

Route Optimization Workshop

Alina Constantin
24.11.2022

ORTEC

Powered by
**Global
Operations**

Optimization

Contents

Introductions

What technology do we use?

How do we configure our optimizers?

How do we implement our optimizers?

Algorithm Knowledge Team for Routing



Presales and Proof of Concept Support

- Prove Value of ORTEC
- Business Analysis
- Data Analytics
- Presales Modeling
- Feasibility of Projects
- Optimization Approach
- Estimations
- Knowledge Sharing of Optimization Technology



Solution Design

- Workshops
- Optimization Framework
- Business Analysis
- Business Rules and Workflows
- Data Analytics
- Optimization Solution Design
- Review of Designs



Project Implementation

- Implementation or Consulting Role
- Support the Optimization Framework
- Baseline Approach and Validation
- Optimization Configuration, Modeling, and Validation
- Review of Configuration and Modeling



Trainings and Workshops

- Optimization Workshops for Customers
- Optimization Training for Sales, Business Consultants, Solution Architects, and Consultants

ORTEC

ORTEC stands for Operations Research Technology



Operations Research

World War II

Development of Operations Research

Planning Military Operations

Scientific Methods

Quantitative Base

Advanced Analytics

Close to 1,000 people working on it

Quantitative or Operational Management



What is Operations Research?

Combination of disciplines



Statistics

Logistics
Finance
Marketing

Mathematics

Economic
problem

Solution

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Information
Technology

```
main( ) {  
    printf("hello, world");  
}
```

What is Operations Research used for?

Almost anything that requires a decision about a real world problem
How to do something better!

- **Scheduling**: hospital patients, classes, buses, planes, sporting events
- **Marketing**: store layout, advertising, social media, online ad placement, recommendations on a website
- **Product development**: product features, pricing, sales forecasts
- **Inventory**: how many to build; how many touchpads store should have in stock
- **Organizations**: business management, cross-cultural issues, social networks
- **Queueing**: waiting lines at amusement parks, banks, movie theaters; line at store to buy new electronic gadget; traffic

Travelling Salesman Problem (TSP)

Well known problem in the Operations Research

Salesman

- 15 cities
- Distance table for all 15 cities

Assignment

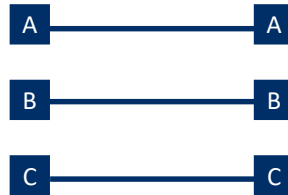
- Visit each city just once
- Find the shortest route and return to the start location
- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?



Vehicle Routing Problem

3 Vehicles and 3 Customers

- 3 vehicles: A, B and C



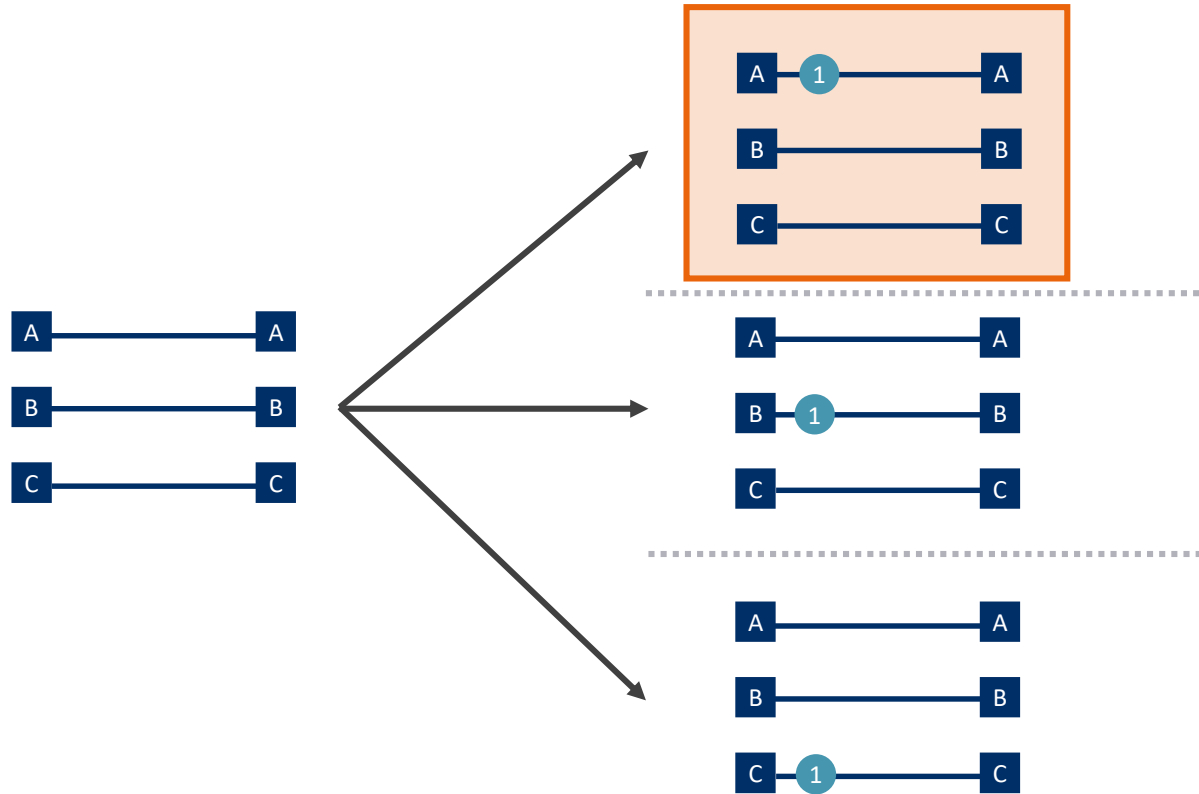
- 3 customers: 1, 2 and 3



- How many solutions are there to plan 3 customers into 3 vehicles?

Vehicle Routing Problem

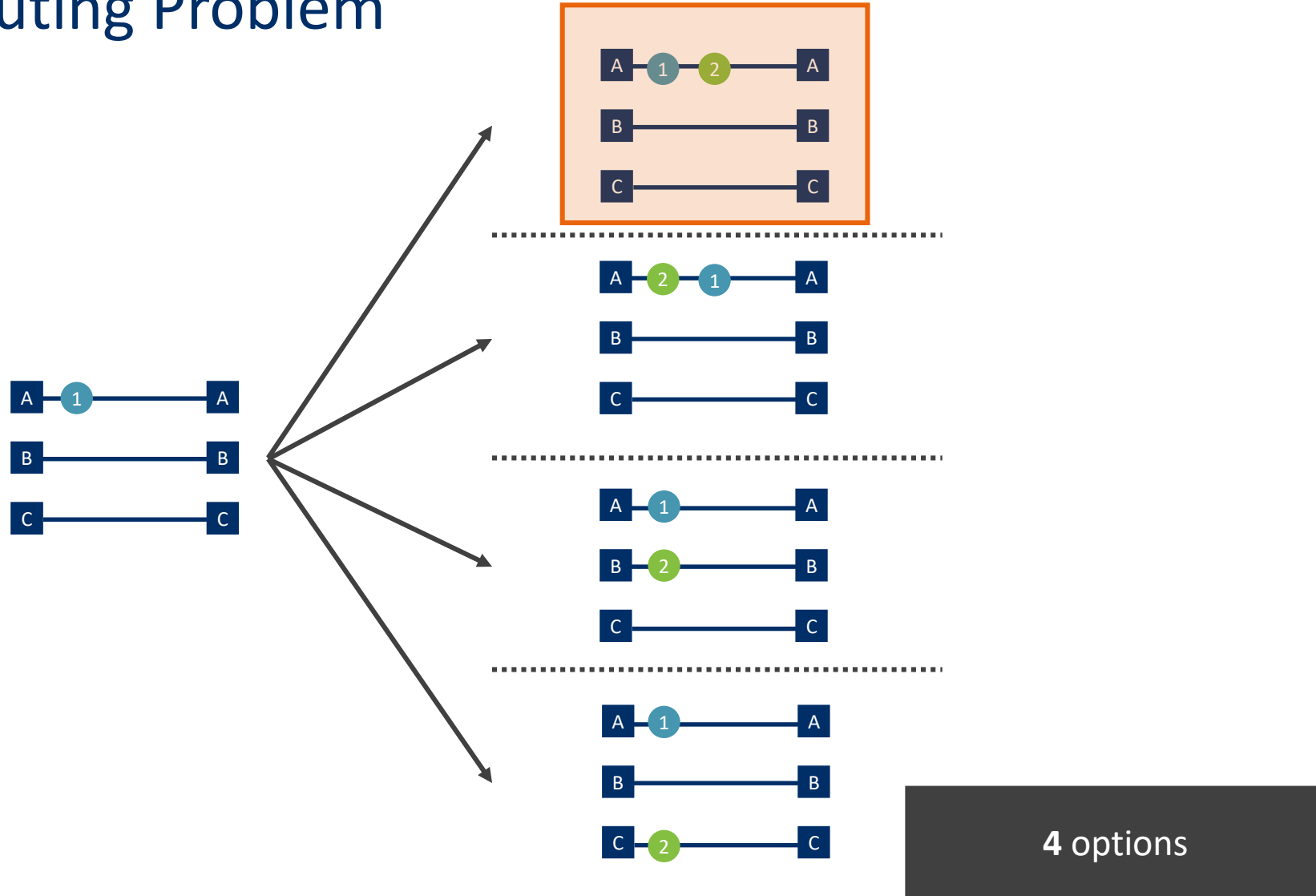
Plan Customer 1



3 options

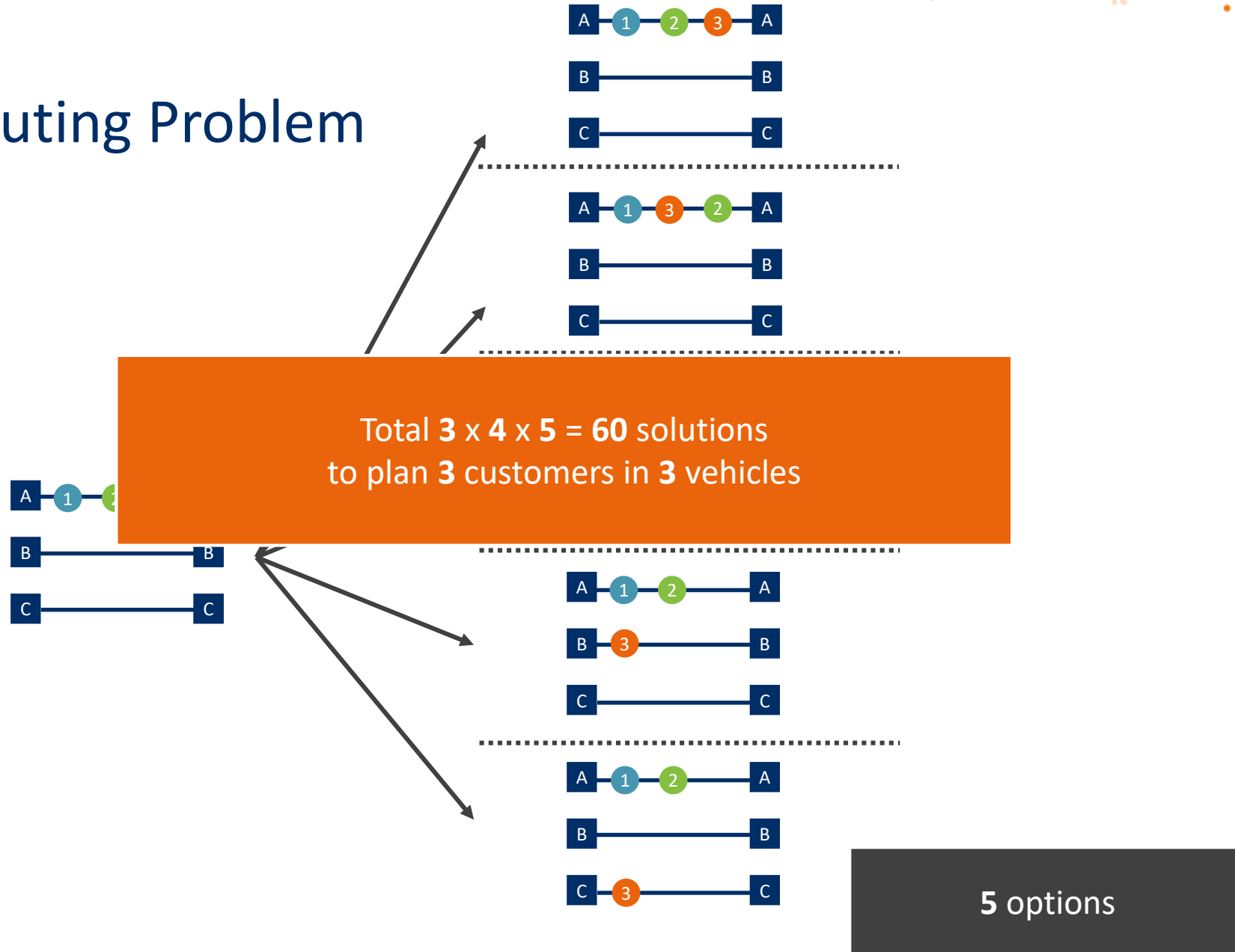
Vehicle Routing Problem

Plan Customer 2



Vehicle Routing Problem

Plan Customer 3



Vehicle Routing Problem

Solutions

Vehicles	Stops	Formula	Solutions
3	3	$3 \times 4 \times 5$	60
3	6	$3 \times 4 \times 5 \times 6 \times 7 \times 8$	20160
3	9	$3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11$	19.958.400
5	25	$5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12 \times 13 \times 14 \times 15 \times 16 \times 17 \times 18 \times 19 \times 20 \times 21 \times 22 \times 23 \times 24 \times 25 \times 26 \times 27 \times 28 \times 29$	$368.406.749.739.154.000.000.000.000.000$ $= 368 \times 10^{27}$
m	n	$m \times (m+1) \times \dots \times (m+n-1)$	

For 5 vehicles and 25 stops →
more than 10^{13} years of calculation time

Theory

Methods

Exact Methods

- Very small problem instances
- In principle: examine all solutions

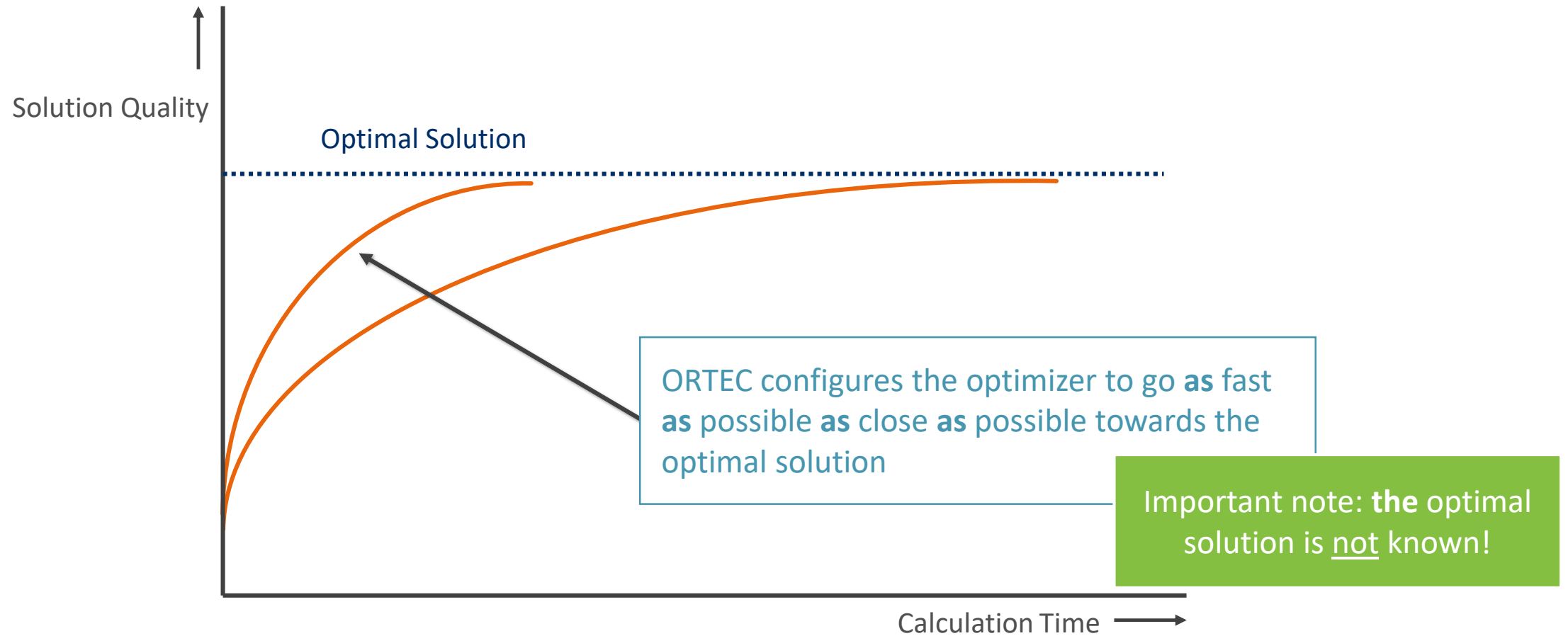
Heuristics

- According to a 'common sense' reasoning
- But sometimes also with guarantee on the quality of solutions
(Example: max. 3% of the best solution)



Theory

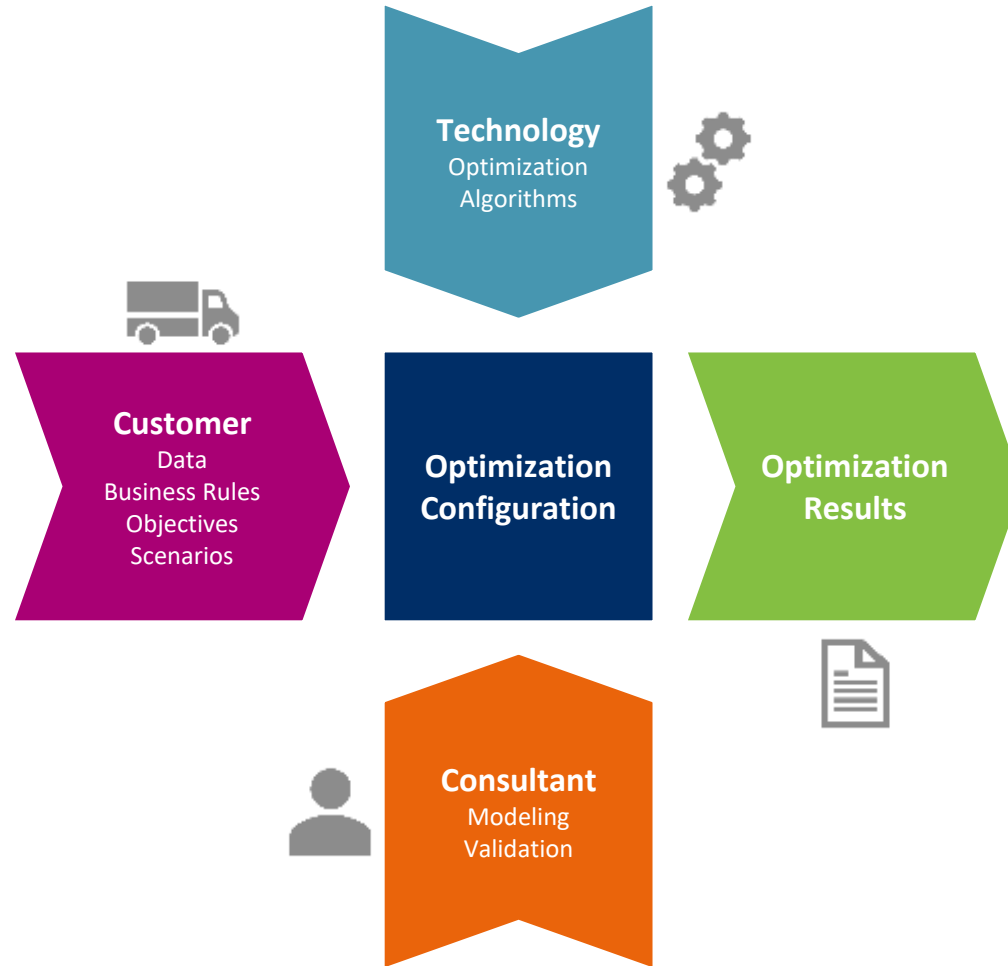
Heuristics choose from many possibilities



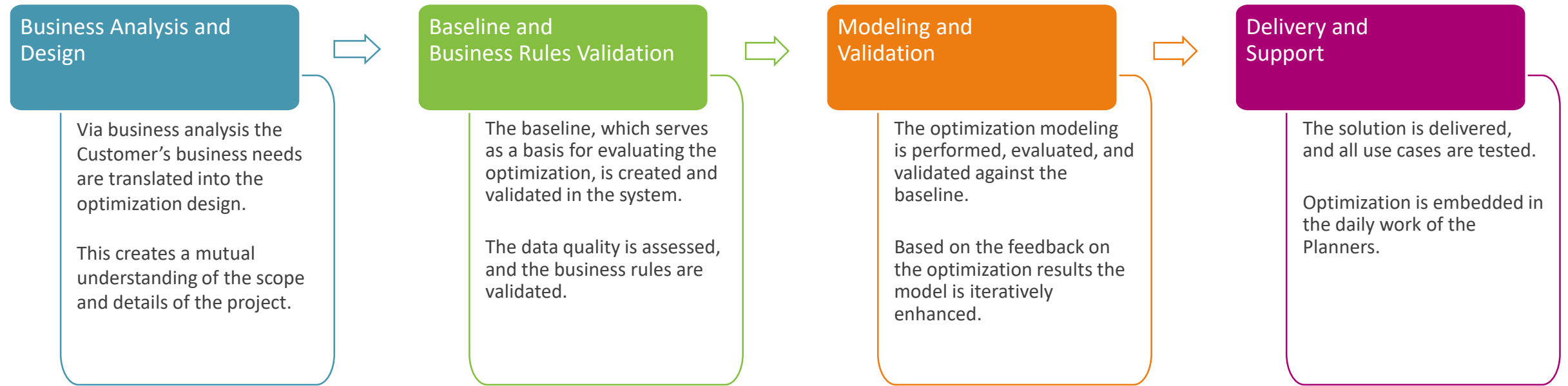


Optimization Framework

Optimization Configuration



Optimization Framework



Project Roles



Project Manager

Enforce usage of framework, scoping



Business Consultant

Input on business processes, business case



Project Solution Architect

Solution design, change requests, big picture



Consultant

Optimization focus, data and business rules validation, scenario running, reporting, automation



AKT Consultant

Assist in baselining, check configuration, optimization modeling, scenarios



AKT Solution Architect

Optimization design, change requests, workshops, framework

Depending on the project, there are several project roles. The most common ones for an optimization project are included here. Not all roles carry out all steps. Per activity there are different roles leading and involved, which is detailed in the following slides.

This only serves as a guideline, as each project is different. For example, complex business and optimization cases require more effort from the Project SA and AKT SA in guiding the Consultant and AKT Consultant, and in cloud projects the Consultant will have more focus on data and automation. Of course, not all projects have all project roles fulfilled by different people, so it might happen that a person has multiple roles.

Australia

- Baptistcare
- Coca Cola Amatil
- Fonterra
- G4S
- Hyne Timber
- Linfox

Belgium

- Ariel
- Servilux
- Vincotte
- Woonhaven Antwerpen

Brazil

- Carrefour
- Danone

CEE

- Aquila
- ATB
- Carrefour RO
- GDF Suez/Engie
- Holiday Classic
- Monетка
- Urgent Cargus
- X5

France

- Abrisud
- Carrefour
- Macadam
- Pomona

Germany

- CCE BE
- CCE Europe
- CCE NL
- CCE NO
- CCE UK
- Dodenhof
- Hermes
- Ostendorf

Italy

- Carrefour IT
- Carrefour TU
- CEVA
- EKOL
- Jamnica
- Ktosovolos
- Roto Dinamic

Netherlands - OGC

- Benegas
- Engen

Netherlands – TTL

- Bakker
- Dobbe
- Heineken
- Kwantum
- PostNL
- Rabelink
- Schotpoort
- Steenbergen
- Van den Anker
- Van Zaal
- Westerman

Netherlands - PPS

- Aboma
- Accolade
- Autobar
- Bavaria
- Benesto
- Breijer
- Carglass
- Coca Cola Enterprises
- CRV
- De Alliantie
- Dekra
- Dela
- Duijvelaar Pompen
- Electrolux
- Facilicom
- Fri-jado
- Generali
- Grolsch
- Harting Bank
- Heijmans
- Imtech Access & Security
- Maas
- Medux
- Motrac Linde
- RPS
- RRS
- Saval
- Shield Group
- Siers
- Stadgenoot
- Tapwacht
- Toshiba
- Unica
- Vereniging Eigen Huis
- Welzorg
- Woonzorg

Nordics

- Arla
- Electrolux
- Nature Energy
- NorgesGruppen

Poland

- Univeg

United Kingdom

- Coca Cola Enterprises
- Waitrose

United States

- Ashland
- CHS
- DHL
- GHI
- McPherson
- Nalco Champion
- Quest
- RTI
- Sanimax
- Walmart

Licenses



Optimization Technology

Optimization Configuration

Technology



Objectives



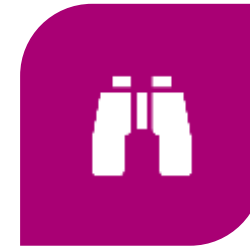
Sorting



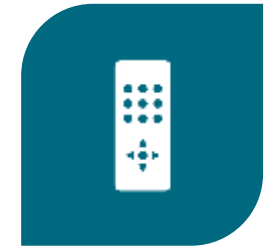
Selecting



Construction



Local Search



Control

Optimization Technology

Solution Approach

Construction

- Rule based & geographical based strategy to build initial routes
- Plan as many orders as possible (or the ones with most profit)
- Initial focus on 'difficult orders'

Local Search

- Heuristics that search 'locally' to minimize cost: reduce km/miles, reduce hours, reduce overtime, reduce routes, etc.

Large Neighborhood Search

- Metaheuristics that search on both a 'local' and 'global' scale for improvements: Ruin and Recreate

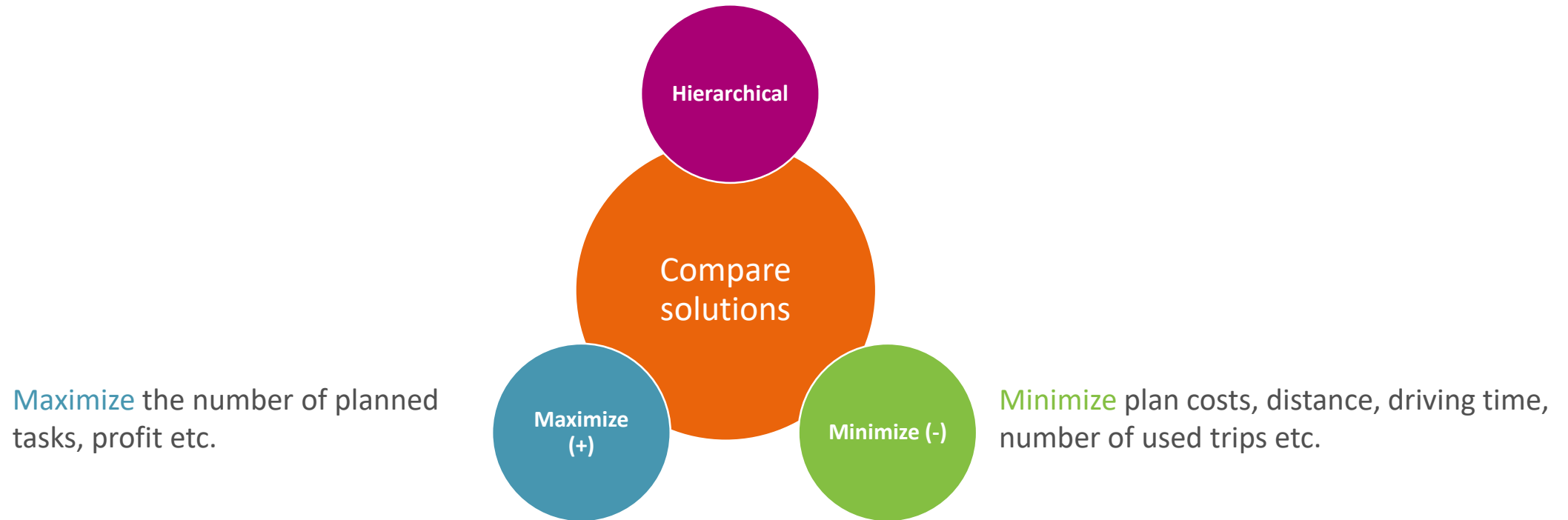


Optimization Objectives

Optimization Objectives

CVRS

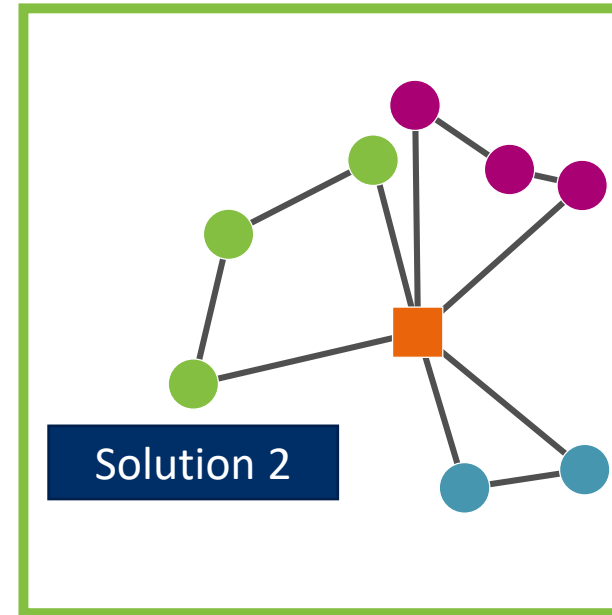
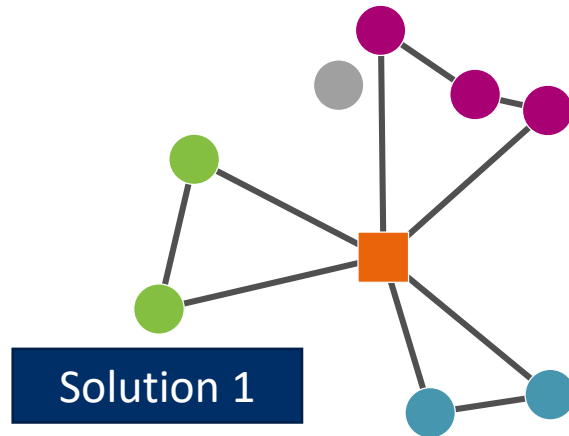
Compare solutions based on the **first objective**. If they are the same, compare them according to the **second objective** etc.



Optimization Objectives

Example

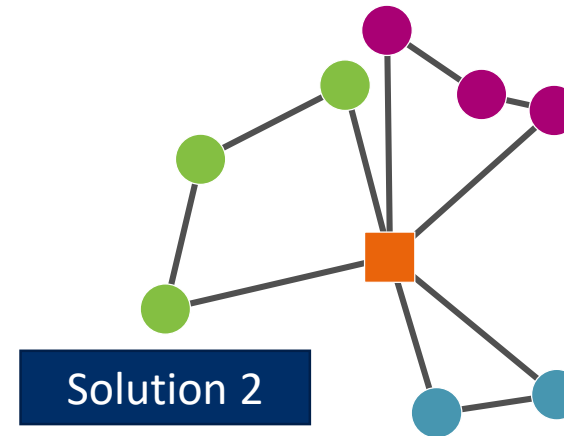
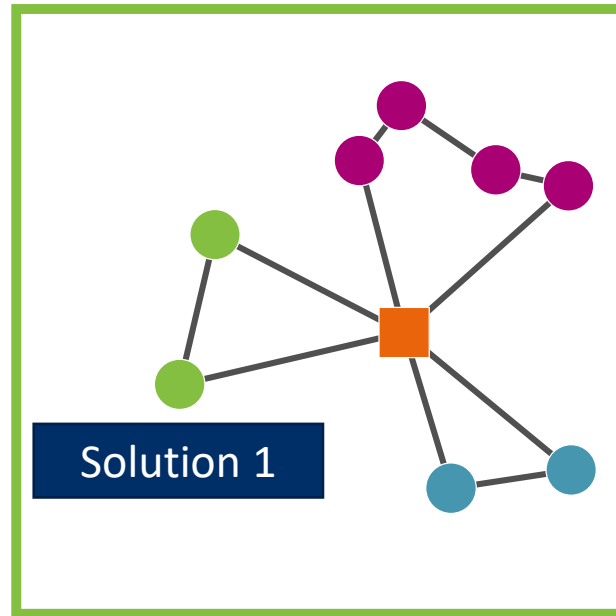
1. Maximize the number of tasks
2. Minimize the distance



Optimization Objectives

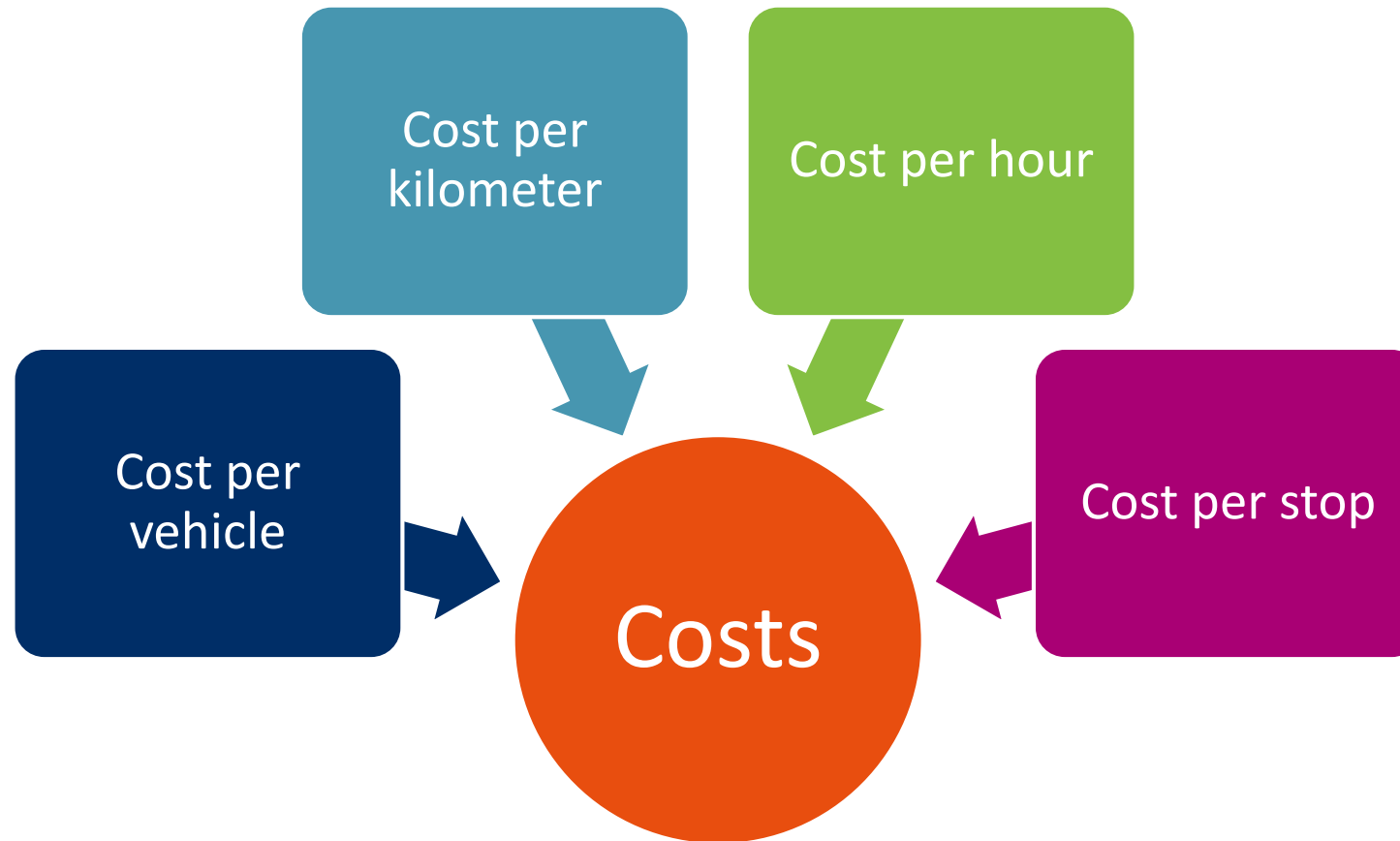
Example

1. Maximize the number of tasks
2. Minimize the distance



Optimization Objectives

PlanCosts



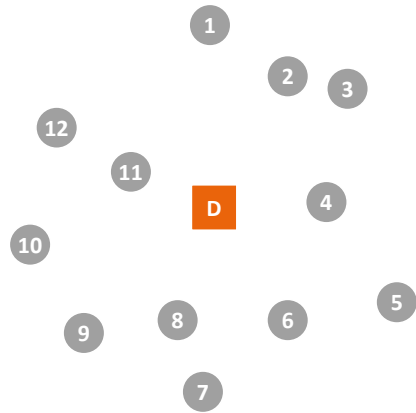


Construction Strategies

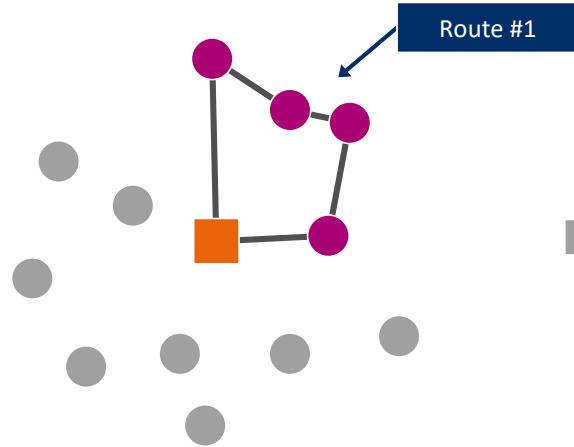
Theory · Basics

Construction · Build Routes

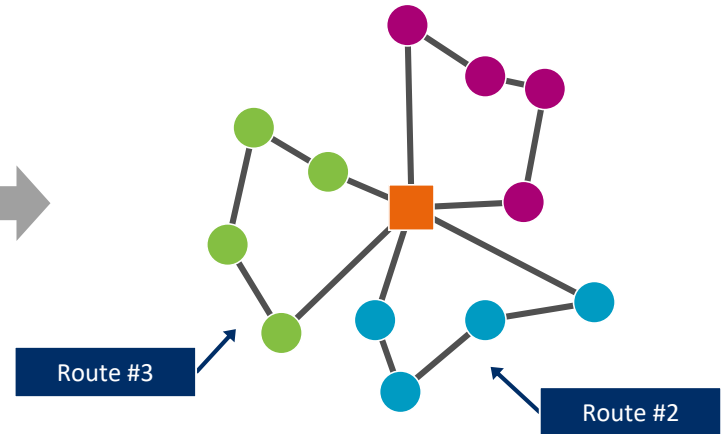
Taking into account the business rules



Unplanned Stops



Build Route #1

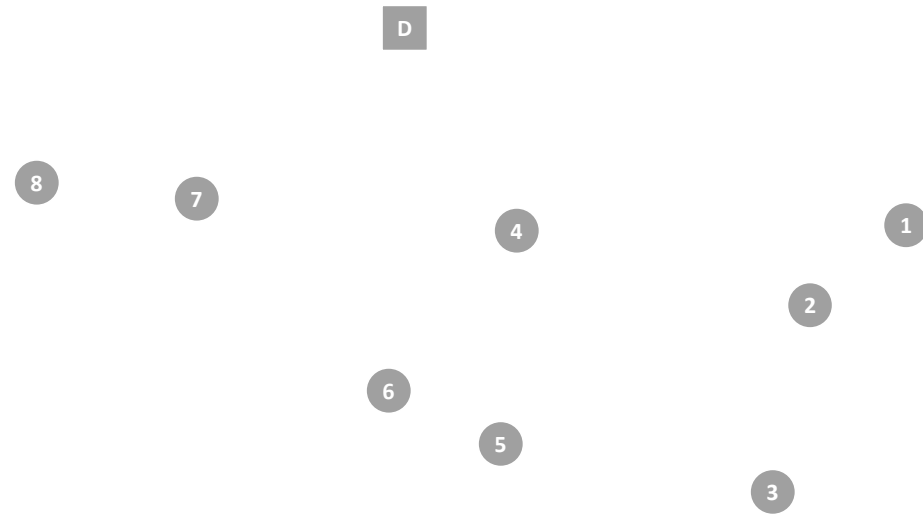


Build Route #2 and #3



Example Construction

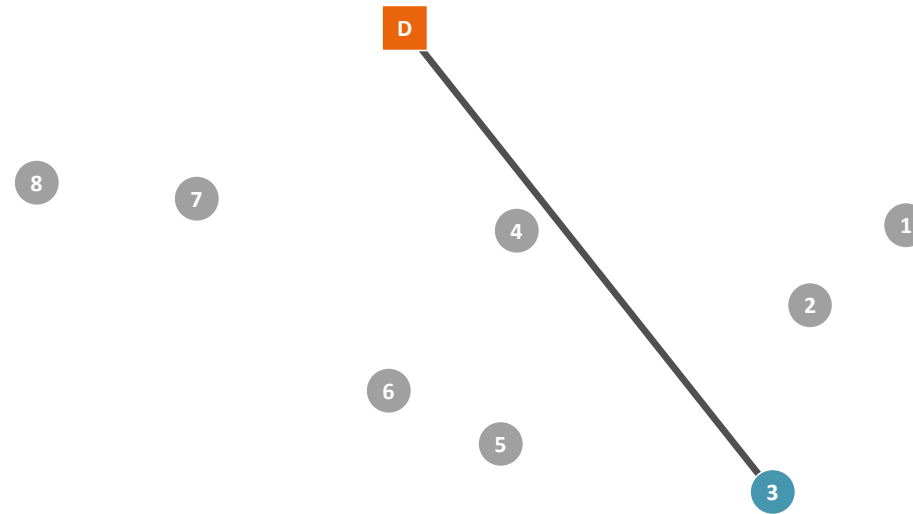
Sequential Insertion



1 Depot · 8 Orders · Maximum 4 orders per trip

Example Construction

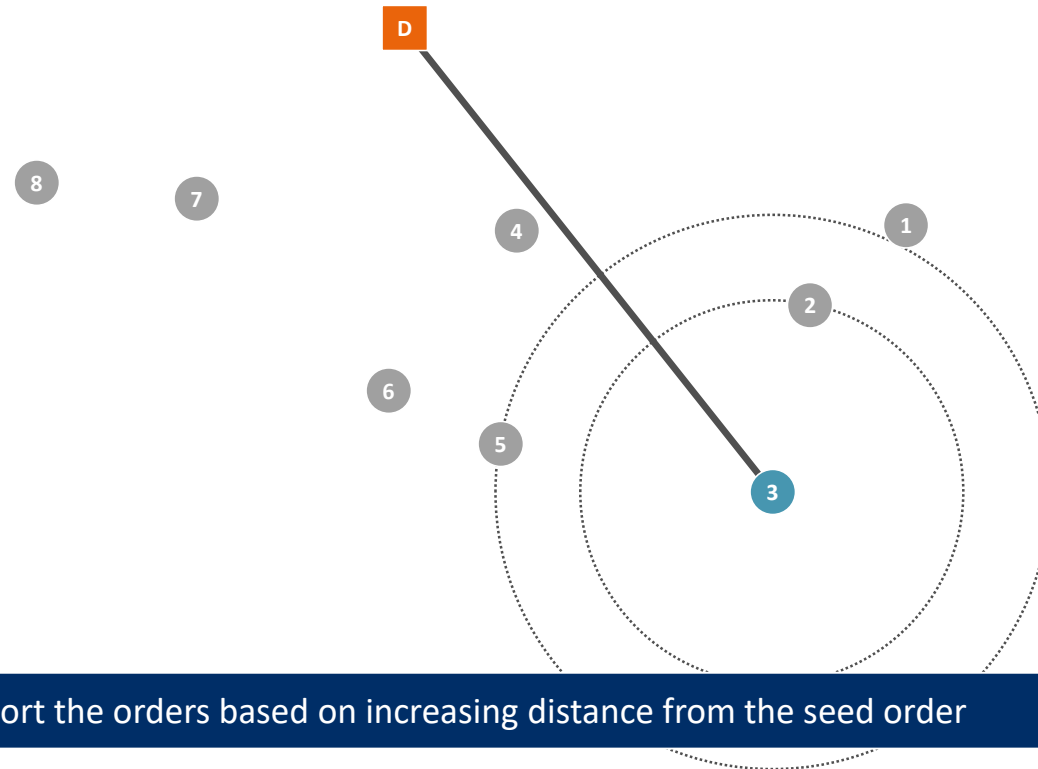
Sequential Insertion



Determine the most difficult (based on distance) order, the seed order

Example Construction

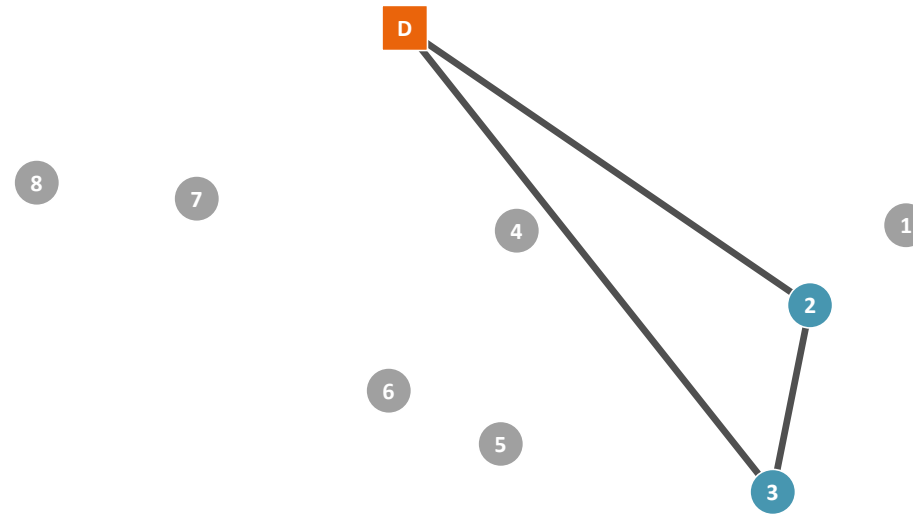
Sequential Insertion



Sort the orders based on increasing distance from the seed order

Example Construction

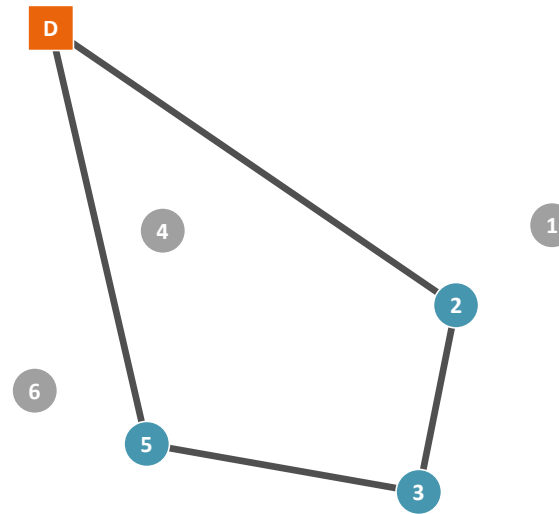
Sequential Insertion



Add the orders in sequence at the best position: 2, 5 and 1

Example Construction

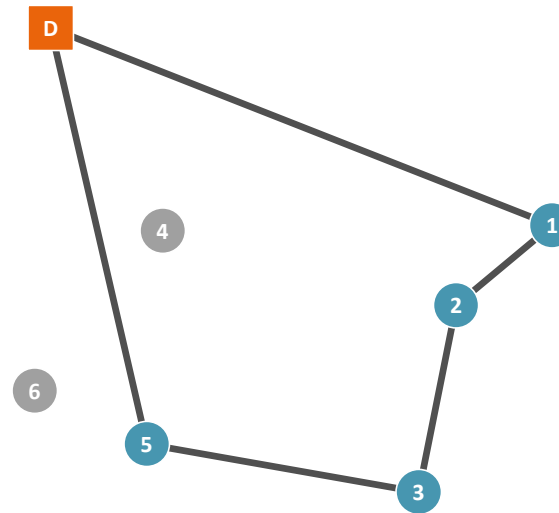
Sequential Insertion



Add the orders in sequence at the best position: 2, 5 and 1

Example Construction

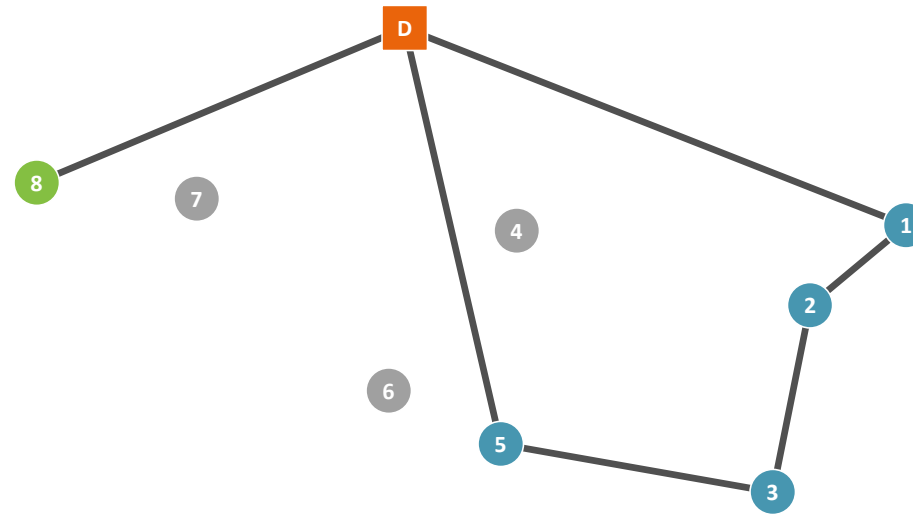
Sequential Insertion



Add the orders in sequence at the best position: 2, 5 and 1

Example Construction

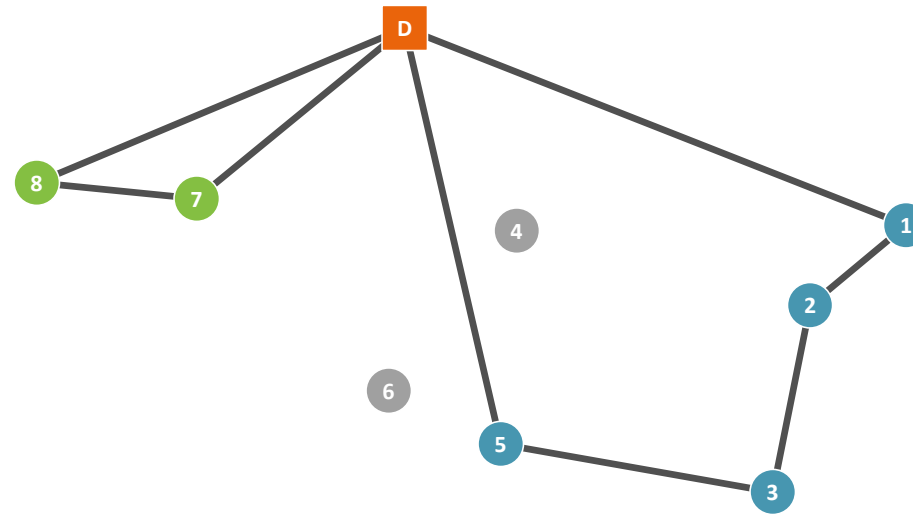
Sequential Insertion



For the 2nd route, the seed order is order 8

Example Construction

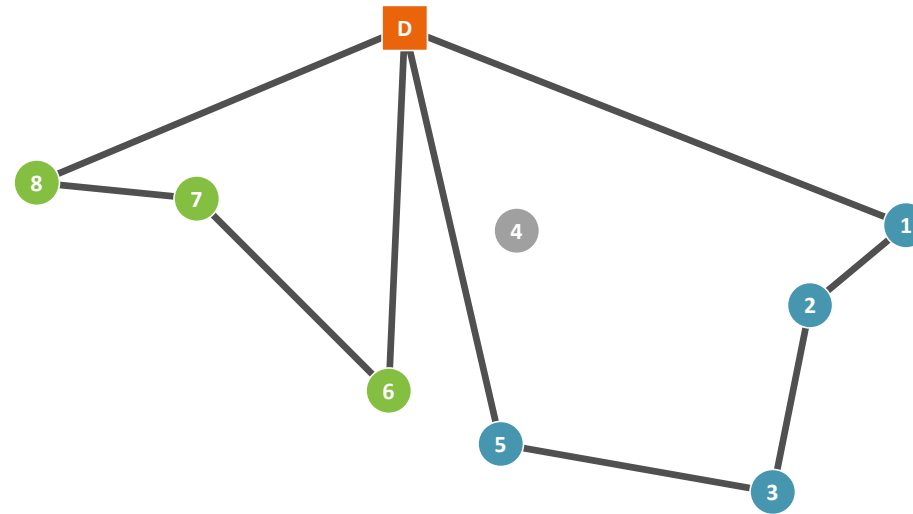
Sequential Insertion



Add the orders in sequence at the best position: 7, 6 and 4

Example Construction

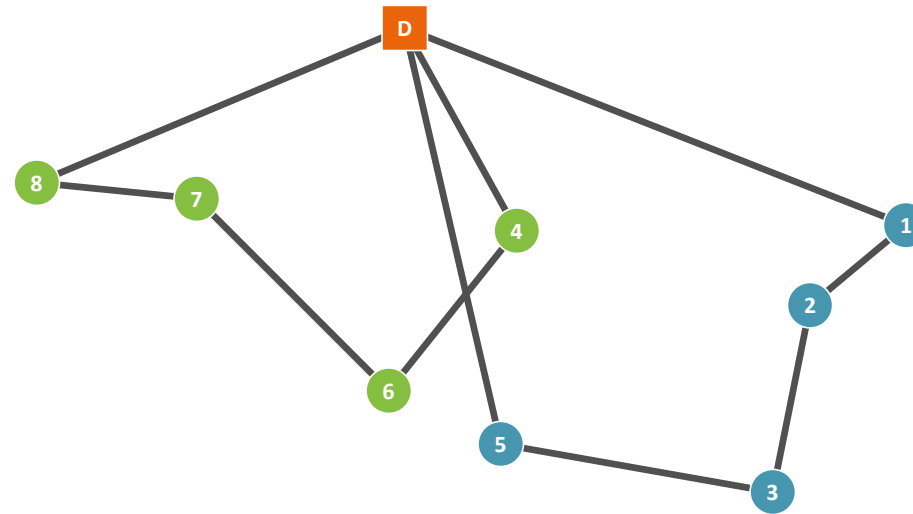
Sequential Insertion



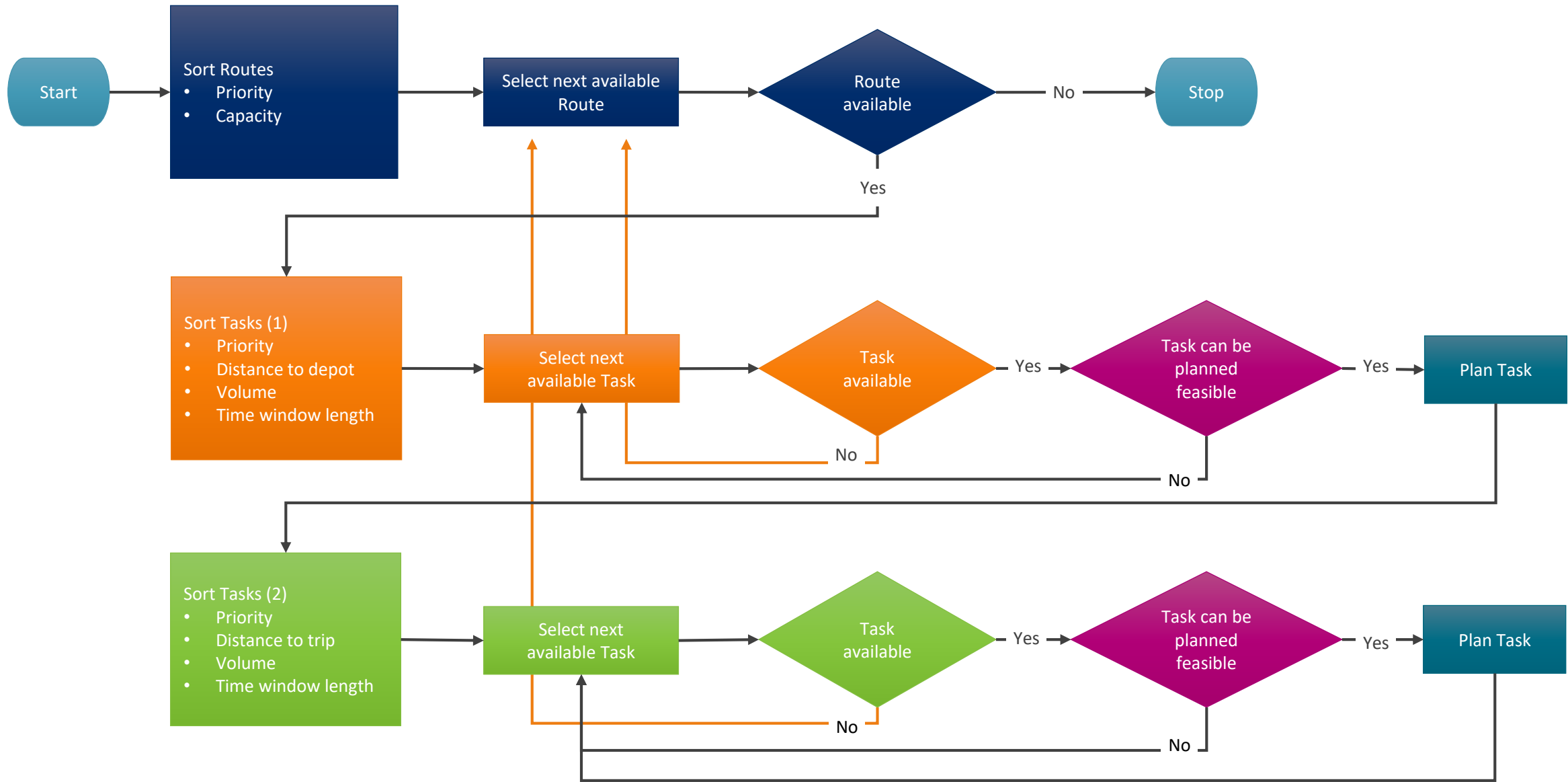
Add the orders in sequence at the best position: 7, 6 and 4

Example Construction

Sequential Insertion



Add the orders in sequence at the best position: 7, 6 and 4



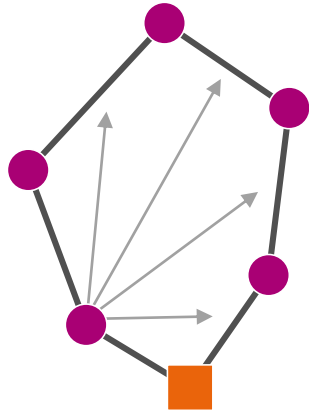


Local Search

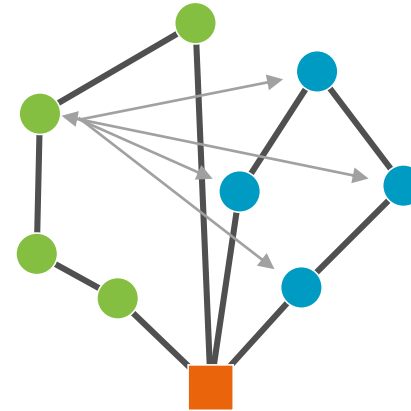
Theory

Local Search · Local Improvements

Multiple improvements methods
Repeated multiple times



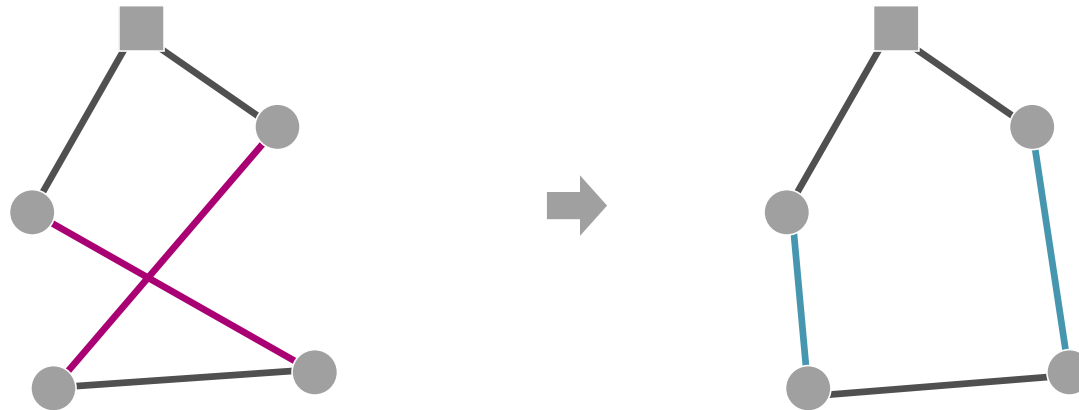
Within Routes



Between Routes

Local Search

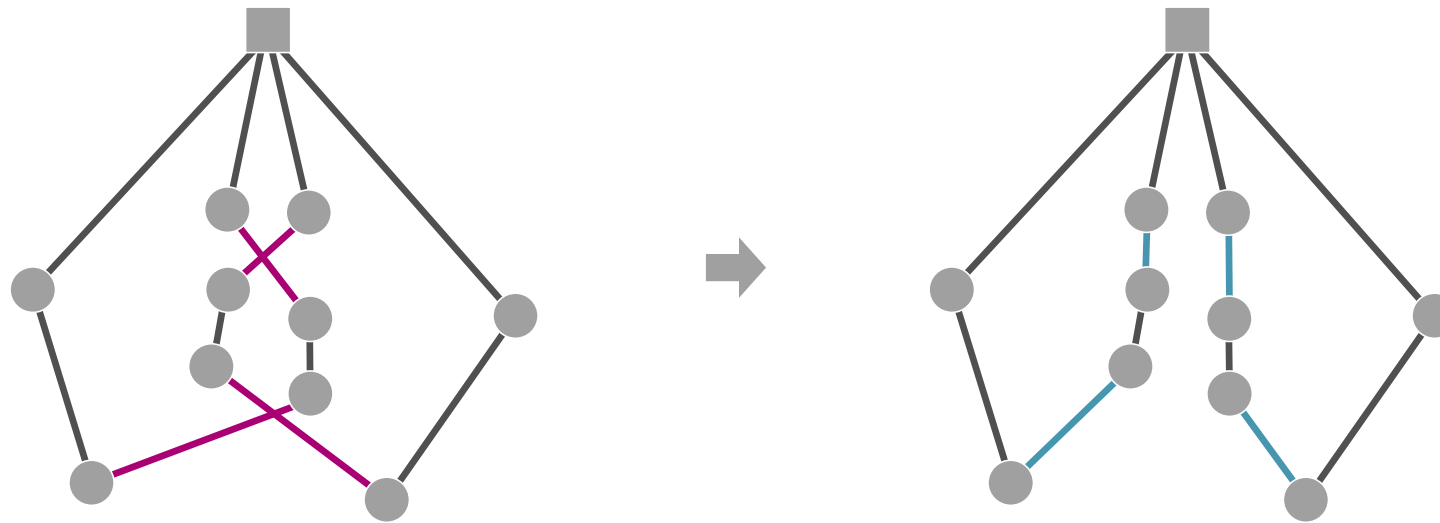
2 Opt · Single Trip



The 2 Opt algorithm improves the solution by redirecting 2 travels

Local Search

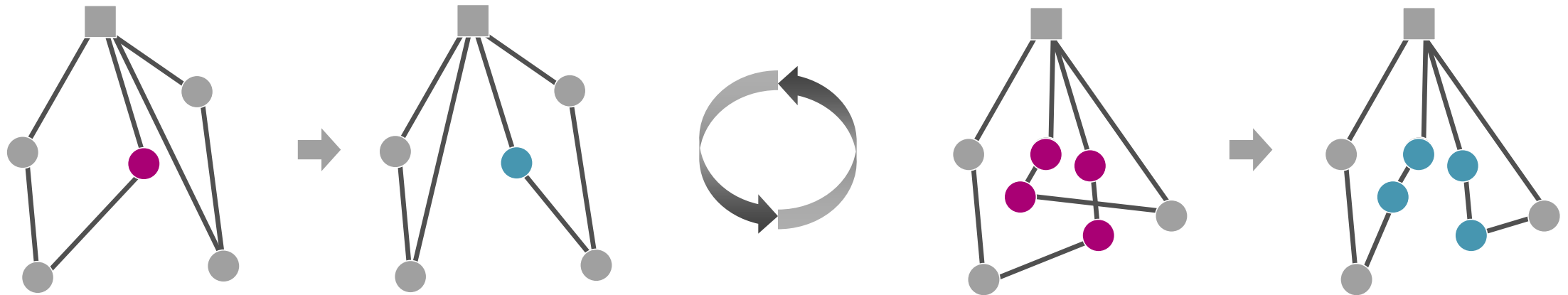
Cross Exchange



The Cross Exchange algorithm improves the solution by redirecting 4 travels

Local Search

Large Neighborhood Move and Swap



Move and Swap algorithm integrates the Move and Swap algorithms into 1 algorithm

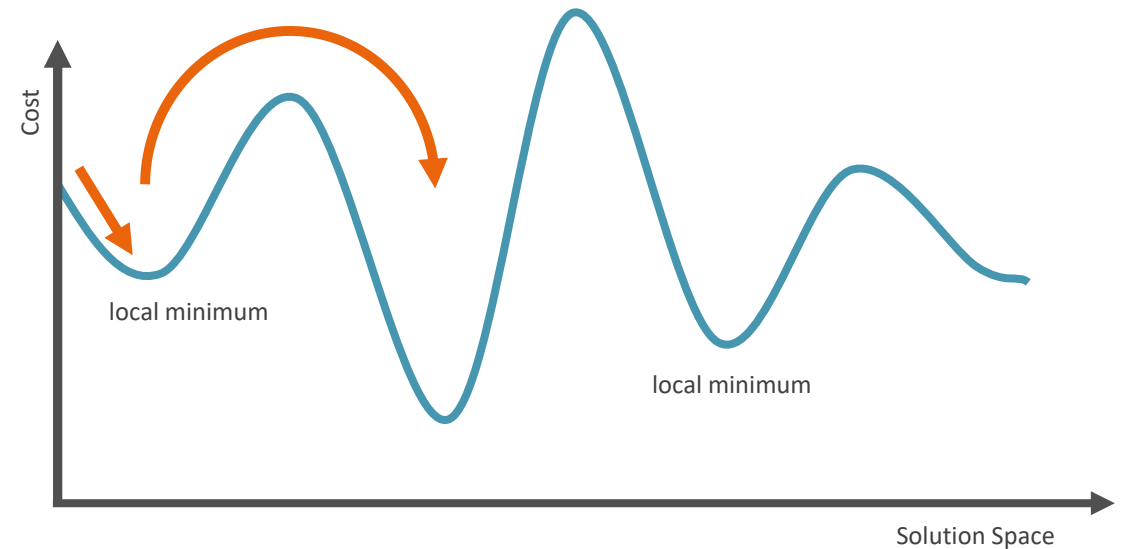


Ruin and Recreate

Ruin and Recreate

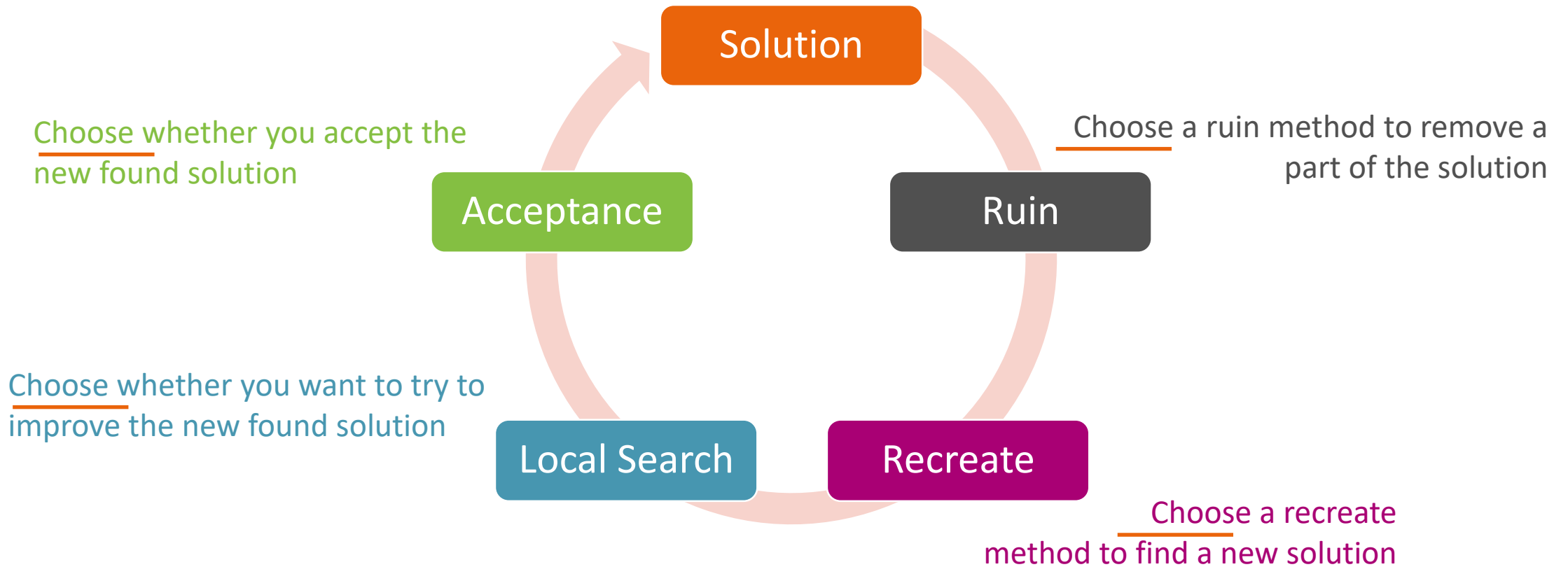
Large Neighborhood Search

- In CVRS, ORTEC implemented technology to escape from local minima: see picture
- This enables the optimizer to find better solutions and approach the global minimum



Ruin and Recreate

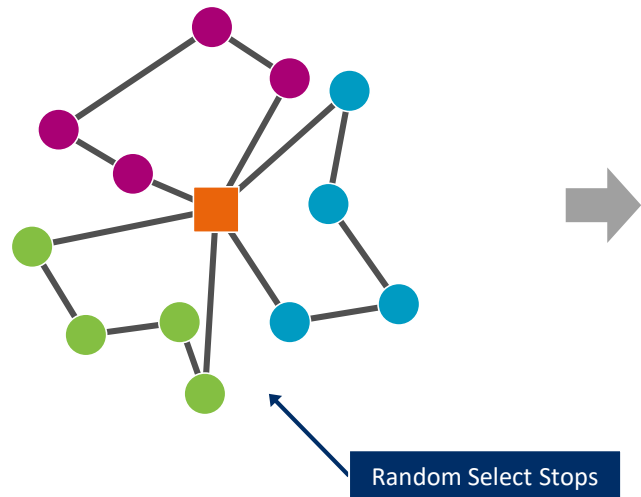
Process



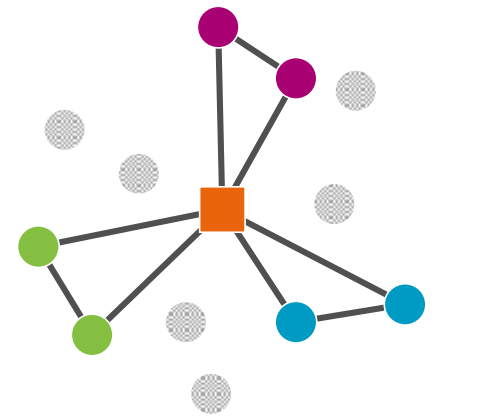


Ruin and Recreate

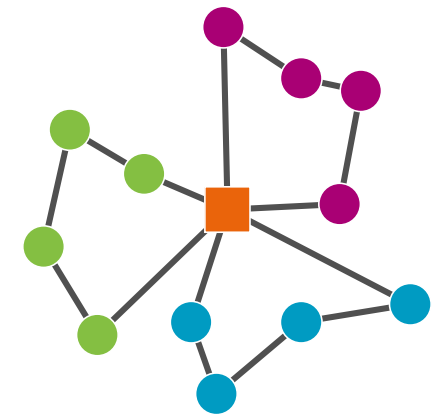
Example



Initial solution



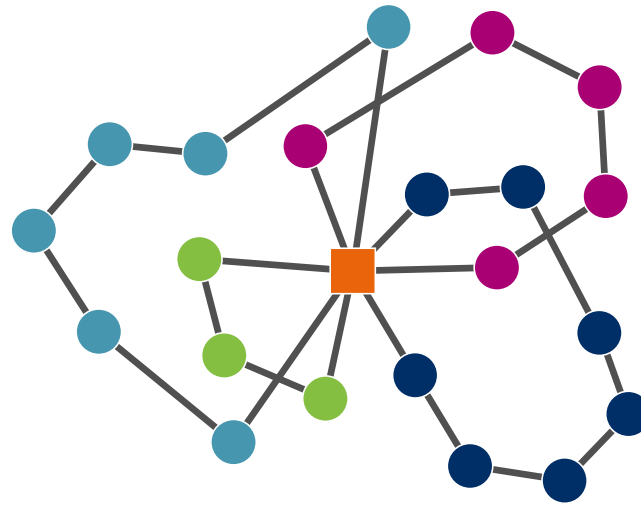
After the Ruin



After the Recreate

Ruin and Recreate

Random Removal

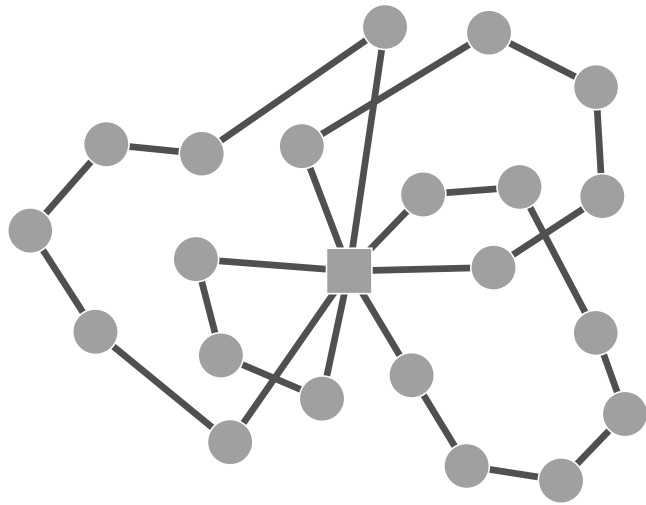


Start Situation

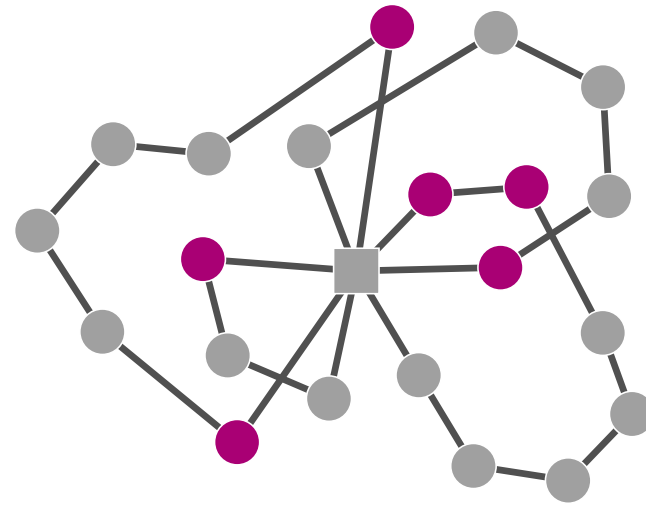
* Example: Pisinger and Ropke, Large Neighborhood Search, Handbook of Metaheuristics, 2010

Ruin and Recreate

Random Removal



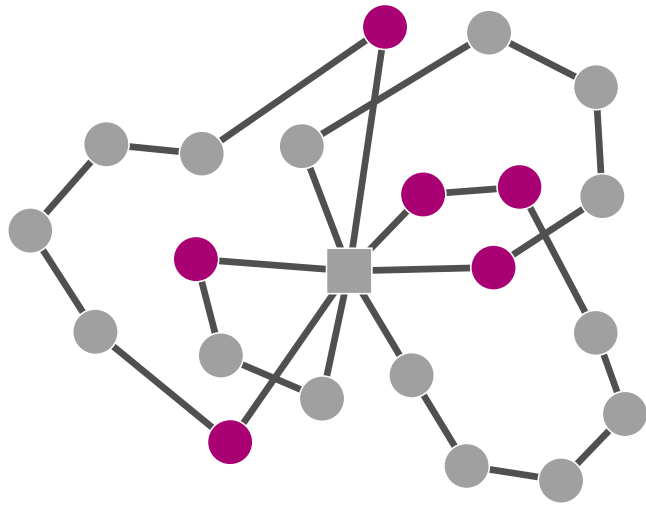
Start Situation



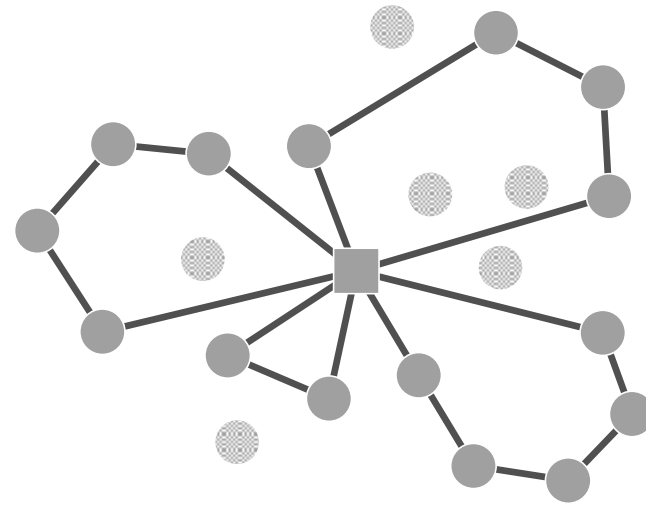
Random Select Tasks

Ruin and Recreate

Random Removal



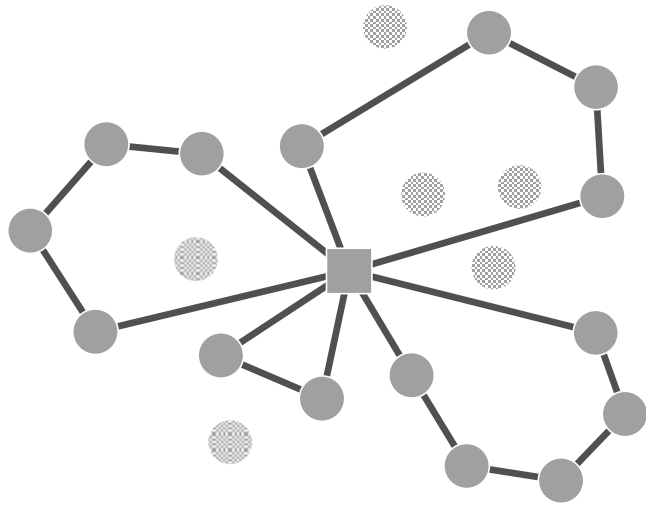
Random Select Tasks



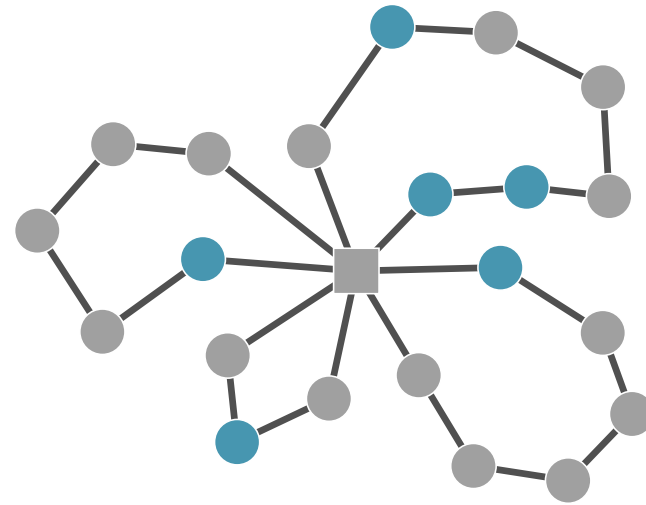
Remove Tasks

Ruin and Recreate

Random Removal



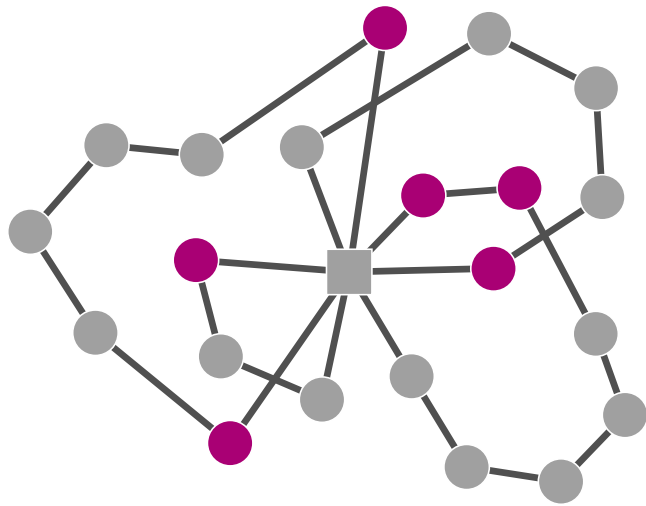
Remove Tasks



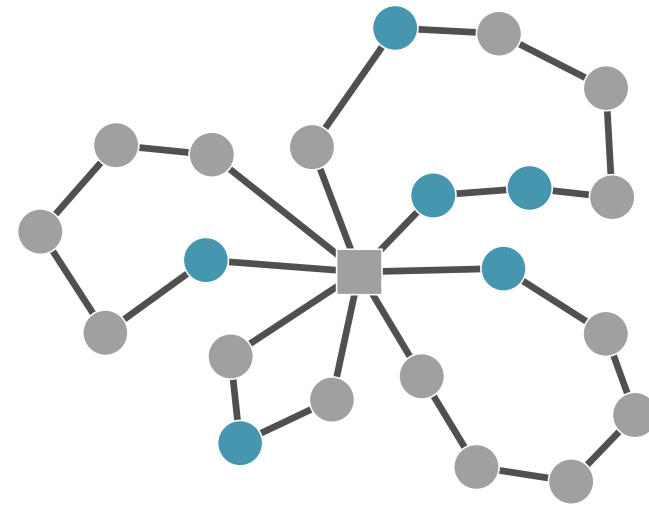
Reinsert Tasks

Ruin and Recreate

Random Removal

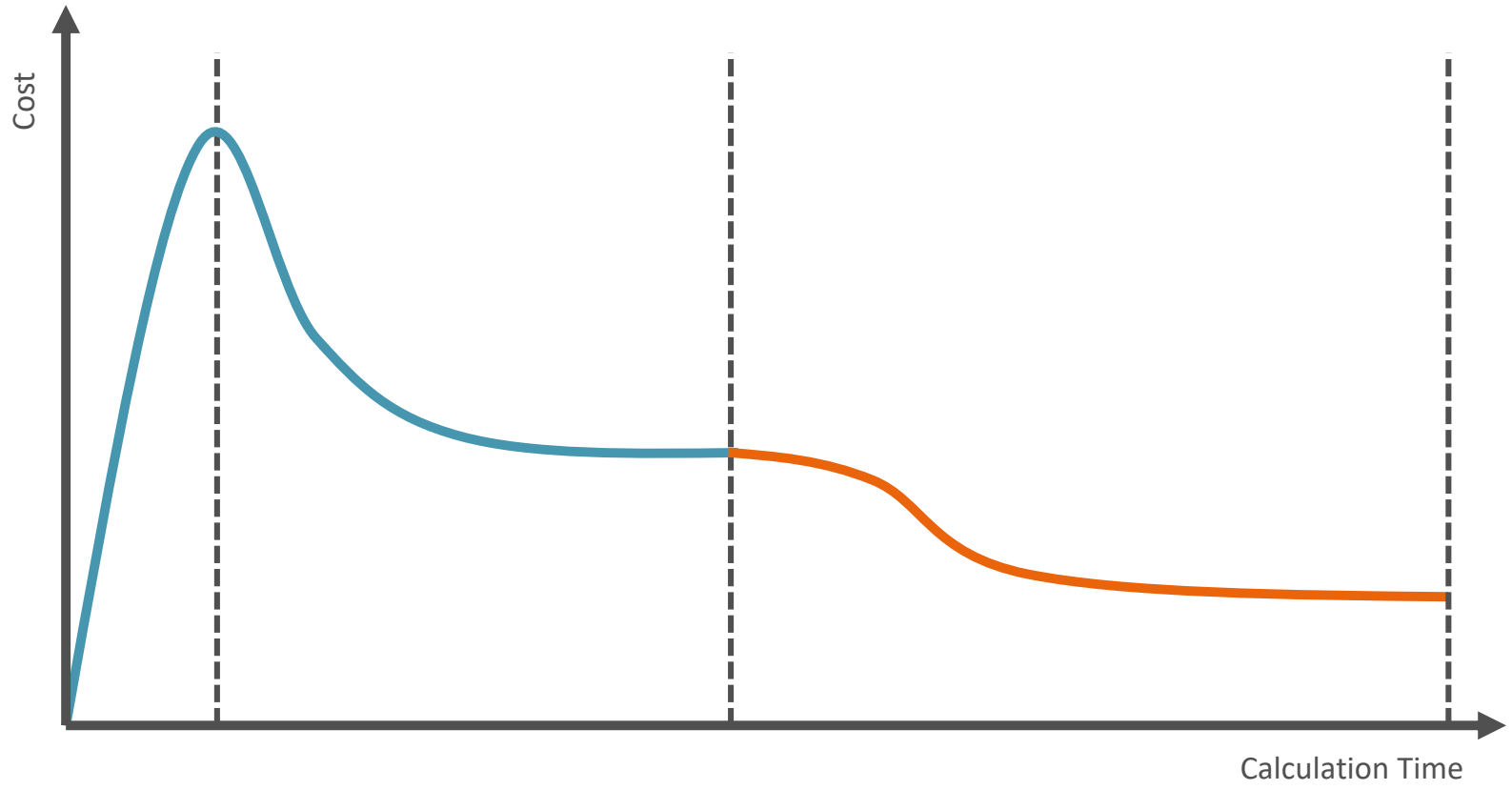


Before



After

Overview of optimization flow



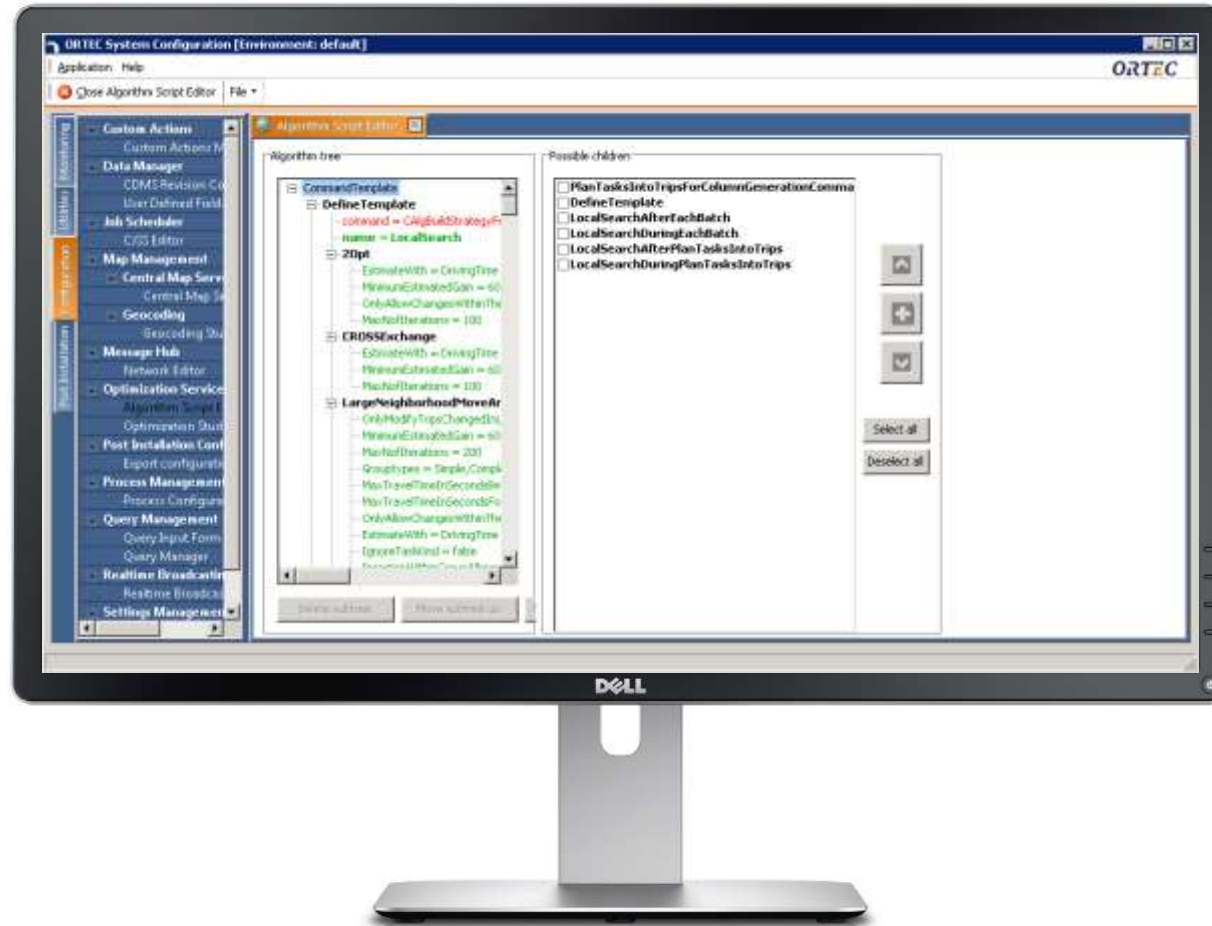


Creation & Activation scripts

Algorithm Script Editor

The Algorithm Script Editor is a general tool for constructing algorithm scripts.

You can find this tool in CSCS under Configuration > Optimization Service > Algorithm Script Editor



- Monitoring
 - Custom Actions
 - Custom Actions Manager (Web)
 - Data Manager
 - Data Cleaner
 - CDMS Revision Configuration
 - User Defined Fields Manager
 - Job Scheduler
 - CJSS Editor
 - Map Management
 - Central Map Server
 - Central Map Server configuration
 - Geocoding
 - Geocoding Studio
 - Message Hub
 - Network Editor
 - Optimization Service
 - Algorithm Script Editor
 - Post Installation Configuration
 - Export configuration
 - Process Management
 - Process Configuration
 - Query Management
 - Query Manager
 - Search Form Designer
 - Realtime Broadcasting Service
 - Realtime Broadcasting Designer
 - Settings Management
 - Settings Manager
 - System Monitor
 - System Monitor Configurator

Algorithm Script Editor

Algorithm tree

- CommandTemplate
 - GlobalSettings
 - DefineTemplate name = LocalSearchWithinTripAlgorithms
 - DefineTemplate name = LocalSearchBetweenTripAlgorithms
 - DefineTemplate name = LocalSearchWithinTrip
 - DefineTemplate name = LocalSearchBetweenTrip
 - DefineTemplate name = LocalSearch
 - DefineTemplate name = LocalSearchFast
 - DefineTemplate name = Recreate_A
 - DefineTemplate name = Recreate_B
 - DefineTemplate name = RelatedRemoval_1_A
 - DefineTemplate name = RelatedRemoval_1_B
 - DefineTemplate name = RelatedRemoval_2_A
 - DefineTemplate name = RelatedRemoval_2_B
 - DefineTemplate name = RandomRemoval_1_A
 - DefineTemplate name = RandomRemoval_1_B
 - DefineTemplate name = RandomRemoval_2_A
 - DefineTemplate name = RandomRemoval_2_B
 - DefineTemplate name = RouletteWheel_1
 - DefineTemplate name = RouletteWheel_2
 - DefineTemplate name = RuinAndRecreate
 - DefineTemplate name = PlanRemainingTasks
 - DefineTemplate name = PlanTasksCloseToSeed
 - DefineTemplate name = PlanSeedTask
 - DefineTemplate name = SequentialInsertionConstruction
 - DefineTemplate name = BatchOptimization
 - PlanTasksInTripsCommand
 - ReoptimizeTasksInTripsCommand

Objectives

Local Search

Ruin and Recreate

Construction

Main

Possible children

- UserDefinedTemplate
- AddSolutionIdentifier
- Recursion
- InsertTasksUsingCheapestInsertion
- InsertTasksUsingParallelCheapestInsertion
- InsertTasksUsingParallelRegretInsertion
- ClusterTasksAtSameAddress
- AlgModifyObjective
- DefineNearestNeighbors
- DefineClosestTrips
- AdjustRestrictionSet
- 2Opt
- RearrangeTasksAtSameAddress
- CROSSExchange
- LargeNeighborhoodSwap
- LargeNeighborhoodMove
- LargeNeighborhoodMoveAndSwap
- CyclicTransfer
- OptimizeResources
- RemoveTasksUsingParallelCheapestRemoval
- RandomRemoval
- RelatedRemoval

Select all

Deselect all

DefineTemplate:
This user defined template enables you to define a template with a specific name. Under the template you can add an arbitrary number of commands. You can then use the template anywhere you like by adding a UserDefinedTemplate with the same specific name. The commands under the template will then be executed at all the places where the UserDefinedTemplate is used.

Optimization Script

Delete subtree Move subtree up Move subtree

Algorithm Script Editor

Standard Scripts

Retail

```
CommandPalette
Define Template
  command = C:\git\algorithms\bin\get
  name = LocalSearch
  20pt
    EstimateWith = DrivingTime
    MinimumEstimatedSize = 60
    OnlyAllowChangesWhenTheServer
    MaxNumberOfItems = 100
  CROSSExchange
    EstimateWith = DrivingTime
    MinimumEstimatedSize = 0
    MaxNumberOfItems = 100
  LargeNeighborhoodMoveA
    OnlyModifyTripsChangedIn
    MinimumEstimatedSize = 0
    MaxNumberOfItems = 100
    GroupTypes = Simple,Comp
    MaxTravelTimeInSeconds
    MaxTravelTimeInSeconds
    OnlyAllowChangesWhenThe
    EstimateWith = DrivingTime
    IgnoreTaskIn = false
    InsertionWithGroupAllowed
  20pt
    EstimateWith = DrivingTime
    MinimumEstimatedSize = 0
    OnlyAllowChangesWhenThe
    MaxNumberOfItems = 100
  CROSSExchange
    EstimateWith = DrivingTime
    MinimumEstimatedSize = 0
    MaxNumberOfItems = 100
  LargeNeighborhoodMoveA
    OnlyModifyTripsChangedIn
    MinimumEstimatedSize = 0
    MaxNumberOfItems = 100
    GroupTypes = Simple,Comp
    MaxTravelTimeInSeconds
    MaxTravelTimeInSeconds
```

Distribution

But standard scripts are often not good enough!

Planning

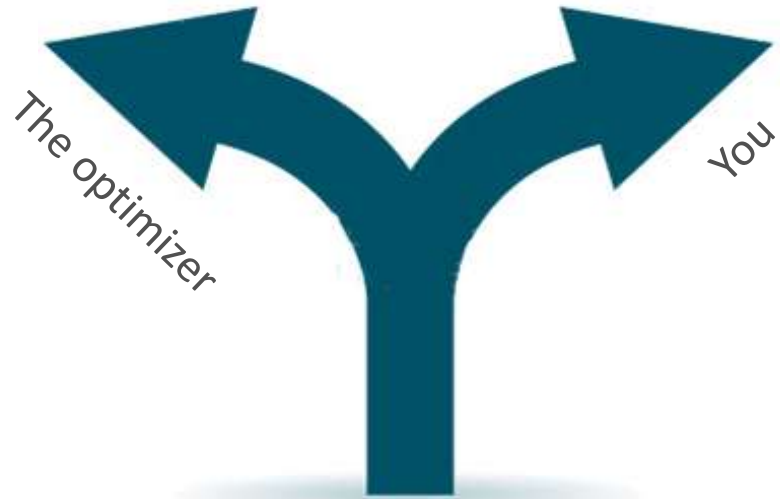
```
GroupTypes = Simple,CompleteTrips
MaxTravelTimeInSecondsBetweenConsecutiveTasksInAGroup
MaxTravelTimeInSecondsForSameTaskWithTabbyToSameGroup
OnlyAllowChangesWhenTheServerIs = false
MaxNumberOfItems = 100
CROSSExchange
  EstimateWith = DrivingTime
  MinimumEstimatedSize = 60
  MaxNumberOfItems = 100
LargeNeighborhoodMoveA
  OnlyModifyTripsChangedInLastIteration = false
  MinimumEstimatedSize = 60
  MaxNumberOfItems = 100
  GroupTypes = Simple,CompleteTrips
  MaxTravelTimeInSecondsBetweenConsecutiveTasksInAGroup = 300
  MaxTravelTimeInSecondsForSameTaskWithTabbyToSameGroup = pickup=300,deliver=300,drive_through=300
```



Unexpected Results

It's not working!

Unexpected results



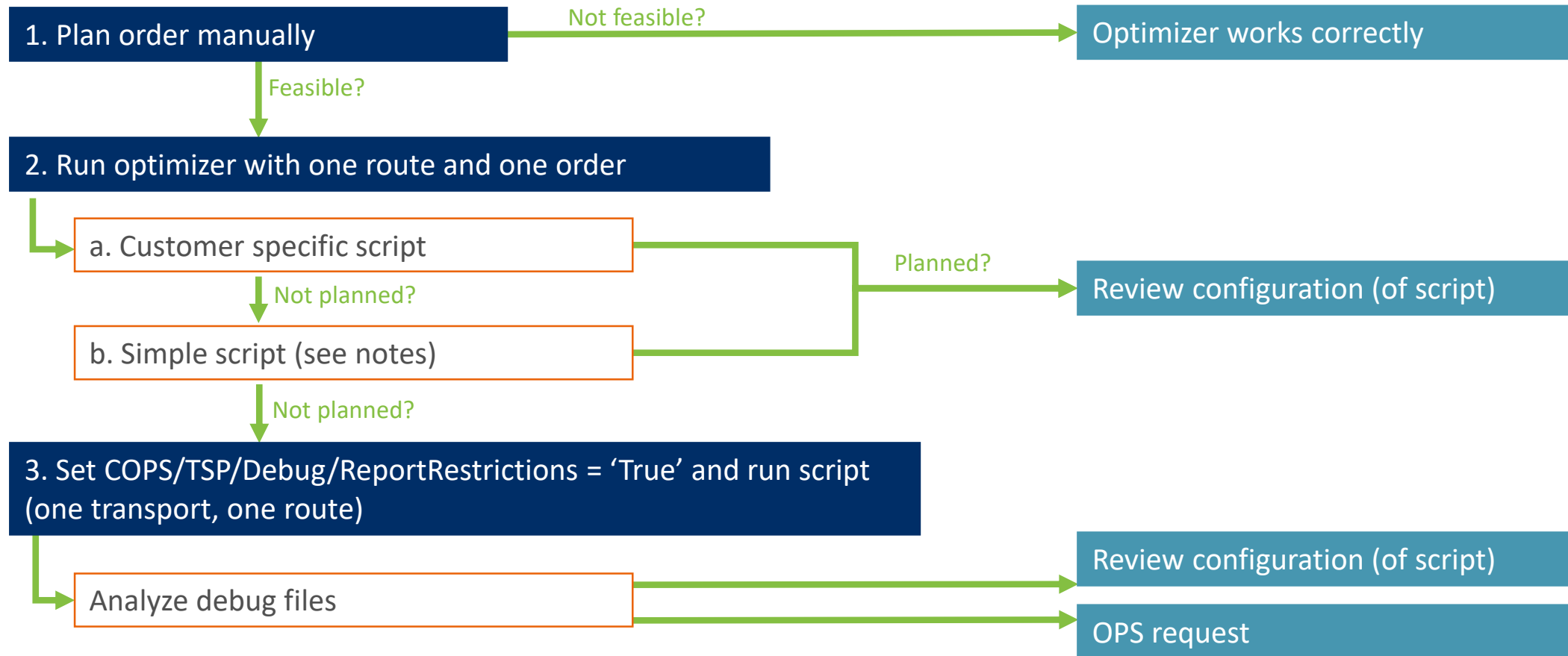
Investigate

Transport/Task not
planned

Transport/Task
planned at
unexpected position

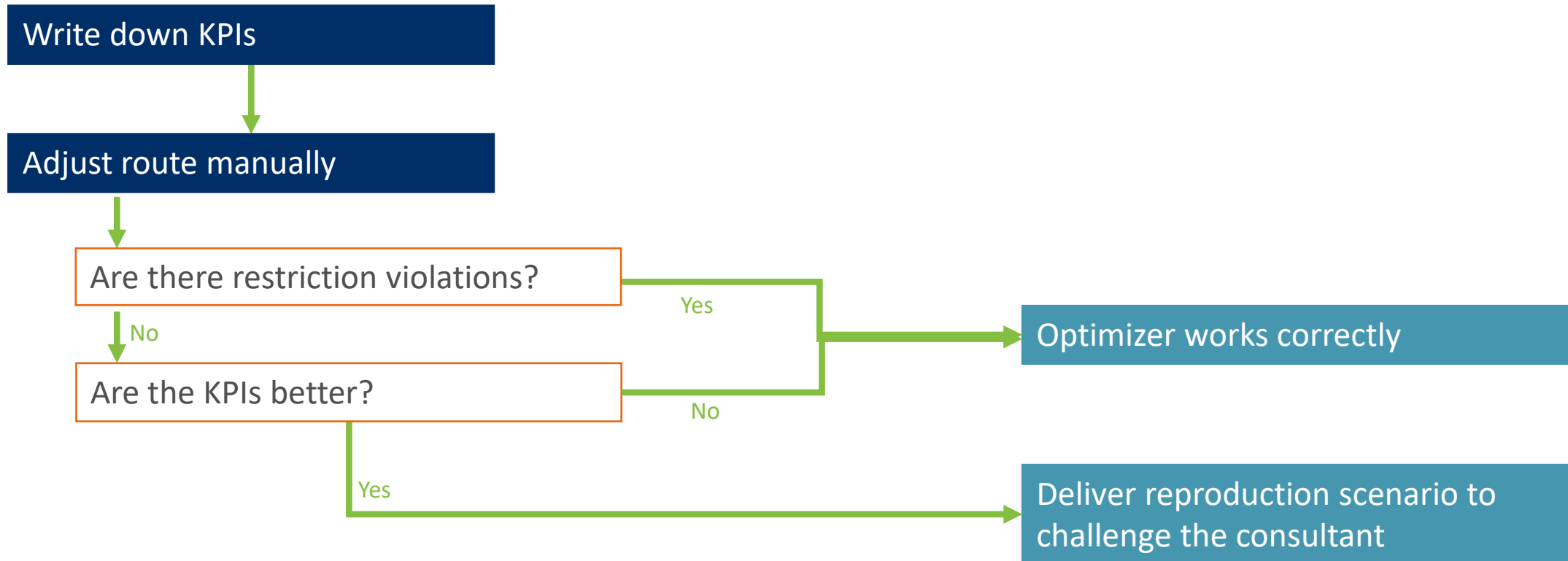
Transport/Task not planned

Steps for analysis



Unexpected result

Challenge the planner

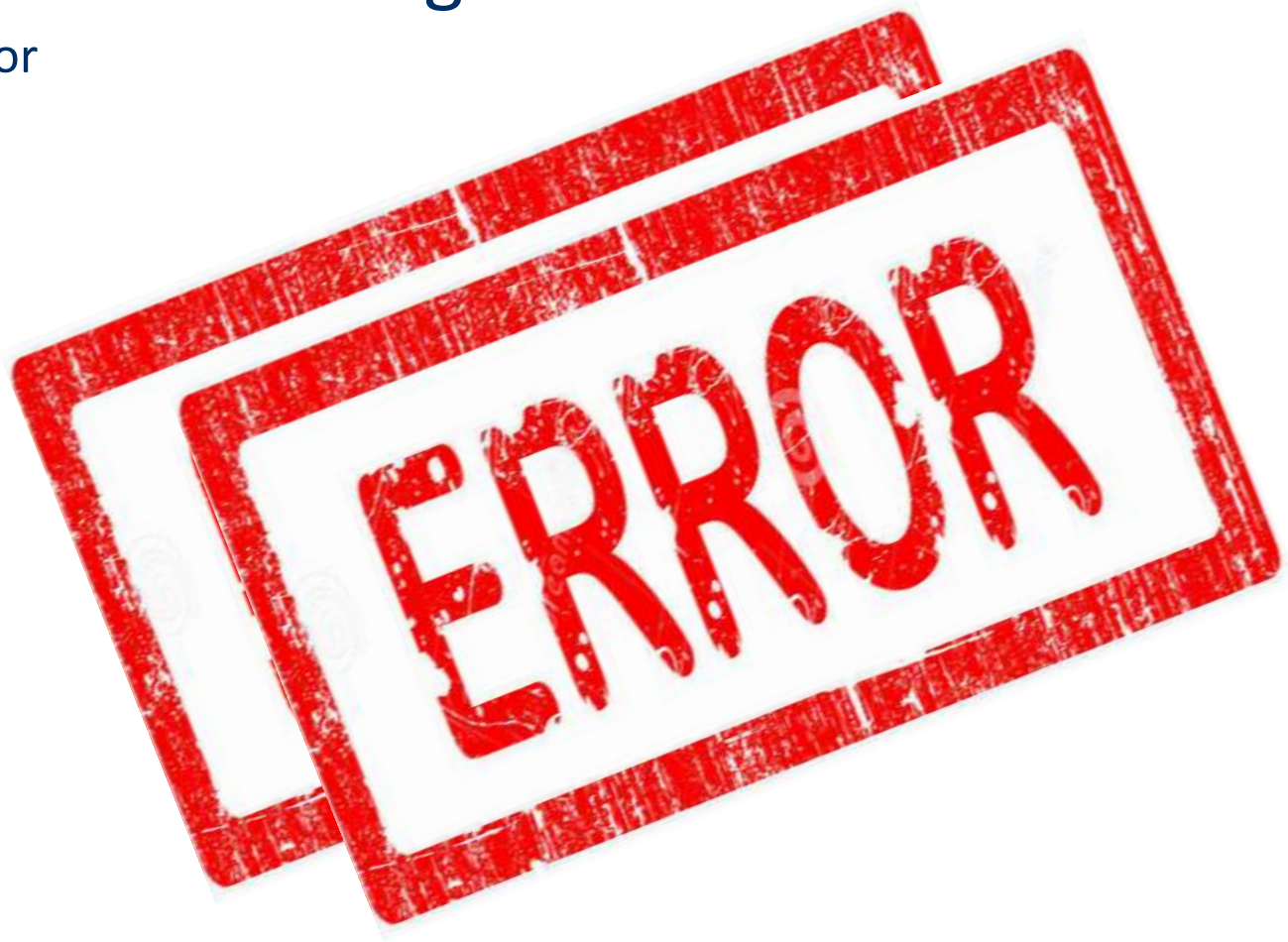




Errors

It's not working!

Error



Read the error message

Configuration

Optimization

Configuration error

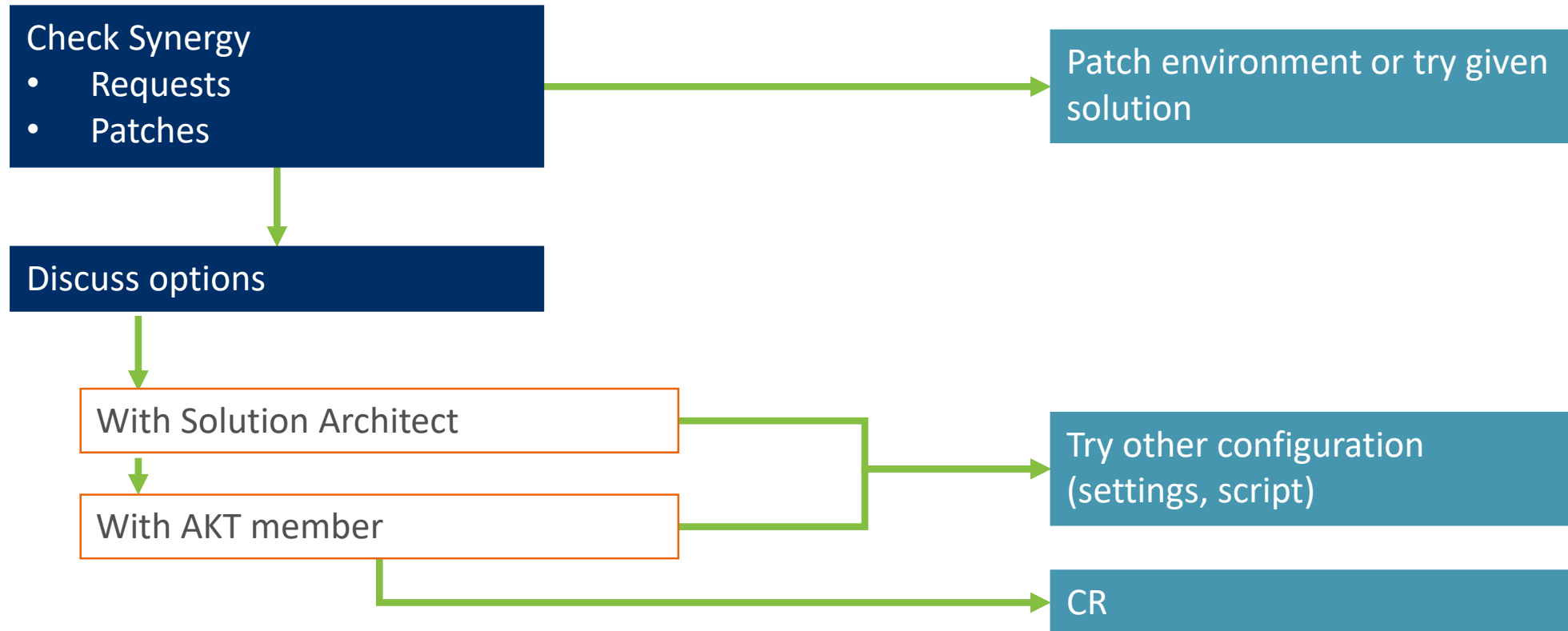
Not supported configuration

Approved	Severity	Type	Description
<input type="checkbox"/>	Error	Optimalisatie	route 16270ORS bevat verandering in de resources

Approved	Severity	Type	Description
<input type="checkbox"/>	Error		Using CVRS with sections requires a valid section model as input. The following inconsistency was found: Section 16443 contains fixations. Addresses, start

Configuration error

Not supported configuration



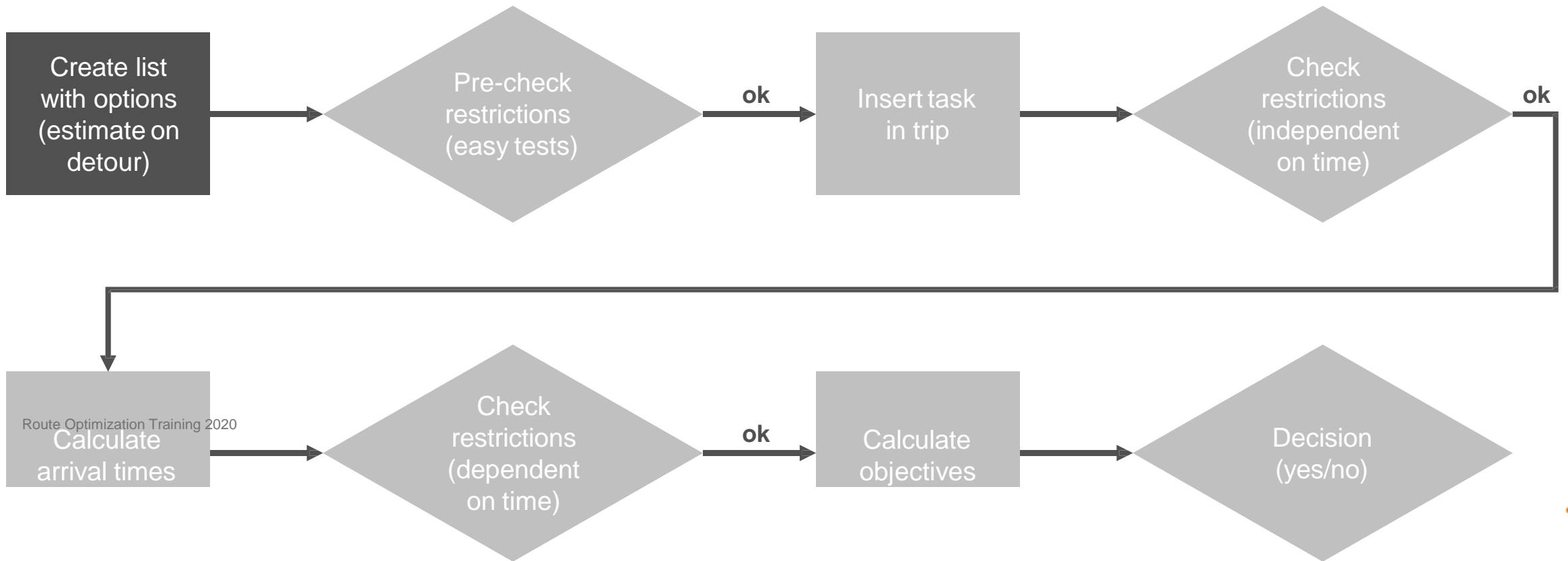


Overview

ORTEC

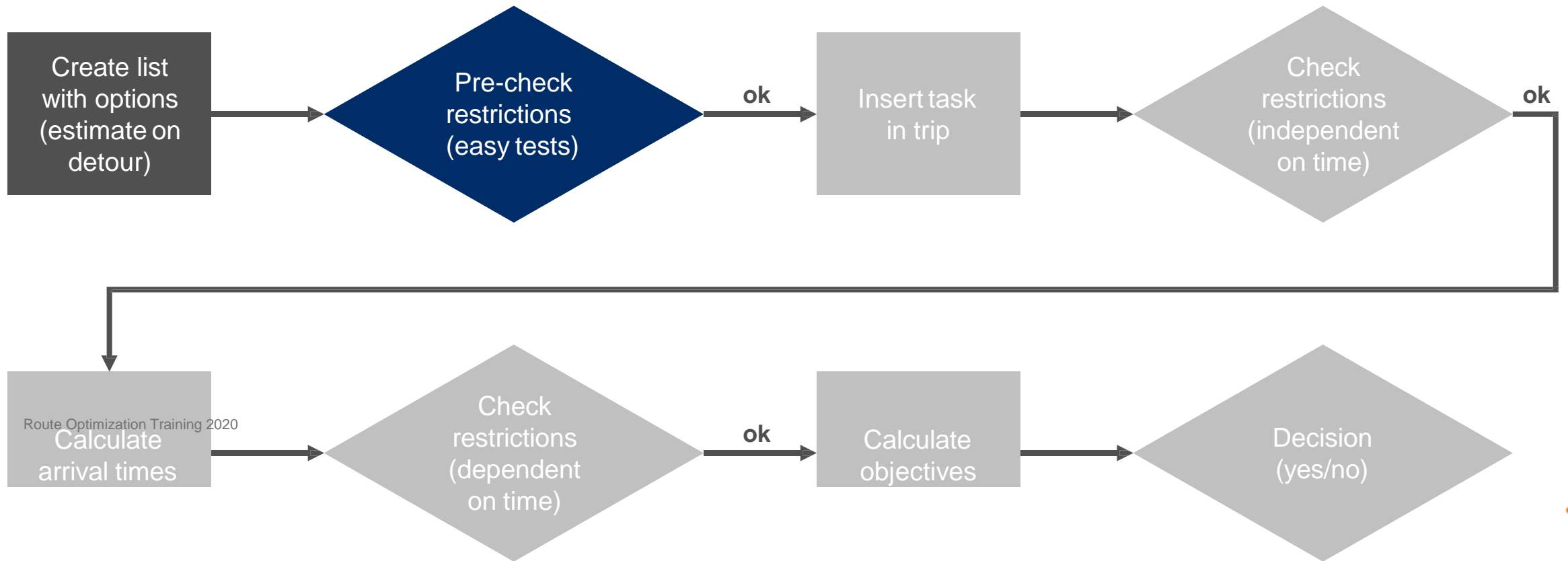
Route Optimization

Under the Hood



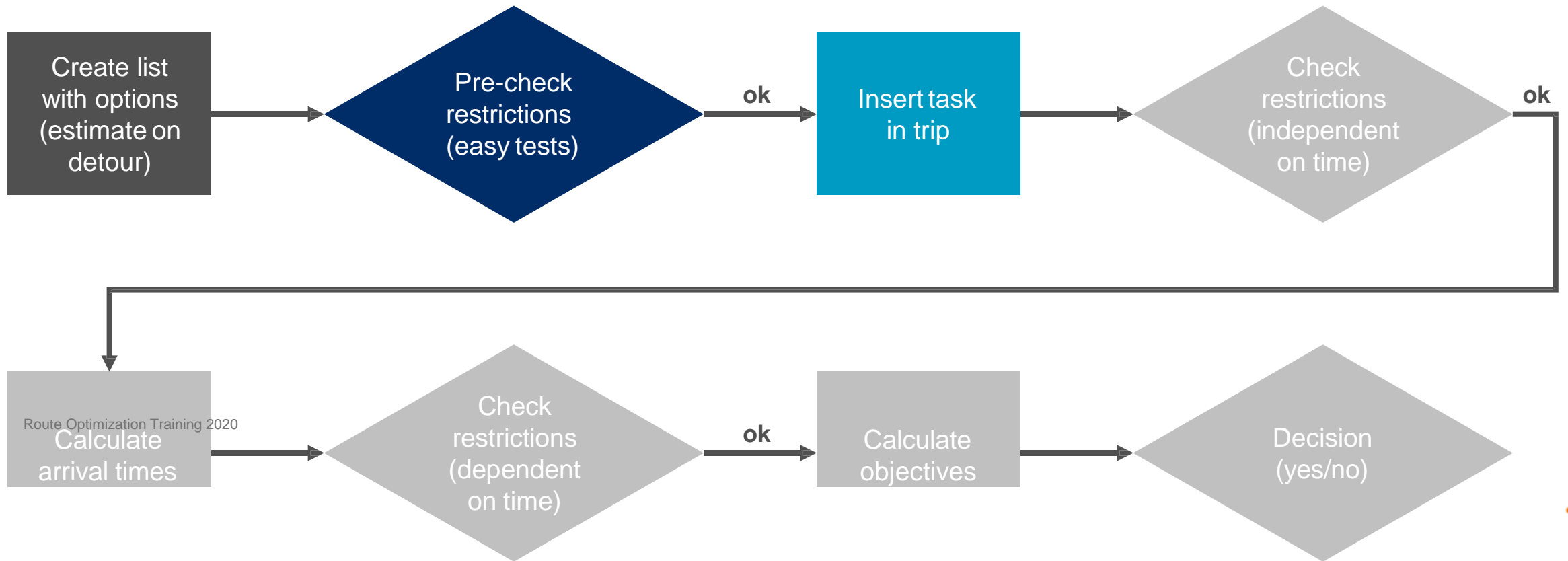
Route Optimization

Under the Hood



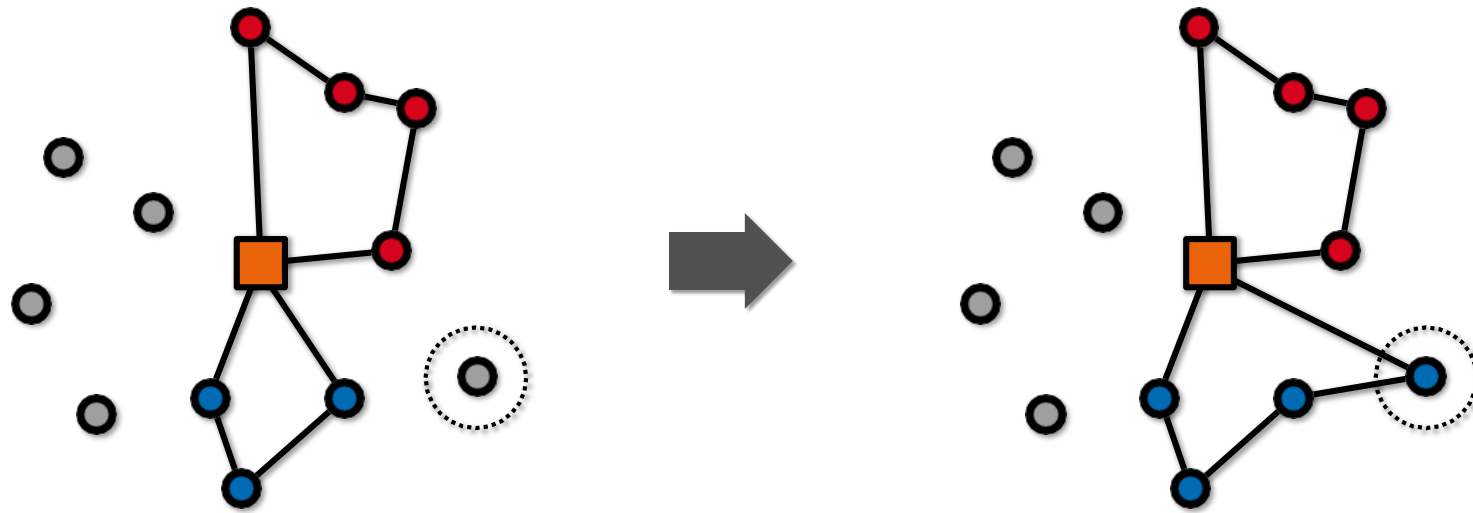
Route Optimization

Under the Hood



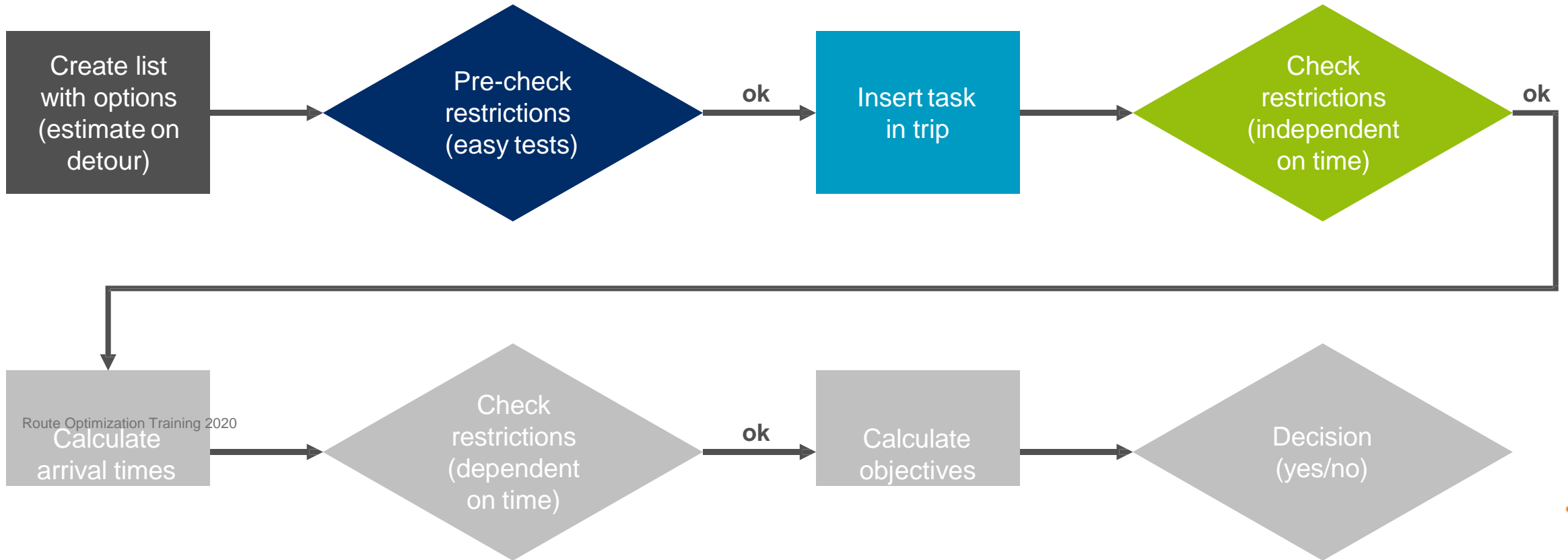
Route Optimization

Insert Task in Trip



Route Optimization

Under the Hood



Route Optimization

Check Restrictions · Independent on Time

Vehicle

- Capacity of the vehicle: volume, weight, etc.
- Equipment restrictions: does a vehicle has the right equipments? (e.g. meter equipment)
- Forbidden resource kinds: can a truck enter a location?

Driver

- Capabilities: fixed routes for drivers
- Plan regions: regions to which a driver 'normally' goes
- Maximum number of stops per route

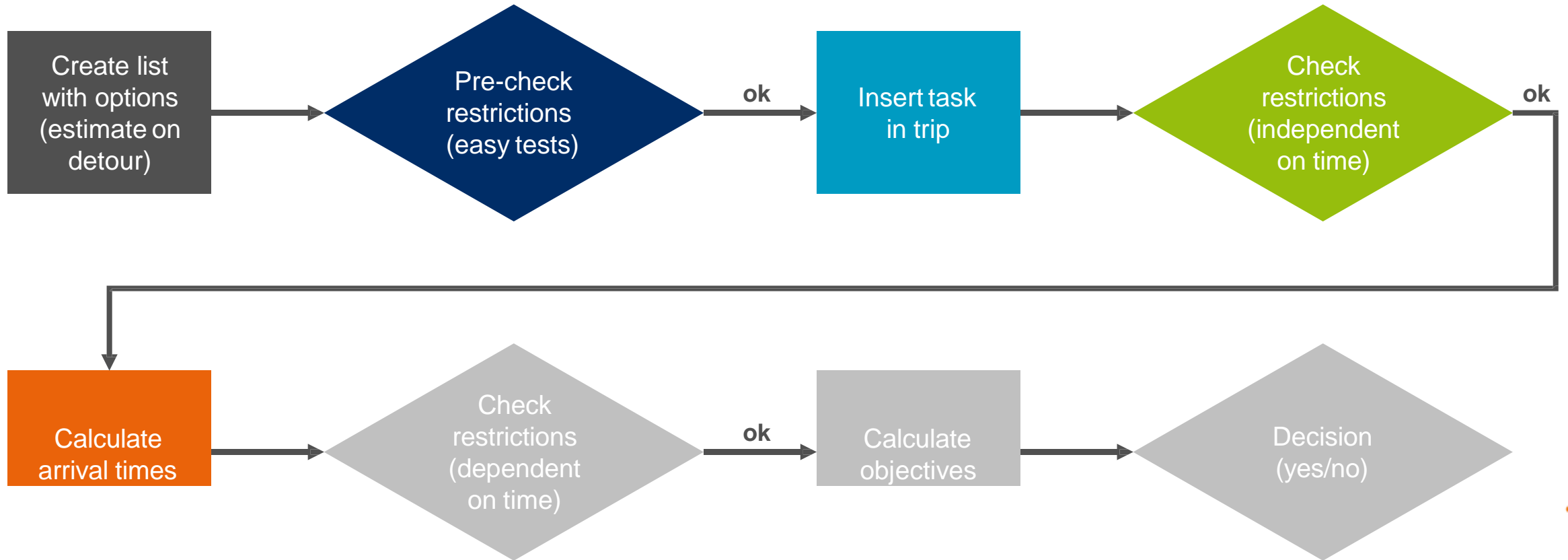
Order

- Position in the trip: first, last, no preference



Route Optimization

Under the Hood



79



Route Optimization

Calculate Arrival Times

Taking into account:

- Start time of the driver
- Depot open and close times
- Address open and close times
- Order time windows
- EU Legislation for pauses and rests

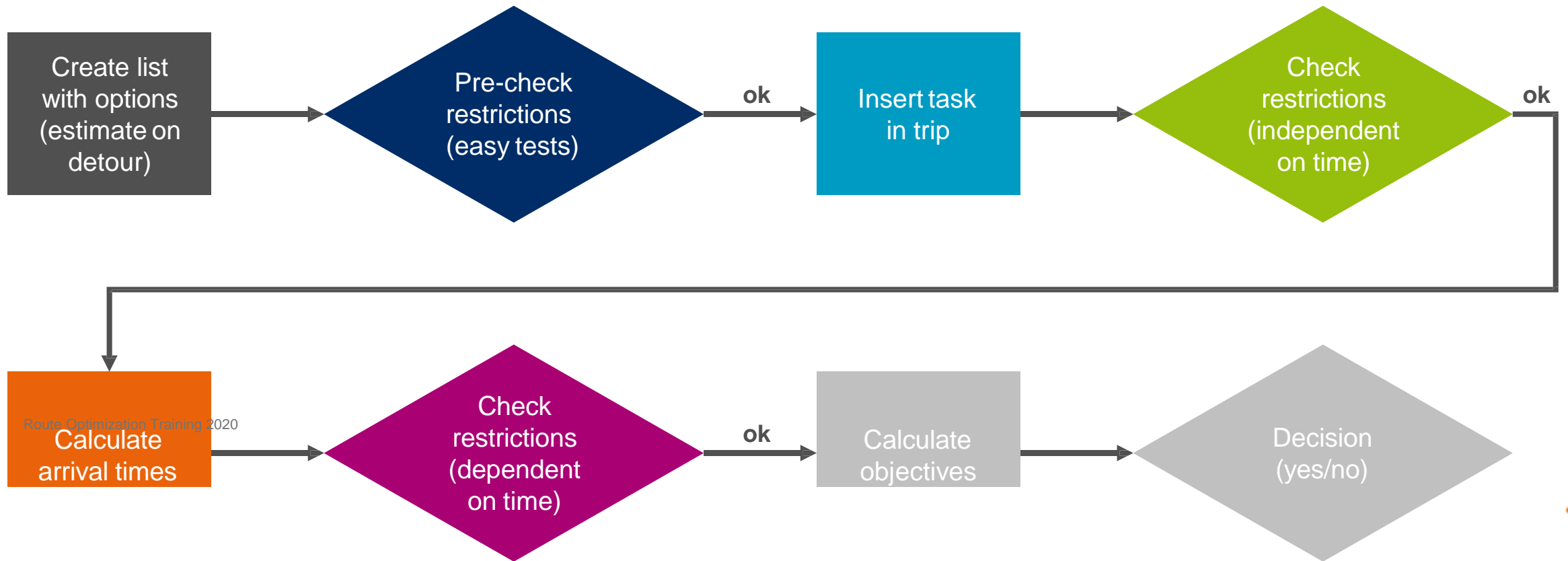
Objectives:

- Finish the trip as early as possible
- Given the above: start as late as possible
- ⑦ make trip as compact as possible

Rule kind	Drivers Legislation
Rule name:	EU: Dayrules
Rule kind group:	
Content:	
	The maximum driving time before a break is 270 minutes.
	The minimum driving time before a break is 240 minutes.
	The maximum driving time before a rest is 540 minutes.
	The minimum driving time before a rest is 510 minutes.
	The maximum working time before a break is 360 minutes.
	The minimum working time before a break is 330 minutes.
	The maximum working time before a rest is 780 minutes.
	The minimum working time before a rest is 750 minutes.
	The minimum break duration is 45 minutes.
	A wait action can be extended to a break when its duration is at least 45 minutes.
	The minimum rest duration is 660 minutes.
	A wait action can be extended to a rest when its duration is at least 660 minutes.

Route Optimization

Under the Hood



Route Optimization

Check Restrictions · Dependent on Time

Driver

- Start and end times of the drivers
- Maximum work time
- Maximum drive time
- EU rules for pause and rest/night break times

Order

- Time windows for the order
- Address calendar: open and close times to visit the location

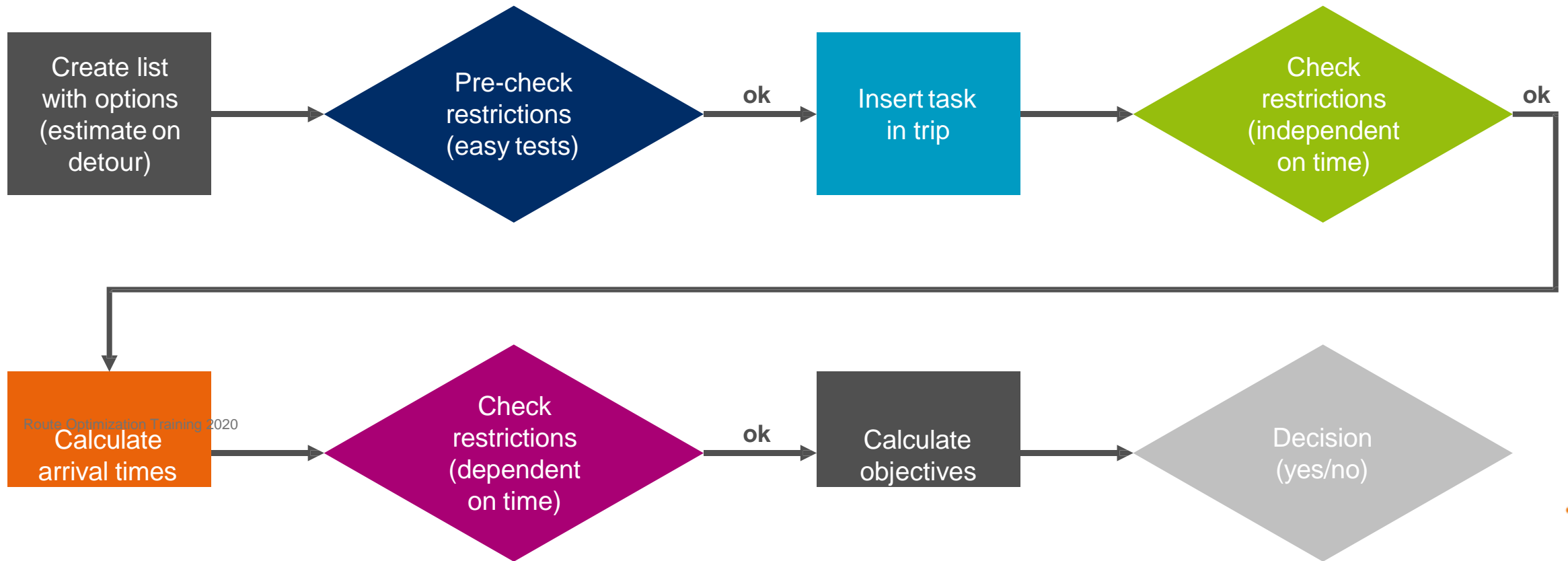
Depot

- Depot open and close times



Route Optimization

Under the Hood



83

Route Optimization Training 2020

Route Optimization

Calculate Objectives



Objectives:

- Plan as many orders as possible
- Minimize total transportation cost

Transportation cost consists of:

- Fixed cost
- Variable cost per kilometer
- Variable cost per hour
- Variable cost per stop

Example:

€ 100 fixed

€ 30 per hour

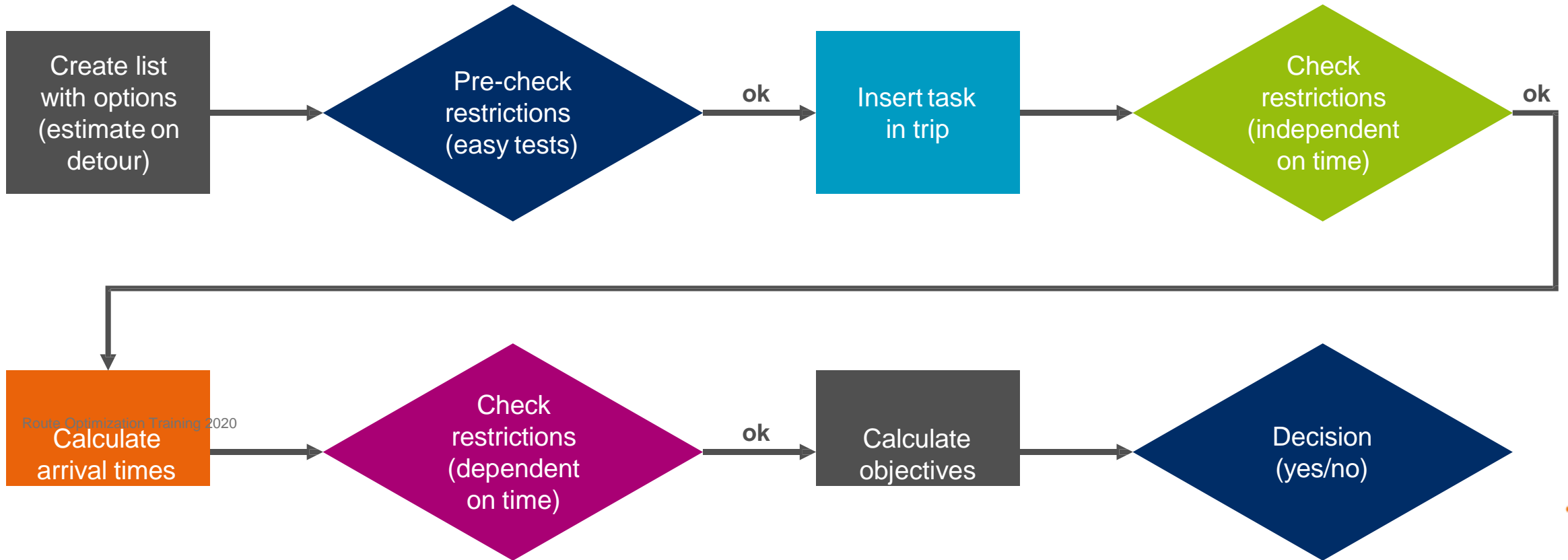
€ 1 per km

Note:

- All decisions during optimization are taken based on these objectives!

Route Optimization

Under the Hood





Questions

e^x $(k!)^x$ $\frac{1}{\pi}$ π



ORTEC