# Lab 1: Scheduling and Assigning Technicians to Tasks Using Decision Optimization for Watson Studio

*Using Modeling Assistant (natural language to build DO model)*

*January 2nd, 2019*

*Author: Sumeet Parashar , email: s.parashar@us.ibm.com*

## Table of contents

# Contents

# Overview

This Lab exercise will explain how to use and demo the ***Decision Optimization (DO)*** for Watson Studio (DO4WS) add-on. It will build upon the use case of assigning and scheduling technicians to repair tasks at various locations, taking into account the technicians' skills and availability, while optimizing for total time for completion, driving distances, etc. You will learn how to:

- Load data.
- Build a model (objectives and constraints) using natural language modeling assistant.
- Build and compare scenarios.
- Visualize results.

# Pre-requisites, access, and files

- To complete this lab, you will need access to a **Watson Studio Local** cluster.
- You will also need to complete the following steps to import the sample project:
    - Download and unzip this GitHub repository: https://github.com/elenalowery/WSL-Workshop (if you haven't already done it in the previous lab)
- Stop all environments that were used for other projects – this is important when sharing the cluster with other users (it will help minimize workload on the cluster).
    - In **Watson Studio** click on your project (for example, *WSL_Demos*), then **Environments**. Stop all environments that are running.

*Note: do not select Save Custom Image when stopping the environment.*

# The Business Problem

This use case is applicable for many industries where there are scheduling challenges to assign limited resources to the right job at the right location. For example, a telecommunications company or a utility provider may need to assign technicians to services calls for the day/week at various locations.

**Data**: Technicians and their skills, service requests, service request locations, estimated time to fix the service issue, penalty or cost of unfinished work, distances between locations.

**Objectives and Constraints:**

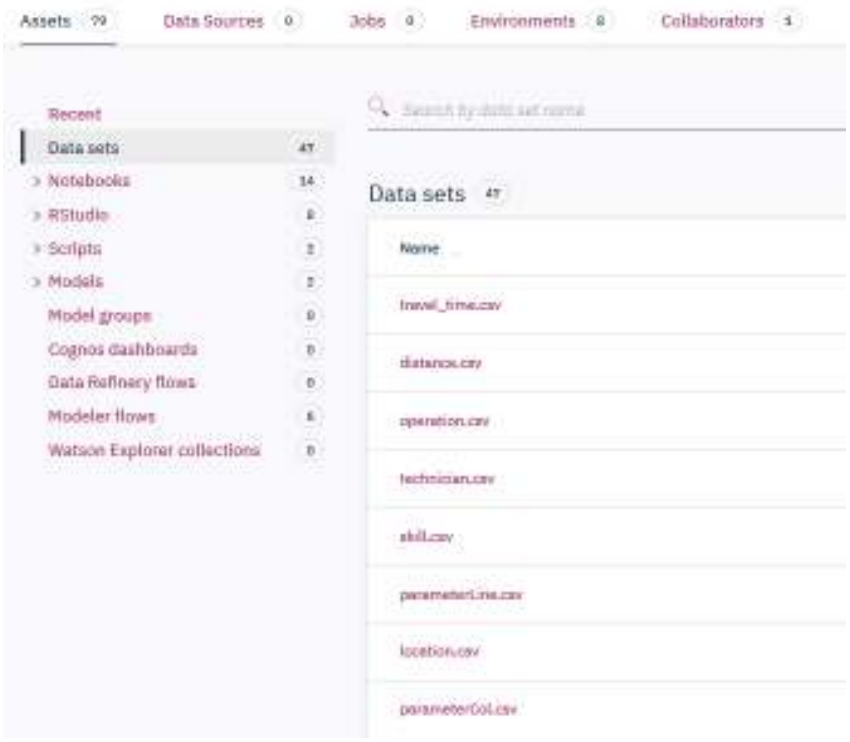
- Maximize service assignments
- Minimize time to finish service calls
- Minimize distance traveled

These constraints must be met while satisfying business rules, such as one technician assignment to each service call, assigning technicians of right skill for the right job, etc.

# Part 1: Load Data

1. In **Watson Studio Local** navigate to your project (for example, *WSL_Demos*).

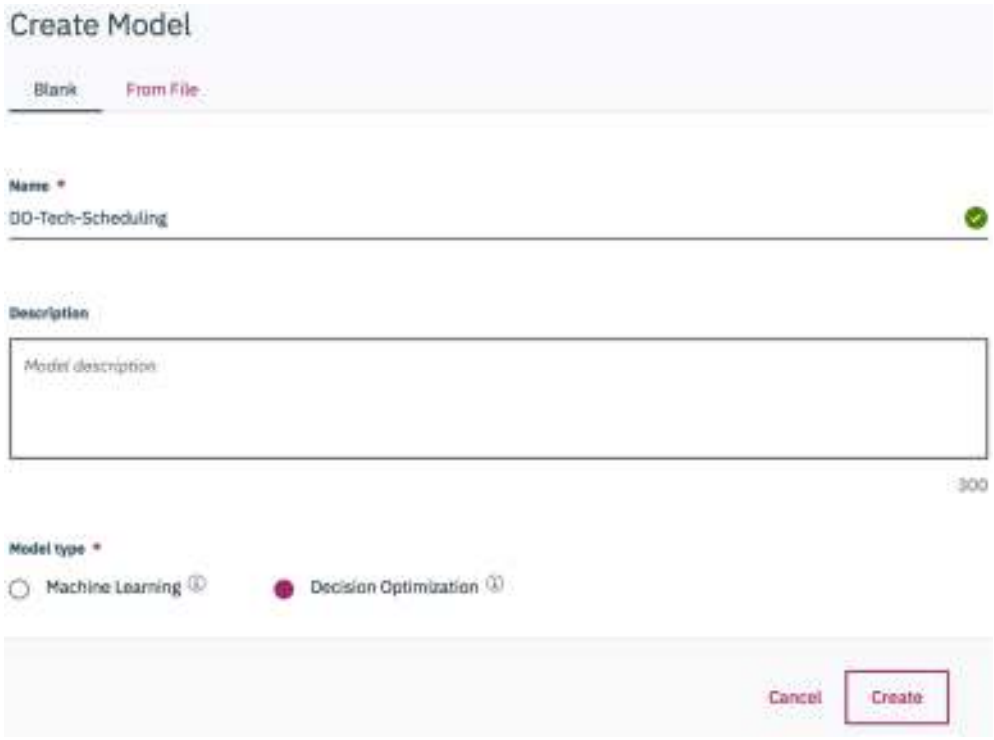
2. Navigate to **Data sets** view and upload all csv files from the *DO* folder of the unzipped git repository (8 files).



# Part 2: Build Decision Optimization model

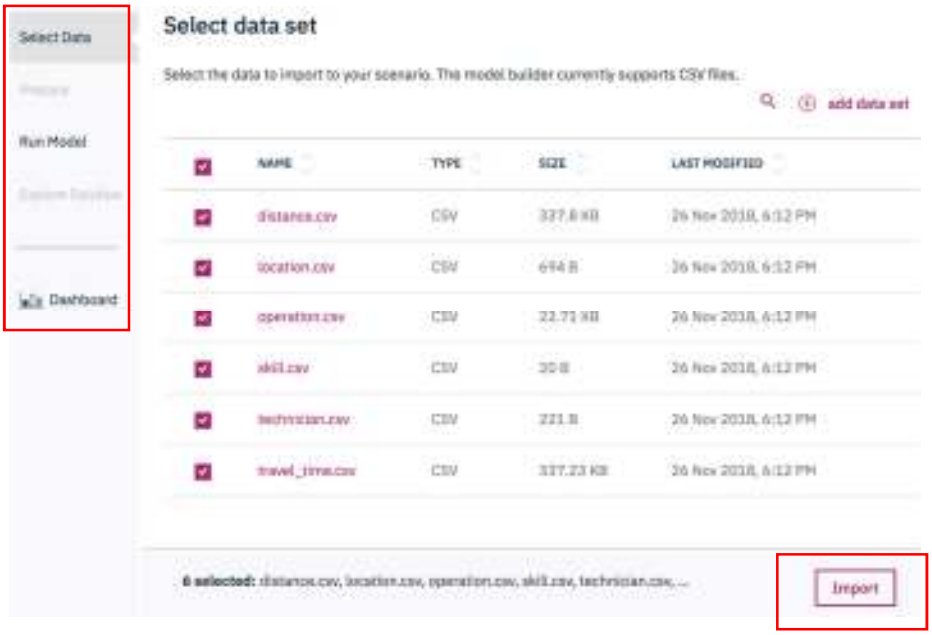
1. In the **Assets** view select **Models**, *then* click **Add Model**.


2. Provide model name, and select *Decision Optimization* as the model type. Click **Create**.



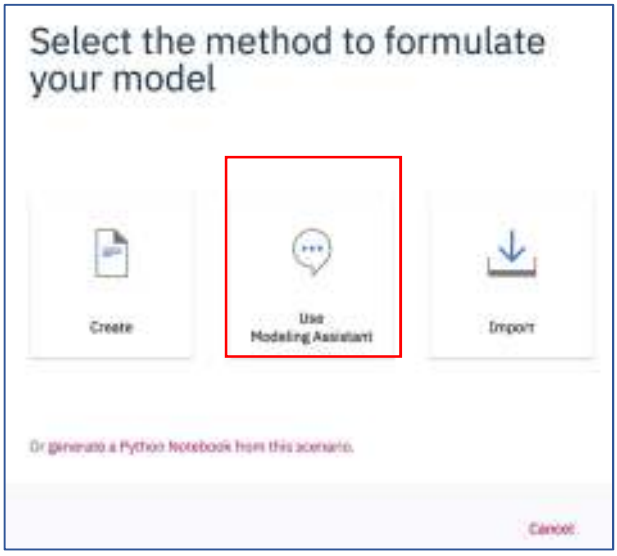

The **Decision Optimization Modeling Assistant** wizard is displayed. First, we need to select data that will be used for building the model.

3. Select the following 6 *csv* files and click **Import**.



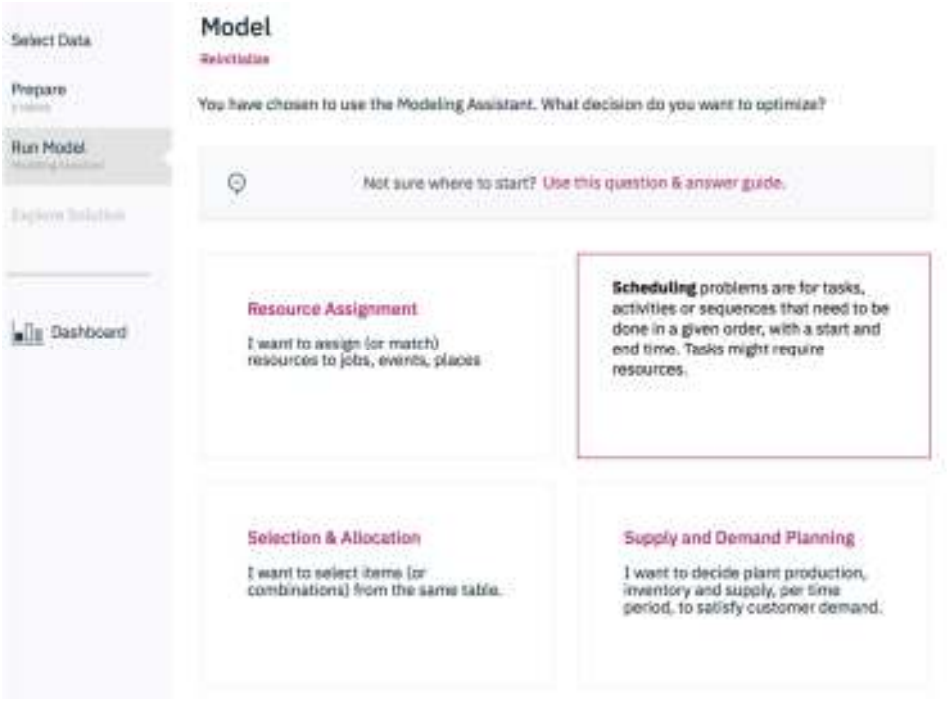

4. After the import the wizard displays the **Prepare** tab, which shows the preview of all the tables. We don't need to make changes in this step.

5. Click on the **Run Model** tab. This will automatically take you to a pop-up window to **Select the method to formulate your Model**. You can also access this function by clicking on the hyper link with the same text.




6. Choose the option in the middle: **Use Modeling Assistant**

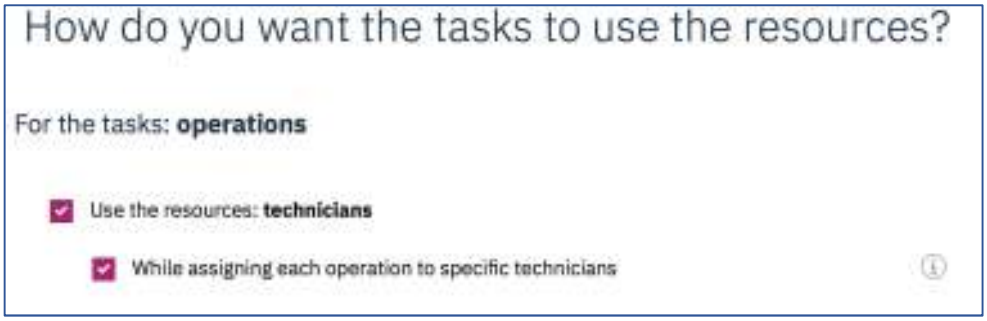
7. The modeling assistant provides four domain templates for building DO models with the natural language modeling assistant. Since we are not just assigning technicians to a job, but also want to schedule the order of the jobs on a time scale, we will choose *Scheduling* template.




8. Next, will select *tasks* and *resources*.

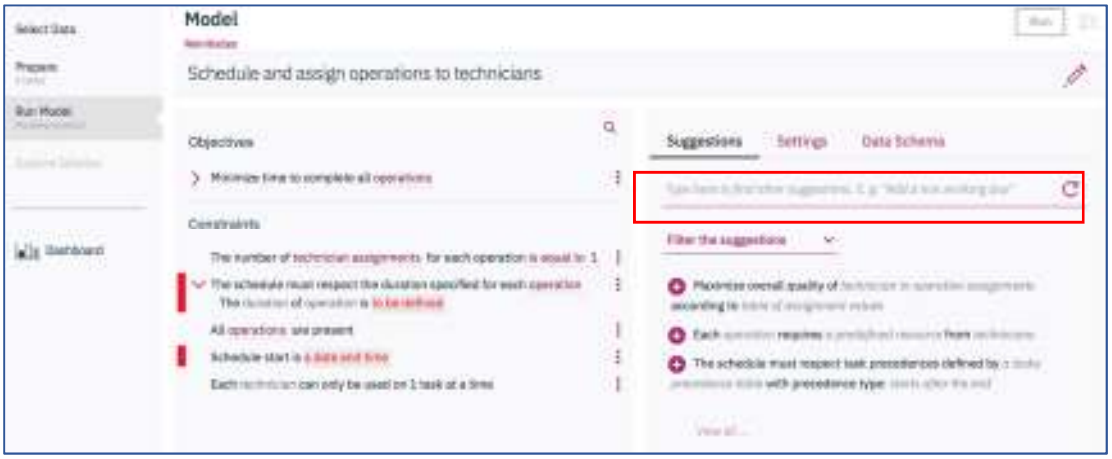
   - Select the table *operation* from the drop-down list for TASKS
   - Select the table *technician* from the drop-down list for RESOURCES
   - Click **Continue** button at the bottom.

9. Leave the two check-boxes checked on the next screen and click **Continue**. Click **Finish** on the next screen.




10. Next, we will configure *objectives* and *constraints*.

    Some basic objectives and constraints are populated on the left side based on the domain template. More suggestions are available on the right side of screen. You can also use the text box on the right side (highlighted in picture below) to write an objective or constraint in natural language and then find a closest suggested match for it.



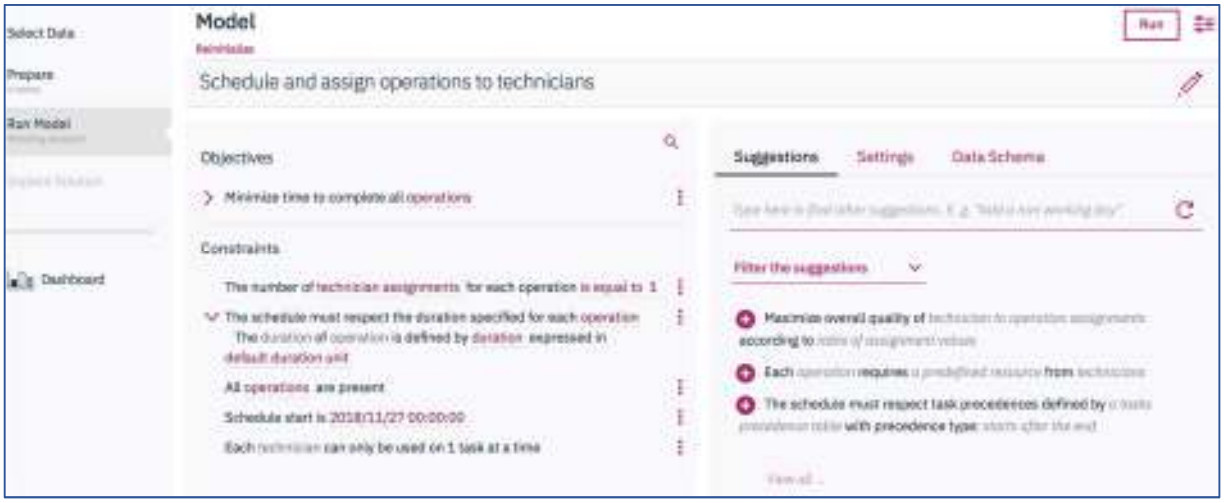

    Notice the objective *Minimize time to complete all operations*. It is a typical objective for scheduling models. But this can be disabled or replaced. We can also add more objectives.

    Out of the four constraints already populated, notice the red bar in front of two constraints. That means that these constraints need some additional input from the user.

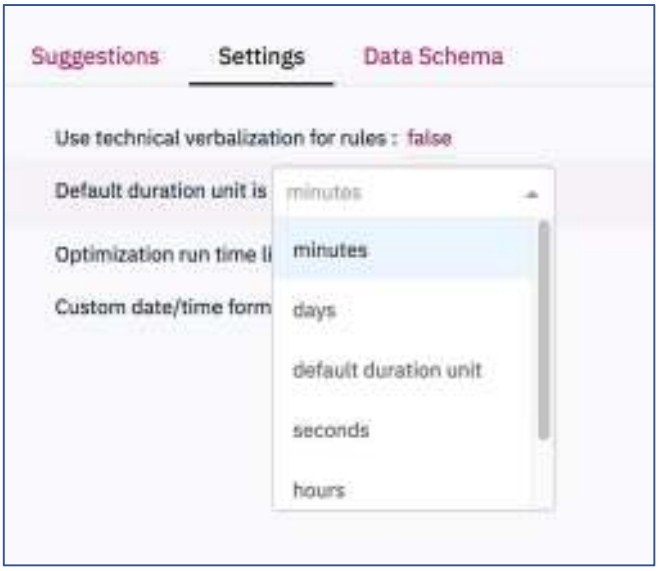
    - **Constraint 1**: *The duration of operation is to be defined*.
      - Click on *to be defined* and pick *duration* as the column name that contains the value of time to complete the task/operation.

    - **Constraint 2**: *Schedule start is a date and time*".
      - Click on "*a date and time*" and set the date. You can pick the current date and time.

The completed model should look like the following screenshot.



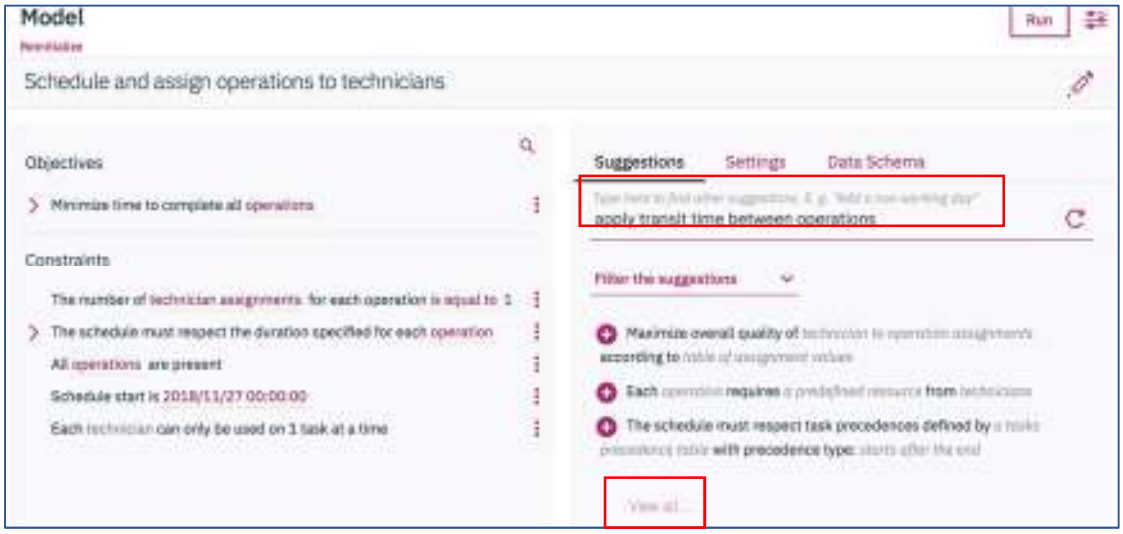

11. Next, check and set the time duration unit in **Settings**. Click **Settings** on right side of screen, set the "*Default duration unit is minutes*"




At this point we can run the model and get a basic solution. However the solution will not be of good quality since few other logical constraints are not set. If you want to take a look at the basic solution, you can skip to **step #16**. Later you can come back to **step #12** to add more constraints.

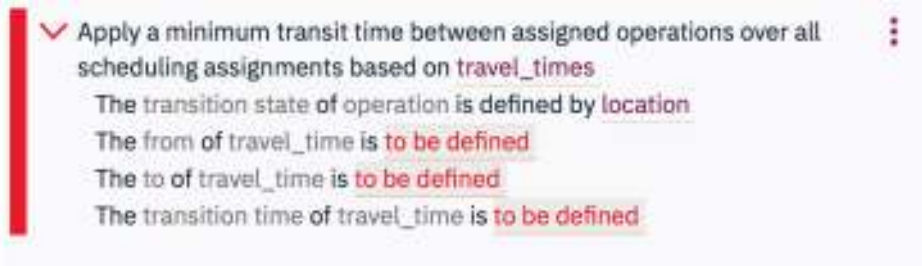
12. Add transit time constraint using natural language:

- Type *apply transit time between operations* (you can try your own variation to apply transit time between operations according to the table travel_time) in the text box on right side of screen, and hit enter.
- Click **View all** to see more matching suggestions.




13. Pick the suggestion: click on the **"+"** sign in front of *Apply a minimum transit time between assigned operations over all scheduling assignments based on travel_times*

Expand the newly added constraint and configure it as shown below




To configure the constraint, pick the correct column from the table *travel_times* as

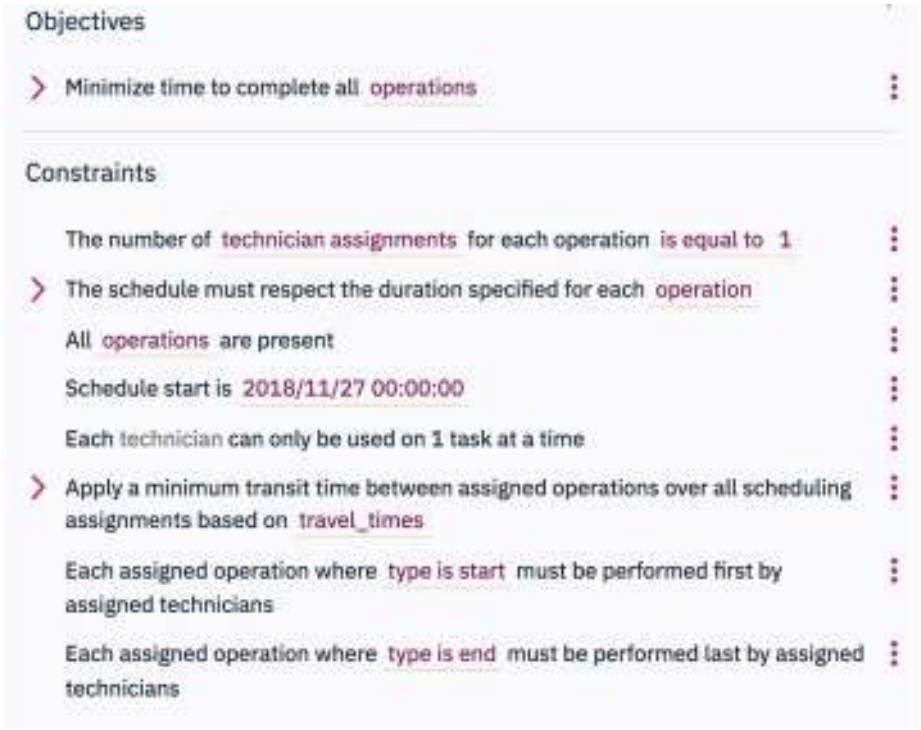
- The transition state of operation is defined by ***location***
- The from of travel_time is defined by ***from***
- The to of travel_time is defined by ***to***
- The transition time of travel_time is defined by ***time***

14. Add two more constraints for start and end tasks: the operation table has a few tasks which are of type *start* or *end*. These tasks can be assumed as sign-in and sign-out operations that each technician should perform.

    - In the natural language box write *operation with type "start" must be performed first* and select "Each assigned operation where type is start must be performed first by assigned technicians"

    - Similarly, type *operation with type "end" must be performed last* in natural language box and select "Each assigned operation where type is end must be performed last by assigned technicians"

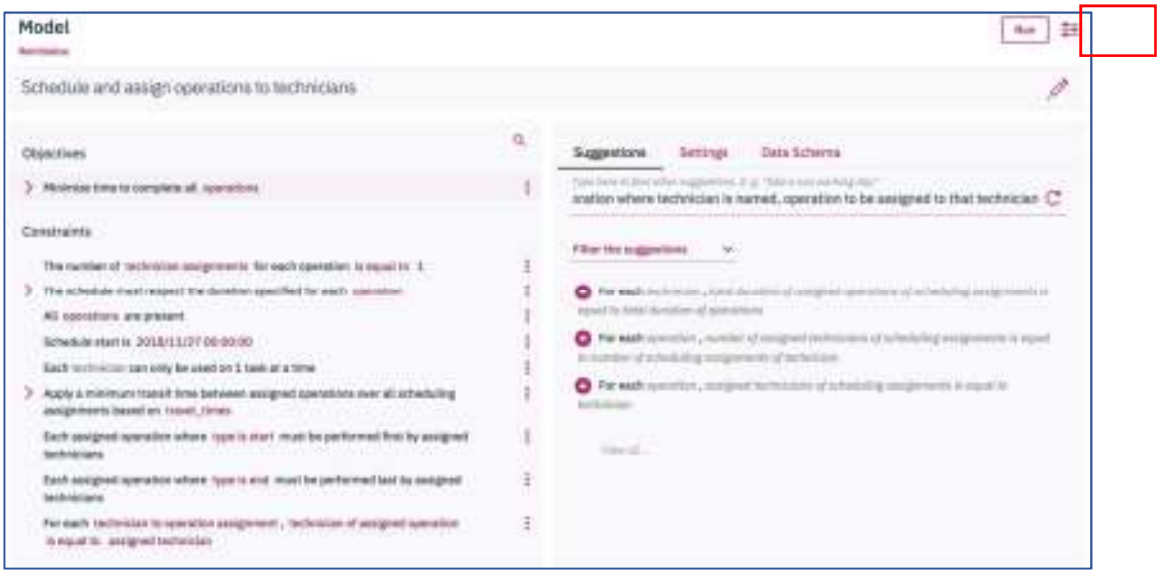
    At this stage the model should look similar to this screenshot:



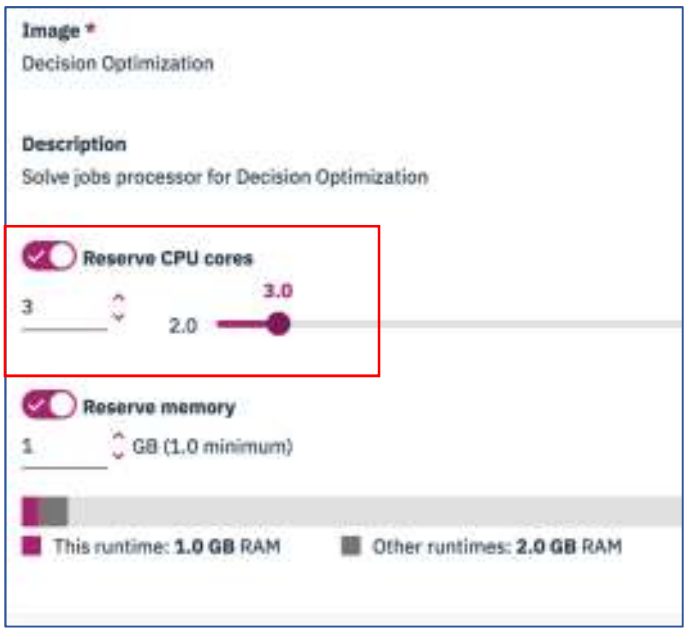
15. Confirm that where technician is named for an operation (specifically for start and end operation types), those named technicians are assigned to the operations:

    - Write in the natural language box *for each operation where technician is named, operation to be assigned to that technician*
    - Pick the suggestion: *For each technician to operation assignment, technician of assigned operation is equal to assigned technician*

    Final model screenshot:

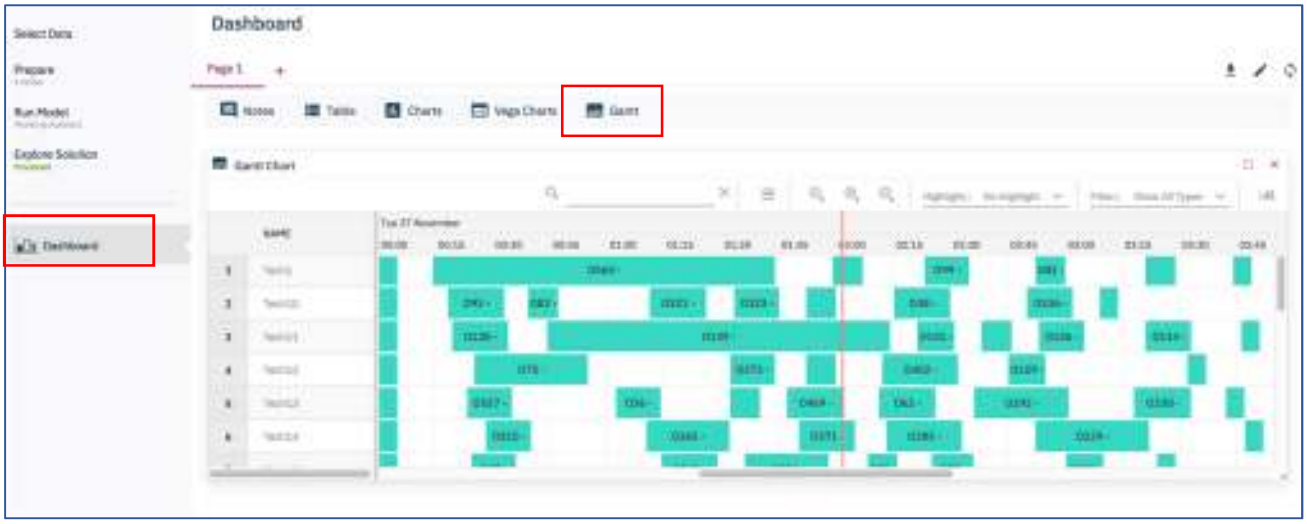



16. We are almost ready to run the model. Click the **Run Configuration** button on top right which is next to the **Run** button.
    - The *Run Configuration* screen is displayed. Change the number of cores to 3. We need to complete this step because there is currently a bug – the run fails with 2 cores.
    - Click **Save and restart**.

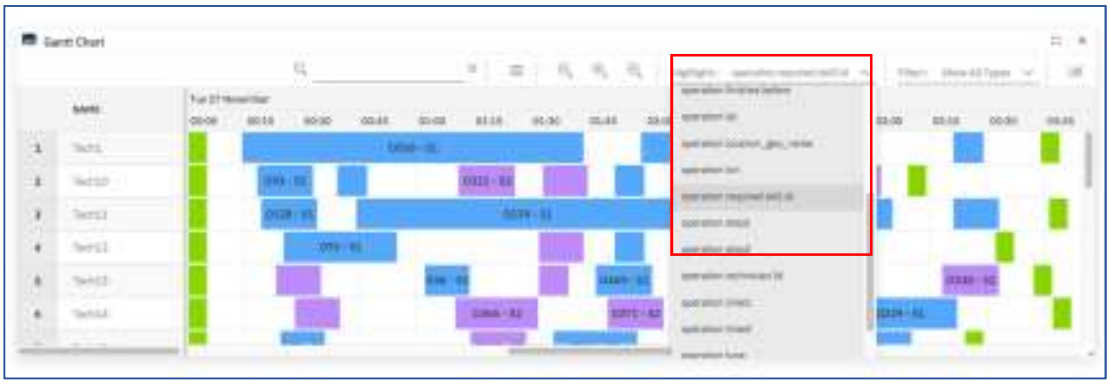


17. Run the model. This will automatically take the screen to the **Explore** tab. The **Explore** tab will show the *objective value* and *solution tables*. Log information is also available.


18. Navigate to the **Dashboard** tab and click on **Gantt** to show a Gantt chart of operations and resource assignment.



19. (Optional) Explore various highlight options of the Gantt chart. Use the Gantt chart menu, use Highlight option and choose "*Operation required skill id*" to highlight operations according to skill required.

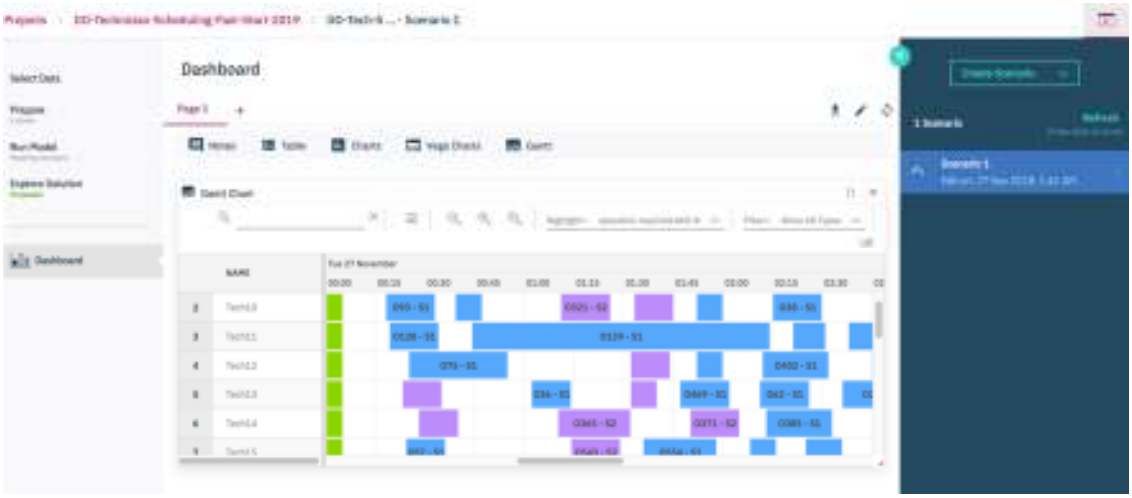


# Part 3: Build and Compare Scenarios

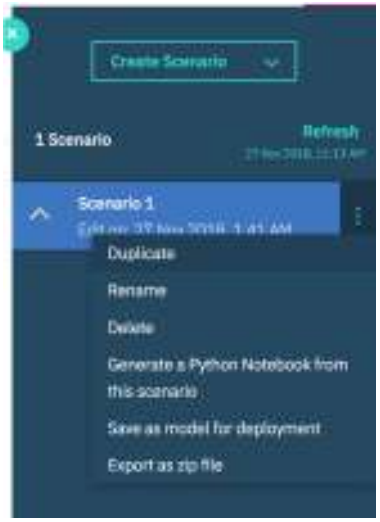
DO4WS also provides functionality for building and comparing scenarios of the same problem. The scenarios could be different in terms of model (objectives and constraints) and/or data. Let's build a scenario with an additional objective and compare the results.

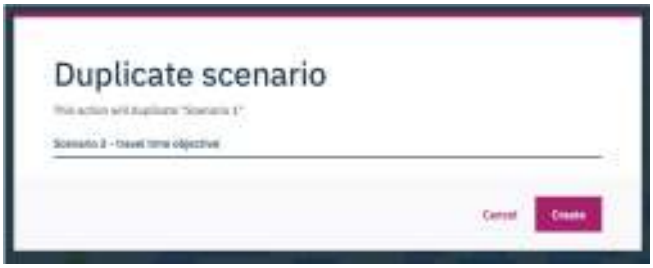
1. While in the DO4WS wizard, click on  near the top right corner of the screen.

   This will open the scenario management area as shown below.




2. Click on the three dots next to **Scenario 1** and choose **Duplicate** from the options. Provide a name for the new scenario and click **Create**.




3. Make sure the new scenario is highlighted in the **Scenario** frame and go to **Run Model** tab from the left frame.

4. In the natural language input box, type *minimize transit cost based on distance*.

   Expand the suggestions and find an entry which may say

*"Minimize transit cost between assigned operations over all scheduling assignments based on travel_times"*
*OR*
*"Minimize transit cost between assigned operations over all scheduling assignments based on distance"*

*The last entry travel_times or distance is table name and can be changed.*

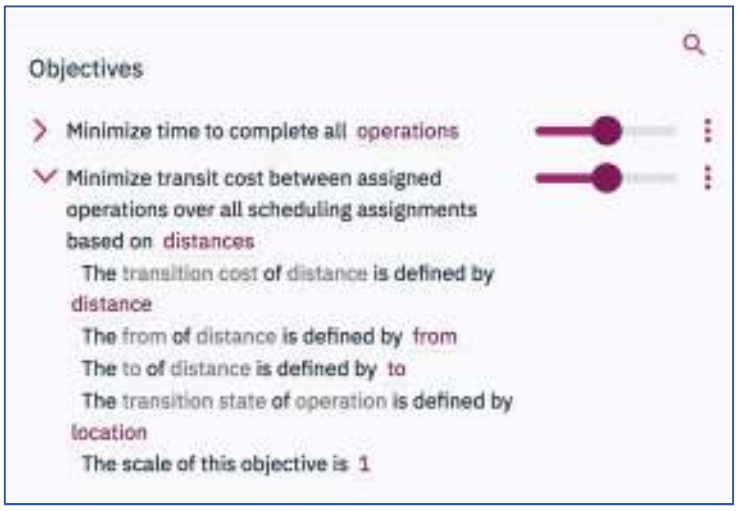
5. Click the **"+"** sign next to the suggested objective *Minimize transit cost between assigned operations over all scheduling assignments based on travel_times.*

Since we used the keyword "minimize", this will automatically get added as an *objective*.

6. Expand the red arrow next to this new objective and configure the entries.


- Click on travel_times and change it to *distances*.
  - The from of distance is *from*
  - The to of distance is *to*
  - The transition cost of distance is defined by *distance*
  - The transition state of operation is defined by *location*

The final set of objectives will look similar to this screenshot:

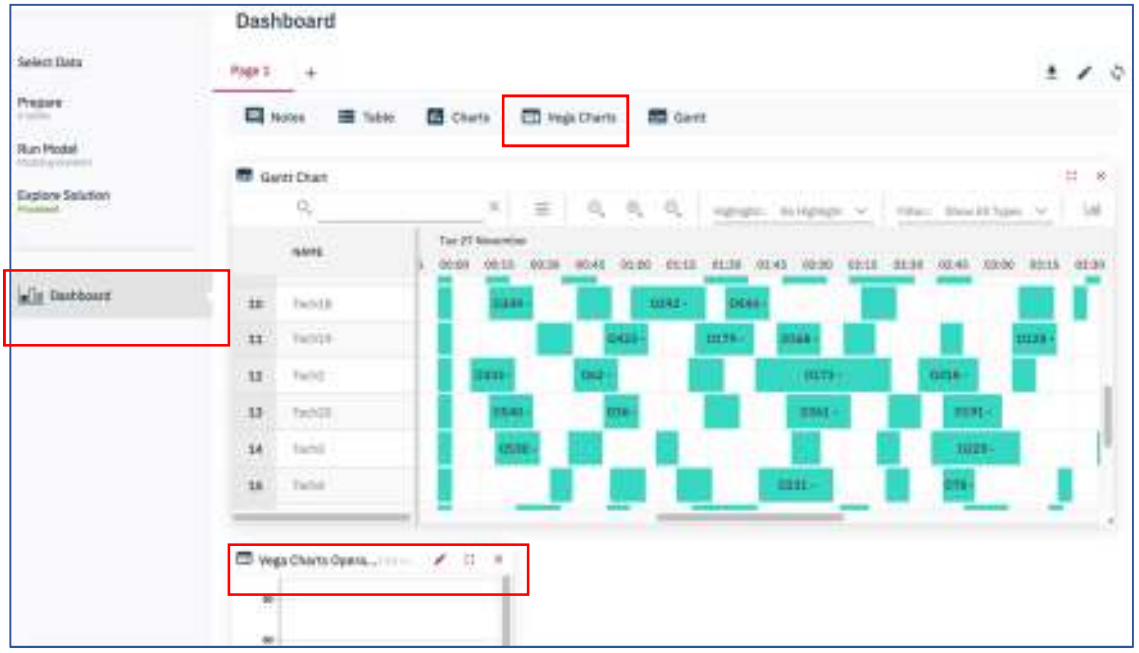


7. Click the **Run** button on top right corner to solve the new model.

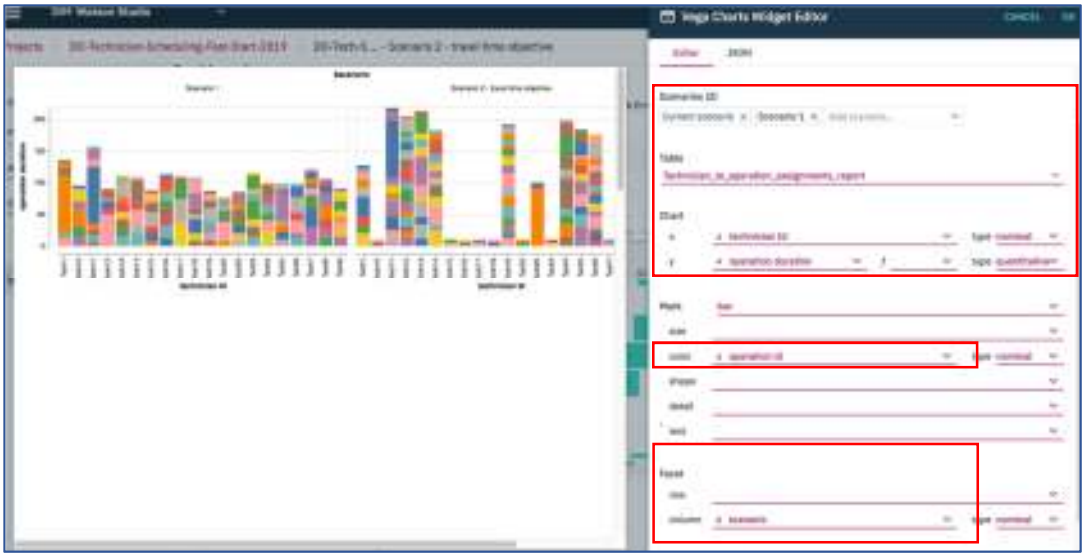
Once successfully solved, the screen will be at the **Explore Solution** tab and show the new objective table and solution tables.


8. Compare the objective value for *time to complete all operations* between the two scenarios. Notice the slight difference. Let's compare the results with some more visualization.

   Navigate to the **Dashboard** tab and click **Vega Charts**. This will add a chart window in the main dashboard area. On top frame of that chart window, click the pencil icon to open the configure widget for the chart.

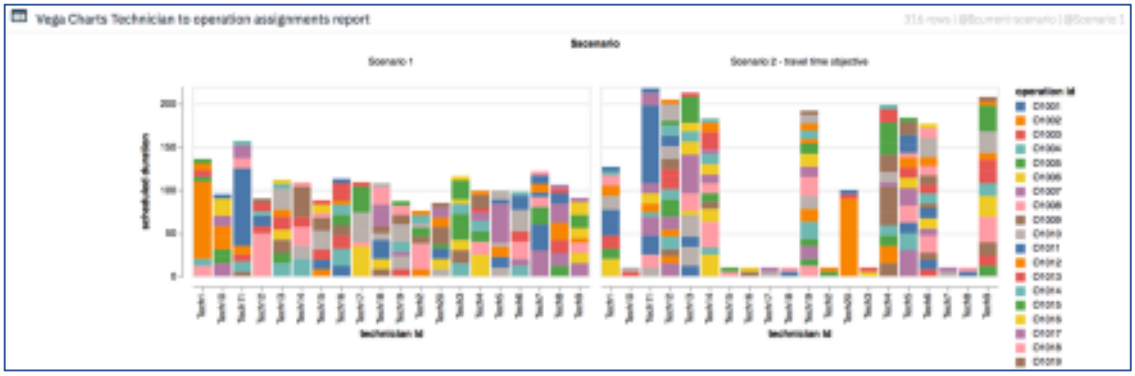


9. In the chart configuration widget make the following changes:
    - Add both scenarios
    - Choose the table *technician to operation assignment report*
    - For **Chart x axis**, pick *Technician Id*
    - For **Chart y axis**, pick *operation duration*
    - Under **Mark**, for color, choose *operation id*
    - Under **Facet**, for column, choose *scenario*



10. The final chart should look similar to the following screenshot.

    *Scenario 2* included an additional objective to reduce total distance travelled. As a result, the total time to complete all operations has suffered. Technician load is also less optimal: *scenario 2* has more uneven work distribution among technicians compared to scenario 1.




**CONGRATULATIONS!**
**You have successfully completed the DO4WS Technician Scheduling Lab.**