**Fashion Reverse Logistics Optimization - Tutorial for Non-Expert Users**

This tutorial will guide you through setting up and using the Fashion Reverse Logistics Optimization tool to plan efficient routes for collecting items from multiple shops and returning them to a central depot.

**Installation Guide**

1. **Install Python**:
   * Download and install Python 3.8 or higher from [python.org](vscode-file://vscode-app/c:/Users/jpern/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)
   * During installation, make sure to check "Add Python to PATH"
2. **Download the Project**:
   * Download or clone the project files to a folder on your computer
3. **Install the Package**:
   * Open Command Prompt (Windows) or Terminal (Mac/Linux)
   * Navigate to the project folder:

*cd path/to/CVRP-FashionRL*

* + Install the package in development mode:

*pip install -e .*

* + This will also install all required dependencies automatically

1. **Additional Solvers (Optional)**:
   * **For Gurobi (Exact Solver)**:
     + Install Gurobi from [gurobi.com/downloads](vscode-file://vscode-app/c:/Users/jpern/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)
     + For academic users: Obtain a free academic license from Gurobi's website
     + For commercial users: Contact Gurobi for a commercial license
   * The program will work without Gurobi using the heuristic or OR-Tools solvers

**Using the Application**

**Starting the Application**

1. Open Command Prompt (Windows) or Terminal (Mac/Linux)
2. Navigate to the project folder if you're not already there:

*cd path/to/CVRP-FashionRL*

1. Launch the application with the graphical interface:

*python -m src.main --gui*

**Creating Sample Data**

1. In the application window, click the "Create Sample Instances" button
2. This will generate sample problems in the "examples" folder
3. A confirmation message will appear when samples are created

**Solving a Problem**

**Using Sample Data:**

1. Make sure "Generate random instance" is unchecked
2. Click "Browse" and navigate to the "examples" folder
3. Select one of the sample instances (e.g., small\_instance.json)

**Creating a Custom Problem:**

1. Check "Generate random instance"
2. Enter the number of shops (e.g., 15)
3. Enter the vehicle capacity (e.g., 10)
4. Enter a random seed (any number, e.g., 42)

**Configure Solver Settings:**

1. **Solution method**:
   * **"Exact (Gurobi)"** - For optimal solutions (best but can be slow for large problems)
   * **"Heuristic (ILS)"** - For quick approximate solutions
   * **"CP (OR-Tools)"** - Alternative solver approach
2. Enter a time limit in seconds (e.g., 30)
   * For small instances (10 shops): 10-30 seconds is usually enough
   * For medium instances (20-30 shops): 60-300 seconds recommended
   * For large instances (40+ shops): 300+ seconds recommended
3. If using the heuristic method, set the "Max iterations" (default: 100)

**Run the Solver:**

1. Click "Run Solver"
2. The status area will show progress and results
3. You can stop the optimization process anytime by clicking "Stop Solver"
4. When complete, solution files will be automatically saved in a timestamped results folder (e.g., results\_20250523\_120145)

**Interpreting Results**

The solution information is displayed in the status area and shows:

* Number of vehicles/routes needed
* Total distance traveled
* Details for each route including:
  + Sequence of shops visited
  + Distance for each route
  + Load on each vehicle

The saved files include:

* **[results folder]/reverse\_logistics\_solution.png** - Visual map of routes
* **[results folder]/reverse\_logistics\_solution.csv** - Data file of the solution
* **[results folder]/solution\_details.txt** - Detailed report of routes

**Understanding the Visualization**

The route visualization shows:

* **Red square**: Central depot (starting and ending point)
* **Blue circles**: Shop locations (with numbers)
* **Colored lines**: Different routes for vehicles
* **Arrows**: Direction of travel
* **Legend**: Route identifiers
* **Title**: Method used, date/time, and key metrics

**Command Line Usage**

If you prefer using the command line or need to process multiple instances, you can use:

*python -m src.main --file examples/small\_instance.json --method heuristic --time-limit 60*

Additional options:

* --shops 15 - Number of shops for random instance
* --capacity 10 - Vehicle capacity for random instance
* --seed 42 - Random seed for reproducibility
* --iterations 200 - Maximum iterations for heuristic solver

**Tips for Best Results**

* Start with small problems (10-20 shops) before trying larger ones
* For exact solutions to larger problems, allow longer running times
* If an optimal solution takes too long, try the heuristic method
* Use the same random seed to get reproducible results when generating random instances
* The OR-Tools solver often provides a good balance between speed and solution quality

**Troubleshooting**

* **"Module not found" errors**: Make sure you installed the package with pip install -e .
* **GUI doesn't appear**: Check if Tkinter is properly installed with your Python
* **Gurobi errors**: The program will automatically use other solvers if Gurobi is not available
* **No solution found**: Try increasing the time limit or switching to a different solver
* **Slow performance**: For large problems, use the heuristic solver instead of the exact solver

**Solution Files**

All solution files are organized in timestamped folders (e.g., results\_20250523\_120145) to keep track of different solution attempts. Each folder contains:

1. **solution\_details.txt** - Contains:
   * Solver method used
   * Timestamp
   * Number of shops
   * Vehicle capacity
   * Total distance
   * Number of routes
   * Detailed route information
2. [reverse\_logistics\_solution.csv](vscode-file://vscode-app/c:/Users/jpern/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Contains the raw solution data (from-to nodes)
3. [reverse\_logistics\_solution.png](vscode-file://vscode-app/c:/Users/jpern/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Visual representation of all routes