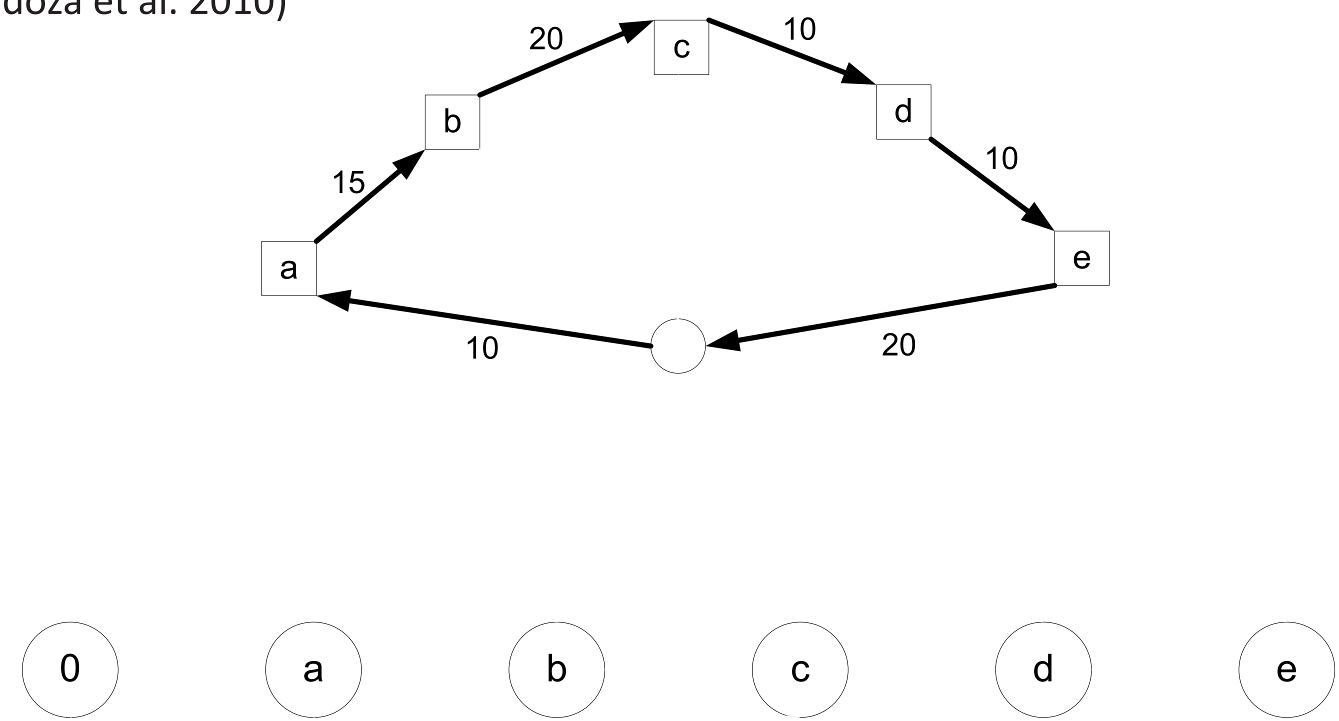
S-split (Mendoza et al. 2010)



GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

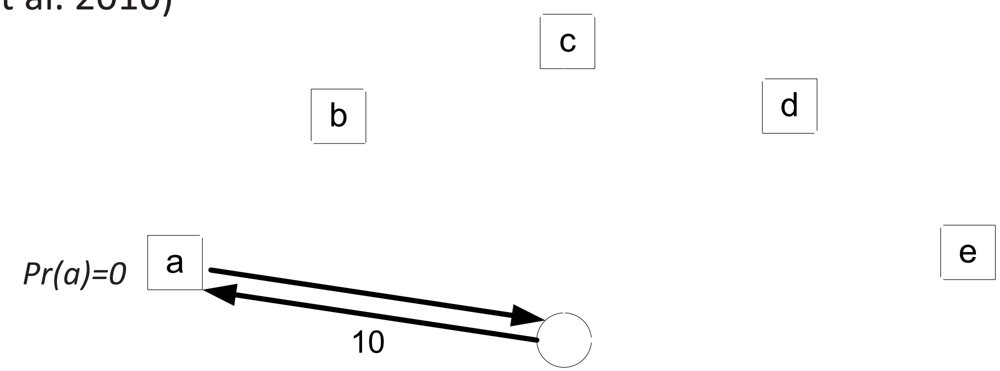
S-split (Mendoza et al. 2010)



GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)



$$l_r = 10 + 10 = 20$$

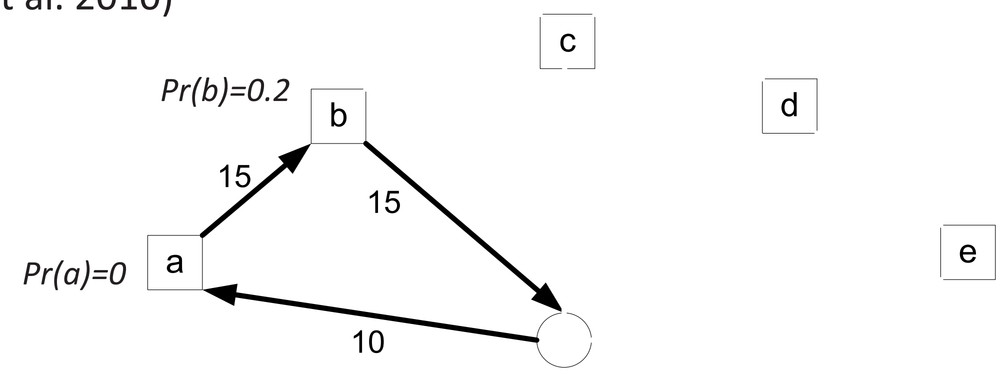
$$E\left[G_r(\xi)\right] = 2 \times 10 \times 0 = 0$$

Expected load constraint: checked

GRASP + HC: building blocs

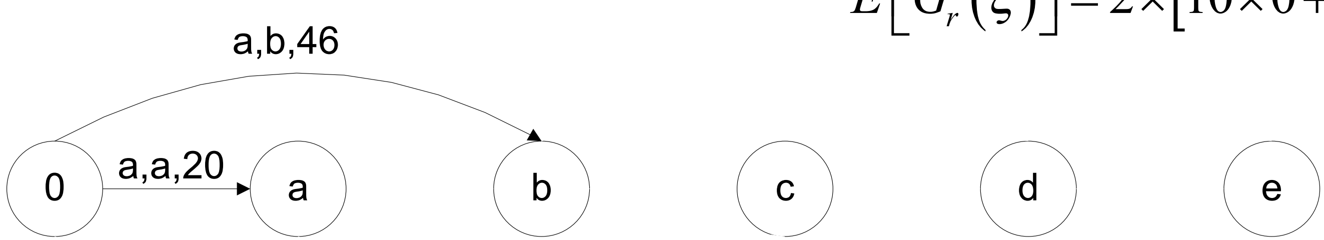
Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)



$$l_r = 10 + 15 + 15 = 40$$

$$E\left[G_r(\xi)\right] = 2 \times \left[10 \times 0 + 15 \times 0.2\right] = 6$$

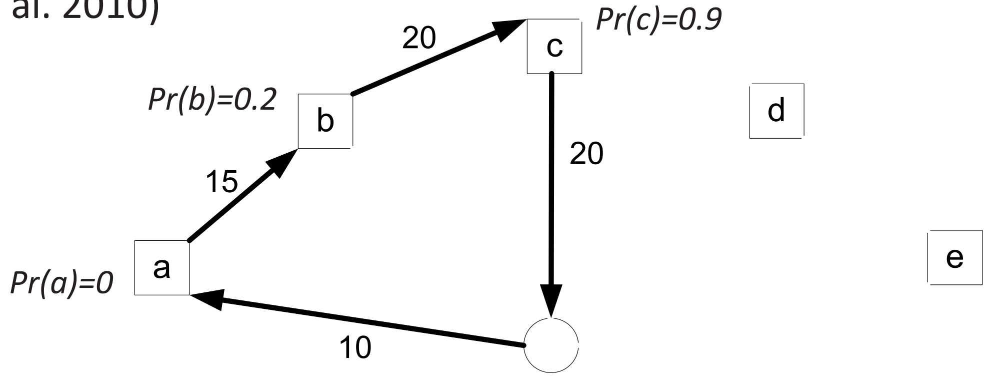


Expected load constraint: checked

GRASP + HC: building blocs

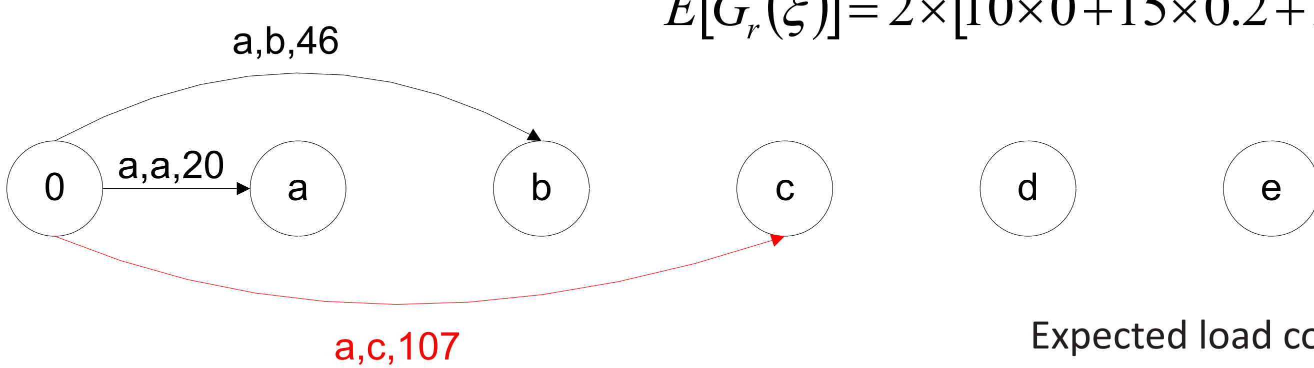
Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)



$$l_r = 10 + 15 + 20 + 20 = 65$$

$$E[G_r(\xi)] = 2 \times [10 \times 0 + 15 \times 0.2 + 20 \times 0.9] = 42$$

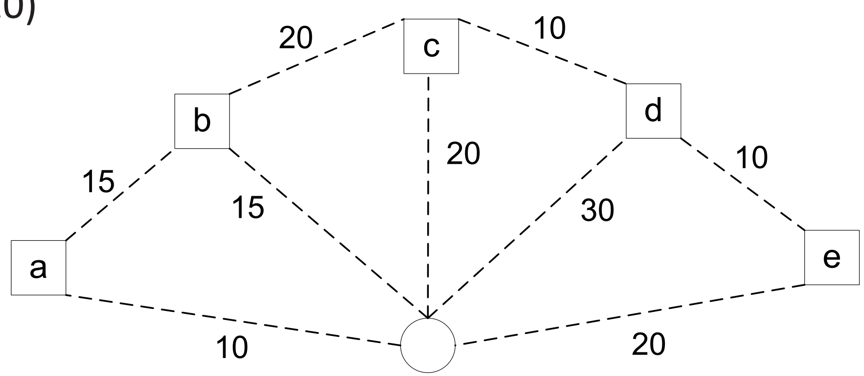


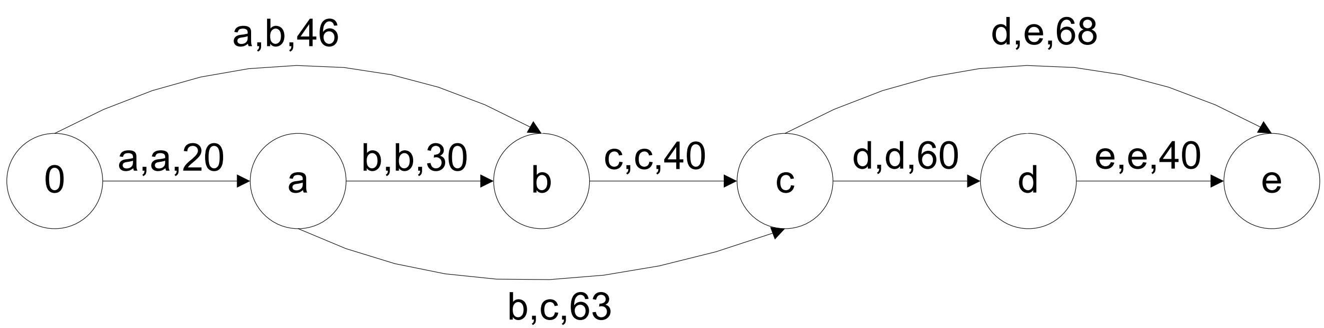
Expected load constraint: failed

GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)

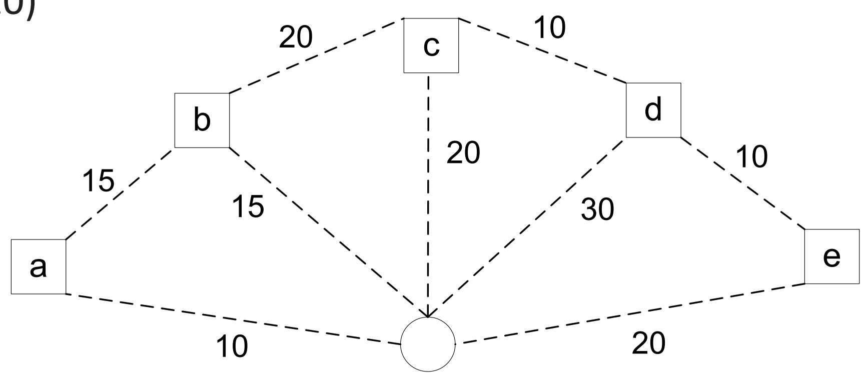


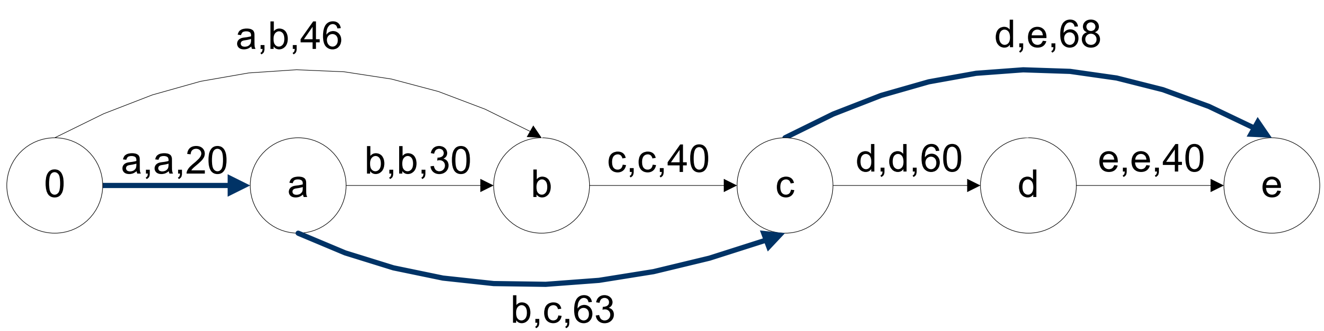


GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)

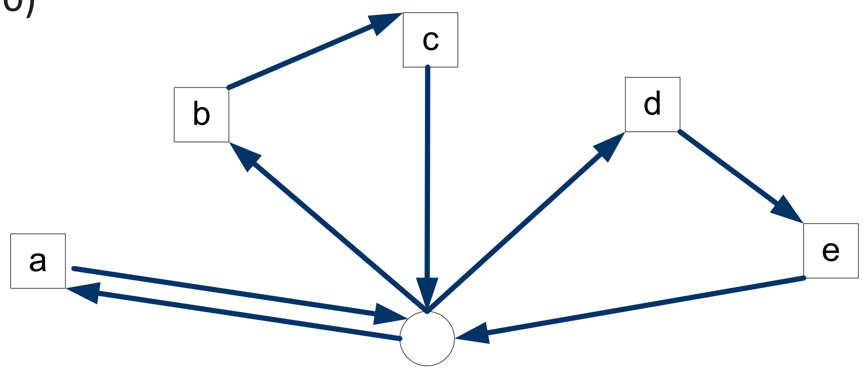


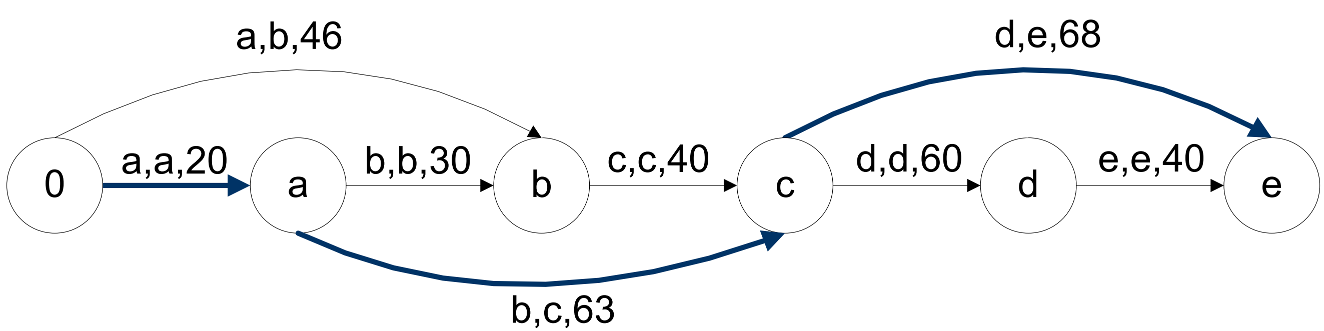


GRASP + HC: building blocs

Randomized route first-cluster second heuristics: cluster-second

S-split (Mendoza et al. 2010)

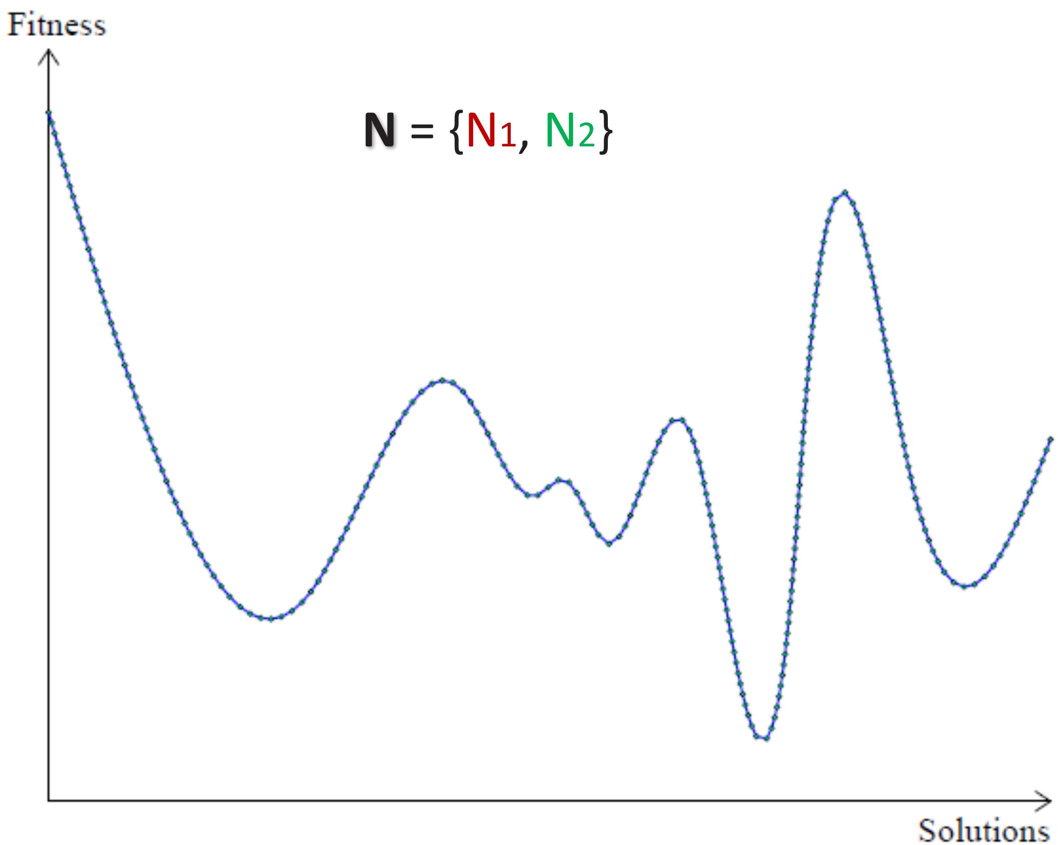




GRASP + HC: building blocs

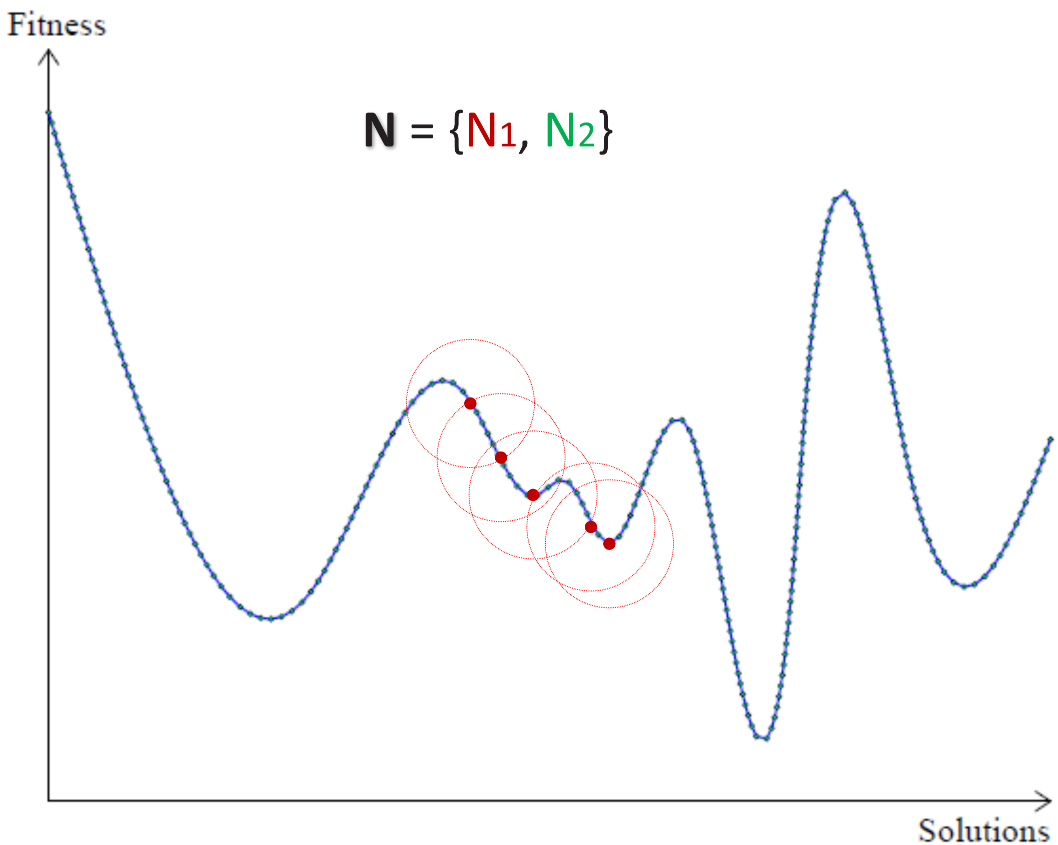
VND: recall

$\mathbf{N} = \{N_1, N_2\}$



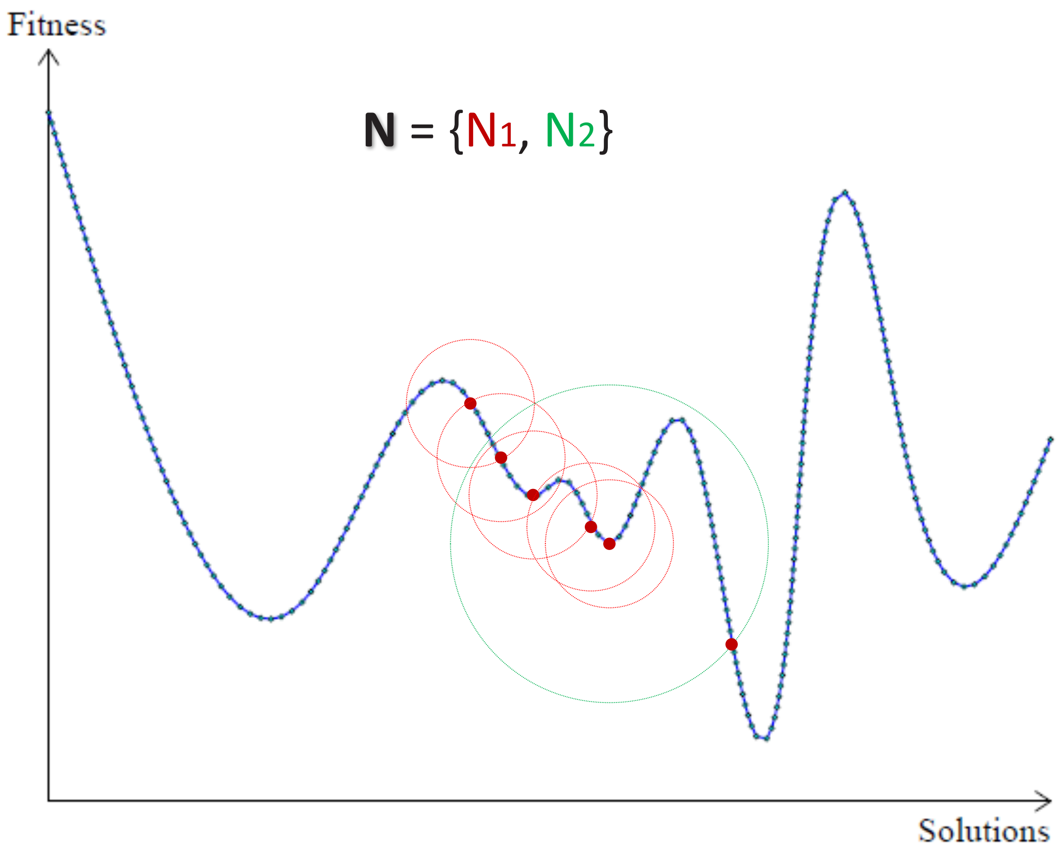
GRASP + HC: building blocs

VND: recall



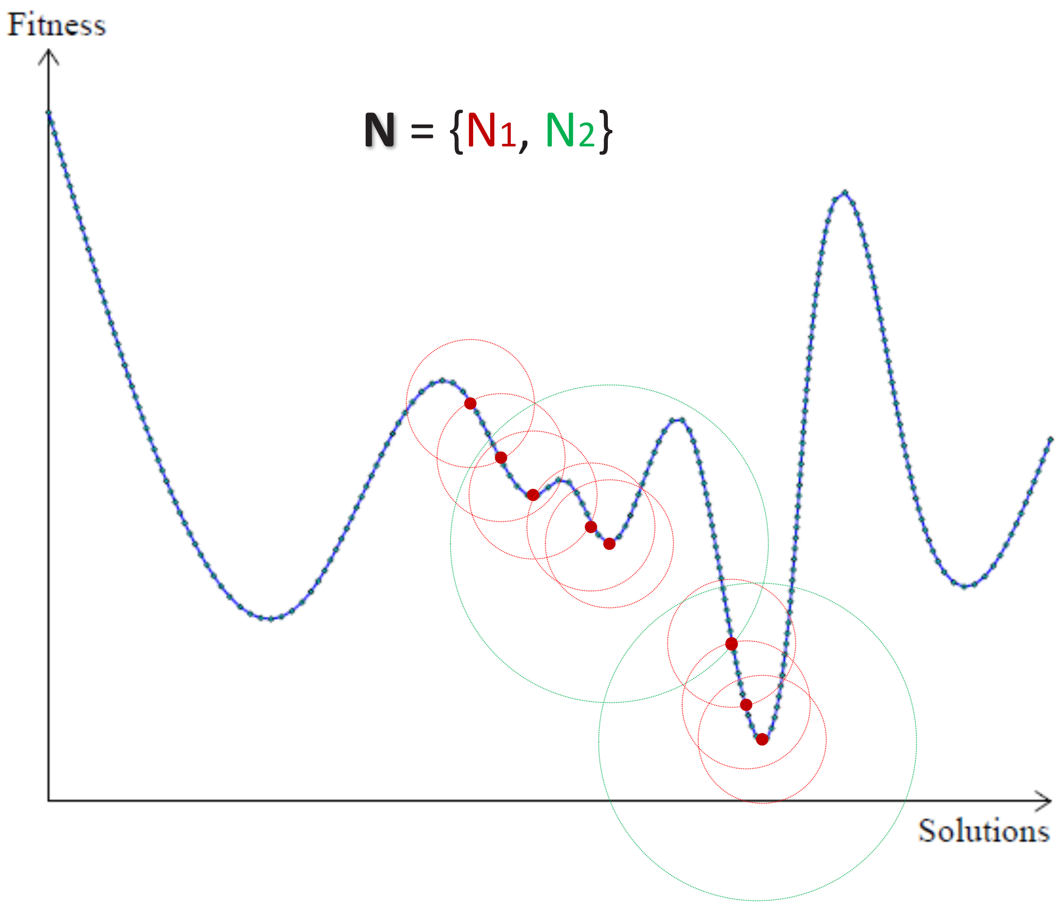
GRASP + HC: building blocs

VND: recall



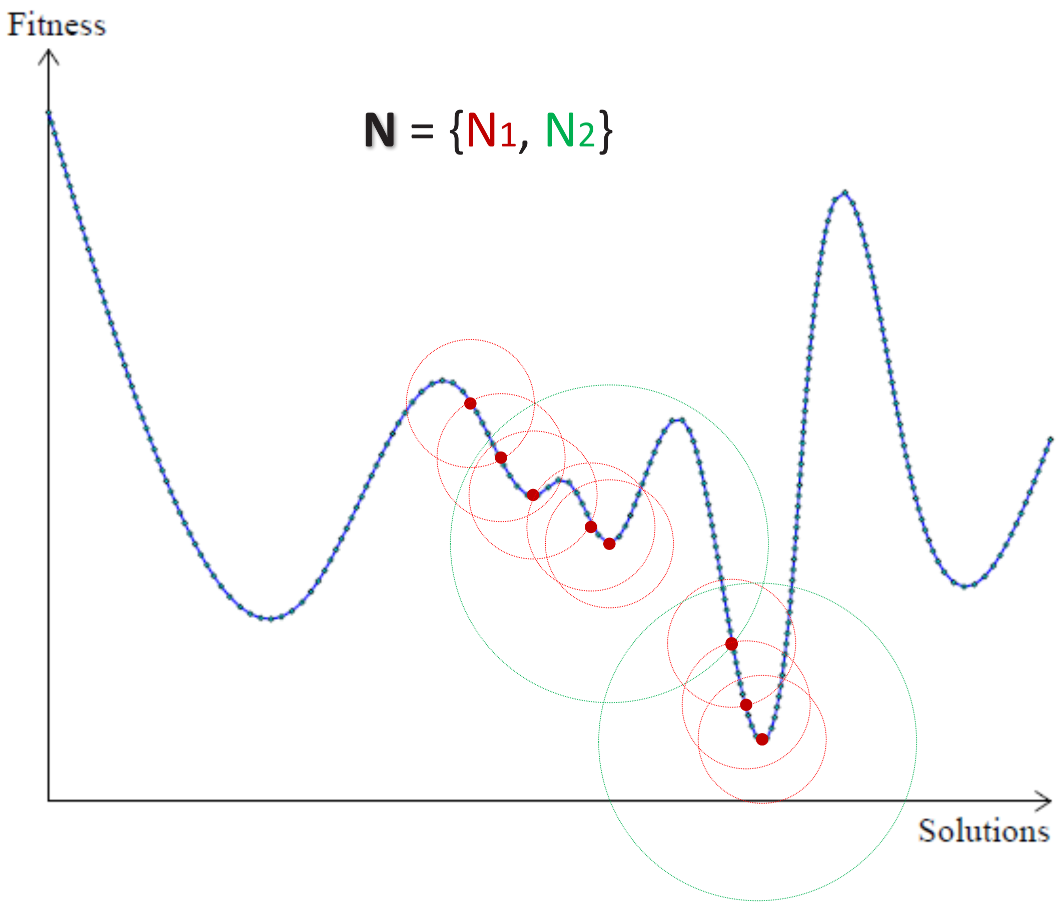
GRASP + HC: building blocs

VND: recall



GRASP + HC: building blocs

VND

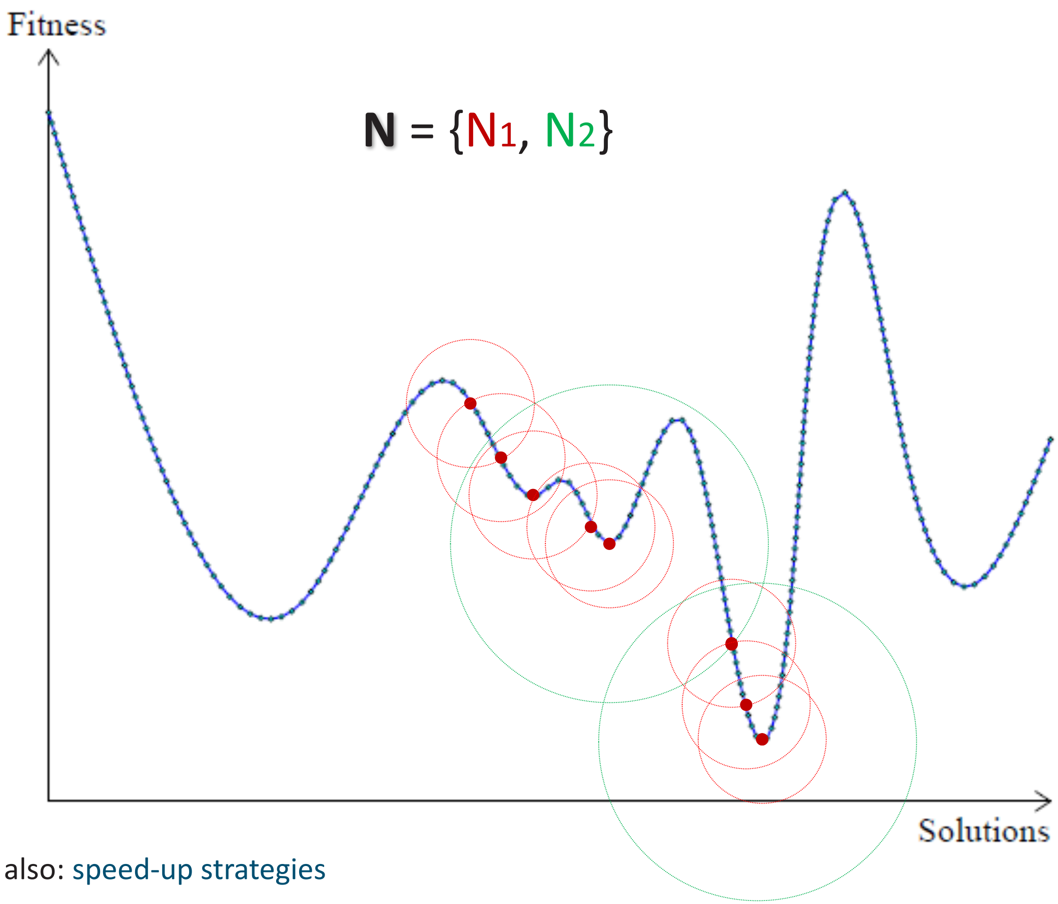


Our neighborhoods

1. Relocate
2. 2-Opt

GRASP + HC: building blocs

VND



Our neighborhoods

1. Relocate
2. 2-Opt

Chance constrained

- Move feasibility

Two-stage stochastic programming

- Move evaluation

See also: speed-up strategies

GRASP + HC: building blocs

Heuristic Concentration

- Set-partitioning formulation

$$\text{Min } Z = \sum_{r \in \Omega} E[C_r] \times x_r \quad (1)$$

Minimize the total expected cost of the solution

S.T

$$\sum_{r \in \Omega} a_{v,r} \times x_r = 1 \; \forall v \in \{1, \ldots v, \ldots n\} \quad (2)$$

Every customer must be serviced by exactly one route

$$x_r \in \{0,1\} \; \forall r \in \Omega \quad (3)$$

Nature of the decision variables

Agenda

- The vehicle routing problem with stochastic demands and duration constraints (VRPSDDC)
  - Chance constraint programming formulation
  - Stochastic programming with recourse formulation
- GRASP + HC
  - General structure
  - Components
- Computational experiments
  - VRPSD
  - **VRPSDDC**
- Conclusions and perspectives

Computational experiments: VRPSDDC

Benchmark instances

- 39 instances adapted from Christiansen and Lysgaard (2007)
  - Adding a duration constraint (L)
  - GRASP+HC VRPSD solutions verify
    - $E[C_r] \leq L \; \forall \, r \in \mathcal{R}$ (Yang et al. 2000, Mendoza et al. 2010, 2011) – (**ED approach**)
- 19 to 60 customers
- Poisson-distributed demands

Computational experiments: VRPSDDC

CC formulation

> Comparison with Best solutions of VRPSD

Metric	Overall	Best run	Worst run
Avg. Gap	2.17%	2.10%	2.34%
Max. Gap	8.44%	8.44%	8.44%
Min. Gap	0.05%	0.00%	0.30%
Std. Dev. Gap	1.90%	1.91%	1.88%

Computational experiments: VRPSDDC

CC formulation

> Average running time

Metric	Overall	Best run	Worst run	Metric	Overall
Avg. Gap	2.17%	2.10%	2.34%	Avg. CPU (s)	233.99
Max. Gap	8.44%	8.44%	8.44%	Max. CPU (s)	1015.05
Min. Gap	0.05%	0.00%	0.30%	Min. CPU (s)	5.98
Std. Dev. Gap	1.90%	1.91%	1.88%	Std. CPU (s)	225.46

Computational experiments: VRPSDDC

CC formulation

- Post-hoc evaluation of the VRPSD (ED) solutoins for β = 0.05

Metric	% Infeasible routes	Max. Pr	Min. Pr	Avg. Pr
Avg.	34.96%	0.217	0.005	0.067
Max.	100.00%	0.446	0.169	0.299
Min.	0.00%	0.010	0.000	0.003

Only 3/39 VRPSD (ED) solutions are feasible

Computational experiments: VRPSDDC

DR formulation: experiment configuration

- > $\phi(\cdot)$ functions
  - Linear
  - Quadratic
  - Piecewise linear
- > GRASP+HC(DR)
  - T=500
- Post-hoc evaluation of VRPSD solutions

Computational experiments: VRPSDDC

DR formulation

> Comparison with best solutions of VRPSD.
> Increase in the objective function due to the violation of the constraint

Metric	Linear	Piece-wise linear	Quadratic
Avg. Increase Obj. Function	2.12%	3.26%	6.10%
Max. Increase Obj. Function	4.49%	9.57%	15.46%
Min. Increase Obj. Function	0.37%	0.54%	0.80%
Std. Dev. Increase Obj. Function	1.22%	2.02%	3.44%

Computational experiments: VRPSDDC

DR formulation

- Post-hoc evaluation VRPSD (ED) solutions for the three penalizations with respect to best DR solution
- Increase in objective function due to overtime

Metric	Linear	Piece-wise linear	Quadratic
Avg. Gap	1.18%	4.56%	81.33%
Max. Gap	3.69%	11.20%	283.21%
Min. Gap	0.00%	0.00%	2.85%
Std. Dev. Gap	1.01%	3.31%	73.72%
Unchanged BKS	10	2	0

Computational experiments: VRPSDDC

DR formulation

> Running time comparison DR vs CC Formulations

Metric	Linear	Piece-wise linear	Quadratic	CC Formulation
Avg. CPU (s)	326.48	346.43	355.45	233.99
Max. CPU (s)	1389.65	1432.22	1510.69	1015.05
Min. CPU (s)	8.19	8.63	9.42	5.98
Std. CPU (s)	315.66	338.42	347.61	225.46


« UNCERTAINTY IS AN UNCOMFORTABLE POSITION. BUT CERTAINTY IS AN ABSURD ONE. »
-Voltaire