The Anylogic Conference 2021

DESIGNING AND SIMULATING A MULTI-DEPOT OMNICHANNEL LAST-MILE DELIVERY OPERATION USING A VRP SOLVER AND ANYLOGIC

Guilherme Coelho
Projects Specialist
guilherme.coelho@lojasrenner.com.br

LOJAS RENNER S.A.

RENNER CAMICADO YOUCOM realize

anylogic®

COMPANY OVERVIEW

RENNER CAMICADO YOUCOM realize

5 RECOGNIZED BUSINESS
Renner, Camicado, Youcom, Ashua, Realize

633 STORES
402 Renner / 9 Ashua / 119 Camicado / 103 Youcom

4 DISTRIBUTION CENTER
SC and RJ automated

24,8 THOUSAND
employees

+800MM in 2020
visits in our e-commerce

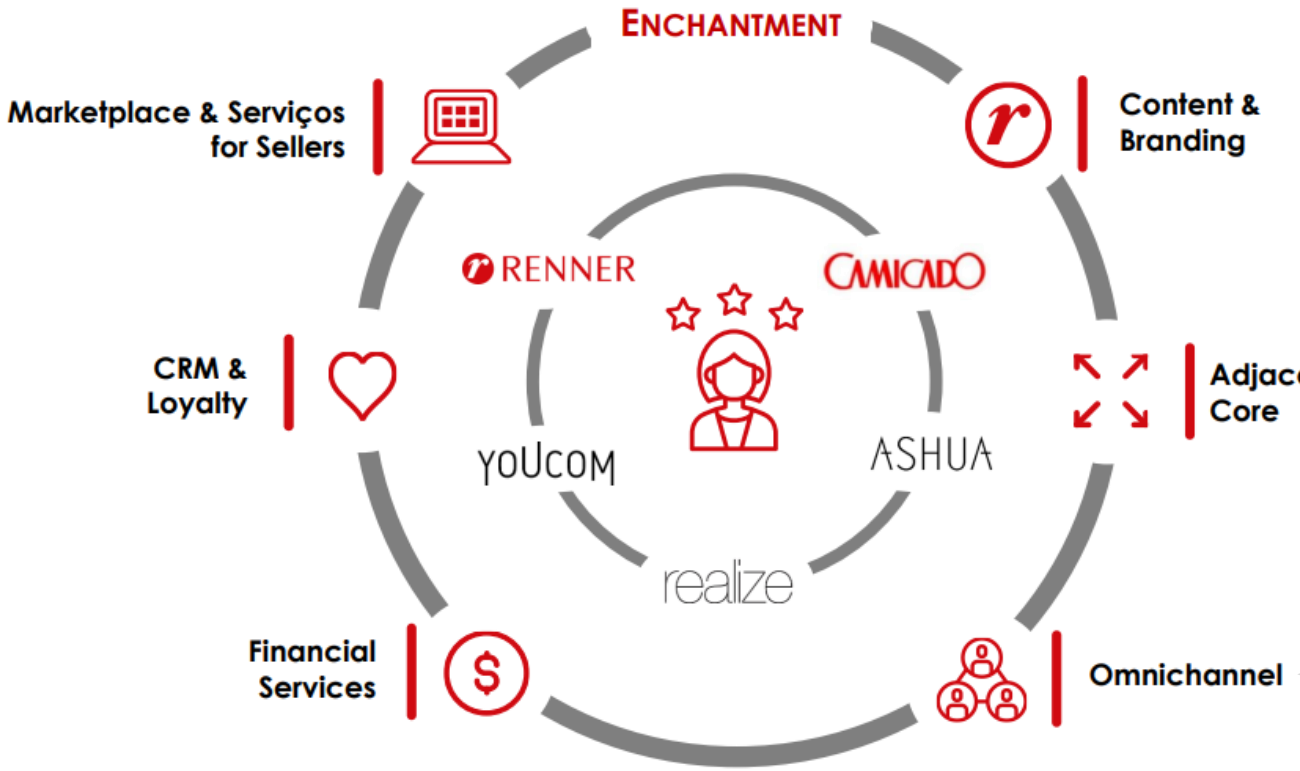
17MM (+205% in 2020)
downloads of the app Renner

~110 SELLERS
in Camicado marketplace

~50 SELLERS
in Renner marketplace

"ENCHANTING EVERYONE IS OUR REALIZATION"

OMNICHANNEL LASTMILE DELIVERY



- Inventory on the right location
- Delivering faster and cheaper
- Improve assortment availability
- Improve inventory accuracy
- Exploit store capillarity

LAST MILE DELIVERY OPERATION

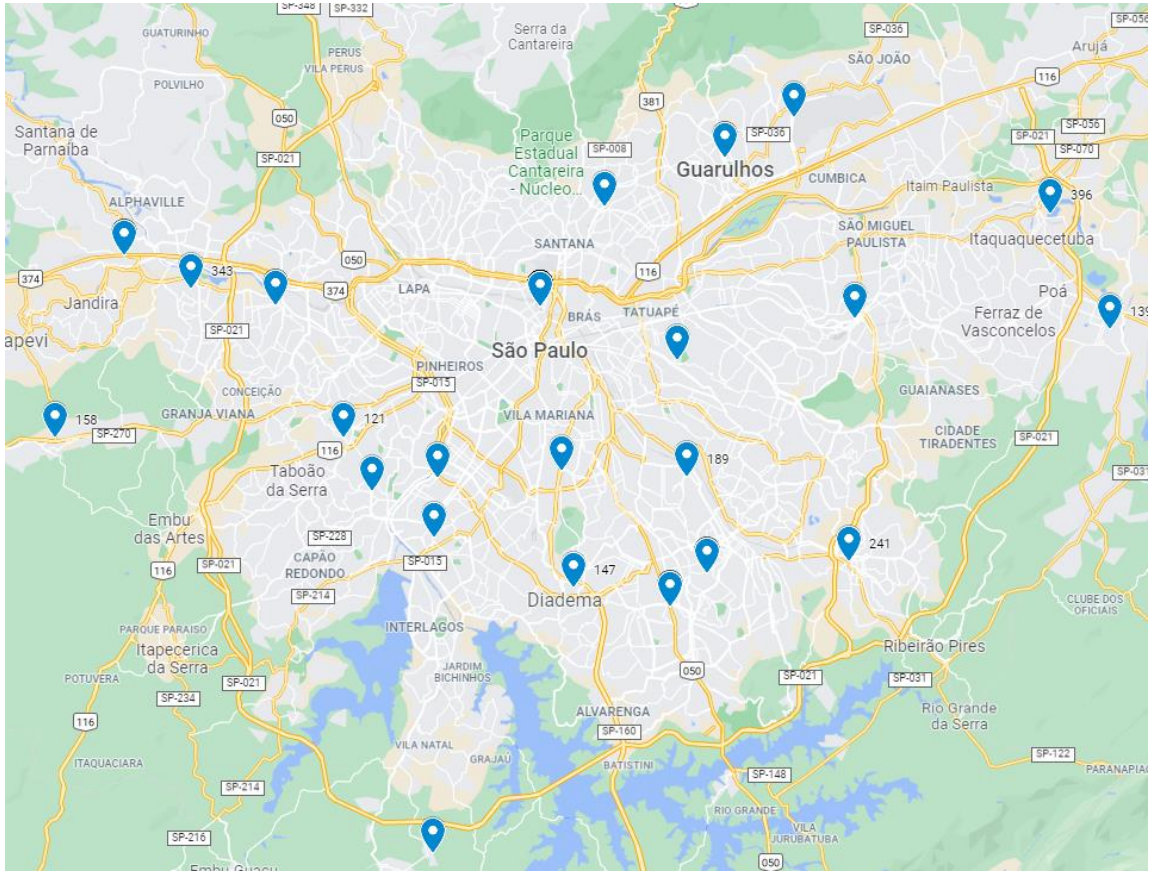
CURRENT OPERATION

- Mgmt. of few large distribution centers
- Mgmt. of retail stores
- Dispatching packages exclusive to 3rd-party carriers
- High cost and long leadtimes
- Ecommerce order tracking is carrier dependent

PROPOSED OPERATION

- Mgmt. of few large distribution centers
- Mgmt. of retail stores
- **Mgmt. of dedicated last-mile couriers**
- **Using retail stores as transit points to last-mile delivery**
- **Standardized ecommerce order tracking**

LAST MILE DELIVERY OPERATION

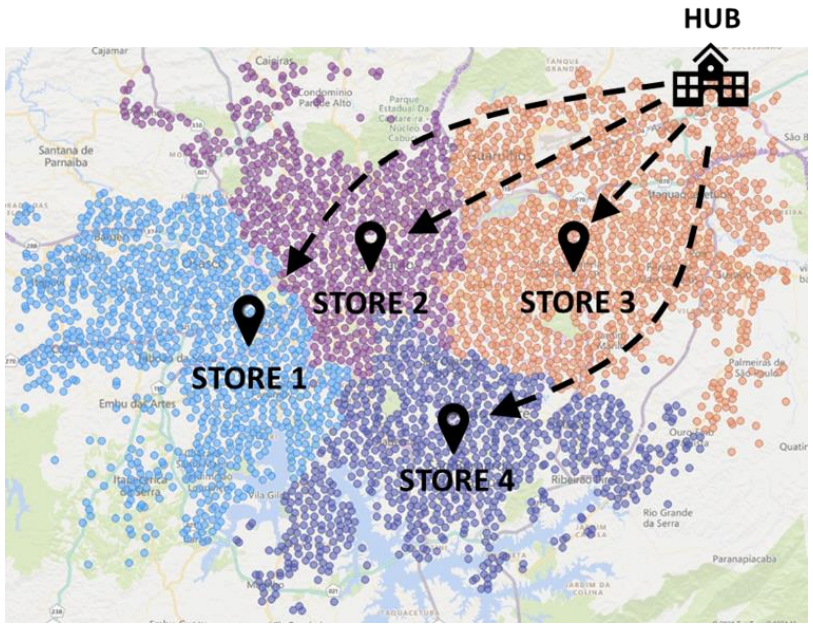


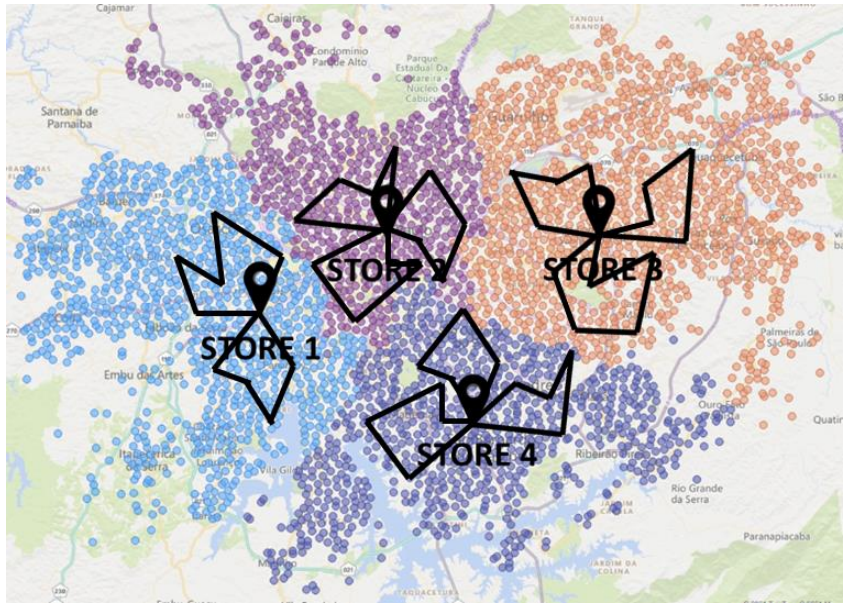
Renner retail stores in São Paulo, Brazil

DESIGNING THE LAST MILE OPERATION

- How to exploit our store capillarity as transit points?
- How many couriers should be used per transit point?
- What is the cost of this operation?
- How long will the couriers work?
- How to account for failed deliveries?
- How to handle special sized deliveries?
- How to simulate the operation considering actual delivery routes?

CASE STUDY





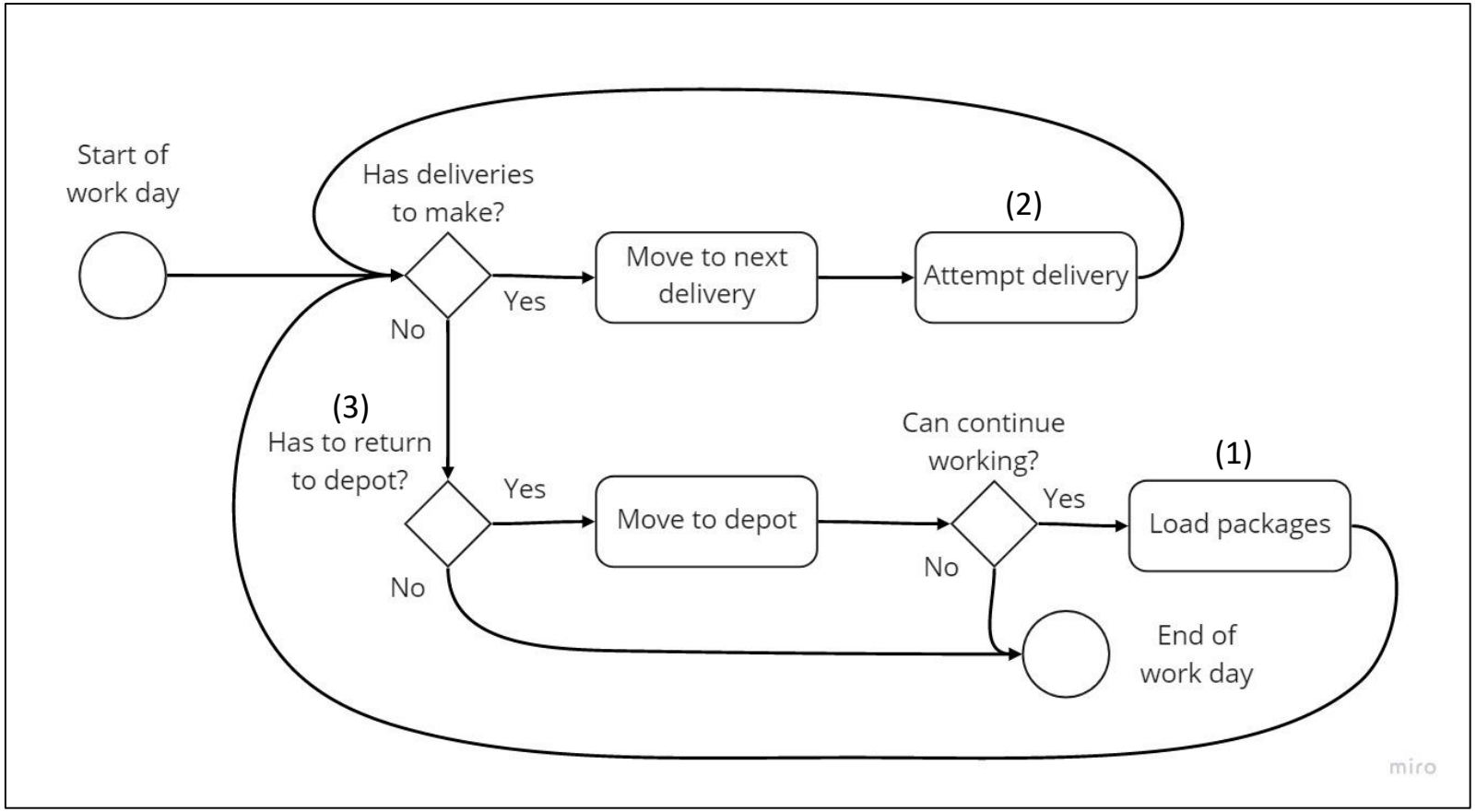
Assuming:

- A fixed set of stores as transit points
- A fixed coverage per transit point
- Hub transfers packages to transit points regularly

Simulate:

- Solving a VRP for each depot
- Delivery routes for random packages
- Fixed couriers per transit point
- Random delivery times
- Random delivery failures

CASE STUDY



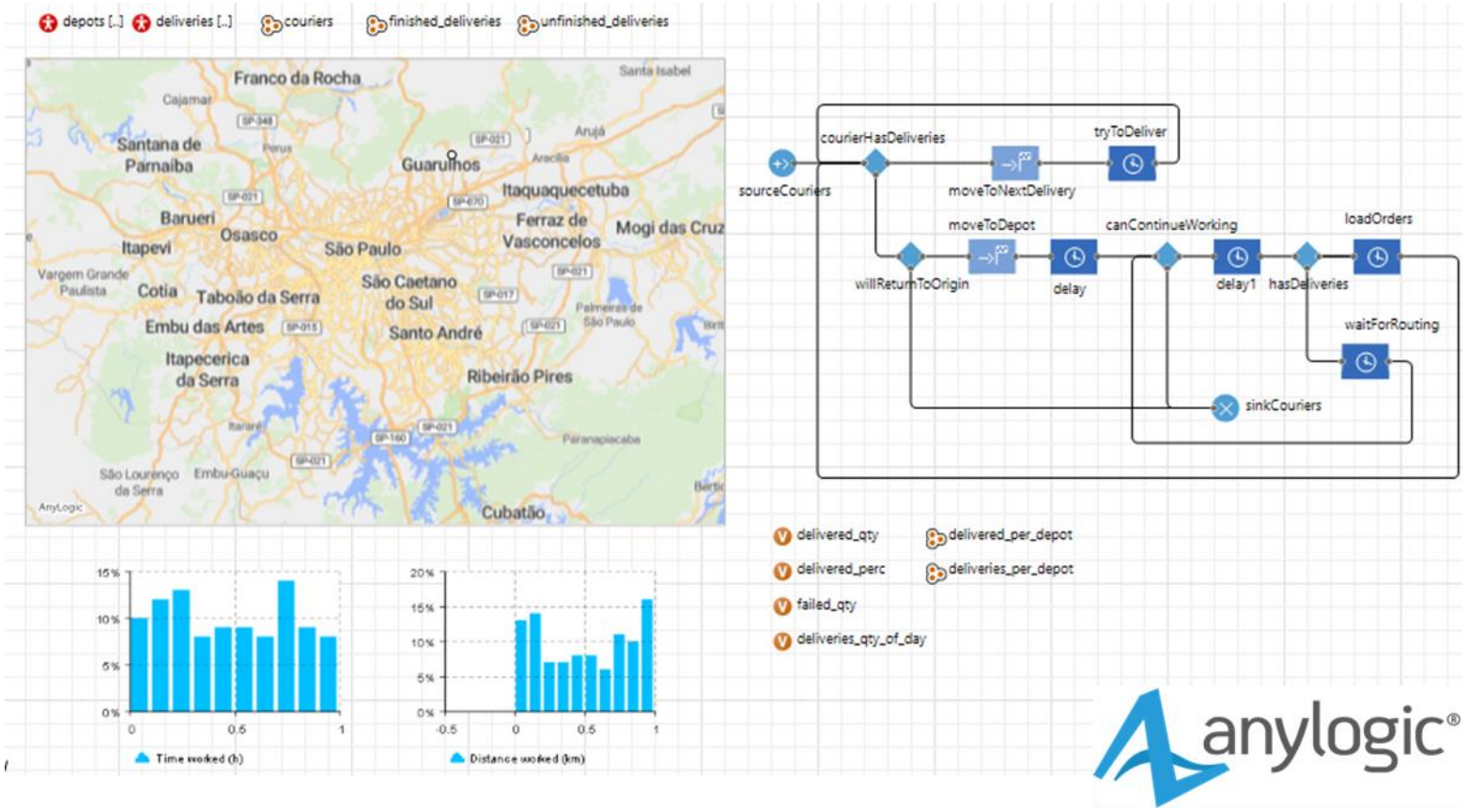
General courier workflow

(1) After loading packages, a new route must be defined for each courier.

(2) There is a probability of failure for each delivery. Also, the time to deliver follows any chosen probability distribution.

(3) When the route is finished, the courier have to return to its depot only if there is still available packages to be delivered or if there was any failed delivery.

THE SIMULATION MODEL

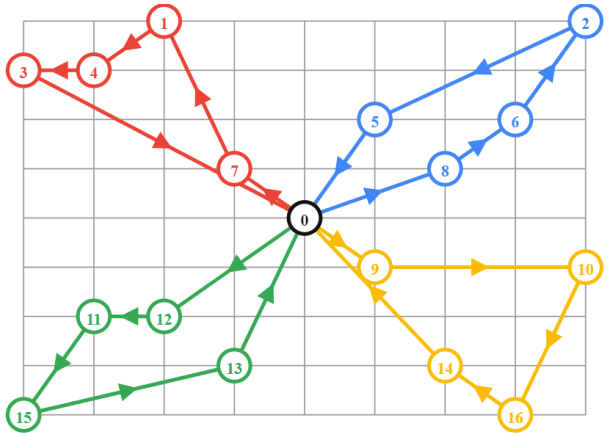


- Anylogic can easily simulate the last mile operation for given routes.
- GIS features are very intuitive and easy to set up.
- Easy to expand with Java.

But..
No library/package for VRP solving is included

THE VEHICLE ROUTING PROBLEM

Vehicle Routing Problem



$$\text{Min} \sum_{k=1}^{K} \sum_{(i,j)\in E} c_{ij} x_{ijk}$$

$$\sum_{k\in K} \sum_{j\in N} x_{ijk} = 1, \forall \mathrm{i} \in \mathrm{C}$$

$$\sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \le Q, \forall k \in \mathrm{K}$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0, \forall h \in \mathrm{C}, \forall k \in \mathrm{K}$$

$$\sum_{i,j\in S} x_{ijk} \le |S| - 1, S \subset C; 2 \le |S| \le \left\lfloor \frac{\mathrm{n}}{2} \right\rfloor; \forall k \in K$$

$$\sum_{j\in N} x_{0jk} = 1, \forall k \in K$$

$$\sum_{i\in N} x_{i,n+1,k} = 1, \forall k \in K$$

$$x_{ijk} \in B^{K|E|}$$

Open source VRP solvers

Jsprit (Graphhopper) - Java

OR-Tools (Google) – C++, Java, Python, C#

OptaPlanner (Red Hat) – Java

General purpose solvers

Gurobi (Gurobi) – C, C++, Java, .NET, Python, Matlab, R

CPLEX (IBM) – C++, Java, .NET

Xpress (FICO) – Java, C/C++, .NET, Python, VB

THE CHOSEN VRP SOLVER: JSPRIT

“jsprit is a java based, open source toolkit for solving rich traveling salesman (TSP) and vehicle routing problems (VRP). It is lightweight, flexible and easy-to-use.”

★ Features

jsprit can solve problems with pickups and deliveries, back hauls, heterogeneous fleets, finite and infinite fleets, multiple depots, time windows, open routes, different start and end locations, multiple capacity dimensions, initial loads, skills ...

★ Tested

jsprit is fit for change due to comprehensive unit and integration tests.

★ Documented

jsprit is documented and additional code-examples illustrate its use.

★ Constraints

jsprit allows you to define additional stateful and stateless constraints to account for the richness of your problem.

★ Benchmarked

jsprit is benchmarked against classical VRP instances (e.g. Solomon instances).

★ Open Source

jsprit is released under Apache License v2.

https://jsprit.github.io/

Register vehicle types

```
List<VehicleType> vrp_vehicle_types = new ArrayList<VehicleType>();
vehicle_types.forEach(vehicle_type -> {
    vrp_vehicle_types.add(
        VehicleTypeImpl
        .Builder
        .newInstance(vehicle_type.name)
        .addCapacityDimension(0, vehicle_type.capacity)
        .setCostPerDistance(vehicle_type.cost_per_km)
        .setFixedCost(vehicle_type.cost_per_day)
        .setUserData(vehicle_type.getTypeVrpParams())
        .build()
    );
});
return vrp_vehicle_types;
```

→ Any number of capacity dimensions can be set

→ Other custom data can also be defined

Register couriers for each vehicle type

```
return couriers
    .stream()
    .map(courier -> {
        RouteVehicleType vehicle_type = courier.toRouteVehicleType();
        VehicleType vrp_vehicle_type = findFirst(
            vrp_vehicle_types,
            type -> type.getTypeId().equals(vehicle_type.name)
        );
        return VehicleImpl.Builder
            .newInstance("courier."+courier.courier_id)
            .setType(vrp_vehicle_type)
            .setStartLocation(Location.newInstance(courier.depot.toRoutePoint().name))
            .setReturnToDepot(true)
            .addAllSkills(courier.getSkills())
            .setUserData(courier.getAditionalVrpParams())
            .setEarliestStart(courier.total_time_worked)
            .setLatestArrival(courier.max_work_time + courier.max_work_time_slack)
            .build();
    })
    .collect(Collectors.toList());
```

Get the related vehicle type

Define the starting point

Register specific courier skills

Register current total work time and set total time constraint

Register each delivery

```
return Delivery.Builder
        .newInstance(point.name)
        .addSizeDimension(0,1)
        .setLocation(Location.newInstance(point.name))
        .setServiceTime(delivery.time_to_deliver[1])
        .addAllRequiredSkills(delivery.getRequiredSkills())
        .build();
```

→ Define the delivery time

→ Register the required skills. I will be checked against the courier skills

Register distance matrix and cost function

```
return new AbstractForwardVehicleRoutingTransportCosts() {

    @Override
    public double getDistance(Location from, Location to, double departureTime, Vehicle vehicle) {
        if(from == null || to == null || from.getId().equals(to.getId()))
            return 0.0;

        String gis_id_1 = findFirst(points, p -> p.name.equals(from.getId())).gis_id;
        String gis_id_2 = findFirst(points, p -> p.name.equals(to.getId())).gis_id;

        return vrp_distance_matrix.get(gis_id_1+"."+gis_id_2);

    }

    @Override
    public double getTransportCost(Location from, Location to, double departureTime, Driver driver, Vehicle vehicle) {
        if(from == null || to == null || from.getId().equals(to.getId()))
            return 0.0;

        double distance = this.getDistance(from, to, departureTime, vehicle);
        if (vehicle == null) return distance;

        VehicleType vehicle_type = vehicle.getType();

        VehicleCostParams cost_params = vehicle_type.getVehicleCostParams();

        return cost_params.perDistanceUnit * distance
            + cost_params.perTransportTimeUnit * distance/vehicle_type.getMaxVelocity();

    }
```

The distance matrix can be obtained by calculating the straight line distance between all points or using any routes apis, such as Google Maps or Open Street Maps.

Register the any cost function

Register the transport time function

```
return new AbstractForwardVehicleRoutingTransportCosts() {

    @Override
    public double getTransportTime(Location from, Location to, double departureTime, Driver driver, Vehicle vehicle) {

        if(from == null || to == null || from.getId().equals(to.getId()))
            return 0.0;

        double pickup_time = from.getId().equals(depot.toRoutePoint().name) ? this.getPickupTime(vehicle) : 0.;

        double distance = this.getDistance(from, to, departureTime, vehicle);

        double speed = this.getSpeed(vehicle, departureTime);

        return pickup_time + distance/speed;

    }
```

→ Get the distance between points

→ Get the vehicle speed

→ Any transport time function can be modeled

JSPRIT MODELING OVERVIEW

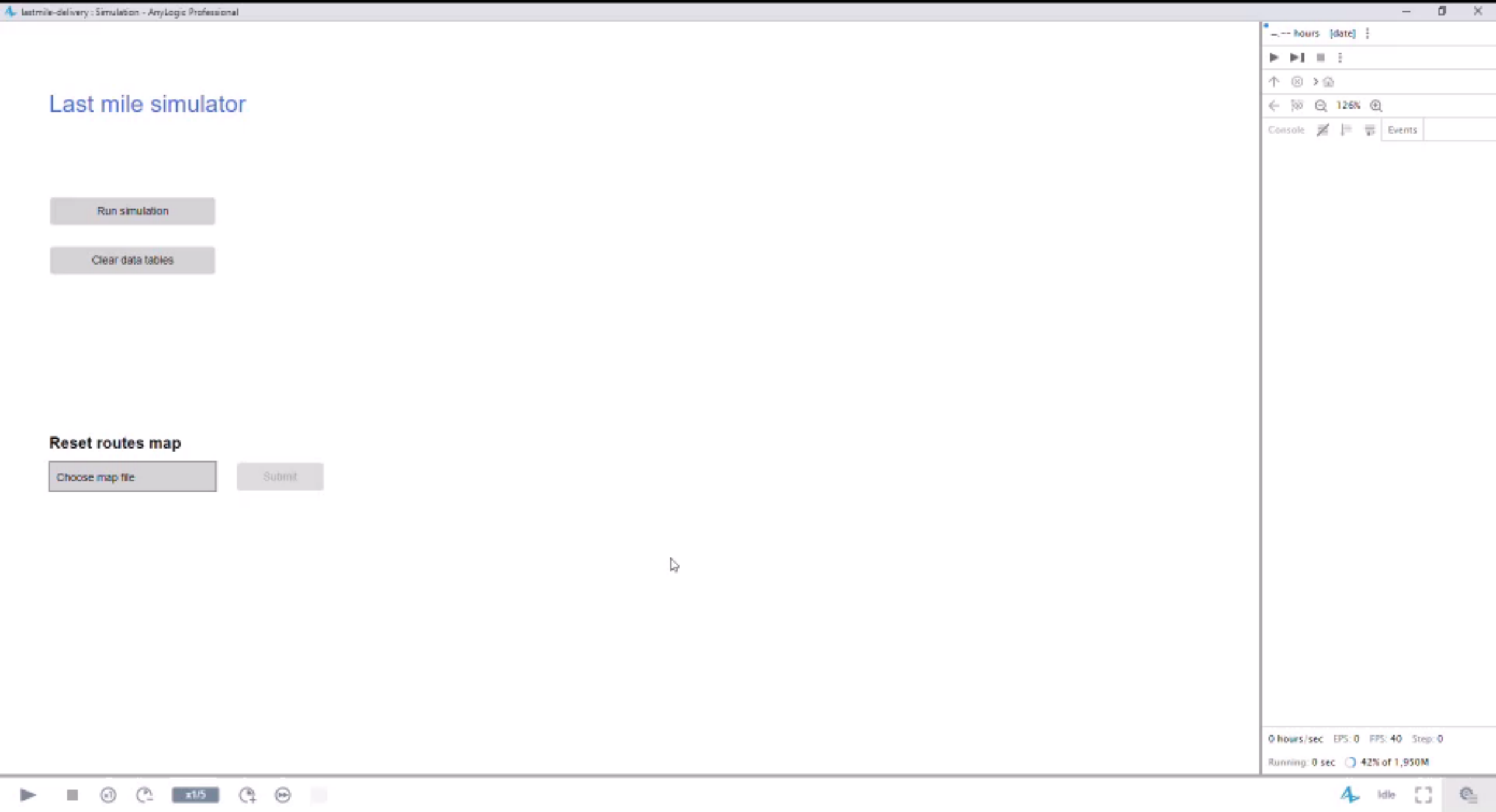
Finally, build the VRP and solve it

```
VehicleRoutingProblem.Builder vrpBuilder = VehicleRoutingProblem
    .Builder
    .newInstance()
    .setRoutingCost(cost_matrix);

vehicles.forEach(v -> vrpBuilder.addVehicle(v));
shipments.forEach(s -> vrpBuilder.addJob(s));

vrpBuilder.setFleetSize(FleetSize.FINITE);

return vrpBuilder.build();
```

Add all structures to the VRP

```
private VehicleRoutingProblemSolution findBestSolution (VehicleRoutingProblem problem) {

    VehicleRoutingAlgorithm algorithm = Jsprit.Builder.newInstance(problem)
            .setProperty(Jsprit.Parameter.THREADS, ManagementFactory.getThreadMXBean().getThreadCount()+"")
            .setObjectiveFunction(this.getObjectiveFunction(problem))
            .buildAlgorithm();

    TimeTermination prematureTermination = new TimeTermination((long)this.getTimeout()*1000); //in miliseconds
    algorithm.setPrematureAlgorithmTermination(prematureTermination);
    algorithm.addListener(prematureTermination);

    Collection<VehicleRoutingProblemSolution> solutions = algorithm.searchSolutions();
    return Solutions.bestOf(solutions);

}
```

Retrieve the best solution


lastmile-delivery : Simulation - AnyLogic Professional
Last mile simulator
Run simulation
Clear data tables
Reset routes map
Choose map file
Submit
hours [date]
126%
Console
Events
0 hours/sec EPS: 0 FPS: 40 Step: 0
Running: 0 sec 42% of 1,950M
x1/5
Idle

FINAL THOUGHTS

- Designing any logistics operation is a challenging endeavor: many decision variables, many good different strategies and many specifics for each locality.
- Choosing the right tool is essential, specially considering time and budget constraints to deliver a project.
- Anylogic provides various modeling alternatives for the same problem. There is no real right answer here.
- Complex problems might require complex solution processes, but there is no the need to “reinvent the wheel” every time. There might be an open source project related to your problem.

THANK YOU. QUESTIONS?