Fleet Optimization Problem

By: Mo Hessien & John Pflugh

Introduction

Introduction

- This is a transportation optimization problem.
- The fleet is tasked with delivering oil products to multiple points of sale, or gas stations, distributed throughout the region.
- Each truck begins its route by loading oil products from an assigned depot. Trucks need to return to a depot after each delivery to be checked then reload before heading to the next station. As a result, a typical shift for a truck involves a sequence of trips such as: Depot → Station 1 → Depot → Station 2 → Depot → Station 3, and so forth.

Introduction

Under the fleet leasing contract, the operational cost structure for trucks consists of two main components:

1. Variable costs: depend directly on mileage (fuel, wear and tear, etc.).
2. Fixed costs: constitute the majority of the total cost and are incurred per truck regardless of how much it is used.

Because the fixed cost component is substantial, the number of trucks deployed becomes a key decision variable in the optimization model.

Problem Description

The primary strategic goal is to maximize the total volume of products shipped, ensuring the company maintains its market share and competitive positioning. Leadership has emphasized that delivering the highest possible quantities is the top priority.

A secondary objective is to minimize total mileage, thereby reducing variable operating costs and improving fleet efficiency.

Together, the optimization goal can be viewed as:

Maximize shipped volume while minimizing the total distance traveled, with priority placed on maximizing shipments.

Problem Description

Constraints:

1. **Supply Constraint:** Each depot has a limited amount of product available, based on monthly purchase quantities. For every product type, the total volume shipped from a depot cannot exceed its available supply.
2. **Capacity Constraint:** Each gas station can store only a certain amount of each product. Deliveries must respect these storage capacity limits, ensuring that no station receives more product than it can store safely.
3. **Driver Shift Constraint:** For every truck, the total working time (including driving time and loading/unloading activities) must not exceed the allowable 12-hour driver shift limit.

Problem Assumptions

Constant Speed (45 km/hr): We assume that the time it takes to travel a distance (in kilometres) along city roads is exactly the product of the distance and the average speed parameter of the truck (in kilometres/hour).

We make this assumption purely because it simplifies our model and is ‘good enough’ for the scenario we are modelling.

This assumption affects our driver shift-time constraint.

Similarly, we make the assumption that it takes a constant amount of time to load and unload a given truck at a given depot-pair, also to simplify our model.

Problem Assumptions

Full Shipments Only: We assume that if a truck is used to make a trip, that truck is filled completely up to the capacity of each product.

We make this assumption because:

- ❏ Moving trucks that have an abnormal center of gravity can be dangerous, especially with petroleum products, and
- ❏ Our client company does not want to pay transportation costs for half-empty trucks, even if it would improve shipping efficiency.

This assumption affects:

- ❏ Our objective function, and
- ❏ Our depot supply & station capacity constraints.

Problem Assumptions

Supply is Not Refilled (No Mid Month Purchases): The supply for each depot is purchased before the beginning of the month and is not refilled in any way during the month, though it is depleted by trucks in each shift, so that each shift has less supply than before.

This assumption reflects the situational reality, except for some rarely-occurring exceptions that we have chosen not to model.

This assumption affects our depot supply constraint.

Sales are Constant and Uniform: We assume the uniformity of sales, so the rate of sales remains constant between all different shifts. This assumption simplifies the model formulation, but also reflects the general trends of the situation.

This assumption affects the calculation of updated station capacity between shifts.

Most Important Question:

If the trucking vendor offers multiple leasing options, how can we evaluate and select the most efficient one?

Methodology - Sets, Parameters, & Variables

In our single-shift model, we use:

- DEPOTS, the set of depots to store and distribute product,
- STATIONS, the set of stations to transport product to for sale,
- TRUCKS, the set of trucks available to transport product, and
- PRODUCTS, the different types of product to be transported and sold.

There are many numerical parameters indexed by these sets and used in the model. These are defined in detail in our ‘prod.mod’ AMPL model file.

In a single shift, each truck can take trips from depots to stations to transport product. For each combination of a depot, a station, and a truck, the nonnegative integer variable assign_truck represents how many trips from that depot to that station the given truck takes during the shift.

Methodology - Objective

Our objective function in the single-shift model actually contains two objective functions - we want to use **lexicographic methods** for **multi-objective optimization** to find a solution that most importantly transports as much product as possible, and within this paradigm uses the least total distance among all trucks.

We rank our objectives:

1. **Maximize** total product capacity transported during the shift
2. **Minimize** distance used by all trucks during the shift

We call the first objective function f_1 and the second f_2.

We always prioritize the first objective - the second objective is only used to determine out of the possible solutions for the first objective, which solution should be used.

Methodology - Objective

There exists some set of weights $w_1 > w_2 > \cdots > w_n$ of the n objective functions such that the single-objective weighted sum of the objectives has **an identical solution** to the lexicographic multi-objective formulation.[1]

We can get close to perfect lexicographic optimization. Specifically, we:

- Compute f_2^* to be the **maximum possible value of** f_2 for the given data. Since f_2 is defined based off parameters that are not updated between iterations, we can calculate f_2^* once at the beginning of the problem and reuse it across all shifts.
- In practice, we obtain the maximum value of f_2^* by minimizing an IP with the same feasible region as the multi-objective formulation but with only the objective f_2.
- We replace our multiple objectives with the **single objective** $\max f_1 - \frac{m}{1+f_2^*} \cdot f_2$.

Methodology - Objective

Then a decrease by any possible amount of f_2 contributes less to the overall objective value than increasing f_1 by at least m. We set $m < 1$; we can think of m to be the smallest increase in f_1 that we care to be significant, since for any smaller increase, we cannot guarantee that an increase in f_2 of some amount will not produce an improved overall objective value.

In practice, the solution is very robust no matter the value of m chosen. We have selected a default value of $m = 0.01$.

This is a modification of the weight-calculation method proposed by Yager.[2]

Methodology - Code Implementation

- All python code runs in the ‘model.py’ script.
- Our base single-shift model is written into an AMPL model file (‘proj.mod’).
- We read this AMPL model into each of the 60 shifts in our model function. We initialize the model data before the first shift, and run the auxiliary IP to determine the best possible value of the secondary objective. We then run each shift iteration, updating supply and store capacity in between runs.
- We repeat this process for each of the desired subsets of trucks.
- Data processing, and plotting is done with the pandas and matplotlib packages.

Methodology - Computational Error

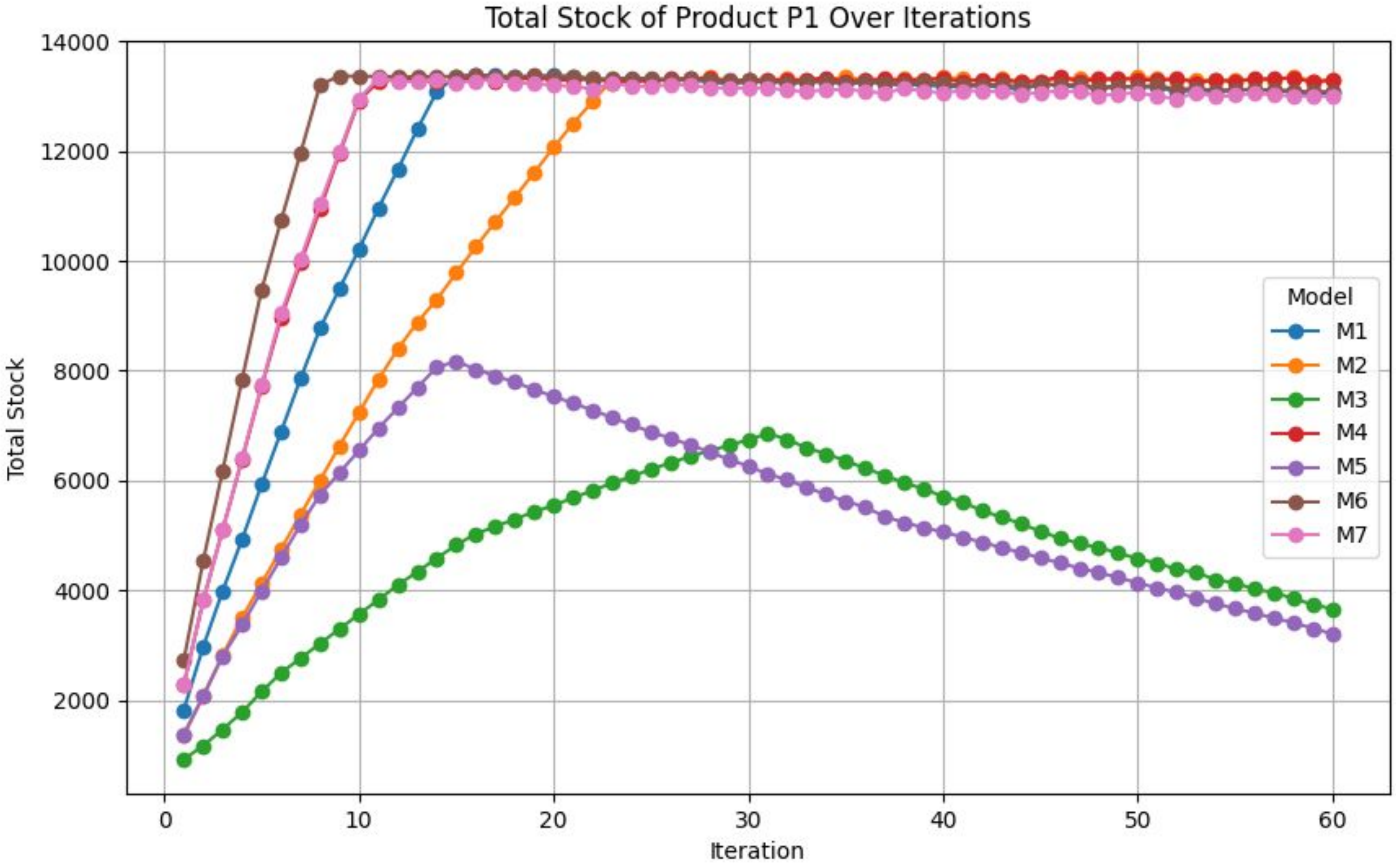
- When using so many iterations, slight sources of error in the MIP solving process build up.
- We use the Gurobi optimizer within the AMPL interface - Gurobi can cut off integer program solutions before total optimality (see Gurobi documentation for MIPGap options).
- In practice, MIP solvers will introduce some error unless tightly controlled, which can worsen solving times significantly.
- For us, the level of error is acceptable - anecdotally, the error is of the order of 10^{-4} relative to the derived quantities.

Results - Different Trucks

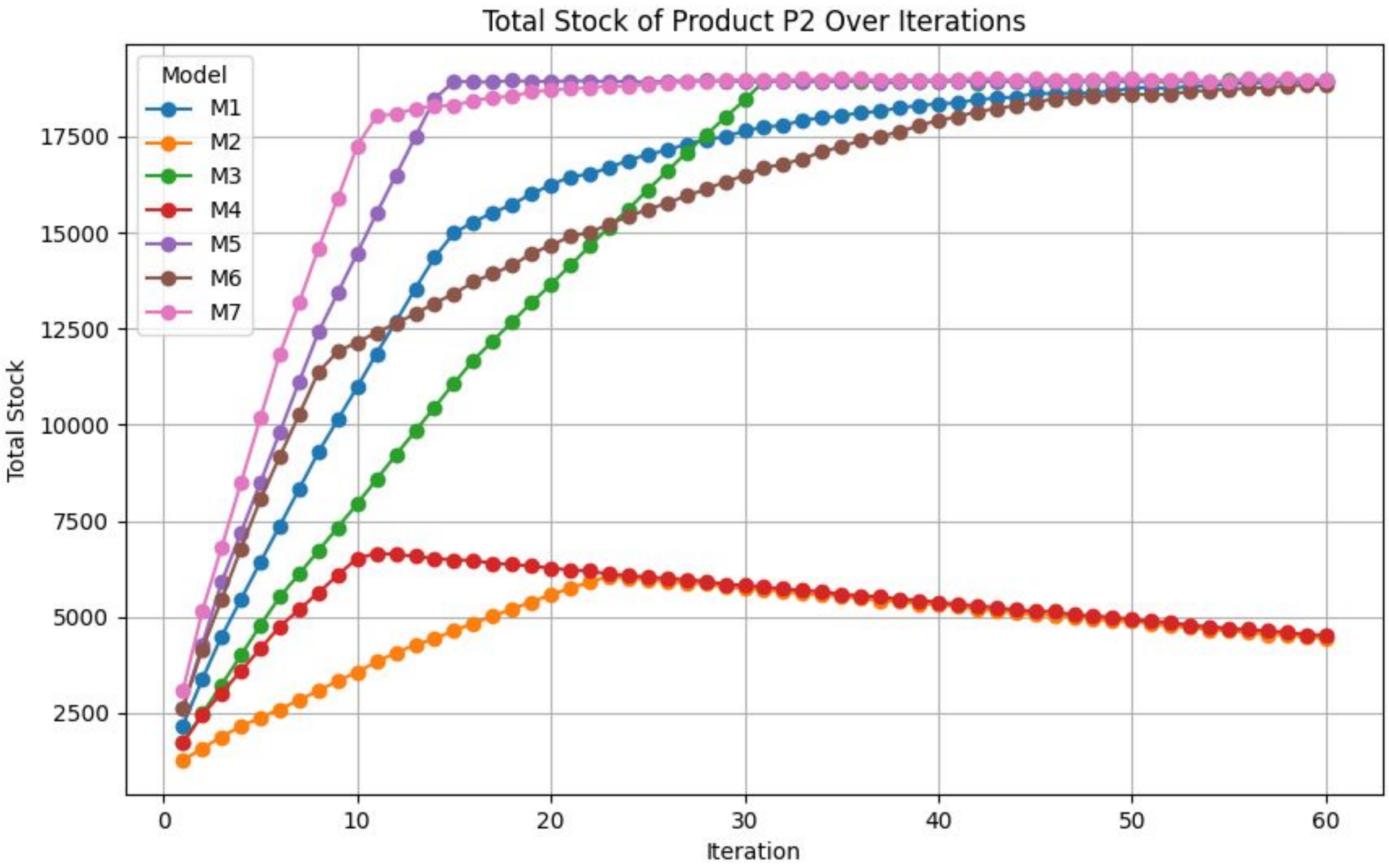
To determine which slate of trucks to purchase, we compare different models using different sets of available trucks.

A ‘type 1’ truck has a capacity of 30 product P1 and 15 product P2, and a ‘type 2’ truck has a capacity of 15 product P1 and 30 product P2.

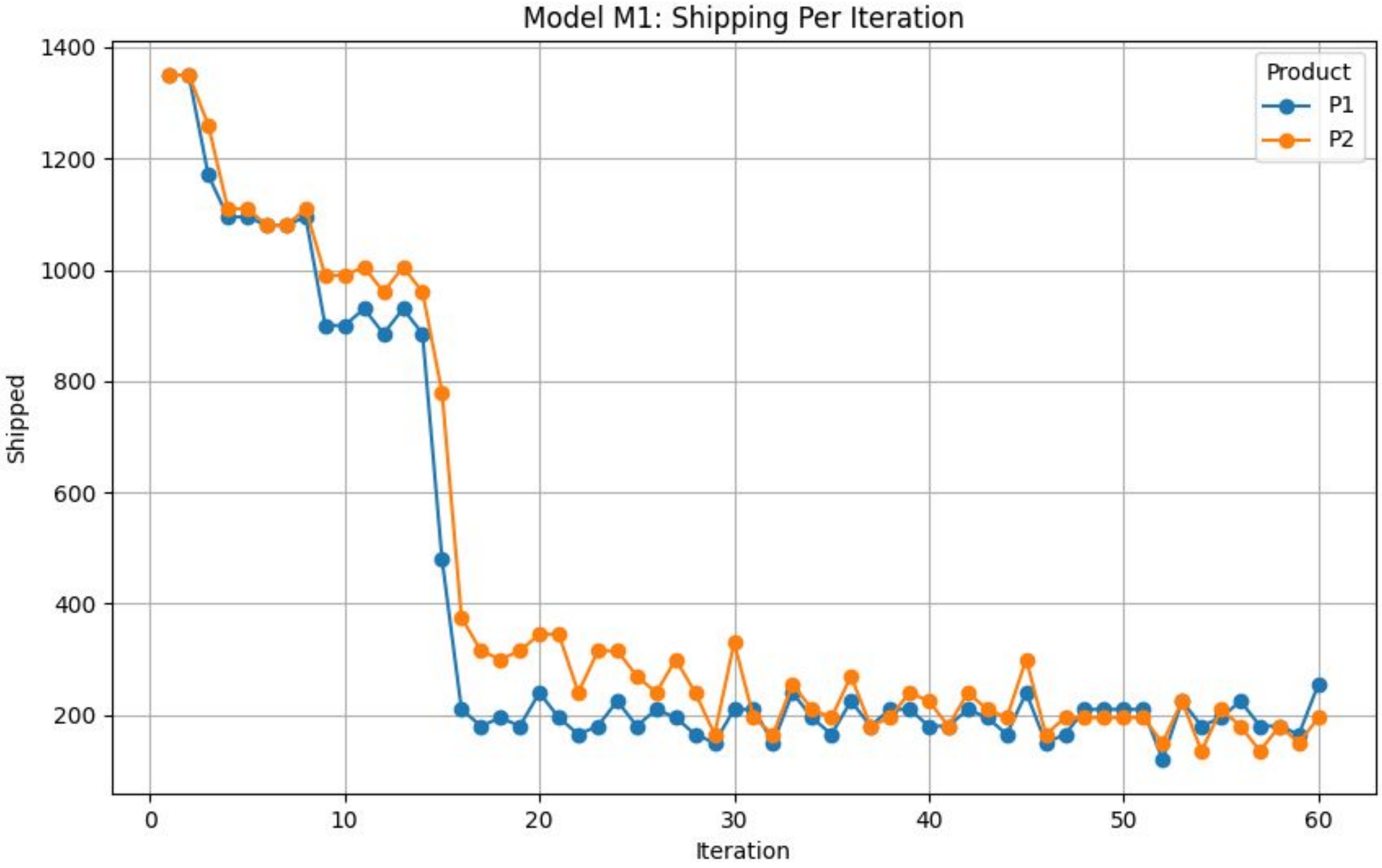
M1: 6 type 1 trucks, 6 type 2 trucks.	Total: 12 trucks.
M2: 6 type 1 trucks.	Total: 6 trucks.
M3: 6 type 2 trucks.	Total: 6 trucks.
M4: 12 type 1 trucks.	Total: 12 trucks.
M5: 12 type 2 trucks.	Total: 12 trucks.
M6: 12 type 1 trucks, 6 type 2 trucks.	Total: 18 trucks.
M7: 6 type 1 trucks, 12 type 2 trucks.	Total: 18 trucks.



Total Stock of Product P1 Over Iterations
Total Stock
Iteration
14000
12000
10000
8000
6000
4000
2000
0
10
20
30
40
50
60
Model
M1
M2
M3
M4
M5
M6
M7



Total Stock of Product P2 Over Iterations
Model
M1
M2
M3
M4
M5
M6
M7
Total Stock
Iteration
2500
5000
7500
10000
12500
15000
17500
0
10
20
30
40
50
60


Model M1: Shipping Per Iteration
Product
P1
P2
Shipped
Iteration
1400
1200
1000
800
600
400
200
0
10
20
30
40
50
60

Appendix - Single-Shift Algebraic Formulation

For each shift in our month-long timespan, we solve a single Integer Program.

Here, we discuss the specific formulation of this IP using the following algebraic conversion from our project code:

- ❏ Sets DEPOTS, STATIONS, PRODUCTS, TRUCKS become D,S,P,T, and
- ❏ Parameters distance, supply, capacity_station, full_capacity_station, capacity_truck become g,h,i,j,k and
- ❏ Parameters avg_speed, shift_duration, load_unload_time become a,b,c and
- ❏ Parameter diff_error becomes m, and
- ❏ Variable assign_truck becomes x, and
- ❏ Constant f_2^* is as previously defined - this is equivalent to max_secondary_objective.

These are intended to improve readability and accessibility of the algebraic formulation for later reference.

Appendix - Single-Shift Algebraic Formulation

$$
\begin{aligned}
\text{max} \quad & \sum_{d\in D}\sum_{s\in S}\sum_{t\in T}\sum_{p\in P} k_{t,p}\cdot x_{d,s,t} - \frac{m}{1+f_2^*}\sum_{d\in D}\sum_{s\in S}\sum_{t\in T} g_{d,s}\cdot x_{d,s,t} \\
\text{subject to} \quad & \sum_{s\in S}\sum_{t\in T} k_{t,p}\cdot x_{d,s,t} \le h_{d,p} && \forall d\in D, p\in P \\
& \sum_{d\in D}\sum_{t\in T} k_{t,p}\cdot x_{d,s,t} \le j_{s,p} - i_{s,p} && \forall s\in S, p\in P \\
& \sum_{d\in D}\sum_{s\in S}\left(c_{d,s,t} + \frac{g_{d,s}}{a_t}\right)\cdot x_{d,s,t} \le b_t && \forall t\in T \\
& x_{d,s,t} \in \mathbb{Z} && \forall d\in D, s\in S, t\in T \\
& x_{d,s,t} \ge 0 && \forall d\in D, s\in S, t\in T
\end{aligned}
$$

Bibliography

1. Sherali, H. D., and A. L. Soyster. 1983. "Preemptive and Nonpreemptive Multi-Objective Programming: Relationship and Counterexamples." *Journal of Optimization Theory and Applications* 39 (2): 173–86. https://doi.org/10.1007/bf00934527.
2. Yager, Ronald R. 1997. "On the Analytic Representation of the Leximin Ordering and Its Application to Flexible Constraint Propagation." *European Journal of Operational Research* 102 (1): 176–92. https://doi.org/10.1016/s0377-2217(96)00217-2.

Thank you!

Do you have any questions?