Table of Contents

1.0 Abstract

WARP Shoe Company seeks to optimize their production process to meet growing demands and increase their overall profitability. To meet the guidelines and requirements set by the company, a production model is created to identify the optimal manufacturing system, aimed at maximizing profit. With 557 different shoe types to be produced, the model uses a decision variable, x_i, that represents the quantity of type *i* shoe to be manufactured. The plan incorporates five constraints: budget for operating costs, machine resource usage, machine runtime, available warehouse capacity, and product demands. After formulating the model, A Mathematical Programming Language (AMPL) is coded to simulate the system, projecting an optimal profit of $11,789,734.80.

2.0 Introduction

WARP Shoe Company, one of the oldest shoe manufacturers in Canada, faced a significant market shift in January 2006 when one of their major competitors went bankrupt. This led WARP to predict that the demand for their shoes would double during February. To address this, the company has asked the Industrial Engineering Department at the University of Toronto to develop a model that can determine the optimal maximized profit for February 2006, while adhering to specific constraints and requirements, as outlined in Appendix A.

Using the data provided by the client, an integer programming model is formulated in AMPL, which uses Gurobi solver, to model the maximized profit.

3.0 Methodology

As part of the formulation process for this integer program, it is important to determine the sets, parameters, objective function and constraints, which can then be represented in AMPL for solving.

Sets (Figure 1)

The model initializes some sets that will later extract data from the MS Access database:

1. Products (indexed by *i*, the number of unique shoe types produced by WARP)
2. Raw Materials (indexed by *j*, the number of distinct raw materials needed to create a shoe)
3. Machines (indexed by *k*, the number of machines in the company)
4. Warehouses (indexed by *l*, the number of warehouses in the company)
5. Years (range of the years for which information about the demand is given: 1997 to 2003)
6. Stores (range of the store numbers: 1 to 10)

Restrictions:

$$i \in \{1, \ldots, 557\}$$
$$j \in \{1, \ldots, 165\}$$
$$k \in \{1, \ldots, 72\}$$
$$l \in \{1, \ldots, 8\}$$

```
# Defining sets for the variables involved
set products;                                   # Set of products
set raw_material;                               # Set of raw materials
set machines;                                   # Set of machines
set warehouse;                                  # Set of warehouse
set years = 1997..2003;                         # Range of years for calculating demand
set stores = 1..10;                             # Range of store numbers
```

Figure 1. Excerpt from AMPL code that displays how the sets were initialized.

Parameters (Figure 2)

The model involves ten different parameters that were needed to gather the optimal solution:

1. Selling price: the sale price for a pair of type *i* shoes (in dollars)
2. Demand: expected number of shoe type *i* needed to meet the market demand
3. Cost: the cost of raw material *j* (in dollars)
4. Average duration: average time it takes to manufacture shoe type *i* with raw material *j*
5. Quantity: the number of type *i* shoes created by machine *k*
6. Operating cost: cost to operate machine *k*
7. Total capacity: total capacity at warehouse *l*
8. Resource limit: resource limit for warehouse *l*
9. House Capacity: number of shoes that can be stored at warehouse *l*
10. Month demand: demand for shoe *i* for every February (from 1997 to 2003) for each store number

```
# Defining parameters for the input data
param selling_price {products};                        # Selling price for each product
param demand {products};                               # Demand for each product
param cost {raw_material};                             # Cost for each raw material
param average_duration {products, raw_material};       # Average processing time per product and raw material
param quantity {products, machines};                   # Quantity of product processed by each machine
param operating_cost {machines};                       # Operating cost per machine
param total_capacity {warehouse};                      # Total capacity for each warehouse
param resource_limit {machines};                       # Resource limits for each machine
param house_capacity {warehouse};                      # Capacity of each warehouse
param month_demand {products, years, {2}, stores};     # Monthly demand for products in February
```

Figure 2. Excerpt from AMPL code that displays how the parameters were initialized.

Decision Variables (Figure 3)

The model incorporates two types of decision variables: x_i and y_l. The first decision variable represents how many type *i* shoes can be manufactured in the optimal solution (this is later rounded in the .run file, as there cannot be fractions of a shoe). The *round* function is chosen, instead of an integer cast in the .mod file, in order to relax the model and reduce computation time. The second decision variable is a binary variable that decides whether a specific warehouse is being capacitated.

Restrictions:

$$x_i \in \mathbb{Z}^+$$
$$y_l \in \{0, 1\}$$

```
# Defining decision variables
var x {products} >= 0;                        # Number of units of each product for production (non-negative)
var y {warehouse} binary default 0;           # Binary variable for warehouse usage (1 if used, 0 if unused)
```

Figure 3. Excerpt from AMPL code that displays how the decision variable was initialized.

Objective Function Value

The objective function value, which represents the optimal profit in dollars, is created by considering all sales and expenses. The AMPL code for the objective function can be found in Appendix B.

$$\begin{aligned} maximize\ z \ = \ & \left(\sum_{i=1}^{557} ((selling\ price \cdot x_i) - 10(x_i - demand_i))\right. \\ & - \left(\sum_{j=1}^{165} \sum_{i=1}^{557} \left(\frac{25}{3600} + \frac{cost_j}{60}\right) \cdot x_i \cdot averageDuration_{ij}\right) \\ & - \left(\sum_{k=1}^{72} \sum_{i=1}^{557} x_i \cdot quantity_{ik} \cdot operatingCost_k\right) \\ & - \left(\sum_{l=1}^{8} totalCapacity_l \cdot y_l\right) \end{aligned}$$

The sales are:

- $(\sum_{i=1}^{557} ((selling\ price \cdot x_i) - 10(x_i - demand_i)$, which denotes how much revenue WARP can make per shoe type *i* (based on each shoe type's selling price) and accounts for the cost of $10.00/shoe for unmet demand.

The expenses are (in order of appearance in the objective function):

- $(\sum_{j=1}^{165} \sum_{i=1}^{557} (\frac{25}{3600} + \frac{cost_j}{60}) \cdot x_i \cdot averageDuration_{ij})$, which denotes the operating cost for the machines depending on how long (time) they can run, as well as labour cost of the workers.
- $(\sum_{k=1}^{72} \sum_{i=1}^{557} x_i \cdot quantity_{ik} \cdot operatingCost_k)$, which denotes the material costs, made up of the quantity of shoe type *i* made by each machine *k*, and the operating costs for those machines.
- $(\sum_{l=1}^{8} totalCapacity_l \cdot y_l)$, which denotes the costs related to storing products at each used warehouse.

Constraints (Figure 4)

Five constraints are implemented into the model to accommodate the client's requirements:

1. **Budget Constraint:** The cost of the raw materials and all operating costs must not exceed $10,000,000. This constraint is indexed by 557 product types (*i*) and 72 machine types (*k*).
2. **Resource Constraint:** There is an upper bound to the availability of the raw materials, which means the resource usage by the machines must remain within their respective limits. This constraint is indexed by 72 machine types (*k*) and 557 product types (*i*).
3. **Runtime Constraint:** The maximum duration that the machines can work for is 14 days. This is further restricted by the working hours of the machines: 12 hours x 60 minutes x 60 seconds x 28 days in February = 1,209,600 seconds. This constraint is indexed by 557 product types (*i*) and 165 raw material types (*j*).
4. **Warehouse Capacity Constraint:** The quantity of products that are produced must fit the available capacities of the individual warehouses. This is indexed by 557 product types (*i*) and each warehouse (*l*).
5. **Demand Constraint:** The demand for a specific shoe type must be met; there cannot be overproduction, as this would lead to losses. This is indexed by the 557 product types (*i*).

```
# Defining constraints

# Budget constraint... total operating costs for machines and cannot exceed $10,000,000
subject to budget: sum {i in products, k in machines} (x[i] * quantity[i,k] * operating_cost[k]) <= 10000000;
# for question 7) change total operating costs to $17,000,000

# Resource constraint... resource usage by machines must not exceed their limit
subject to resources {k in machines}: sum {i in products} x[i] * quantity [i,k] <= resource_limit[k];

# Runtime constraint... total processing time for raw materials must not exceed 14 days (in seconds)
# 14 days = 12 hours * 28 days * 60 minutes * 60 seconds = 1,209,600
subject to runtime {j in raw_material}: sum {i in products} (x[i] * average_duration[i,j]) <= 1209600;
# for question 6) replace with 8 hours * 28 days * 60 minutes * 60 seconds = 806,400

# Warehouse capacity constraint... total product quantity must fit within available warehouse capacity
subject to warehouse_capacity: sum {i in products} x[i] <= sum {l in warehouse} house_capacity[l] * y[l];

# Demand constraint... the production of each product cannot exceed its demand
subject to demand_limit {i in products}: x[i] <= demand[i];
```

Figure 4. Excerpt from AMPL code that displays how the constraints were initialized.

4.0 Results

The integer programming model is solved using AMPL's Gurobi solver, generating four files in the process: .mod, .dat, .run, and .out.

Table 1. Files in the AMPL model.

File	Description
.mod	Defines the sets, parameters, decision variables, objective function, and constraints required for the mathematical model.
.dat	Collects the necessary input data for solving the model, referencing specific tables from the datasets (provided by the company's .mdb database file).
.run	Executes the .mod and .dat files, using Gurobi solver to find a solution.
.out	Records and presents all the results and solutions obtained from the solver.

After executing the model in AMPL, the optimal objective value is determined to be z^* = **$11,789,734.80**, representing the profit in dollars.

1. **How should you estimate the demand for the month of February?**

The client has requested a projection for February 2006. However, the database only contains information from the years 1997 to 2003. This means that the data for every February during these seven years can be extracted and then averaged to extrapolate a demand for February 2006.

To apply this in the AMPL code, an SQL query must be added to the .dat file where the data for

```
############# Read data from Product_Demand table here. For example, The following code reads the demand for month 1 of each year.
table Product_Demand IN "ODBC" "W:\\WARP2011W.mdb" "SQL=SELECT * FROM Product_Demand WHERE Month=2":
    [Product_Num, Year, Month, Store_Num], month_demand ~ Demand;
read table Product_Demand;

for {i in products} {
    let demand[i] := 2 * (sum {year in years, store in stores} month_demand[i, year, 2, store]) / 7;
}

####################################################
```

the demand is taken from the database:

Figure 5. Excerpt from .dat file in AMPL model that uses the SQL query

The query can be translated as:

1. Only the products associated with February data (month = 2) are selected from the Product Demand table in the database.

Later, in the for loop:

2. The data is divided by 7 to take the average across the seven years.
3. The data is also multiplied by 2 because the demand for February 2006 is double that of January 2006, as projected by the client.

2. **How many variables and constraints do you have?**

- **Variables:**
    - The model includes two types of decision variables. The first variable, x_i, represents the number of shoes of types i to be produced. There are 557 distinct types of shoes, resulting in 577 decision variables. Therefore, i = 557. The second variable is y_l, which is a binary variable that determines whether a warehouse is in use or not.
- **Constraints:**
    - The model incorporates five constraints:
        1. **Budget Constraint**: The total operating costs for machines must not exceed $10,000,000. This constraint is indexed by 557 product types (i) and 72 machine types (k).

2. **Resource Constraint**: Resource usage by machines must remain within their limits. This is indexed by 72 machine types (*k*) and 557 product types (*i*).
3. **Runtime Constraint**: The total processing time for raw materials must not exceed 14 days (equivalent to 1,209,600 seconds). This constraint is indexed by 557 product types (*i*) and 165 raw material types (*j*).
4. **Warehouse Capacity Constraint**: The quantity of products must fit within the available capacities of warehouses. This is indexed by 557 product types (*i*) and each warehouse (*l*).
5. **Demand Constraint**: The production of each product type cannot exceed its demand. This is indexed by the 557 product types (*i*).

3. **If you had to relax your integer program to an LP, how many constraints were violated after rounding the LP solution to the closest integer solution?**

The code originally does not assume fractional values for x_i (which represent production quantities) to avoid a solution that is not feasible. Since a fraction of a shoe cannot be manufactured, the optimal values are rounded to the nearest integer using the *round* function. This reflects an integer program (IP).

By relaxing to a linear program (LP), the values of x_i are left in fractional values. To do so, the rounding is removed to allow x_i to take on non-integer values. We can identify the violating constraints by checking which are restricted, as those constraints will be violated if more units are added to them. This means that the constraints that are currently binding will be violated first. After solving the relaxed LP, it is observed that the **two constraints**, related to available raw material and maximum demand, are the ones that are violated.

4. **Which constraints are binding, and what is the real-world interpretation of those binding constraints?**

- **Binding Constraints:**
  - Identified by constraints with zero slack, the binding constraints are the available raw material constraint and the demand constraint (Figure 6).

- **Real-World Interpretation:**
  1. Available Raw Material Constraint: This indicates that the solution is limited by the total supply of raw materials for the specific month. In practical terms, all available raw materials have been fully utilized, and no additional resources are available to increase production. This suggests a resource bottleneck, where the

lack of additional raw materials prevents further production, and in turn, stunts the maximum profit.

2. Demand Constraint: This constraint implies that the production for certain shoe types exactly meets the market demand. Producing more than the demand would incur unnecessary costs without generating additional profit. Therefore, production is strictly capped to avoid overproduction and minimize waste.

Overall, these binding constraints highlight the key limitations in the production process: the availability of raw materials, as well as the market's demand for specific products. By addressing limitations to overcome these bottlenecks, such as sourcing additional raw materials or expanding the demand, a more profitable solution can be reached. Sensitivity analysis can be applied to quantitatively determine the amount by which said solutions could improve the objective value.

```
3) Binding constraints (constraints at their limits):

   - Resource availability for machine 1 is fully utilized (binding).
   - Resource availability for machine 5 is fully utilized (binding).
   - Resource availability for machine 11 is fully utilized (binding).
   - Resource availability for machine 13 is fully utilized (binding).
   - Resource availability for machine 19 is fully utilized (binding).
   - Resource availability for machine 21 is fully utilized (binding).
   - Resource availability for machine 24 is fully utilized (binding).
   - Resource availability for machine 28 is fully utilized (binding).
   - Resource availability for machine 31 is fully utilized (binding).
   - Resource availability for machine 41 is fully utilized (binding).
   - Resource availability for machine 43 is fully utilized (binding).
   - Resource availability for machine 44 is fully utilized (binding).
   - Resource availability for machine 50 is fully utilized (binding).
   - Resource availability for machine 56 is fully utilized (binding).
   - Resource availability for machine 57 is fully utilized (binding).
   - Resource availability for machine 59 is fully utilized (binding).
   - Resource availability for machine 61 is fully utilized (binding).
   - Resource availability for machine 62 is fully utilized (binding).
   - Resource availability for machine 63 is fully utilized (binding).
   - Resource availability for machine 64 is fully utilized (binding).
   - Resource availability for machine 65 is fully utilized (binding).
   - Resource availability for machine 66 is fully utilized (binding).
   - Resource availability for machine 67 is fully utilized (binding).
   - Resource availability for machine 70 is fully utilized (binding).
   - Resource availability for machine 71 is fully utilized (binding).
   - Resource availability for machine 73 is fully utilized (binding).
   - Resource availability for machine 76 is fully utilized (binding).
   - Resource availability for machine 77 is fully utilized (binding).
   - Resource availability for machine 80 is fully utilized (binding).
   - Resource availability for machine 81 is fully utilized (binding).
   - Resource availability for machine 83 is fully utilized (binding).
   - Resource availability for machine 84 is fully utilized (binding).
   - Resource availability for machine 85 is fully utilized (binding).
   - Resource availability for machine 86 is fully utilized (binding).
   - Resource availability for machine 87 is fully utilized (binding).
   - Resource availability for machine 88 is fully utilized (binding).
   - Resource availability for machine 97 is fully utilized (binding).
   - Resource availability for machine 101 is fully utilized (binding).
   - Resource availability for machine 105 is fully utilized (binding).
   - Resource availability for machine 106 is fully utilized (binding).
   - Resource availability for machine 108 is fully utilized (binding).
   - Resource availability for machine 113 is fully utilized (binding).
   - Resource availability for machine 115 is fully utilized (binding).
   - Resource availability for machine 116 is fully utilized (binding).
   - Resource availability for machine 117 is fully utilized (binding).
   - Resource availability for machine 119 is fully utilized (binding).
   - Resource availability for machine 120 is fully utilized (binding).
   - Resource availability for machine 123 is fully utilized (binding).
   - Resource availability for machine 129 is fully utilized (binding).
   - Resource availability for machine 131 is fully utilized (binding).
```

```
   - Resource availability for machine 132 is fully utilized (binding).
   - Resource availability for machine 133 is fully utilized (binding).
   - Resource availability for machine 134 is fully utilized (binding).
   - Resource availability for machine 135 is fully utilized (binding).
   - Resource availability for machine 137 is fully utilized (binding).
   - Resource availability for machine 139 is fully utilized (binding).
   - Resource availability for machine 140 is fully utilized (binding).
   - Resource availability for machine 142 is fully utilized (binding).
   - Resource availability for machine 144 is fully utilized (binding).
   - Resource availability for machine 145 is fully utilized (binding).
   - Resource availability for machine 146 is fully utilized (binding).
   - Resource availability for machine 147 is fully utilized (binding).
   - Resource availability for machine 149 is fully utilized (binding).
   - Resource availability for machine 150 is fully utilized (binding).
   - Resource availability for machine 151 is fully utilized (binding).
   - Resource availability for machine 153 is fully utilized (binding).
   - Resource availability for machine 154 is fully utilized (binding).
   - Resource availability for machine 158 is fully utilized (binding).
   - Resource availability for machine 159 is fully utilized (binding).
   - Resource availability for machine 163 is fully utilized (binding).
   - Resource availability for machine 164 is fully utilized (binding).
   - Maximum demand for product SH025 is fully met (binding).
   - Maximum demand for product SH027 is fully met (binding).
   - Maximum demand for product SH044 is fully met (binding).
   - Maximum demand for product SH058 is fully met (binding).
   - Maximum demand for product SH062 is fully met (binding).
   - Maximum demand for product SH063 is fully met (binding).
   - Maximum demand for product SH087 is fully met (binding).
   - Maximum demand for product SH095 is fully met (binding).
   - Maximum demand for product SH105 is fully met (binding).
   - Maximum demand for product SH113 is fully met (binding).
   - Maximum demand for product SH121 is fully met (binding).
   - Maximum demand for product SH134 is fully met (binding).
   - Maximum demand for product SH165 is fully met (binding).
   - Maximum demand for product SH176 is fully met (binding).
   - Maximum demand for product SH201 is fully met (binding).
   - Maximum demand for product SH221 is fully met (binding).
   - Maximum demand for product SH247 is fully met (binding).
   - Maximum demand for product SH256 is fully met (binding).
   - Maximum demand for product SH287 is fully met (binding).
   - Maximum demand for product SH320 is fully met (binding).
   - Maximum demand for product SH329 is fully met (binding).
   - Maximum demand for product SH367 is fully met (binding).
   - Maximum demand for product SH369 is fully met (binding).
   - Maximum demand for product SH415 is fully met (binding).
   - Maximum demand for product SH417 is fully met (binding).
   - Maximum demand for product SH430 is fully met (binding).
   - Maximum demand for product SH456 is fully met (binding).
   - Maximum demand for product SH479 is fully met (binding).
   - Maximum demand for product SH490 is fully met (binding).
   - Maximum demand for product SH497 is fully met (binding).
   - Maximum demand for product SH511 is fully met (binding).
   - Maximum demand for product SH534 is fully met (binding).
   - Maximum demand for product SH557 is fully met (binding).
```

Figure 6. The model identifies the binding constraints.

5. **Assume that some additional warehouse space is available at the price of $10/box of shoes. Is it economical to buy it? What is the optimal amount of space to buy in this situation?**

To evaluate whether purchasing additional warehouse space would impact the model's profitability, the dual value of the maximum capacity constraint is analyzed in AMPL. The dual value is calculated to be 0, indicating a shadow price of 0 (Figure 7). This implies that increasing or decreasing the warehouse space will have no effect on the objective value (profit) of the model.

```
4) Dual value (shadow price) for warehouse capacity constraint:
   - Warehouse capacity dual value (shadow price): 0.00
```

Figure 7. The dual value calculated by AMPL is 0.

This conclusion is supported by the fact that the binding constraints are available raw materials and maximum demand, rather than maximum capacity. Since the maximum capacity constraint is not binding, any variation in this constraint will not impact the profit.

Consequently, the optimal profit remains fixed at $11,789,734.80. Investing in additional warehouse space at $10/box is therefore not economical, as it does not result in any profit increase, and instead, it is an additional expense. Hence, the optimal decision is to purchase no additional warehouse space.

6. **Imagine that machines were available for only 8 hours per day. How would your solution change? Which constraints are binding now? Does the new solution seem realistic to you?**

In the AMPL model, reducing the maximum machine operating hours directly tightens the maximum duration constraint. Previously, machines operated for 1,209,600 seconds. With the new limit, they now operate for 806,400 seconds (8 hours/day x 60 minutes/hour x 60 seconds/minute x 28 days in February).

However, even with this adjustment, the model indicates that the optimal profit remains unchanged, and the optimal values are also unaffected. This outcome occurs because the maximum duration constraint is not binding, so modifying it does not influence the objective function. The binding constraints remain the same as before; available raw materials and demand are the real bottlenecks.

This solution seems realistic and logical since the costs associated with machine operation will proportionally adjust to the number of hours: fewer hours result in lower costs, while more hours lead to higher costs. Therefore, this proportional change means that the reduction in machine hours neither improves nor worsens the optimal profit because the limiting factors will still be a lack of raw materials and restricted demand.

7. **If in addition there was a $7,000,000 budget available to buy raw materials, what would you do? Change your formulation and solve it again.**

Re-solving the formulation with a revised budget constraint of $17,000,000 (a $7,000,000 increase from the initial $10,000,000) yields identical results to the original. The optimal profit remains unchanged at $11,789,734.80.

This outcome can be attributed to the fact that the available raw materials constraint is not binding in the initial solution. Since it did not restrict the outcome previously, making it less restrictive by increasing the budget does not alter the model. As a result, the objective value remains unaffected.

5.0 Conclusion

The AMPL model developed for WARP Shoe Company successfully identifies the optimal production plan to maximize profit for February 2006, achieving an objective value of $11,789,734.80. The model uses Gurobi solver to determine the optimal quantity of each shoe type *i* that needs to be produced in order to achieve this profit, as detailed in Appendix C. The results show that raw material availability and demand are the binding constraints, which provides insight into what factors are limiting the profit. Overall, this report outlines the strategies and decisions needed to maximize profitability and simultaneously optimize production for WARP Shoe Company.

6.0 Appendices

Appendix A: Assumptions based on the client's requirements (adapted from the assignment).

- Raw materials budget = $10,000,000
- Demand must be met to avoid losses. The demand for February 2006 can be found by taking the average of every February from 1997 to 2003.
- The cost to make each pair is $10.
- Machines can work a maximum of 12 hours a day, 28 days a month, and the setup times and costs are negligible.
- Workers are paid at a fixed rate of $25/hour, such that there is an equal number of machines and workers.
- Transportation costs can be ignored throughout the entire model.
- The closing inventory of January 2006 was 0.

Appendix B: Excerpt of objective function from AMPL code.

```
# Defining the objective function to maximize profit
maximize profit:
  sum {i in products}
    (selling_price[i] * x[i] + (10 * (x[i] - demand[i])))          # Revenue minus penalty for unmet demand
  - sum {i in products, j in raw_material}
    ((25 / 3600 + cost[j] / 60) * x[i] * average_duration[i,j])     # Raw matrials and processing costs
  - sum {i in products, k in machines}
    (x[i] * quantity[i,k] * operating_cost[k])                      # Machine operating costs
  - sum {l in warehouse}
    (total_capacity[l] * y[l]);                                     # Fixed costs for using warehouses
```

Appendix C: The number of pairs of each shoe type required to meet the optimal solution is determined from the .out file in AMPL.

x [*] :=

```
SH001 0     SH002 449   SH003 0     SH004 0     SH005 33    SH006 0     SH007 0
SH008 0     SH009 27    SH010 0     SH011 433   SH012 289   SH013 425   SH014 0
SH015 0     SH016 0     SH017 0     SH018 247   SH019 200   SH020 0     SH021 0
SH022 0     SH023 0     SH024 399   SH025 404   SH026 0     SH027 434   SH028 0
SH029 410   SH030 0     SH031 306   SH032 491   SH033 0     SH034 373   SH035 0
SH036 0     SH037 0     SH038 479   SH039 389   SH040 433   SH041 339   SH042 215
SH043 0     SH044 442   SH045 374   SH046 0     SH047 411   SH048 292   SH049 0
SH050 437   SH051 0     SH052 0     SH053 0     SH054 0     SH055 0     SH056 0
SH057 205   SH058 432   SH059 8     SH060 191   SH061 168   SH062 458   SH063 446
SH064 18    SH065 369   SH066 0     SH067 0     SH068 0     SH069 0     SH070 0
SH071 339   SH072 7     SH073 346   SH074 0     SH075 0     SH076 65    SH077 177
SH078 407   SH079 195   SH080 409   SH081 164   SH082 289   SH083 0     SH084 0
SH085 0     SH086 0     SH087 430   SH088 423   SH089 0     SH090 475   SH091 0
SH092 0     SH093 285   SH094 307   SH095 422   SH096 453   SH097 0     SH098 0
SH099 0     SH100 0     SH101 0     SH102 0     SH103 0     SH104 0     SH105 418
SH106 201   SH107 1     SH108 0     SH109 0     SH110 417   SH111 0     SH112 0
SH113 454   SH114 0     SH115 0     SH116 200   SH117 21    SH118 294   SH119 0
SH120 0     SH121 508   SH122 0     SH123 0     SH124 0     SH125 0     SH126 76
SH127 381   SH128 0     SH129 0     SH130 94    SH131 55    SH132 0     SH133 135
SH134 490   SH135 0     SH136 0     SH137 0     SH138 0     SH139 0     SH140 0
SH141 0     SH142 399   SH143 0     SH144 449   SH145 0     SH146 489   SH147 68
SH148 0     SH149 88    SH150 425   SH151 0     SH152 0     SH153 130   SH154 437
SH155 0     SH156 0     SH157 0     SH158 0     SH159 0     SH160 21    SH161 0
SH162 0     SH163 0     SH164 0     SH165 460   SH166 0     SH167 0     SH168 380
SH169 0     SH170 0     SH171 0     SH172 82    SH173 308   SH174 0     SH175 0
SH176 450   SH177 0     SH178 140   SH179 108   SH180 0     SH181 0     SH182 0
SH183 11    SH184 105   SH185 371   SH186 68    SH187 115   SH188 246   SH189 0
SH190 0     SH191 81    SH192 0     SH193 0     SH194 304   SH195 243   SH196 0
SH197 437   SH198 0     SH199 0     SH200 0     SH201 476   SH202 0     SH203 501
SH204 0     SH205 144   SH206 0     SH207 0     SH208 0     SH209 436   SH210 0
SH211 0     SH212 0     SH213 437   SH214 0     SH215 0     SH216 0     SH217 206
SH218 0     SH219 0     SH220 0     SH221 468   SH222 3     SH223 409   SH224 443
SH225 404   SH226 419   SH227 256   SH228 0     SH229 0     SH230 70    SH231 0
SH232 419   SH233 449   SH234 405   SH235 471   SH236 497   SH237 0     SH238 396
SH239 110   SH240 107   SH241 419   SH242 0     SH243 0     SH244 242   SH245 0
SH246 0     SH247 422   SH248 79    SH249 405   SH250 0     SH251 0     SH252 0
SH253 381   SH254 479   SH255 203   SH256 434   SH257 0     SH258 0     SH259 0
SH260 2     SH261 409   SH262 411   SH263 0     SH264 0     SH265 413   SH266 0
SH267 0     SH268 0     SH269 429   SH270 76    SH271 0     SH272 0     SH273 267
SH274 119   SH275 0     SH276 280   SH277 423   SH278 0     SH279 19    SH280 0
SH281 308   SH282 415   SH283 0     SH284 461   SH285 0     SH286 437   SH287 410
SH288 162   SH289 430   SH290 130   SH291 0     SH292 136   SH293 0     SH294 0
SH295 403   SH296 0     SH297 0     SH298 111   SH299 408   SH300 427   SH301 0
SH302 0     SH303 0     SH304 0     SH305 0     SH306 0     SH307 81    SH308 266
```

```
SH309 0     SH310 0     SH311 0     SH312 477   SH313 483   SH314 391   SH315 0
SH316 168   SH317 316   SH318 0     SH319 0     SH320 462   SH321 0     SH322 291
SH323 0     SH324 0     SH325 475   SH326 0     SH327 0     SH328 194   SH329 400
SH330 0     SH331 0     SH332 0     SH333 0     SH334 387   SH335 0     SH336 0
SH337 0     SH338 109   SH339 0     SH340 439   SH341 429   SH342 299   SH343 30
SH344 0     SH345 318   SH346 0     SH347 177   SH348 489   SH349 41    SH350 0
SH351 0     SH352 397   SH353 0     SH354 0     SH355 0     SH356 186   SH357 0
SH358 0     SH359 21    SH360 0     SH361 417   SH362 419   SH363 0     SH364 0
SH365 260   SH366 198   SH367 452   SH368 0     SH369 378   SH370 0     SH371 0
SH372 274   SH373 417   SH374 0     SH375 0     SH376 24    SH377 149   SH378 423
SH379 254   SH380 0     SH381 18    SH382 0     SH383 0     SH384 505   SH385 0
SH386 0     SH387 0     SH388 0     SH389 0     SH390 0     SH391 128   SH392 0
SH393 0     SH394 0     SH395 448   SH396 0     SH397 0     SH398 0     SH399 443
SH400 0     SH401 0     SH402 44    SH403 0     SH404 0     SH405 0     SH406 0
SH407 437   SH408 0     SH409 0     SH410 262   SH411 400   SH412 88    SH413 0
SH414 0     SH415 438   SH416 0     SH417 482   SH418 343   SH419 290   SH420 90
SH421 417   SH422 0     SH423 419   SH424 0     SH425 459   SH426 0     SH427 477
SH428 0     SH429 27    SH430 430   SH431 0     SH432 0     SH433 0     SH434 0
SH435 246   SH436 0     SH437 0     SH438 97    SH439 0     SH440 0     SH441 0
SH442 0     SH443 0     SH444 175   SH445 463   SH446 0     SH447 459   SH448 0
SH449 210   SH450 0     SH451 455   SH452 0     SH453 471   SH454 0     SH455 1
SH456 430   SH457 253   SH458 219   SH459 265   SH460 0     SH461 0     SH462 182
SH463 0     SH464 403   SH465 270   SH466 441   SH467 126   SH468 141   SH469 0
SH470 379   SH471 394   SH472 292   SH473 194   SH474 282   SH475 0     SH476 417
SH477 423   SH478 354   SH479 488   SH480 0     SH481 0     SH482 112   SH483 435
SH484 0     SH485 0     SH486 0     SH487 83    SH488 0     SH489 441   SH490 537
SH491 184   SH492 39    SH493 0     SH494 65    SH495 136   SH496 0     SH497 450
SH498 239   SH499 104   SH500 425   SH501 421   SH502 0     SH503 0     SH504 0
SH505 0     SH506 0     SH507 99    SH508 416   SH509 0     SH510 0     SH511 398
SH512 30    SH513 327   SH514 397   SH515 0     SH516 425   SH517 0     SH518 0
SH519 0     SH520 123   SH521 232   SH522 54    SH523 135   SH524 97    SH525 152
SH526 0     SH527 259   SH528 0     SH529 0     SH530 0     SH531 0     SH532 475
SH533 0     SH534 406   SH535 441   SH536 32    SH537 280   SH538 0     SH539 0
SH540 170   SH541 4     SH542 0     SH543 332   SH544 0     SH545 0     SH546 276
SH547 0     SH548 0     SH549 61    SH550 0     SH551 0     SH552 181   SH553 0
SH554 0     SH555 0     SH556 423   SH557 470
```