Transportation Optimizer – *Mojo*

……..sustaining the environment one mile at a time

MercuryGate's transportation optimizer (Mojo) was built to provide both shippers and logistics providers the ability to reduce greenhouse gases while saving costs for transportation. Mojo shows demonstrable CO^2 reductions for transportation plans allowing the transportation department to be a major contributor in the overall green initiatives of corporations. MercuryGate built Mojo to solve three critical issues:

1. Enable multiple pickup and drop scenarios for consideration (continuous moves) along with backhauls and cross-docking scenarios (most optimizers are built for single location optimization).
2. Provide continuous feedback as to why the system was not able to optimize the shipments.
3. Easily load data in and get results out.

Mojo can take shipments/orders from multiple locations/customers and determine the most cost effective way to deliver those shipments. The optimizer engine will consider cross-docking, continuous moves, backhauls and simple consolidations. Complex algorithms are fed by a simple set of spreadsheets that may be generated from the TMS or merely uploaded into the Mojo tool. Mojo considers all designated options and allows the user to specify rules that must be adhered to as the alternatives are considered. If Mojo is not able to optimize an order, the tool will tell the user what criteria was a limiting factor in not being able to optimize the shipment. All of these factors combine to make Mojo a user friendly optimizer with continuous feedback on ways to improve the solution.

The process begins with a set of data to consider for optimization. Information such as shipments, rates, cross-dock facilities, carriers, and parameters all feed into the engine to determine the optimal solution.

The data may be loaded via a spreadsheet or from the MercuryGate TMS. Either way, the information in the database is now ready to feed into the optimizer. If the data comes from an external source, Mojo has simple wizards that allow the user to map and identify to Mojo the columns that contain the information necessary to run Mojo (see figure 1).

Multi-pickup And drop scenarios – backhauls – cross-docking and more…

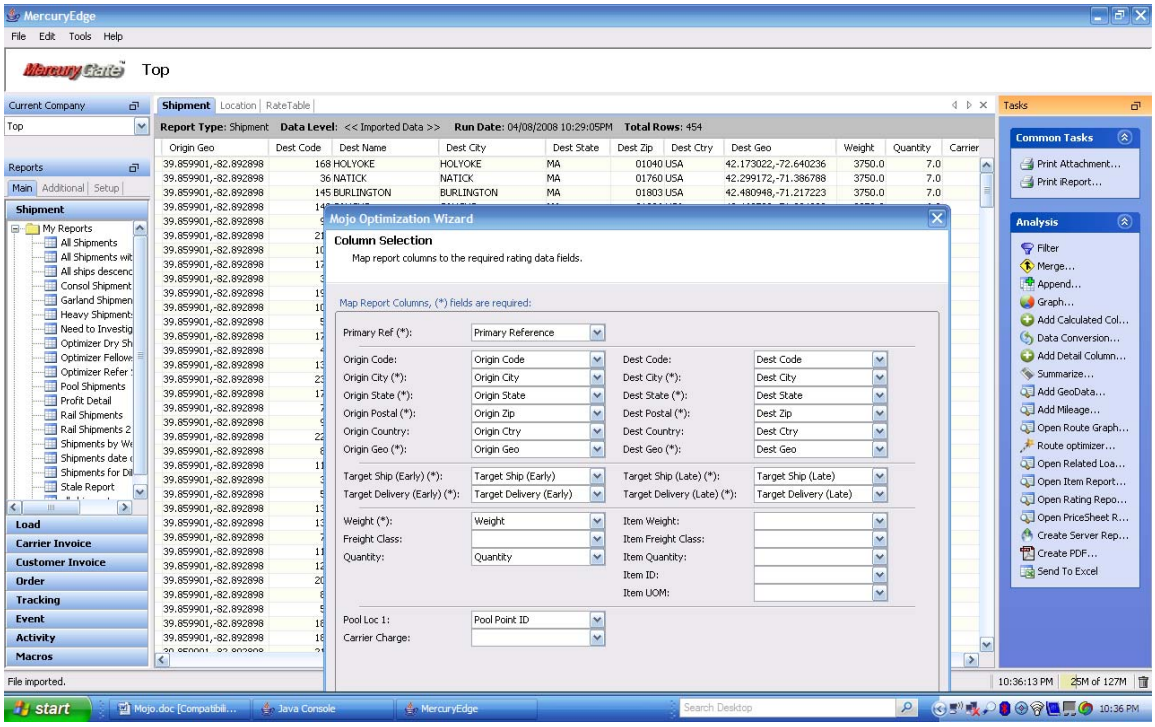

Figure 1: Mapping wizard allows users to easily import data for running Mojo.

Once the data has been successfully loaded and mapped, the user now needs to specify parameters for the optimizer. The best way to describe what Mojo does is by the use of scenarios.

Scenario 1: Multi-pickup/multi-drop truckload

The first scenario is a simple one with a twist. There are multiple shipping locations and multiple customer locations. This would be a typical 3PL world where they are managing the freight for multiple customers or this may be a shipper that has multiple distribution centers and is running a transportation control center that is optimizing and determining carrier routing. In this example, there are multiple LTL and truckload shipments that are all shipping on different dates. The user has specified that the ship date is only important from the perspective of meeting the delivery date allowing more opportunities to consolidate loads. In this example, 1154 shipments are shipping out of Charlotte, Atlanta area, and Dallas area to all numerous destinations. Mojo indicates to the user that in this scenario it created 780 loads that saved over $210k in transportation costs. **Mojo also shows that this optimized routing eliminated almost 25 tons of CO^2 emissions helping to sustain the environment.** The loads created were a mix of multi-drop, continuous moves, and of course simple point to point moves that could not be optimized. Notice on the results page the optimizer shows you the event sequence (i.e. PDDPDD – pickup, drop, drop, pickup, drop, drop) for easy viewing and understanding of what the load looks like (see figure 2 below).

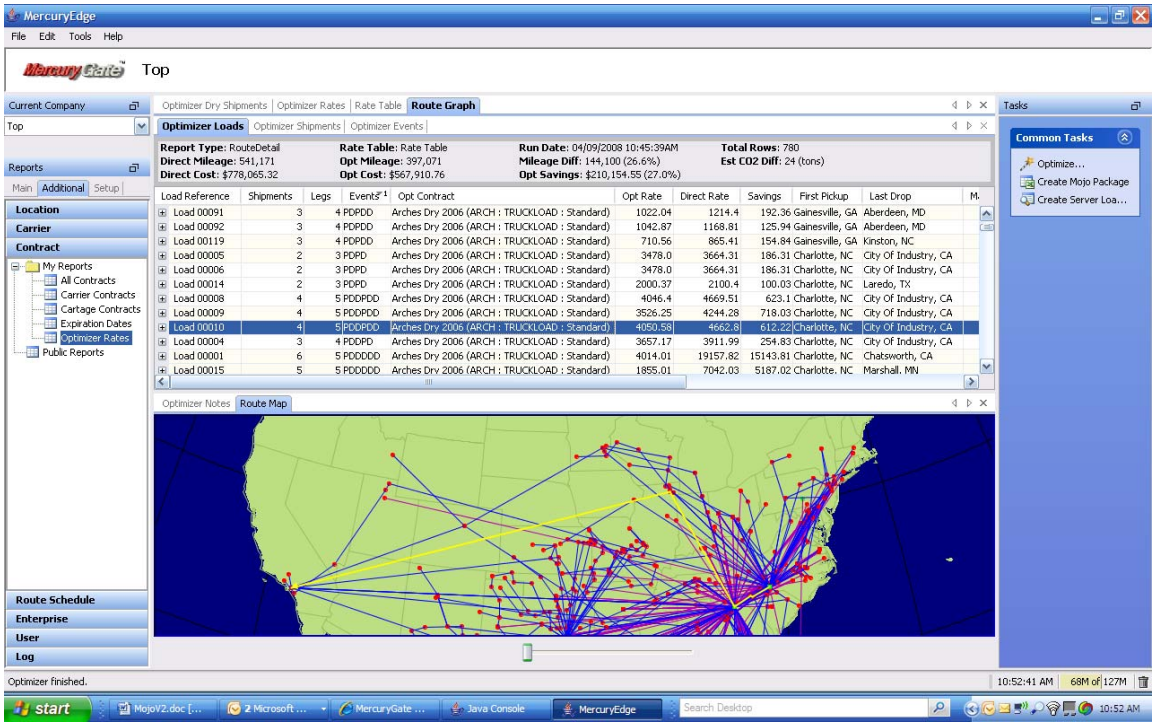

Figure 2: Scenario 1 showing continuous moves involving multiple shipping points. Note Event column showing PDDPDD as an example.

Scenario 2: Forced Pooling (Cross-docking)

In the second scenario, a shipper is delivering goods to a retail outlet in a mall situation. The goods are all originating out of Ohio and going into various mall locations throughout the United States. This example is called forced pooling because the 3PL in this case has setup multiple pool locations to feed the shipments into the stores. There is a line haul out of the distribution center into the pool and then the pool may deliver to one or many stores. The big issue to solve here was one of consolidating the line hauls into a multi-stop load into multiple pooling locations. There are also some savings in consolidating shipments from the pool facility to the store. In this scenario the savings being measured are against the cost of shipping the freight directly to each pool vs. consolidating freight as well as stopping at multiple pool locations. The example took 454 shipments and created 253 loads (the load count includes outbound loads from the pool as well as into the pool location, see figure 3 below.) Mojo saved over $41k and reduced green house emissions of CO^2 by over 60 tons!

Finally, Figure 4 below shows one aspect of the continuous improvement part of Mojo. In scenario 1 described above the maximum weight was set at 45,000 pounds. However, many of the shipments exceeded that rule just as standalone shipments and therefore were excluded from optimization. There may be nothing you as a user can do about the restriction but it is important to note that certain shipments were excluded from being optimized based on any rule specified. Additional improvements can be calculated by Mojo. You may designate for Mojo to determine the optimum number of stops to set as a rule. By indicating for Mojo to start at a maximum of eight stops and step down by a single stop to four, Mojo will utilize the best number of stops and then indicate to the user which number of stops is optimum. Mojo will do these types of analysis on many different types of rules allowing the user to determine based on their data the proper settings and/or contract terms to negotiate to maximize their transportation savings.

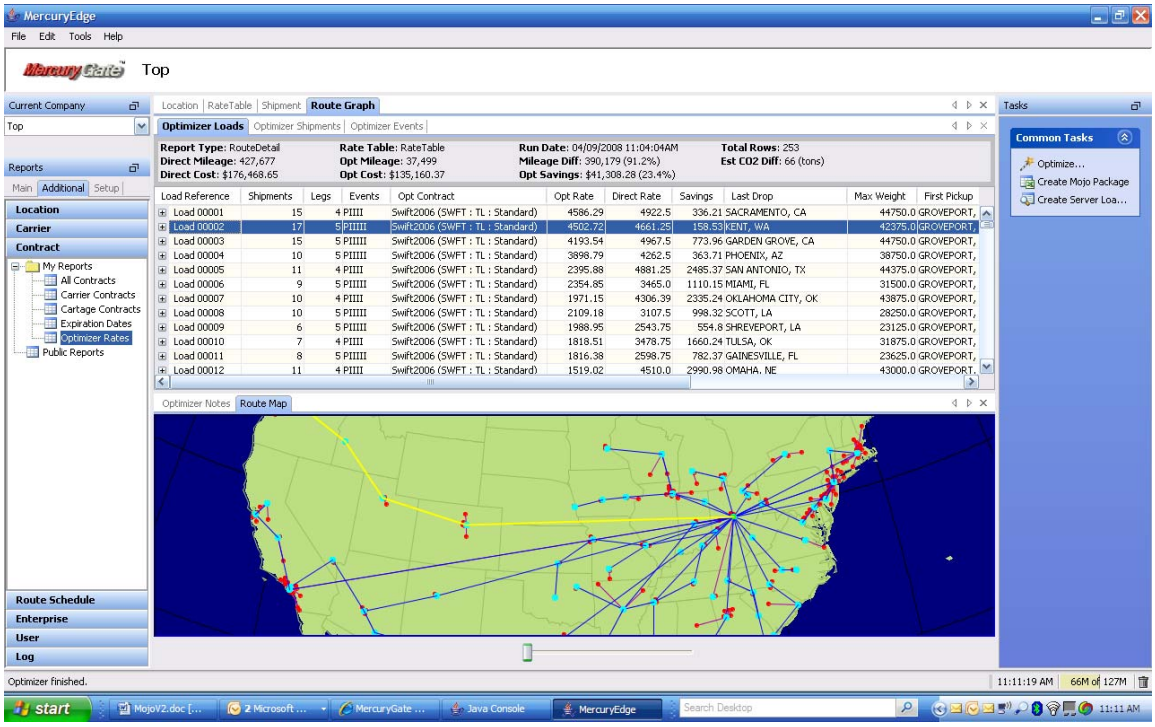

Figure 3: Scenario 2 showing forced pooling

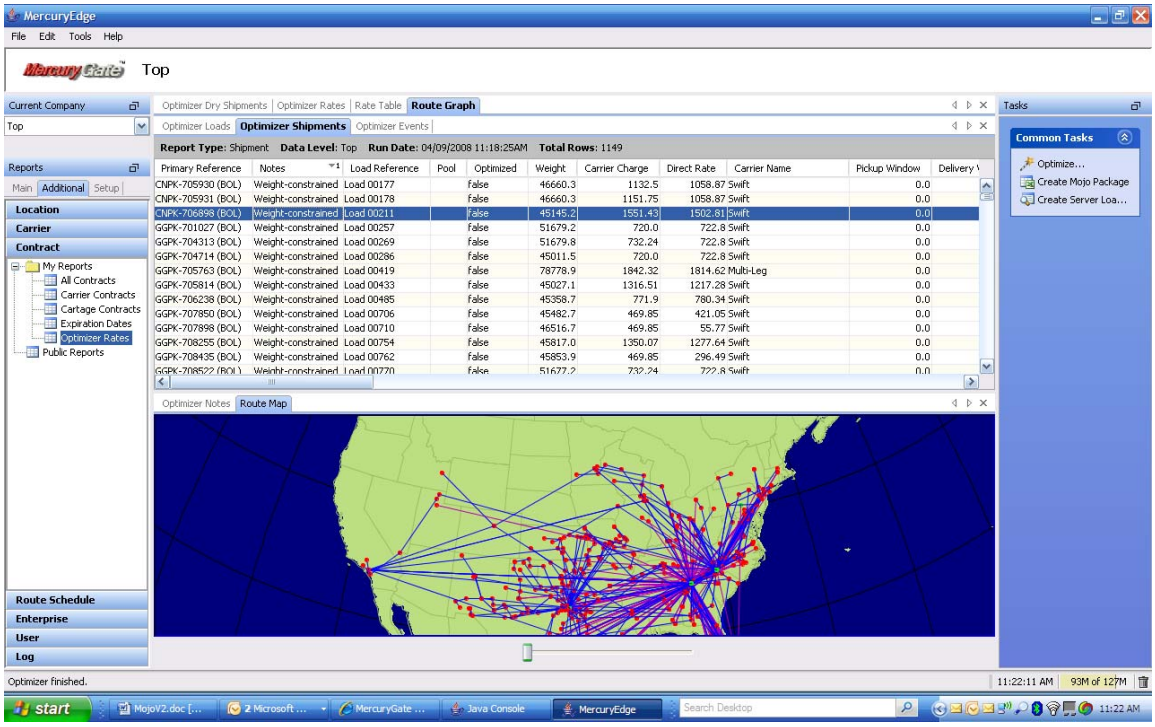

Figure 4: Optimizer indicates why it cannot optimize certain shipments so user may understand implications easily.

Scenario 3: Sailing Schedule Optimizer

MercuryGate prepares you to really get the supply chain sizzling by addressing your Ocean transportation. The great misconception is that there is little anybody can do to optimize such a simple transportation move. However when you start to look at the containers, chassis, schedules, cut dates, slots on vessels and product availability the problem becomes very complex. All you need is MercuryGate's Schedule Optimizer Mojo! Mojo takes your orders with their dates and products and combines them with your available equipment to make sure product is delivered to the port by the cut date to maximize sailing slots based on cost and still meet customer demand. The optimizer even takes into account weekends, holidays, and equipment turnaround.

Scenario

The issue is one where a manufacturer is shipping product to China. The manufacturer may ship out of two different ports on many different carriers but only has so much capacity on each ocean vessel. To further add complexity, the manufacturer only has so many chassis to utilize for each ocean carrier. They can use third party chassis but this cost additional monies. The manufacturer wants to maximize the low cost voyages as well as minimize the use of third party chassis. Constraints such as chassis turnaround, cut dates, product availability, holidays and weekends must all be taken into account when trying to solve the problem.

Scenario Solution

The solution begins with the loading of the data. A spreadsheet can be loaded that contains different workbooks for orders (shipments), schedules, rates, and constraints.

Once the optimizer runs, it presents a shipment-to-schedule assignment on the "Optimizer Loads" tab indicating when they need to arrive at the port for which vessels (see Figure 5 below). The beauty of Mojo is that it takes into account all of the constraints and determines based on the mix of product, shipments, equipment, and schedules to get the best cost while maximizing equipment and vessels. The "Optimizer Shipments" tab shows the user how a shipment was optimized or why the shipment failed to get assigned to any load. The "Notes" and "Constraint Notes" columns provide the detail of why the shipment failed to optimize. The "Calendar" tab provides a graphical view of the results allowing the user to see calendar constraints (see Figure 6 below), etc.

Mojo considers containers, chassis, schedules, cut dates, slots on vessels and other complex variables.

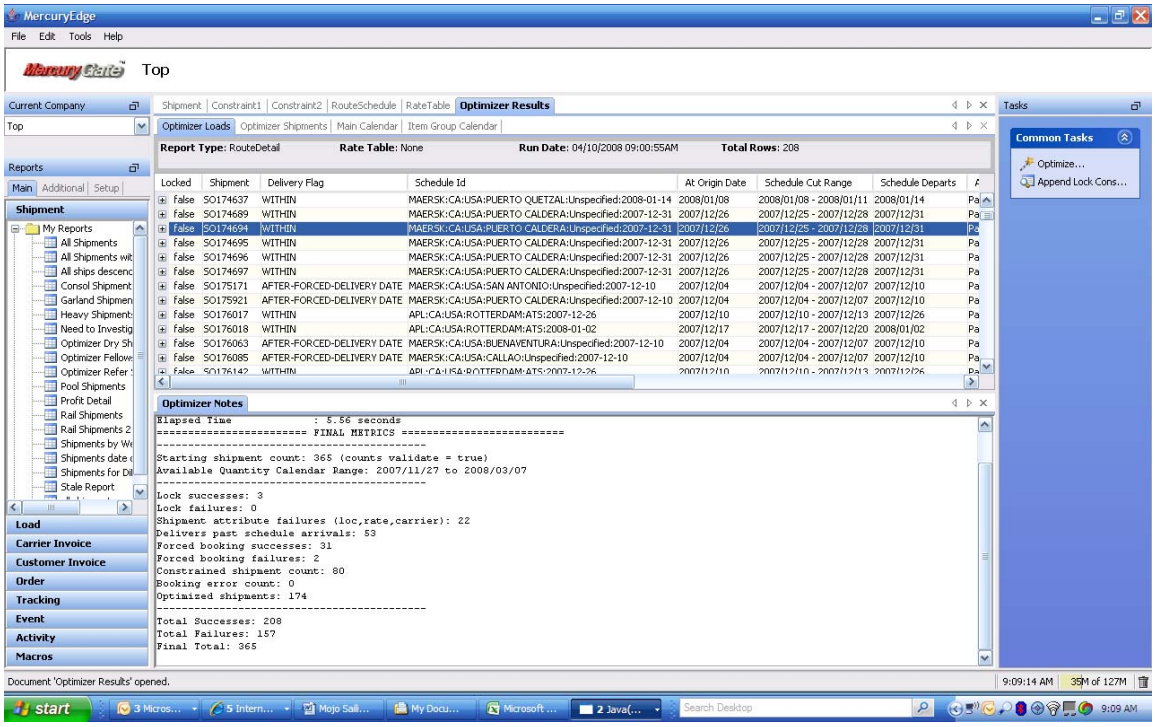

Figure 5: Results of optimization showing loads, schedules used, etc.

Once the user likes the loads that are planned they may be locked on that schedule so that as future shipments are considered, previous committed capacity will be taken into account.

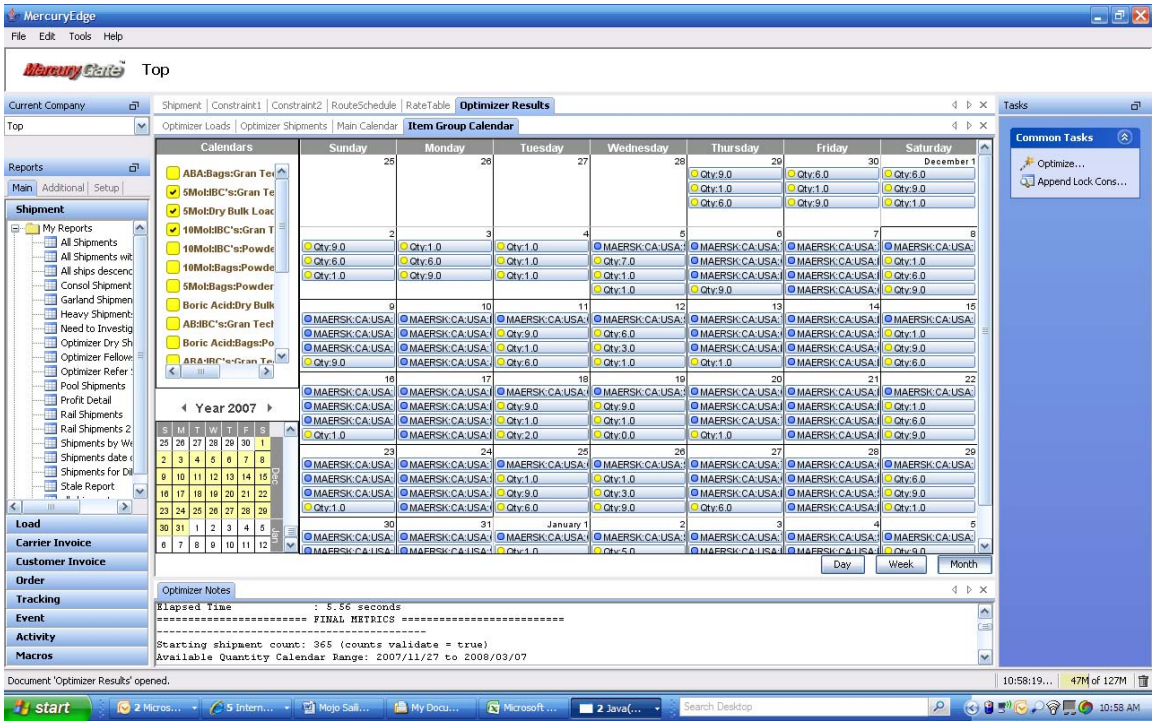

Figure 6: Calendar showing product and Maersk voyages

For more information email sales@mercurygate.com