Math6120 - Nonlinear Optimisation
Coursework #2

Emma Tarmey - 2940 4045

May 2022

Contents

Chapter 1

The Problem

1.1 Formulation of Model and Assumptions Made

1.1.1 Deriving our Decision Variables

In this assignment, we seek to maximise the total sales value of final products to be made within an oil refinery. We seek to do so by means of optimising over the three stages of production: distillation, pooling and blending. This will in turn allow us to make optimal decisions about the quantity of product produced at the end of every stage. Consequently, we may make optimal decisions about the quantities of final product produced. These three stages of production separate four sets of product: crude oils, intermediate products, tank products and final products.

This gives us the following four sets of products to consider:

- Four Crude Oils
- Sixteen Intermediate Products
- Two Tank Products
- Two Final Products

This gives us 24 total products we may make decisions about within our formulation. Additionally, there are three aspects of any given product to consider:

- Quantity
- Sulphur Content
- Viscosity

As a result, we have a total $3 \times 24 = 72$ variables in our formulation. Of these, it makes sense to consider only the quantities of product as decision variables as the sulphur contents and viscosities of our products are determined from these quantities within the constraints below. Consequently, we have 24 decision variables in our formulation, given by the quantities of each of the 24 products present in the production process.

For notation purposes, to properly group all 72 variables by the relevant value, we write the following:

$$\begin{pmatrix} \mathrm{HS1}_{\mathrm{quantity}} \\ \mathrm{HS1}_{\mathrm{sulphur}} \\ \mathrm{HS1}_{\mathrm{viscosity}} \end{pmatrix} = \begin{pmatrix} \mathrm{HS1}_{\mathrm{q}} \\ \mathrm{HS1}_{\mathrm{s}} \\ \mathrm{HS1}_{\mathrm{v}} \end{pmatrix}$$

The variable names for each of the other 23 substances are similar. Using this notation to identify each of our 72 variables, we may organise these into three vectors of size (24×1): $\mathbf{q}$, $\mathbf{s}$ and $\mathbf{v}$. Since we are optimising over the quantities of substance we choose to produce, we consider $\mathbf{q}$ as our decision vector. The vector $\mathbf{q}$ takes the following form:

$$
\mathbf{q} = \begin{pmatrix} \mathrm{HS1_q} \\ \mathrm{HS2_q} \\ \mathrm{LS1_q} \\ \mathrm{LS2_q} \\ \\ \mathrm{HS1SR_q} \\ \mathrm{HS1CR_q} \\ \mathrm{HS1HGO_q} \\ \mathrm{HS1VGO_q} \\ \\ \mathrm{HS2SR_q} \\ \mathrm{HS2CR_q} \\ \mathrm{HS2HGO_q} \\ \mathrm{HS2VGO_q} \\ \\ \mathrm{LS1SR_q} \\ \mathrm{LS1CR_q} \\ \mathrm{LS1HGO_q} \\ \mathrm{LS1VGO_q} \\ \\ \mathrm{LS1SR_q} \\ \mathrm{LS2CR_q} \\ \mathrm{LS2HGO_q} \\ \mathrm{LS2VGO_q} \\ \\ \mathrm{T1_q} \\ \mathrm{T2_q} \\ \\ \mathrm{HSFO_q} \\ \mathrm{LSFO_q} \end{pmatrix} = \begin{pmatrix} \mathrm{CRUDE_q} \\ \\ \mathrm{INTER_q} \\ \\ \mathrm{T1_q} \\ \mathrm{T2_q} \\ \\ \mathrm{HSFO_q} \\ \mathrm{LSFO_q} \end{pmatrix} \in \mathbb{R}^{24}
$$

Where we have $\mathrm{CRUDE_q} \in \mathbb{R}^4$ and $\mathrm{INTER_q} \in \mathbb{R}^{16}$. The (24×1) vectors for all product sulphur contents $\mathbf{s}$ and viscosities $\mathbf{v}$ are similar. For modelling purposes, it is sensible to require that $\mathbf{q} \geq 0$, $\mathbf{s} \geq 0$ and $\mathbf{v} \geq 0$.

1.1.2 Deriving our Objective Function

We seek to maximise the final total sales value of our two final products: high-sulphur fuel oil (HSFO) and low-sulphur fuel oil (LSFO). The total sales value is variable and depends on the unit price and quantity of our two final products. This results in the following formula:

$$\begin{aligned} \mathit{totalvalue} &= \mathit{value}(\mathrm{HSFO}) + \mathit{value}(\mathrm{LSFO}) \\ &= \Big(\mathrm{HSFO_q} \times \mathit{unitprice}(\mathrm{HSFO_s})\Big) + \Big(\mathrm{LSFO_q} \times \mathit{unitprice}(\mathrm{LSFO_s})\Big) \end{aligned}$$

Additionally, the unit price of our oils is variable and depends upon the sulphur content of the oil produced. The unit price is calculated in the following way:

$$\mathit{unitprice}(\mathrm{HSFO_s}) = \mathit{base}_{\mathrm{HSFO}} \times \left(2 - \frac{\mathrm{HSFO_s}}{\mathrm{HSFO_{sulphurmax}}}\right)$$

$$\mathit{unitprice}(\mathrm{LSFO_s}) = \mathit{base}_{\mathrm{LSFO}} \times \left(2 - \frac{\mathrm{LSFO_s}}{\mathrm{LSFO_{sulphurmax}}}\right)$$

Where we have the following parameters given by the problem specification:

$$\begin{aligned} \mathit{base}_{\mathrm{HSFO}} &= \$100 \\ \mathit{base}_{\mathrm{LSFO}} &= \$150 \\ \mathrm{HSFO_{sulphurmax}} &= 1.5\% \\ \mathrm{LSFO_{sulphurmax}} &= 3.5\% \end{aligned}$$

We hence have the following objective function for our formulation:

$$\max_{\mathbf{q}} \quad \Big(\mathrm{HSFO_q} \times \mathit{unitprice}(\mathrm{HSFO_s})\Big) + \Big(\mathrm{LSFO_q} \times \mathit{unitprice}(\mathrm{LSFO_s})\Big)$$

The above also shows us that the fuel oils are at their most valuable when we seek to minimise their sulphur content, as the unit price of fuel oils with zero sulphur content will be double the unit price of fuel oils with maximum sulphur content. However, it will not be sufficient to simply set sulphur content to zero. This is due to constraints on the sulphur content and viscosity of our intermediate products below.

1.1.3 Deriving our Constraints

In this problem, we have constraints on each of the three properties that a given product has: quantity, sulphur content and viscosity. To begin, we have the following inequality constraints on the quantities of our 24 products:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \\ 1 \\ 0 \\ 0 \\ 0 \\ \\ 1 \\ 0 \\ 0 \\ 0 \\ \\ 1 \\ 0 \\ 0 \\ 0 \\ \\ 1 \\ 0 \\ 0 \\ 0 \\ \\ 0 \\ 0 \\ \\ 10 \\ 11 \end{pmatrix} \leq \begin{pmatrix} \mathrm{HS1_q} \\ \mathrm{HS2_q} \\ \mathrm{LS1_q} \\ \mathrm{LS2_q} \\ \\ \mathrm{HS1SR_q} \\ \mathrm{HS1CR_q} \\ \mathrm{HS1HGO_q} \\ \mathrm{HS1VGO_q} \\ \\ \mathrm{HS2SR_q} \\ \mathrm{HS2CR_q} \\ \mathrm{HS2HGO_q} \\ \mathrm{HS2VGO_q} \\ \\ \mathrm{LS1SR_q} \\ \mathrm{LS1CR_q} \\ \mathrm{LS1HGO_q} \\ \mathrm{LS1VGO_q} \\ \\ \mathrm{LS1SR_q} \\ \mathrm{LS2CR_q} \\ \mathrm{LS2HGO_q} \\ \mathrm{LS2VGO_q} \\ \\ \mathrm{T1_q} \\ \mathrm{T2_q} \\ \\ \mathrm{HSFO_q} \\ \mathrm{LSFO_q} \end{pmatrix} \leq \begin{pmatrix} \infty \\ \infty \\ \infty \\ \infty \\ \\ 3 \\ 3 \\ 3 \\ 3 \\ \\ 3 \\ 3 \\ 3 \\ 3 \\ \\ 3 \\ 3 \\ 3 \\ 3 \\ \\ 3 \\ 3 \\ 3 \\ 3 \\ \\ 15 \\ 15 \\ \\ 11 \\ 17 \end{pmatrix}$$

Additionally, we have the following inequality constraints on the sulphur contents and viscosities of our final products:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} \mathrm{HSFO_s} \\ \mathrm{LSFO_s} \end{pmatrix} \leq \begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix} \qquad \begin{pmatrix} 32.0 \\ 30.0 \end{pmatrix} \leq \begin{pmatrix} \mathrm{HSFO_v} \\ \mathrm{LSFO_v} \end{pmatrix} \leq \begin{pmatrix} 40.0 \\ 34.0 \end{pmatrix}$$

We also have equality constraints on the sulphur contents and viscosities of our intermediate products:

$$\begin{pmatrix} \text{HS1SR}_\text{s} \\ \text{HS1CR}_\text{s} \\ \text{HS1HGO}_\text{s} \\ \text{HS1VGO}_\text{s} \\ \\ \text{HS2SR}_\text{s} \\ \text{HS2CR}_\text{s} \\ \text{HS2HGO}_\text{s} \\ \text{HS2VGO}_\text{s} \\ \\ \text{LS1SR}_\text{s} \\ \text{LS1CR}_\text{s} \\ \text{LS1HGO}_\text{s} \\ \text{LS1VGO}_\text{s} \\ \\ \text{LS1SR}_\text{s} \\ \text{LS2CR}_\text{s} \\ \text{LS2HGO}_\text{s} \\ \text{LS2VGO}_\text{s} \\ \\ \end{pmatrix} = \begin{pmatrix} 5.84 \\ 5.40 \\ 0.24 \\ 2.01 \\ \\ 5.85 \\ 5.38 \\ 0.26 \\ 2.04 \\ \\ 0.64 \\ 0.57 \\ 0.02 \\ 0.14 \\ \\ 0.93 \\ 0.85 \\ 0.03 \\ 0.26 \\ \\ \end{pmatrix} \qquad \begin{pmatrix} \text{HS1SR}_\text{q} \\ \text{HS1CR}_\text{q} \\ \text{HS1HGO}_\text{q} \\ \text{HS1VGO}_\text{q} \\ \\ \text{HS2SR}_\text{q} \\ \text{HS2CR}_\text{q} \\ \text{HS2HGO}_\text{q} \\ \text{HS2VGO}_\text{q} \\ \\ \text{LS1SR}_\text{q} \\ \text{LS1CR}_\text{q} \\ \text{LS1HGO}_\text{q} \\ \text{LS1VGO}_\text{q} \\ \\ \text{LS1SR}_\text{q} \\ \text{LS2CR}_\text{q} \\ \text{LS2HGO}_\text{q} \\ \text{LS2VGO}_\text{q} \\ \\ \end{pmatrix} = \begin{pmatrix} 43.7 \\ 36.8 \\ 12.8 \\ 15.4 \\ \\ 47.3 \\ 39.2 \\ 13.1 \\ 15.9 \\ \\ 39.9 \\ 38.2 \\ 13.5 \\ 16.3 \\ \\ 38.1 \\ 34.1 \\ 13.2 \\ 15.5 \\ \\ \end{pmatrix}$$

For simplicity, we may denote some of these constraints in vector form. Rewriting the above gives us the following:

$$\begin{aligned} LB_q \leq \mathbf{q} &\leq UB_q \\ 0 \leq \text{HSFO}_s &\leq 3.5 \\ 0 \leq \text{LSFO}_s &\leq 1.5 \\ 32.0 \leq \text{HSFO}_v &\leq 40.0 \\ 30.0 \leq \text{LSFO}_v &\leq 34.0 \\ \text{INTER}_s &= V_s \\ \text{INTER}_v &= V_v \end{aligned}$$

To model the production process itself, we have the following constraints controlling the distillation of our 4 crude oils into our 16 intermediate products:

$$\begin{array}{lcllll}
\mathrm{HS1}_q & \geq & \mathrm{HS1SR}_q & + \mathrm{HS1CR}_q & + \mathrm{HS1HGO}_q & + \mathrm{HS1VGO}_q \\
\mathrm{HS2}_q & \geq & \mathrm{HS2SR}_q & + \mathrm{HS2CR}_q & + \mathrm{HS2HGO}_q & + \mathrm{HS2VGO}_q \\
\mathrm{LS1}_q & \geq & \mathrm{LS1SR}_q & + \mathrm{LS1CR}_q & + \mathrm{LS1HGO}_q & + \mathrm{LS1VGO}_q \\
\mathrm{LS2}_q & \geq & \mathrm{LS2SR}_q & + \mathrm{LS2CR}_q & + \mathrm{LS2HGO}_q & + \mathrm{LS2VGO}_q
\end{array}$$

This will in-turn ensure that the total quantity of intermediates never exceeds the quantity of the given crude which produced them. An inequality constraint is used as opposed to an equality constraint to prevent starting points from being infeasible too easily. This is discussed further in (2.3).

Additionally, we have the following constraints controlling the pooling of our 16 intermediate products into our 2 pooling tanks:

$$\mathrm{T1}_q = high\_sulphur\_total\_quantity$$

$$\mathrm{T2}_q = low\_sulphur\_total\_quantity$$

$$\mathrm{T1}_s = \frac{high\_sulphur\_weighted\_sulphur}{high\_sulphur\_total\_quantity}$$

$$\mathrm{T2}_s = \frac{low\_sulphur\_weighted\_sulphur}{low\_sulphur\_total\_quantity}$$

$$\mathrm{T1}_v = \frac{high\_sulphur\_weighted\_viscosity}{high\_sulphur\_total\_quantity}$$

$$\mathrm{T2}_v = \frac{low\_sulphur\_weighted\_viscosity}{low\_sulphur\_total\_quantity}$$

Where, for ease of notation, we define the following values:

$$
\begin{aligned}
\textit{high\_sulphur\_weighted\_sulphur} = {} & (\mathrm{HS1SR}_q \times \mathrm{HS1SR}_s) + (\mathrm{HS1CR}_q \times \mathrm{HS1CR}_s) + \\
& (\mathrm{HS1HGO}_q \times \mathrm{HS1HGO}_s) + (\mathrm{HS1VGO}_q \times \mathrm{HS1VGO}_s) + \\
& (\mathrm{HS2SR}_q \times \mathrm{HS2SR}_s) + (\mathrm{HS2CR}_q \times \mathrm{HS2CR}_s) + \\
& (\mathrm{HS2HGO}_q \times \mathrm{HS2HGO}_s) + (\mathrm{HS2VGO}_q \times \mathrm{HS2VGO}_s)
\end{aligned}
$$

$$
\begin{aligned}
\textit{low\_sulphur\_weighted\_sulphur} = {} & (\mathrm{LS1SR}_q \times \mathrm{LS1SR}_s) + (\mathrm{LS1CR}_q \times \mathrm{LS1CR}_s) + \\
& (\mathrm{LS1HGO}_q \times \mathrm{LS1HGO}_s) + (\mathrm{LS1VGO}_q \times \mathrm{LS1VGO}_s) + \\
& (\mathrm{LS2SR}_q \times \mathrm{LS2SR}_s) + (\mathrm{LS2CR}_q \times \mathrm{LS2CR}_s) + \\
& (\mathrm{LS2HGO}_q \times \mathrm{LS2HGO}_s) + (\mathrm{LS2VGO}_q \times \mathrm{LS2VGO}_s)
\end{aligned}
$$

$$
\begin{aligned}
\textit{high\_sulphur\_weighted\_viscosity} = {} & (\mathrm{HS1SR}_q \times \mathrm{HS1SR}_v) + (\mathrm{HS1CR}_q \times \mathrm{HS1CR}_v) + \\
& (\mathrm{HS1HGO}_q \times \mathrm{HS1HGO}_v) + (\mathrm{HS1VGO}_q \times \mathrm{HS1VGO}_v) + \\
& (\mathrm{HS2SR}_q \times \mathrm{HS2SR}_v) + (\mathrm{HS2CR}_q \times \mathrm{HS2CR}_v) + \\
& (\mathrm{HS2HGO}_q \times \mathrm{HS2HGO}_v) + (\mathrm{HS2VGO}_q \times \mathrm{HS2VGO}_v)
\end{aligned}
$$

$$
\begin{aligned}
\textit{low\_sulphur\_weighted\_viscosity} = {} & (\mathrm{LS1SR}_q \times \mathrm{LS1SR}_v) + (\mathrm{LS1CR}_q \times \mathrm{LS1CR}_v) + \\
& (\mathrm{LS1HGO}_q \times \mathrm{LS1HGO}_v) + (\mathrm{LS1VGO}_q \times \mathrm{LS1VGO}_v) + \\
& (\mathrm{LS2SR}_q \times \mathrm{LS2SR}_v) + (\mathrm{LS2CR}_q \times \mathrm{LS2CR}_v) + \\
& (\mathrm{LS2HGO}_q \times \mathrm{LS2HGO}_v) + (\mathrm{LS2VGO}_q \times \mathrm{LS2VGO}_v)
\end{aligned}
$$

$$
\begin{aligned}
\textit{high\_sulphur\_total\_quantity} = {} & \mathrm{HS1SR}_q + \mathrm{HS1CR}_q + \mathrm{HS1HGO}_q + \mathrm{HS1VGO}_q + \dots \\
& \dots \quad \mathrm{HS2SR}_q + \mathrm{HS2CR}_q + \mathrm{HS2HGO}_q + \mathrm{HS2VGO}_q
\end{aligned}
$$

$$
\begin{aligned}
\textit{low\_sulphur\_total\_quantity} = {} & \mathrm{LS1SR}_q + \mathrm{LS1CR}_q + \mathrm{LS1HGO}_q + \mathrm{LS1VGO}_q + \\
& \dots \quad \mathrm{LS2SR}_q + \mathrm{LS2CR}_q + \mathrm{LS2HGO}_q + \mathrm{LS2VGO}_q
\end{aligned}
$$

These pooling constraints ensure that the sulphur contents and viscosity of the 2 tank products are calculated properly. The formula being that we take the weighted mean of the sulphur contents and viscosities across each of the 8 relevant intermediate products. This mean is weighted by the chosen quantity of intermediate, reflecting the nature of the pooling process.

Finally, we have the following constraints controlling the blending of our 2 pooled substances into our 2 final products:

$$\begin{aligned}
\text{HSFO}_q &= (t \times \text{T1}_q) + ((1-t) \times \text{T2}_q) \\
\text{HSFO}_s &= (t \times \text{T1}_s) + ((1-t) \times \text{T2}_s) \\
\text{HSFO}_v &= (t \times \text{T1}_v) + ((1-t) \times \text{T2}_v) \\
\text{LSFO}_q &= ((1-t) \times \text{T1}_q) + (t \times \text{T2}_q) \\
\text{LSFO}_s &= ((1-t) \times \text{T1}_s) + (t \times \text{T2}_s) \\
\text{LSFO}_v &= ((1-t) \times \text{T1}_v) + (t \times \text{T2}_v)
\end{aligned}$$

Where $t \in \mathbb{R}$ is an additional variable allowing for the linear interpolation of our tank substances into our final products. The idea being that we allow the fuel oils to blend an optimal ratio of the tank products, but also ensure that none is wasted in the process. By varying t between 0 and 1, the solver will be able to determine an optimal blend. t hence becomes an additional decision variable, bringing our number of decision variables up from 24 to 25. This is because we seek to determine the optimal quantities of every substance to produce, but also the optimal blend of sulphur content and viscosity across the final products. To reflect this, we propose an updated decision vector $\mathbf{q}'$, containing both our new variable t and the contents of our prior decision vector q:

$$\mathbf{q}' = \begin{pmatrix} t \\ \mathbf{q} \end{pmatrix} \in \mathbb{R}^{25}$$

$$0 \leq t \leq 1$$

1.1.4 Full Statement of Model

$$\max_{\mathbf{q}'} \quad \Big(\mathrm{HSFO_q} \times \mathit{unitprice}(\mathrm{HSFO_s})\Big) + \Big(\mathrm{LSFO_q} \times \mathit{unitprice}(\mathrm{LSFO_s})\Big)$$

subject to (quantity bounds)

$$LB_q \leq \mathbf{q} \leq UB_q$$

(sulphur content bounds)

$$\mathrm{INTER}_s = V_s$$
$$0 \leq \mathrm{HSFO}_s \leq 3.5$$
$$0 \leq \mathrm{LSFO}_s \leq 1.5$$

(viscosity bounds)

$$\mathrm{INTER}_v = V_v$$
$$32.0 \leq \mathrm{HSFO}_v \leq 40.0$$
$$30.0 \leq \mathrm{LSFO}_v \leq 34.0$$

(distillation constraints)

$$\mathrm{HS1}_q \geq \mathrm{HS1SR}_q + \mathrm{HS1CR}_q + \mathrm{HS1HGO}_q + \mathrm{HS1VGO}_q$$
$$\mathrm{HS2}_q \geq \mathrm{HS2SR}_q + \mathrm{HS2CR}_q + \mathrm{HS2HGO}_q + \mathrm{HS2VGO}_q$$
$$\mathrm{LS1}_q \geq \mathrm{LS1SR}_q + \mathrm{LS1CR}_q + \mathrm{LS1HGO}_q + \mathrm{LS1VGO}_q$$
$$\mathrm{LS2}_q \geq \mathrm{LS2SR}_q + \mathrm{LS2CR}_q + \mathrm{LS2HGO}_q + \mathrm{LS2VGO}_q$$

(pooling constraints)

$$\mathrm{T1}_q = \mathit{high\_sulphur\_total\_quantity}$$
$$\mathrm{T2}_q = \mathit{low\_sulphur\_total\_quantity}$$

$$\mathrm{T1}_s = \frac{\mathit{high\_sulphur\_weighted\_sulphur}}{\mathit{high\_sulphur\_total\_quantity}}$$

$$\mathrm{T2}_s = \frac{\mathit{low\_sulphur\_weighted\_sulphur}}{\mathit{low\_sulphur\_total\_quantity}}$$

$$\mathrm{T1}_v = \frac{\mathit{high\_sulphur\_weighted\_viscosity}}{\mathit{high\_sulphur\_total\_quantity}}$$

$$\mathrm{T2}_v = \frac{\mathit{low\_sulphur\_weighted\_viscosity}}{\mathit{low\_sulphur\_total\_quantity}}$$

(blending constraints)

$$\mathrm{HSFO}_q = (t \times \mathrm{T1}_q) + ((1-t) \times \mathrm{T2}_q)$$
$$\mathrm{HSFO}_s = (t \times \mathrm{T1}_s) + ((1-t) \times \mathrm{T2}_s)$$
$$\mathrm{HSFO}_v = (t \times \mathrm{T1}_v) + ((1-t) \times \mathrm{T2}_v)$$
$$\mathrm{LSFO}_q = ((1-t) \times \mathrm{T1}_q) + (t \times \mathrm{T2}_q)$$
$$\mathrm{LSFO}_s = ((1-t) \times \mathrm{T1}_s) + (t \times \mathrm{T2}_s)$$
$$\mathrm{LSFO}_v = ((1-t) \times \mathrm{T1}_v) + (t \times \mathrm{T2}_v)$$
$$0 \leq t \leq 1$$

$$\mathbf{q} \geq 0, \quad \mathbf{s} \geq 0, \quad \mathbf{v} \geq 0,$$

1.2 Considering Convexity

A nonlinear programming model is convex if it satisfies two conditions:

- The objective function is convex.
- The solution space is a convex set.

A given function $f(x)$ is considered convex if is satisfies the following condition:

$$f''(x) > 0 \text{ everywhere}$$

Conversely, a given function $f(x)$ is considered non-convex if the following holds:

$$f''(x) \leq 0 \text{ at any given point x}$$

To begin, we look at the solution space as defined by the constraints. Clearly, most of the constraints are linear bounds or linear equality constraints on the decision variables. Though, looking at the pooling constraints on sulphur content and viscosity defined above:

$$\text{T1}_s = \frac{high\_sulphur\_weighted\_sulphur}{high\_sulphur\_total\_quantity}$$

$$\text{T2}_s = \frac{low\_sulphur\_weighted\_sulphur}{low\_sulphur\_total\_quantity}$$

$$\text{T1}_v = \frac{high\_sulphur\_weighted\_viscosity}{high\_sulphur\_total\_quantity}$$

$$\text{T2}_v = \frac{low\_sulphur\_weighted\_viscosity}{low\_sulphur\_total\_quantity}$$

In particular, the four values in the numerators of the above constraints all directly involve multiplying quantity variables with sulphur variables or viscosity variables as per their definitions. However, noting that all terms in the definitions of these values have positive coefficients, we hence fail to find any point at which $f''(x) \leq 0$ within these constraints. These pooling constraints are the only non-linear constraints within our formulation given our 25 decision variables. Hence we may claim that the solution space defined by our constraints is a convex set.

Looking at the objective function in more detail gives us the following:

$$
\begin{aligned}
f(q) &= \Big(\mathrm{HSFO_q} \times \mathit{unitprice}(\mathrm{HSFO_s})\Big) + \Big(\mathrm{LSFO_q} \times \mathit{unitprice}(\mathrm{LSFO_s})\Big) \\
&= \left(\mathrm{HSFO_q} \times \mathit{base}_{\mathrm{HSFO}} \times \left(2 - \frac{\mathrm{HSFO_s}}{\mathrm{HSFO_{sulphurmax}}}\right)\right) + \left(\mathrm{LSFO_q} \times \mathit{base}_{\mathrm{LSFO}} \times \left(2 - \frac{\mathrm{LSFO_s}}{\mathrm{LSFO_{sulphurmax}}}\right)\right) \\
&= \Big(2 \times \mathrm{HSFO}_q \times \mathit{base}_{\mathrm{HSFO}}\Big) + \left(-\frac{\mathit{base}_{\mathrm{HSFO}}}{\mathrm{HSFO_{sulphurmax}}} \times (\mathrm{HSFO}_q \times \mathrm{HSFO}_s)\right) + \dots \\
&\quad \dots \Big(2 \times \mathrm{LSFO}_q \times \mathit{base}_{\mathrm{LSFO}}\Big) + \left(-\frac{\mathit{base}_{\mathrm{LSFO}}}{\mathrm{LSFO_{sulphurmax}}} \times (\mathrm{LSFO}_q \times \mathrm{HSFO}_s)\right)
\end{aligned}
$$

We know from the specification that:

$$
\begin{gathered}
\mathit{base}_{\mathrm{HSFO}} > 0 \\
\mathit{base}_{\mathrm{LSFO}} > 0 \\
\mathrm{HSFO_{sulphurmax}} > 0 \\
\mathrm{LSFO_{sulphurmax}} > 0
\end{gathered}
$$

Hence, in the above function, we clearly see negative coefficients on two separate terms each multiplying two decision variables. Hence, $f''(q) \leq 0$ here when differentiating with respect to these decision variables. As a result, the objective function is not a convex function. Consequently, our non-linear program formulation is non-convex.

Chapter 2

Solution

2.1 Solution Method

To solve this non-linear optimization problem, we modelled the formulation specified in (1.1.4) using AMPL. Then, we produced optimal solutions using several solvers. The results of this process are below. The corresponding AMPL code and raw command-line output are provided in (3.1) and (3.2).

2.2 Optimal Production Plan

Running our AMPL model with several appropriate solvers gives us the following optimal solution:

- Total Sales Value = $ 4,066.25
- HSFO Produced: 14.72 ktons, with sulphur content 3.5%
- LSFO Produced: 11.00 ktons, with sulphur content 0.6417%

Looking at the solutions produced by each solver individually gives us the following table of solution:

Solver	Total Sales Value ($)	HSFO Quantity	HSFO Sulphur	LSFO Quantity	LSFO Sulphur
ConOpt	4066.25	14.7215	3.5	11	0.641726
Knitro	4066.25	14.7215	3.5	11	0.641726
Minos	4066.25	14.7215	3.5	11	0.641726
Snopt	4066.25	14.7215	3.5	11	0.641726

Table 2.1: Table of Optimal Solutions

Clearly, the above table shows a very strong consensus of optimal solution across the solvers. Though, looking closely at the output in appendices 3.2.1-3.2.4, we can see a small change in intermediate product quantities in the Knitro solution when compared to solutions from other solvers. Though, this is certainly the result of numerical error ($1.19719e - 09 \approx 0$).

Looking at the above solutions, it seems interesting that the HSFO sulphur was allowed to be maximal, when fuel oil is more valuable when sulphur content is minimised as per (1.1.2). Though, it seems the solvers are seeking to maximise the the value of the LSFO fuel oil produced, which has a higher base price, by means of keeping the LSFO sulphur content as low as possible. The solver attempts to keep the LSFO sulphur content minimal, being constrained by viscosity when doing so. As a result, the less valuable HSFO fuel oil has the composition we see for the purpose of feasibility.

2.3 Considering Different Starting Points

For different starting points of our blending value t, AMPL always produces the same optimal solution and will always set $t = 0.934914$. This suggests that we have found a stable maximum for our problem with respect to the blending of tank products.

Though, setting different starting quantities for the crude oils made the problem much more temperamental. As per the definition of UB_q in (1.1.3), the quantities for our crude oils are unbounded above. However, the quantities of all other products in the formulation are bounded above. As a result, it proved rather easy to accidentally suggest an infeasible starting point and trouble the solvers. Though, this was solved with an amendment to the distillation constraints (1.1.4), where we change our equality constraints on quantity of intermediates to inequality constraints. This in turn made it possible to allow waste of some crude oils if using the entire quantity of crude would force a violation of an upper bound later in production.

2.4 Effect of Removing All Pool Tanks

We can simulate the effect of removing all pool tanks within our solution by disallowing any blending within our constraints. In particular, if we mandate $t = 1$, then the HSFO fuel oil will be composed entirely of the high-sulphur intermediates and the LSFO fuel oil will be composed entirely of the low-sulphur intermediates. In practice, this will be equivalent to the eight high-sulphur intermediate streams pooling directly into the HSFO final product and the eight low-sulphur intermediate streams pooling directly into the LSFO final product. Adding the equality constraint $t = 1$ to our formulation produces the following optimal solution:

- Total Sales Value = \$ 3,610.85
- HSFO Produced: 11 ktons, with sulphur content 3.5%
- LSFO Produced: 10 ktons, with sulphur content 0.4891%

This results in a reduction of \$455.40 in the objective function value. This is explained by the reduced production of both fuel oils. Comparing the optimal solutions, the sulphur contents are identical for HSFO, and reduced for LSFO. This should be beneficial, as fuel oils are more valuable when having lesser sulphur content. However, it seems clear that the viscosity constraints prevent more fuel oils from being produced here. This is corroborated by the optimal solutions produced by the model with two pool tanks:

Solver	HSFO Viscosity (%)	LSFO Viscosity (%)
ConOpt	32	30
Knitro	32	30
Minos	32	30
Snopt	32	30

Table 2.2: Table of Optimal Final Product Viscosities

Clearly, the optimal solutions are always constrained by viscosity, evidenced by the viscosity value always being set to their lower bounds in an optimal solution. Based on this, we conclude that the blending stage of production is primarily useful for controlling the viscosity of the final products, as this allows for a much higher quantity final products to be produced from the intermediate streams. Whereas, without blending, we achieve similarly optimal sulphur content levels but at the cost of viscosity constraints preventing higher quantities of final product from being produced.

2.5 Summary of Recommended Policies

In conclusion, looking in particular at the solution produced by ConOpt, we recommend buying the following quantities of crude oils:

Crude	Quantity (kton)
HS1	6.0
HS2	9.0
LS1	8.91076
LS2	1.81077

Table 2.3: Table of Optimal Crude Oil Quantities

Then, during distillation, we recommend producing the following quantities of the sixteen intermediate products:

Intermediate	Quantity (kton)
HS1 SR	3.0
HS1 CR	0.581033
HS1 HGO	2.41897
HS1 VGO	0.0
HS2 SR	3.0
HS2 CR	3.0
HS2 HGO	3.0
HS2 VGO	0.0
LS1 SR	2.91076
LS1 CR	3.0
LS1 HGO	3.0
LS1 VGO	0.0
LS2 SR	1.0
LS2 CR	0.0
LS2 HGO	0.810774
LS2 VGO	0.0

Table 2.4: Table of Optimal Intermediate Product Quantities

Then, during pooling, we recommend pooling the above quantities of intermediate product to produce the following tank products:

Product	Quantity (kton)	Sulphur (%)	Viscosity (%)
Tank1	15.0	3.71388	32.1497
Tank2	10.7215	0.42785	29.8503

Table 2.5: Table of Optimal Tank Product Properties

Finally, we recommend using the optimal blend of the above tank products specified by the blending variable $t = 0.934914$ to produce the following optimal final products:

Product	Quantity (kton)	Sulphur (%)	Viscosity (%)
HSFO	14.7215	3.5	32.0
LSFO	11.0	0.641726	30.0

Table 2.6: Table of Optimal Final Products

This will in turn produce the maximum possible total sales value of our final products: $ 4,066.25.

2.6 Suggestions for Further Investigation

2.6.1 Re-modelling Distillation

Going further with the detail of the model itself, it may be worth investigating the distillation process in more detail. In the above, we assume that we may always obtain the optimal combination of intermediate products from a given quantity of crude oils. If in reality it is not always feasible to obtain an optimal spread of the four possible intermediates from a crude oil, it would be sensible to model this limitation within the constraints. This would require the distillation constraints of our model to be updated, only permitting a given distribution of intermediates to be produced from a given crude through upper and lower limits on quantities.

2.6.2 Modelling Loss of Yield

Moreover, in the above model, we also assume that there will be zero loss on the yield at any point in the production process. This is separate from allowing some quantity of crude oils to be unused in production. In reality, this assumption may prove poor as one can typically expect loss of product quantity as product is processed. Consequently, it would be worth modelling a loss of quantity at each stage of production (distillation, pooling and blending). This could be implemented with the use of deterioration constants applied to the quantities of obtained products at the end of each stage of production:

$$\begin{aligned}
\mathrm{HS1SR}'_q &= d_{\mathrm{dist}} \times \mathrm{HS1SR}_q \\
&\ldots \\
\mathrm{LS2VGO}'_q &= d_{\mathrm{dist}} \times \mathrm{LS2VGO}_q \\
\\
\mathrm{T1}'_q &= d_{\mathrm{pool}} \times \mathrm{T1}_q \\
\mathrm{T2}'_q &= d_{\mathrm{pool}} \times \mathrm{T2}_q \\
\\
\mathrm{HSFO}'_q &= d_{\mathrm{blend}} \times \mathrm{HSFO}_q \\
\mathrm{HSFO}'_q &= d_{\mathrm{blend}} \times \mathrm{HSFO}_q \\
\\
0 \leq\ & d_{\mathrm{dist}} \leq 1 \\
0 \leq\ & d_{\mathrm{pool}} \leq 1 \\
0 \leq\ & d_{\mathrm{blend}} \leq 1
\end{aligned}$$

Where we have d_{dist}, d_{pool} and d_{blend} as constants determining as a proportion how much substance we yield from a given production process. A value of $d = 1$ giving a perfect yield and a value of $d < 1$ implying some waste of loss in production.

2.6.3 Modelling Cost of Crude Oils

Finally, it may also be worth investigating the costs of our crude oils to be bought. We could then considering maximising profits as opposed to our current model of maximising total sales value (revenue). This would require a small modification to the objective function of our problem, which simply subtracts the total costs of all crude oils bought from the total sales value. The remainder of the model would be the same, so this will have no impact on feasibility. Though, the optimality may be very different, particularly if lower sulphur or lower viscosity crude oils were more expensive to buy prior to production. A resulting new objective function could be:

$$\begin{aligned}\max_{\mathbf{q}'} \quad & \Big(\mathrm{HSFO_q} \times \mathit{unitprice}(\mathrm{HSFO_s})\Big) + \Big(\mathrm{LSFO_q} \times \mathit{unitprice}(\mathrm{LSFO_s})\Big) \\ & - \Big(\mathrm{HS1}_q \times \mathit{cost}_{\mathrm{HS1}}\Big) - \Big(\mathrm{HS2}_q \times \mathit{cost}_{\mathrm{HS2}}\Big) \\ & - \Big(\mathrm{LS1}_q \times \mathit{cost}_{\mathrm{LS1}}\Big) - \Big(\mathrm{LS2}_q \times \mathit{cost}_{\mathrm{LS2}}\Big)\end{aligned}$$

Chapter 3

Appendices

The below code may be run in the AMPL environment by executing the command “include cw2_verbose.run;” using AMPL command line.

3.1 AMPL Code

3.1.1 AMPL Model Code

```
## Math6120 - Nonlinear Optimisation
## Coursework 2 - AMPL Model
## Emma Tarmey, 2940 4045


## This file specifies the mathematical model used to solve our problem


## sets

# PROPERTIES acts as a structure for organising the three variables
# associated with each of our 22 substances
# 1 = quantity / amount (kton)
# 2 = sulphur content (%)
# 3 = viscosity (V50%)
set PROPERTIES := {'quantity', 'sulphur', 'viscosity'};
set QUANTITY   := {'quantity'};


## parameters
param lsfo_base_price;
param hsfo_base_price;
param lsfo_sulphur_upper_bound;
param hsfo_sulphur_upper_bound;
```

```
## variables

# starting product variables
var hs1 {p in QUANTITY} >= 0; # high-sulphur crude oil
var hs2 {p in QUANTITY} >= 0; # high-sulphur crude oil
var ls1 {p in QUANTITY} >= 0; # low-sulphur crude oil
var ls2 {p in QUANTITY} >= 0; # low-sulphur crude oil

# intermediate product variables
var hs1_sr  {p in PROPERTIES} >= 0; # short residue
var hs1_cr  {p in PROPERTIES} >= 0; # cracked residue
var hs1_hgo {p in PROPERTIES} >= 0; # heavy gas oil
var hs1_vgo {p in PROPERTIES} >= 0; # visbroken gas oil

var hs2_sr  {p in PROPERTIES} >= 0;
var hs2_cr  {p in PROPERTIES} >= 0;
var hs2_hgo {p in PROPERTIES} >= 0;
var hs2_vgo {p in PROPERTIES} >= 0;

var ls1_sr  {p in PROPERTIES} >= 0;
var ls1_cr  {p in PROPERTIES} >= 0;
var ls1_hgo {p in PROPERTIES} >= 0;
var ls1_vgo {p in PROPERTIES} >= 0;

var ls2_sr  {p in PROPERTIES} >= 0;
var ls2_cr  {p in PROPERTIES} >= 0;
var ls2_hgo {p in PROPERTIES} >= 0;
var ls2_vgo {p in PROPERTIES} >= 0;

# tank product variables
var tank_one {p in PROPERTIES} >= 0;
var tank_two {p in PROPERTIES} >= 0;

# final product variables
var hsfo {p in PROPERTIES} >= 0; # high-sulphur fuel oil
var lsfo {p in PROPERTIES} >= 0; # low-sulphur fuel oil

# linear interpolation variable (used for blending)
var t >= 0;

# price calculation variables
var lsfo_actual_price =
    lsfo_base_price * (2 - (lsfo['sulphur'] / lsfo_sulphur_upper_bound));
var hsfo_actual_price =
```

```
    hsfo_base_price * (2 - (hsfo['sulphur'] / hsfo_sulphur_upper_bound));

# blending variables
var high_sulphur_intermediate_weighted_sulphur = (
        (hs1_sr['quantity']*hs1_sr['sulphur'])   +
        (hs1_cr['quantity']*hs1_cr['sulphur'])   +
        (hs1_hgo['quantity']*hs1_hgo['sulphur']) +
        (hs1_vgo['quantity']*hs1_vgo['sulphur']) +
        (hs2_sr['quantity']*hs2_sr['sulphur'])   +
        (hs2_cr['quantity']*hs2_cr['sulphur'])   +
        (hs2_hgo['quantity']*hs2_hgo['sulphur']) +
        (hs2_vgo['quantity']*hs2_vgo['sulphur']));

var low_sulphur_intermediate_weighted_sulphur = (
        (ls1_sr['quantity']*ls1_sr['sulphur'])   +
        (ls1_cr['quantity']*ls1_cr['sulphur'])   +
        (ls1_hgo['quantity']*ls1_hgo['sulphur']) +
        (ls1_vgo['quantity']*ls1_vgo['sulphur']) +
        (ls2_sr['quantity']*ls2_sr['sulphur'])   +
        (ls2_cr['quantity']*ls2_cr['sulphur'])   +
        (ls2_hgo['quantity']*ls2_hgo['sulphur']) +
        (ls2_vgo['quantity']*ls2_vgo['sulphur']));

var high_sulphur_intermediate_weighted_viscosity = (
        (hs1_sr['quantity']*hs1_sr['viscosity'])   +
        (hs1_cr['quantity']*hs1_cr['viscosity'])   +
        (hs1_hgo['quantity']*hs1_hgo['viscosity']) +
        (hs1_vgo['quantity']*hs1_vgo['viscosity']) +
        (hs2_sr['quantity']*hs2_sr['viscosity'])   +
        (hs2_cr['quantity']*hs2_cr['viscosity'])   +
        (hs2_hgo['quantity']*hs2_hgo['viscosity']) +
        (hs2_vgo['quantity']*hs2_vgo['viscosity']));

var low_sulphur_intermediate_weighted_viscosity = (
        (ls1_sr['quantity']*ls1_sr['viscosity'])   +
        (ls1_cr['quantity']*ls1_cr['viscosity'])   +
        (ls1_hgo['quantity']*ls1_hgo['viscosity']) +
        (ls1_vgo['quantity']*ls1_vgo['viscosity']) +
        (ls2_sr['quantity']*ls2_sr['viscosity'])   +
        (ls2_cr['quantity']*ls2_cr['viscosity'])   +
        (ls2_hgo['quantity']*ls2_hgo['viscosity']) +
        (ls2_vgo['quantity']*ls2_vgo['viscosity']));


var high_sulphur_total_quantity = (hs1_sr['quantity']  + hs1_cr['quantity'] +
                                   hs1_hgo['quantity'] + hs1_vgo['quantity'] +
```

```
                                   hs2_sr['quantity']  + hs2_cr['quantity'] +
                                   hs2_hgo['quantity'] + hs2_vgo['quantity']);

var low_sulphur_total_quantity  = (ls1_sr['quantity']  + ls1_cr['quantity'] +
                                   ls1_hgo['quantity'] + ls1_vgo['quantity'] +
                                   ls2_sr['quantity']  + ls2_cr['quantity'] +
                                   ls2_hgo['quantity'] + ls2_vgo['quantity']);


## objective function
maximize total_sales_volume: ( (lsfo['quantity'] * lsfo_actual_price) +
                               (hsfo['quantity'] * hsfo_actual_price) ) ;


## constraints

# linear interpolation constraint
subject to linear_interpolation:
        t <= 1;

# this constraint will simulate the removal of all pool tanks from production
# to include this constraint, un-comment the below 2 lines of code
#subject to remove_tanks:
#        t = 1;


## The production process we model takes 3 stages: distillation, pooling and blending
## The following 3 sets of constraints determine how each of these stages occur


# STAGE 1 - distillation constraints
subject to hs1_distillation:
        hs1['quantity'] >= hs1_sr['quantity']  + hs1_cr['quantity'] +
                           hs1_hgo['quantity'] + hs1_vgo['quantity'];
subject to hs2_distillation:
        hs2['quantity'] >= hs2_sr['quantity']  + hs2_cr['quantity'] +
                           hs2_hgo['quantity'] + hs2_vgo['quantity'];
subject to ls1_distillation:
        ls1['quantity'] >= ls1_sr['quantity']  + ls1_cr['quantity'] +
                           ls1_hgo['quantity'] + ls1_vgo['quantity'];
subject to ls2_distillation:
        ls2['quantity'] >= ls2_sr['quantity']  + ls2_cr['quantity'] +
                           ls2_hgo['quantity'] + ls2_vgo['quantity'];


# STAGE 2 - pooling constraints
```

```
# quantity of each tank's pool
subject to pooling_tank_one_quantity:
        tank_one['quantity'] = (hs1_sr['quantity']  + hs1_cr['quantity'] +
                                hs1_hgo['quantity'] + hs1_vgo['quantity'] +
                                hs2_sr['quantity']  + hs2_cr['quantity'] +
                                hs2_hgo['quantity'] + hs2_vgo['quantity']);

subject to pooling_tank_two_quantity:
        tank_two['quantity'] = (ls1_sr['quantity']  + ls1_cr['quantity'] +
                                ls1_hgo['quantity'] + ls1_vgo['quantity'] +
                                ls2_sr['quantity']  + ls2_cr['quantity'] +
                                ls2_hgo['quantity'] + ls2_vgo['quantity']);

# sulphur content of each tank's pool
subject to pooling_tank_one_sulphur:
        tank_one['sulphur'] =
        (high_sulphur_intermediate_weighted_sulphur / high_sulphur_total_quantity);
subject to pooling_tank_two_sulphur:
        tank_two['sulphur'] =
        (low_sulphur_intermediate_weighted_sulphur  / low_sulphur_total_quantity);

# viscosity of each tank's pool
subject to pooling_tank_one_viscosity:
        tank_one['viscosity'] =
        (high_sulphur_intermediate_weighted_viscosity / high_sulphur_total_quantity);
subject to pooling_tank_two_viscosity:
        tank_two['viscosity'] =
        (low_sulphur_intermediate_weighted_viscosity  / low_sulphur_total_quantity);

# max quantity of each tank
subject to tank_one_max_capacity:
        tank_one['quantity'] <= 15;
subject to tank_two_max_capacity:
        tank_two['quantity'] <= 15;


# STAGE 3 - blending constraints

# quantity of final products
subject to blending_hsfo_quantity:
        hsfo['quantity'] =
        ((t * tank_one['quantity']) + ((1 - t) * tank_two['quantity']));
subject to blending_lsfo_quantity:
        lsfo['quantity'] =
        ((t * tank_two['quantity']) + ((1 - t) * tank_one['quantity']));
```

```
# sulphur content of final products
subject to blending_hsfo_sulphur:
        hsfo['sulphur'] =
        ((t * tank_one['sulphur']) + ((1 - t) * tank_two['sulphur']));
subject to blending_lsfo_sulphur:
        lsfo['sulphur'] =
        ((t * tank_two['sulphur']) + ((1 - t) * tank_one['sulphur']));

# viscosity of final products
subject to blending_hsfo_viscosity:
        hsfo['viscosity'] =
        ((t * tank_one['viscosity']) + ((1 - t) * tank_two['viscosity']));
subject to blending_lsfo_viscosity:
        lsfo['viscosity'] =
        ((t * tank_two['viscosity']) + ((1 - t) * tank_one['viscosity']));


# The following constraints are all bounds on our 72 variables
# We organise the following constraints by production stage order


# intermediate product constraints

# quantity of high sulphur 1 intermediates
subject to hs1_sr_quantity_min:
        hs1_sr['quantity']  >= 1;
subject to hs1_sr_quantity_max:
        hs1_sr['quantity']  <= 3;
subject to hs1_cr_quantity_min:
        hs1_cr['quantity']  >= 0;
subject to hs1_cr_quantity_max:
        hs1_cr['quantity']  <= 3;
subject to hs1_hgo_quantity_min:
        hs1_hgo['quantity'] >= 0;
subject to hs1_hgo_quantity_max:
        hs1_hgo['quantity'] <= 3;
subject to hs1_vgo_quantity_min:
        hs1_vgo['quantity'] >= 0;
subject to hs1_vgo_quantity_max:
        hs1_vgo['quantity'] <= 3;

# quantity of high sulphur 2 intermediates
subject to hs2_sr_quantity_min:
        hs2_sr['quantity']  >= 1;
subject to hs2_sr_quantity_max:
```

```
        hs2_sr['quantity']  <= 3;
subject to hs2_cr_quantity_min:
        hs2_cr['quantity']  >= 0;
subject to hs2_cr_quantity_max:
        hs2_cr['quantity']  <= 3;
subject to hs2_hgo_quantity_min:
        hs2_hgo['quantity'] >= 0;
subject to hs2_hgo_quantity_max:
        hs2_hgo['quantity'] <= 3;
subject to hs2_vgo_quantity_min:
        hs2_vgo['quantity'] >= 0;
subject to hs2_vgo_quantity_max:
        hs2_vgo['quantity'] <= 3;

# quantity of low sulphur 1 intermediates
subject to ls1_sr_quantity_min:
        ls1_sr['quantity']  >= 1;
subject to ls1_sr_quantity_max:
        ls1_sr['quantity']  <= 3;
subject to ls1_cr_quantity_min:
        ls1_cr['quantity']  >= 0;
subject to ls1_cr_quantity_max:
        ls1_cr['quantity']  <= 3;
subject to ls1_hgo_quantity_min:
        ls1_hgo['quantity'] >= 0;
subject to ls1_hgo_quantity_max:
        ls1_hgo['quantity'] <= 3;
subject to ls1_vgo_quantity_min:
        ls1_vgo['quantity'] >= 0;
subject to ls1_vgo_quantity_max:
        ls1_vgo['quantity'] <= 3;

# quantity of low sulphur 2 intermediates
subject to ls2_sr_quantity_min:
        ls2_sr['quantity']  >= 1;
subject to ls2_sr_quantity_max:
        ls2_sr['quantity']  <= 3;
subject to ls2_cr_quantity_min:
        ls2_cr['quantity']  >= 0;
subject to ls2_cr_quantity_max:
        ls2_cr['quantity']  <= 3;
subject to ls2_hgo_quantity_min:
        ls2_hgo['quantity'] >= 0;
subject to ls2_hgo_quantity_max:
        ls2_hgo['quantity'] <= 3;
subject to ls2_vgo_quantity_min:
```

```
        ls2_vgo['quantity'] >= 0;
subject to ls2_vgo_quantity_max:
        ls2_vgo['quantity'] <= 3;

# sulphur content of the HS1 intermediates
subject to hs1_sr_sulphur:
        hs1_sr['sulphur']  = 5.84;
subject to hs1_cr_sulphur:
        hs1_cr['sulphur']  = 5.40;
subject to hs1_hgo_sulphur:
        hs1_hgo['sulphur'] = 0.24;
subject to hs1_vgo_sulphur:
        hs1_vgo['sulphur'] = 2.01;

# sulphur content of the HS2 intermediates
subject to hs2_sr_sulphur:
        hs2_sr['sulphur']  = 5.85;
subject to hs2_cr_sulphur:
        hs2_cr['sulphur']  = 5.38;
subject to hs2_hgo_sulphur:
        hs2_hgo['sulphur'] = 0.26;
subject to hs2_vgo_sulphur:
        hs2_vgo['sulphur'] = 2.04;

# sulphur content of the LS1 intermediates
subject to ls1_sr_sulphur:
        ls1_sr['sulphur']  = 0.64;
subject to ls1_cr_sulphur:
        ls1_cr['sulphur']  = 0.57;
subject to ls1_hgo_sulphur:
        ls1_hgo['sulphur'] = 0.02;
subject to ls1_vgo_sulphur:
        ls1_vgo['sulphur'] = 0.14;

# sulphur content of the LS2 intermediates
subject to ls2_sr_sulphur:
        ls2_sr['sulphur']  = 0.93;
subject to ls2_cr_sulphur:
        ls2_cr['sulphur']  = 0.85;
subject to ls2_hgo_sulphur:
        ls2_hgo['sulphur'] = 0.03;
subject to ls2_vgo_sulphur:
        ls2_vgo['sulphur'] = 0.26;

# viscosity of the HS1 intermediates
subject to hs1_sr_viscosity:
```

```
        hs1_sr['viscosity']  = 43.7;
subject to hs1_cr_viscosity:
        hs1_cr['viscosity']  = 36.8;
subject to hs1_hgo_viscosity:
        hs1_hgo['viscosity'] = 12.8;
subject to hs1_vgo_viscosity:
        hs1_vgo['viscosity'] = 15.4;

# viscosity of the HS2 intermediates
subject to hs2_sr_viscosity:
        hs2_sr['viscosity']  = 47.3;
subject to hs2_cr_viscosity:
        hs2_cr['viscosity']  = 39.2;
subject to hs2_hgo_viscosity:
        hs2_hgo['viscosity'] = 13.1;
subject to hs2_vgo_viscosity:
        hs2_vgo['viscosity'] = 15.9;

# viscosity of the LS1 intermediates
subject to ls1_sr_viscosity:
        ls1_sr['viscosity']  = 39.9;
subject to ls1_cr_viscosity:
        ls1_cr['viscosity']  = 38.2;
subject to ls1_hgo_viscosity:
        ls1_hgo['viscosity'] = 13.5;
subject to ls1_vgo_viscosity:
        ls1_vgo['viscosity'] = 16.3;

# viscosity of the LS2 intermediates
subject to ls2_sr_viscosity:
        ls2_sr['viscosity']  = 38.1;
subject to ls2_cr_viscosity:
        ls2_cr['viscosity']  = 34.1;
subject to ls2_hgo_viscosity:
        ls2_hgo['viscosity'] = 13.2;
subject to ls2_vgo_viscosity:
        ls2_vgo['viscosity'] = 15.5;



# final product requirements

# quantity of final products
subject to lsfo_quantity_min:
        lsfo['quantity'] >= 10;
subject to lsfo_quantity_max:
```

```
        lsfo['quantity'] <= 11;
subject to hsfo_quantity_min:
        hsfo['quantity'] >= 11;
subject to hsfo_quantity_max:
        hsfo['quantity'] <= 17;

# sulphur content of final products
subject to lsfo_sulphur_min:
        lsfo['sulphur'] >= 0.0;
subject to lsfo_sulphur_max:
        lsfo['sulphur'] <= lsfo_sulphur_upper_bound;
subject to hsfo_sulphur_min:
        hsfo['sulphur'] >= 0.0;
subject to hsfo_sulphur_max:
        hsfo['sulphur'] <= hsfo_sulphur_upper_bound;

# viscosity of final products
subject to lsfo_viscosity_min:
        lsfo['viscosity'] >= 30.0;
subject to lsfo_viscosity_max:
        lsfo['viscosity'] <= 34.0;
subject to hsfo_viscosity_min:
        hsfo['viscosity'] >= 32.0;
subject to hsfo_viscosity_max:
        hsfo['viscosity'] <= 40.0;
```

3.1.2 AMPL Data File

```
## Math6120 - Nonlinear Optimization
## Coursework 2 - AMPL Model - Data File
## Emma Tarmey, 2940 4045

## This file specifies all parameters required by the corresponding cw2 model

param lsfo_base_price := 150;
param hsfo_base_price := 100;

param lsfo_sulphur_upper_bound := 1.5;
param hsfo_sulphur_upper_bound := 3.5;
```

3.1.3 AMPL Run File

```
reset;
option solver conopt;
option display_1col 1;
model cw2.mod;
data  cw2.dat;

print "";
print "***** Math6120 - Nonlinear Optimization *****";
print "***** Coursework 2 - AMPL Model *****";
print "***** Emma Tarmey, 2940 4045 *****";
print "";

solve;
print "";
print "***** Optimal Solution: *****";
print "";

print "";
print "***** Starting Products: *****";
print "";
display hs1;
display hs2;
display ls1;
display ls2;

print "";
print "***** Intermediate Products: *****";
print "";
display hs1_sr;
display hs1_cr;
display hs1_hgo;
display hs1_vgo;

display hs2_sr;
display hs2_cr;
display hs2_hgo;
display hs2_vgo;

display ls1_sr;
display ls1_cr;
display ls1_hgo;
display ls1_vgo;

display ls2_sr;
```

```
display ls2_cr;
display ls2_hgo;
display ls2_vgo;

print "";
print "***** Tank Products: *****";
print "";
display tank_one;
display tank_two;

print "";
print "***** Final Products: *****";
print "";
display t;
display lsfo;
display hsfo;
display total_sales_volume;
```

3.2 AMPL Output

3.2.1 AMPL Output using ConOpt Solver

```
ampl: include cw2_verbose.run;

***** Math6120 - Nonlinear Optimization *****
***** Coursework 2 - AMPL Model *****
***** Emma Tarmey, 2940 4045 *****

CONOPT 3.17A: Locally optimal; objective 4066.254854
19 iterations; evals: nf = 11, ng = 10, nc = 325, nJ = 33, nH = 0, nHv = 0

***** Optimal Solution: *****


***** Starting Products: *****

hs1 [*] :=
quantity  6
;

hs2 [*] :=
quantity  9
;

ls1 [*] :=
quantity  8.91076
;

ls2 [*] :=
quantity  1.81077
;


***** Intermediate Products: *****

hs1_sr [*] :=
 quantity  3        sulphur  5.84   viscosity 43.7
;

hs1_cr [*] :=
 quantity  0.581033     sulphur  5.4        viscosity 36.8
;

hs1_hgo [*] :=
```

```
 quantity  2.41897     sulphur  0.24      viscosity 12.8
;

hs1_vgo [*] :=
 quantity  0        sulphur  2.01   viscosity 15.4
;

hs2_sr [*] :=
 quantity  3        sulphur  5.85   viscosity 47.3
;

hs2_cr [*] :=
 quantity  3        sulphur  5.38   viscosity 39.2
;

hs2_hgo [*] :=
 quantity  3        sulphur  0.26   viscosity 13.1
;

hs2_vgo [*] :=
 quantity  0        sulphur  2.04   viscosity 15.9
;

ls1_sr [*] :=
 quantity  2.91076     sulphur  0.64      viscosity 39.9
;

ls1_cr [*] :=
 quantity  3        sulphur  0.57   viscosity 38.2
;

ls1_hgo [*] :=
 quantity  3        sulphur  0.02   viscosity 13.5
;

ls1_vgo [*] :=
 quantity  0        sulphur  0.14   viscosity 16.3
;

ls2_sr [*] :=
 quantity  1        sulphur  0.93   viscosity 38.1
;

ls2_cr [*] :=
 quantity  0        sulphur  0.85   viscosity 34.1
;
```

```
ls2_hgo [*] :=
 quantity  0.810774     sulphur  0.03       viscosity 13.2
;

ls2_vgo [*] :=
 quantity  0        sulphur  0.26   viscosity 15.5
;


***** Tank Products: *****

tank_one [*] :=
 quantity 15           sulphur  3.71388   viscosity 32.1497
;

tank_two [*] :=
 quantity 10.7215      sulphur  0.42785   viscosity 29.8503
;


***** Final Products: *****

t = 0.934914

lsfo [*] :=
 quantity 11            sulphur  0.641726   viscosity 30
;

hsfo [*] :=
 quantity 14.7215     sulphur  3.5      viscosity 32
;

total_sales_volume = 4066.25

ampl:
```

3.2.2 AMPL Output using Knitro Solver

```
ampl: include cw2_verbose.run;

***** Math6120 - Nonlinear Optimization *****
***** Coursework 2 - AMPL Model *****
***** Emma Tarmey, 2940 4045 *****

Artelys Knitro 13.0.1:                        Knitro 13.0.1: Locally optimal or satisfactory
objective 4066.254852; feasibility error 1.4e-09
86 iterations; 115 function evaluations

suffix feaserror OUT;
suffix opterror OUT;
suffix numfcevals OUT;
suffix numiters OUT;

***** Optimal Solution: *****


***** Starting Products: *****

hs1 [*] :=
quantity  1.22383e+13
;

hs2 [*] :=
quantity  47786900000
;

ls1 [*] :=
quantity  2848550000
;

ls2 [*] :=
quantity  4786360000
;


***** Intermediate Products: *****

hs1_sr [*] :=
 quantity  3       sulphur  5.84   viscosity 43.7
;

hs1_cr [*] :=
```

```
 quantity  0.581033     sulphur  5.4        viscosity 36.8
;

hs1_hgo [*] :=
 quantity  2.41897     sulphur  0.24      viscosity 12.8
;

hs1_vgo [*] :=
 quantity  1.19719e-09     sulphur  2.01          viscosity 15.4
;

hs2_sr [*] :=
 quantity  3        sulphur  5.85   viscosity 47.3
;

hs2_cr [*] :=
 quantity  3        sulphur  5.38   viscosity 39.2
;

hs2_hgo [*] :=
 quantity  3        sulphur  0.26   viscosity 13.1
;

hs2_vgo [*] :=
 quantity  1.28004e-09     sulphur  2.04          viscosity 15.9
;

ls1_sr [*] :=
 quantity  2.91076     sulphur  0.64      viscosity 39.9
;

ls1_cr [*] :=
 quantity  3        sulphur  0.57   viscosity 38.2
;

ls1_hgo [*] :=
 quantity  3        sulphur  0.02   viscosity 13.5
;

ls1_vgo [*] :=
 quantity  3.45939e-08     sulphur  0.14          viscosity 16.3
;

ls2_sr [*] :=
 quantity  1        sulphur  0.93   viscosity 38.1
;
```

```
ls2_cr [*] :=
 quantity  3.74668e-09     sulphur  0.85          viscosity 34.1
;

ls2_hgo [*] :=
 quantity  0.810774     sulphur  0.03       viscosity 13.2
;

ls2_vgo [*] :=
 quantity  8.38464e-09     sulphur  0.26          viscosity 15.5
;


***** Tank Products: *****

tank_one [*] :=
 quantity 15           sulphur  3.71388   viscosity 32.1497
;

tank_two [*] :=
 quantity 10.7215      sulphur  0.42785   viscosity 29.8503
;


***** Final Products: *****

t = 0.934914

lsfo [*] :=
 quantity 11            sulphur  0.641726   viscosity 30
;

hsfo [*] :=
 quantity 14.7215     sulphur  3.5      viscosity 32
;

total_sales_volume = 4066.25

ampl:
```

3.2.3 AMPL Output using Minos Solver

```
ampl: include cw2_verbose.run;

***** Math6120 - Nonlinear Optimization *****
***** Coursework 2 - AMPL Model *****
***** Emma Tarmey, 2940 4045 *****

MINOS 5.51: optimal solution found.
239 iterations, objective 4066.254855
Nonlin evals: obj = 111, grad = 110, constrs = 111, Jac = 110.

***** Optimal Solution: *****


***** Starting Products: *****

hs1 [*] :=
quantity  6
;

hs2 [*] :=
quantity  9
;

ls1 [*] :=
quantity  8.91076
;

ls2 [*] :=
quantity  1.81077
;


***** Intermediate Products: *****

hs1_sr [*] :=
 quantity  3        sulphur  5.84   viscosity 43.7
;

hs1_cr [*] :=
 quantity  0.581033     sulphur  5.4        viscosity 36.8
;

hs1_hgo [*] :=
 quantity  2.41897     sulphur  0.24      viscosity 12.8
```

```
;

hs1_vgo [*] :=
 quantity  0        sulphur  2.01   viscosity 15.4
;

hs2_sr [*] :=
 quantity  3        sulphur  5.85   viscosity 47.3
;

hs2_cr [*] :=
 quantity  3        sulphur  5.38   viscosity 39.2
;

hs2_hgo [*] :=
 quantity  3        sulphur  0.26   viscosity 13.1
;

hs2_vgo [*] :=
 quantity  0        sulphur  2.04   viscosity 15.9
;

ls1_sr [*] :=
 quantity  2.91076     sulphur  0.64      viscosity 39.9
;

ls1_cr [*] :=
 quantity  3        sulphur  0.57   viscosity 38.2
;

ls1_hgo [*] :=
 quantity  3        sulphur  0.02   viscosity 13.5
;

ls1_vgo [*] :=
 quantity  0        sulphur  0.14   viscosity 16.3
;

ls2_sr [*] :=
 quantity  1        sulphur  0.93   viscosity 38.1
;

ls2_cr [*] :=
 quantity  0        sulphur  0.85   viscosity 34.1
;
```

```
ls2_hgo [*] :=
 quantity  0.810774     sulphur  0.03       viscosity 13.2
;

ls2_vgo [*] :=
 quantity  0        sulphur  0.26   viscosity 15.5
;


***** Tank Products: *****

tank_one [*] :=
 quantity 15           sulphur  3.71388   viscosity 32.1497
;

tank_two [*] :=
 quantity 10.7215      sulphur  0.42785   viscosity 29.8503
;


***** Final Products: *****

t = 0.934914

lsfo [*] :=
 quantity 11            sulphur  0.641726   viscosity 30
;

hsfo [*] :=
 quantity 14.7215     sulphur  3.5      viscosity 32
;

total_sales_volume = 4066.25

ampl:
```

3.2.4 AMPL Output using Snopt Solver

```
ampl: include cw2_verbose.run;

***** Math6120 - Nonlinear Optimization *****
***** Coursework 2 - AMPL Model *****
***** Emma Tarmey, 2940 4045 *****

SNOPT 7.5-1.2 : Optimal solution found.
111 iterations, objective 4066.254882
Nonlin evals: obj = 23, grad = 22, constrs = 23, Jac = 22.

***** Optimal Solution: *****


***** Starting Products: *****

hs1 [*] :=
quantity  6
;

hs2 [*] :=
quantity  9
;

ls1 [*] :=
quantity  8.91076
;

ls2 [*] :=
quantity  1.81077
;


***** Intermediate Products: *****

hs1_sr [*] :=
 quantity  3        sulphur  5.84   viscosity 43.7
;

hs1_cr [*] :=
 quantity  0.581033     sulphur  5.4        viscosity 36.8
;

hs1_hgo [*] :=
 quantity  2.41897     sulphur  0.24      viscosity 12.8
```

```
;

hs1_vgo [*] :=
 quantity  0        sulphur  2.01   viscosity 15.4
;

hs2_sr [*] :=
 quantity  3        sulphur  5.85   viscosity 47.3
;

hs2_cr [*] :=
 quantity  3        sulphur  5.38   viscosity 39.2
;

hs2_hgo [*] :=
 quantity  3        sulphur  0.26   viscosity 13.1
;

hs2_vgo [*] :=
 quantity  0        sulphur  2.04   viscosity 15.9
;

ls1_sr [*] :=
 quantity  2.91076     sulphur  0.64      viscosity 39.9
;

ls1_cr [*] :=
 quantity  3        sulphur  0.57   viscosity 38.2
;

ls1_hgo [*] :=
 quantity  3        sulphur  0.02   viscosity 13.5
;

ls1_vgo [*] :=
 quantity  0        sulphur  0.14   viscosity 16.3
;

ls2_sr [*] :=
 quantity  1        sulphur  0.93   viscosity 38.1
;

ls2_cr [*] :=
 quantity  0        sulphur  0.85   viscosity 34.1
;
```

```
ls2_hgo [*] :=
 quantity  0.810774     sulphur  0.03       viscosity 13.2
;

ls2_vgo [*] :=
 quantity  0        sulphur  0.26   viscosity 15.5
;


***** Tank Products: *****

tank_one [*] :=
 quantity 15           sulphur  3.71388   viscosity 32.1497
;

tank_two [*] :=
 quantity 10.7215      sulphur  0.42785   viscosity 29.8503
;


***** Final Products: *****

t = 0.934914

lsfo [*] :=
 quantity 11            sulphur  0.641726   viscosity 30
;

hsfo [*] :=
 quantity 14.7215     sulphur  3.5      viscosity 32
;

total_sales_volume = 4066.25

ampl:
```

3.2.5 AMPL Output using ConOpt, simulating Removal of All Pool Tanks

```
ampl: include cw2_verbose.run;

***** Math6120 - Nonlinear Optimization *****
***** Coursework 2 - AMPL Model *****
***** Emma Tarmey, 2940 4045 *****

CONOPT 3.17A: Locally infeasible; objective 3610.85342
7 iterations; evals: nf = 2, ng = 1, nc = 59, nJ = 5, nH = 0, nHv = 0

***** Optimal Solution: *****


***** Starting Products: *****

hs1 [*] :=
quantity  3.24536
;

hs2 [*] :=
quantity  6
;

ls1 [*] :=
quantity  4.56042
;

ls2 [*] :=
quantity  5.43958
;


***** Intermediate Products: *****

hs1_sr [*] :=
 quantity  2.36605     sulphur  5.84      viscosity 43.7
;

hs1_cr [*] :=
 quantity  0       sulphur  5.4   viscosity 36.8
;

hs1_hgo [*] :=
 quantity  0.87931     sulphur  0.24      viscosity 12.8
```

```
;

hs1_vgo [*] :=
 quantity  0        sulphur  2.01   viscosity 15.4
;

hs2_sr [*] :=
 quantity  3        sulphur  5.85   viscosity 47.3
;

hs2_cr [*] :=
 quantity  0        sulphur  5.38   viscosity 39.2
;

hs2_hgo [*] :=
 quantity  3        sulphur  0.26   viscosity 13.1
;

hs2_vgo [*] :=
 quantity  0        sulphur  2.04   viscosity 15.9
;

ls1_sr [*] :=
 quantity  2.60235     sulphur  0.64      viscosity 39.9
;

ls1_cr [*] :=
 quantity  1.50544     sulphur  0.57      viscosity 38.2
;

ls1_hgo [*] :=
 quantity  0.452633     sulphur  0.02       viscosity 13.5
;

ls1_vgo [*] :=
 quantity  0        sulphur  0.14   viscosity 16.3
;

ls2_sr [*] :=
 quantity  2.43958     sulphur  0.93      viscosity 38.1
;

ls2_cr [*] :=
 quantity  0        sulphur  0.85   viscosity 34.1
;
```

```
ls2_hgo [*] :=
 quantity  3        sulphur  0.03   viscosity 13.2
;

ls2_vgo [*] :=
 quantity  0        sulphur  0.26   viscosity 15.5
;


***** Tank Products: *****

tank_one [*] :=
 quantity 11       sulphur  3.5   viscosity 32
;

tank_two [*] :=
 quantity 10            sulphur  0.489147   viscosity 30
;


***** Final Products: *****

t = 1

lsfo [*] :=
 quantity 10            sulphur  0.489147   viscosity 30
;

hsfo [*] :=
 quantity 11       sulphur  3.5   viscosity 32
;

total_sales_volume = 3610.85

ampl:
```

3.3 Acknowledgements

- A forum for discussing AMPL and asking questions is available at:
  `https://groups.google.com/g/ampl`
  This forum was consulted to understand the AMPL language and environment during development of our model.

- Programming advice is available at:
  `https://stackoverflow.com/questions/tagged/ampl`
  This forum was consulted to understand the AMPL language syntax during development of our model.