MIE262 2025 Lab Project – WARP Shoes

Gabriel Faustino, Benjamin Zhou

1.0 Abstract

This report investigates a production plan for the WARP Shoe Company in February 2006, assuming a projected doubling in demand for all shoes resulting from the bankruptcy of a competitor. Given initial considerations and data, demand for shoes is estimated to develop a mathematical program with two decision variable classes, an objective function to maximize profit, and constraints constructed under various assumptions. The program is solved through AMPL, yielding an optimal profit of $11 837 500. Further sensitivity analysis of the program is conducted to investigate how changes to program parameters would affect the optimal solution and objective function value.

2.0 Introduction

The WARP Shoe Company, one of the oldest shoe companies in Canada, has consulted with University of Toronto (UofT) students over the past three years for assistance in increasing profit. At the beginning of 2006, one of WARP's main competitors went bankrupt, leading the company's market analysts to predict a doubling in all shoe demand during February 2006; two UofT industrial engineering students have been approached to determine the most profitable production plan. Conducive to this goal, a database containing data related to production, sales, and storage of shoes was provided. Additionally, a set of considerations has been provided:

- Closing inventory of January 2006 is assumed 0 for all shoes,
- All sales are assumed to happen at the end of the month,
- The budget for raw materials is $10 000 000,
- Sales prices remain the same as in the Product_Master table across all stores,
- Failure to meet demand costs $10/pair of shoes due to potential new customer loss,
- The WARP shoe production plant can run up to 12 hours a day, 28 days a month,
- Setup times and costs of all machines are negligible,
- Each machine requires one operator, and workers are paid $25/hour
- Transportation costs and manufacturing sequences can be ignored

A mathematical program is developed using this information and a set of associated assumptions. Section 3.0 presents these assumptions, defines sets, variables and parameters, and describes the formulation of the program. Section 4.0 details results from solving the program through AMPL. Section 5.0 continues by providing real-world interpretations of the optimal solution alongside analysis to answer the set of questions provided concerning changes to program parameters. Lastly, Section 6.0 provides a brief summary of the report.

3.0 Methodology

3.1 Assumptions

From Section 2.0, the initial set of considerations contains several statements prompting the examination of certain assumptions in order to feasibly construct a mathematical program. Listed below are all assumptions held true before formulation:

The production plant runs at maximum capacity for the full 12 hours a day over 28 days—in other words, total downtime across the month of February is 0. This means:

- All machines are able to operate simultaneously,
- All associated components (primarily machines) of the production plant do not require replacement at any point during production hours,
- The time to start producing a new pair of shoes after a prior pair is finished on each machine is negligible,
- Assuming there are several shifts for workers each day, the time it takes for operators to switch out/in is negligible,
- Workers can be paid for fractional hours,
- In between days, production can pick up exactly from where it left off with no time or cost penalty; this allows disregarding actual day-to-day production amounts.

Concerning setup times and costs, transportation costs, and manufacturing sequence:

- Setup of the production plant does not occur during production hours,
- Transportation of materials and products between warehouses, the production plant, and stores does not occur during production hours
- Setup and transportation costs are negligible
- Shoes can be made from the raw materials in any sequence

Concerning closing inventory of January 2006 and the sales period:

- Inventory at the start of February 2006 is zero; all shoes sold at the end of the month are manufactured during the month,
- All sales take place at the end of February 2006, meaning demand is entirely independent from any variables and outcomes associated with production.

Additionally, we note a quirk of writing in the database provided; the use of the term "shoes" is not explicitly defined as a pair of shoes, or an individual shoe. Hence, we assume that we are producing pairs of shoes—i.e. the numerical data provided for a machine refers to the production of both individual shoes.

3.2 Decision Variables and Parameters

With negligible factors excluded, the relevant decision variables and parameters are defined with respect to the following sets in Table 3.2.1:

Table 3.2.1. Relevant Sets

Set	AMPL Name	Definition	Database Table > Column
$i \in \{1, 2, \ldots, 557\}$	`SHOE`	Shoe models	Product_Master > Product_Num
$j \in \{1, 2, \ldots, 165\}$	`MATERIAL`	Raw materials	RM_Master > RM_Num
$k \in \{1, 2, \ldots, 72\}$	`MACHINE`	Machines	Machine_Master > Machine_Num
$l \in \{1, 2, \ldots, 8\}$	`WAREHOUSE`	Warehouses	Warehouse_Master > Warehouse_Num

The two classes of decision variables and nine fixed-value parameter classes are hence defined in Tables 3.2.2 and Table 3.2.3, respectively.

Table 3.2.2. Decision Variables

Variable	AMPL Name and index	Definition	Type
x_i	`shoes_produced{SHOE}`	Pairs of shoes of type `SHOE` produced	real
y_l	`warehouse_used{WAREHOUSE}`	If warehouse `WAREHOUSE` used	binary

Table 3.2.3. Fixed-Value Parameters

Parameter	AMPL Name and index	Definition (units)	Database Table > Column
p_i	`sale_price{SHOE}`	Sale price of shoe *i* ($/pair of shoes)	Product_Master > Sales_Price
d_i	`shoe_demand{SHOE}`	Projected demand of shoe *i* (pair of shoes)	Product_Num > Demand
q_{ij}	`material_quantity{SHOE, MATERIAL}`	Quantity of material *j* to produce shoe *i* (material unit/pair of shoes)	BOM > Quantity
c_j	`material_cost {MATERIAL}`	Cost of using material *j* ($/material unit)	RM_Master > Cost
m_k	`machine_cost{MACHINE}`	Operating cost for machine *k* ($/min)	Machine_Master > OpCost_per_min

Parameter	AMPL Name and index	Definition (units)	Database Table > Column
t_{ik}	`operation_time{SHOE, MACHINE}`	Operation time for machine *k* on shoe *i* (sec/pair of shoes)	Machine_Assign > Avg_Duration
z_l	`warehouse_cost{WAREHOUSE}`	Operation cost for warehouse *l* ($/warehouse)	Warehouse_Master > Op_Cost
s_j	`material_stock{MATERIAL}`	Stock of raw material *j* (material unit)	RM_Master > S_Quantity
w_l	`warehouse_capacity {WAREHOUSE}`	Capacity of warehouse *l* (pairs of shoes)	Warehouse_Master > Capacity

3.3 Mathematical Program Formulation

First, the objective function is constructed. The mathematical program should maximize profit (z), represented by:

$$maximize\ Profit\ (z)\ =\ Revenue\ -\ Costs$$

Revenue from the WARP shoes exclusively comprises sales of the shoes, while costs for the production of WARP shoes constitute five distinct sources; the components of the profit objective function are listed below in Table 3.3.1. Revenue components are added; cost components are subtracted.

Table 3.3.1. Objective Function Components

Component	Expression
Revenue from selling shoes	$\sum_{i=1}^{557} p_i x_i$
Total cost of raw materials needed to produce shoes	$\sum_{i=1}^{557}\sum_{j=1}^{165} c_j q_{ij} x_i$
Total cost of machine operation, based on time spent making shoes	$\sum_{i=1}^{557}\sum_{k=1}^{72} m_k(\frac{1}{60}t_{ik})x_i$
Total cost of labour, based on time spent making shoes	$\sum_{i=1}^{557}\sum_{k=1}^{72} 25(\frac{1}{3600}t_{ik})x_i$

Component	Expression
Total cost of warehouse operation	$\sum_{l=1}^{8} z_l y_l$
Total cost of unmet demand, where applicable	$\sum_{i=1}^{557} 10\ \max\{0, d_i - x_i\}$

Consolidating the revenue and cost components into the profit expression yields the following objective function:

$$max\, z\ =\ \sum_{i=1}^{557} p_i x_i - \sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij} c_j x_i - \sum_{i=1}^{557}\sum_{k=1}^{72} (\frac{1}{60}) m_k t_{ik} x_i - \sum_{i=1}^{557}\sum_{k=1}^{72} (\frac{1}{144}) t_{ik} x_i - \sum_{l=1}^{8} z_l y_l - \sum_{i=1}^{557} 10\ \max\{0, d_i - x_i\}$$

The production of shoes is next subject to certain constraints, listed below in Table 3.3.2. The bounding constraints for our decision variables x_i and y_l is also included.

Table 3.3.2. Constraint Formulation

Constraint	Expression	Notes
Raw material budget	$\sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij} c_j x_i \leq 10\ 000\ 000$	Budget limit of \$10 000 000 as specified
Raw material availability	$\sum_{i=1}^{557} q_{ij} x_i \leq s_j \quad \forall\, j\ =\ 1,\ldots, 165$	—
Machine operation time limit	$\sum_{i=1}^{557} t_{ik} x_i \leq 1\ 209\ 600 \quad \forall\, k\ =\ 1,\ldots, 72$	Express time limit in seconds/month: $(\frac{60s}{min})(\frac{60min}{h})(\frac{12h}{d})(28d)\ =\ 1\ 209\ 600\frac{s}{m}$
Warehouse capacity	$\sum_{i=1}^{557} x_i \leq \sum_{l=1}^{8} w_l y_l$	—
Bounds for shoe production	$x_i \in R, x_i \geq 0\ \ \forall\, i\ =\ 1,\ldots, 557$	Relaxed from initial integer bounds (i.e. $x_i \in Z, x_i \geq 0\ \ \forall\, i\ =\ 1,\ldots, 557$)
Bounds for warehouse use	$y_l \in \{0, 1\}\ \forall\, i\ =\ 1,\ldots, 8$	—

3.4 Mathematical Program

In sum, one objective function along with 4 constraints and two decision variable bound statements are identified. The complete mathematical program is as follows:

$$max\ z = \sum_{i=1}^{557} p_i x_i - \sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij} c_j x_i - \sum_{i=1}^{557}\sum_{k=1}^{72} (\frac{1}{60}) m_k t_{ik} x_i - \sum_{i=1}^{557}\sum_{k=1}^{72} (\frac{1}{144}) t_{ik} x_i - \sum_{l=1}^{8} z_l y_l - \sum_{i=1}^{557} 10\ \max\{0, d_i - x_i\}$$

$$s.t. \quad \sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij} c_j x_i \leq 10\,000\,000$$

$$\sum_{i=1}^{557} q_{ij} x_i \leq s_j \quad \forall\, j = 1,\ldots, 165$$

$$\sum_{i=1}^{557} t_{ik} x_i \leq 1\,209\,600 \quad \forall\, k = 1,\ldots, 72$$

$$\sum_{i=1}^{557} x_i \leq \sum_{l=1}^{8} w_l y_l$$

$$x_i \in R, x_i \geq 0 \ \forall\, i = 1,\ldots, 557$$

$$y_l \in \{0, 1\} \ \forall\, i = 1,\ldots, 8$$

The program was translated into AMPL (see Appendix A), constituting a .dat, .mod, and .run file. Details concerning the implementation can be found in the script comments; the files are included in the project submission. The AMPL implementation of the model was initially run without relaxation. This was not solved in a reasonable amount of time; therefore, the integer constraint on the shoe production variable, x_i, was relaxed to allow for production of fraction quantities of shoes. The binary warehouse constraint was not relaxed.

4.0 Results & Analysis

Optimizing the model in AMPL yields an optimal objective function value of $11 837 500. The optimal solution—specifying values of all of the decision variables—is included in the solution report in Appendix B1. Interpreting the results yields the following key pieces of information for WARP:

- Maximum profit for the month of February 2006 is determined to be $11 837 500.
- Each of the shoes_produced values in the solution report corresponds to the determined optimal quantity of pairs of shoes produced—leading to the $11 837 500 in profit. Many of the values do seem to match projected demand (see Figure 5.1.6. in Section 5.0 for sample values), corresponding to the concept of extremes in optimization. At the same time, many values are equal to 0; this implies regardless of the demand for certain shoe models, it is more profitable to forgo production of certain models entirely, instead shouldering the entirety of the unmet demand penalty—it may be best to discontinue production of these shoe models and repurpose associated production assets. Equivalently, the WARP company would save the costs of production and storage.

- Each of the warehouse_used values in the solution report corresponds to the determined optimal choice of either using (1) or not using (0) a warehouse to store shoes—and incurring the associated warehouse operating cost. Hence, the WARP company should only keep warehouses 2, 3, 4, 6, and 7 in operation for February 2006.

5.0 Discussion

In this section, Questions 1–7 of the assignment are investigated.

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

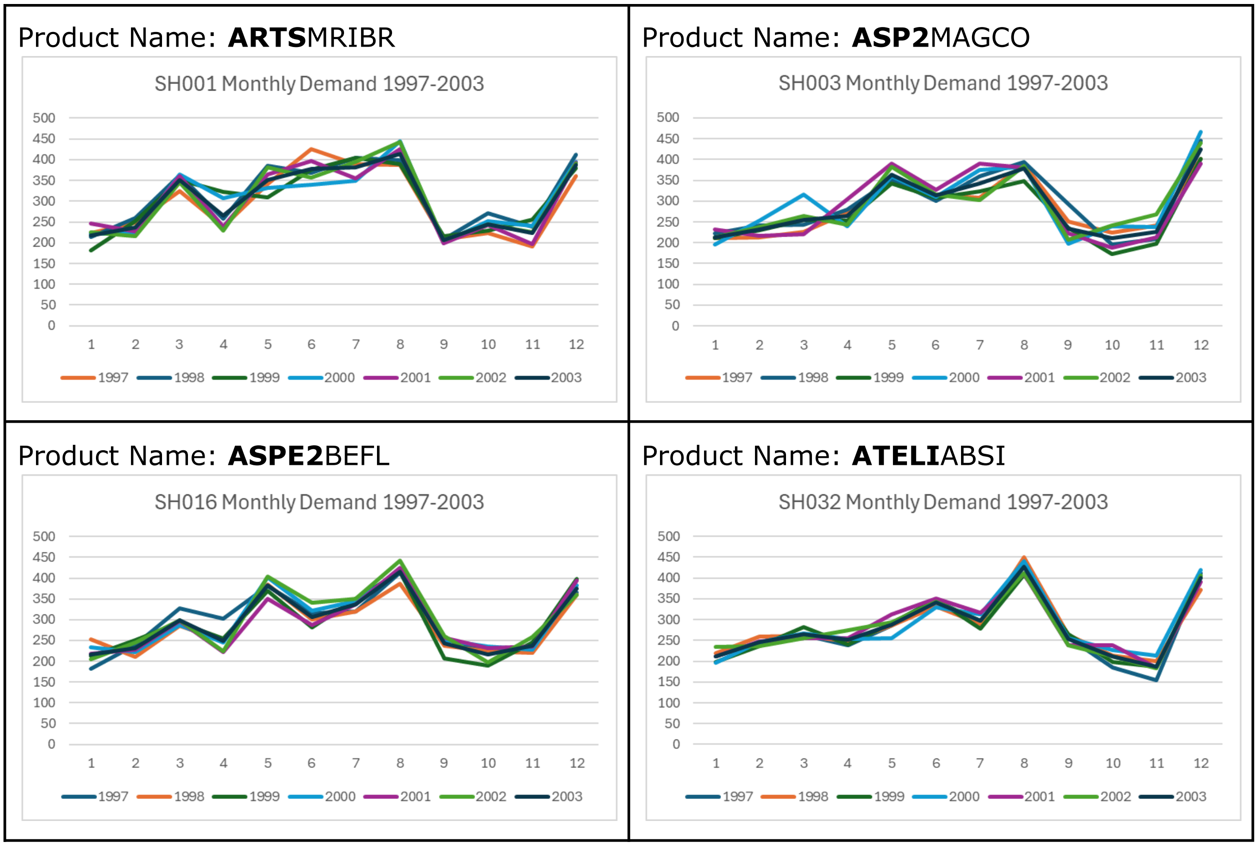
To estimate demand for the month of February, we must first identify the best method of estimating demand. Excel was used to manipulate the demand data and determined this method prior to the formulation of a mathematical program; the figures below illustrate this process.

To begin, the demand for each shoe was summed across all stores for each year and month:

Product_Num	Year	Month	Store_Num	Demand		Product_Num	Year	Month	Total Demand
SH001	1997	1	1	9		SH001	1997	1	223
SH001	1997	1	2	7		SH001	1997	2	248
SH001	1997	1	3	18		SH001	1997	3	324
SH001	1997	1	4	19		SH001	1997	4	238
SH001	1997	1	5	11		SH001	1997	5	342
SH001	1997	1	6	11		SH001	1997	6	425
SH001	1997	1	7	39		SH001	1997	7	389
SH001	1997	1	8	28		SH001	1997	8	388
SH001	1997	1	9	33		SH001	1997	9	209
SH001	1997	1	10	48		SH001	1997	10	224
SH001	1997	2	1	13		SH001	1997	11	191
SH001	1997	2	2	10		SH001	1997	12	360
SH001	1997	2	3	21		SH001	1998	1	213
SH001	1997	2	4	18		SH001	1998	2	259
SH001	1997	2	5	12		SH001	1998	3	362
SH001	1997	2	6	11		SH001	1998	4	258
SH001	1997	2	7	41		SH001	1998	5	385
SH001	1997	2	8	27		SH001	1998	6	369
SH001	1997	2	9	48		SH001	1998	7	404
SH001	1997	2	10	47		SH001	1998	8	398
SH001	1997	3	1	18		SH001	1998	9	209
SH001	1997	3	2	19		SH001	1998	10	271

Figure 5.1.1. Summed shoe demand across all 10 stores for each year and month, Excel

Next, the monthly demand by year was plotted for a sample of four distinct shoe models—selected by identifying distinct common prefix labels from the product names (Product_Name column in the Product_Master table) and choosing four arbitrarily. The graphs are shown below:



Figures 5.1.2. - 5.1.5. Monthly demand by year for four distinct shoe models

First and foremost, a trend for demand across the months of a year is identifiable. This means demand for a certain month across every year is not arbitrary, and there are reasonable grounds to apply an estimation technique on total monthly demand. Since we do not observe a monotonous increasing or decreasing trend for February, linear regression is determined unsuitable to estimate demand. Instead, an average will be taken across February for all years, more suitable to the presumed arbitrary nature of demand data: Averages of total monthly demand are taken from 1997–2003 for February, and when doubled, will be assumed sufficiently accurate estimates for 2006.

Product_Num	Year	Month	Store_Num	Demand
SH001	1997	1	1	9
SH001	1997	1	2	7
SH001	1997	1	3	18
SH001	1997	1	4	19
SH001	1997	1	5	11
SH001	1997	1	6	11
SH001	1997	1	7	39
SH001	1997	1	8	28
SH001	1997	1	9	33
SH001	1997	1	10	48
SH001	1997	2	1	13
SH001	1997	2	2	10
SH001	1997	2	3	21
SH001	1997	2	4	18
SH001	1997	2	5	12
SH001	1997	2	6	11
SH001	1997	2	7	41
SH001	1997	2	8	27
SH001	1997	2	9	48
SH001	1997	2	10	47
SH001	1997	3	1	18
SH001	1997	3	2	19

Product_Num	Year	Month	Total Demand
SH001	1997	1	223
SH001	1997	2	248
SH001	1997	3	324
SH001	1997	4	238
SH001	1997	5	342
SH001	1997	6	425
SH001	1997	7	389
SH001	1997	8	388
SH001	1997	9	209
SH001	1997	10	224
SH001	1997	11	191
SH001	1997	12	360
SH001	1998	1	213
SH001	1998	2	259
SH001	1998	3	362
SH001	1998	4	258
SH001	1998	5	385
SH001	1998	6	369
SH001	1998	7	404
SH001	1998	8	398
SH001	1998	9	209
SH001	1998	10	271

Product_Num	Feb. Total Demand	Historical Feb. Avg Demand	Projected Feb. 2006 Demand
SH001	1661	237.2857143	474.5714286
SH002	1571	224.4285714	448.8571429
SH003	1622	231.7142857	463.4285714
SH004	1475	210.7142857	421.4285714
SH005	1631	233	466
SH006	1493	213.2857143	426.5714286
SH007	1541	220.1428571	440.2857143
SH008	1519	217	434
SH009	1541	220.1428571	440.2857143
SH010	1384	197.7142857	395.4285714
SH011	1517	216.7142857	433.4285714
SH012	1519	217	434
SH013	1489	212.7142857	425.4285714
SH014	1549	221.2857143	442.5714286
SH015	1358	194	388
SH016	1627	232.4285714	464.8571429
SH017	1473	210.4285714	420.8571429
SH018	1797	256.7142857	513.4285714
SH019	1472	210.2857143	420.5714286
SH020	1494	213.4285714	426.8571429
SH021	1631	233	466
SH022	1566	223.7142857	447.4285714

Figure 5.1.6. Estimated shoe demand for February 2006 across all shoe models

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

Per the specified mathematical formulation in Section 3.2–3, there are:

- <u>565 total decision variables</u>, of which:
    - 557 specify shoe production (x_i, `shoe_production{SHOE}`), and
    - 8 specify warehouse usage (y_l, `warehouse_used{WAREHOUSE}`).
- <u>804 total constraints</u>, of which:
    - 239 are production constraints:
        - 1 constraint for the maximum raw materials budget,
        - 165 constraints for raw material stock (one for each material type),
        - 72 constraints for machine operation time limits (one for each machine, and
        - 1 constraint for warehouse capacity and shoe storage.
    - 565 are variable constraints:
        - 557 specify bounds and allowed values for shoe production (x_i, `shoe_production{SHOE}`), and
        - 8 specify bounds and allowed values for warehouse usage (y_l, `warehouse_used{WAREHOUSE}`).

Question 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?

Per Appendix B2, there are 50 constraints with negative slack values; that is, to satisfy the constraints, the left-hand side would have to be decreased (less production) or the right-hand side would have to be increased (more tolerance/resources). Under the current model however, they are technically impossible values. Therefore, there are 50 constraints violated; per Appendix B2, all of them are `MaterialAvailability` (raw material stock) constraints. This suggests that raw materials are the primary limitation on production and bottleneck for profit.

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

Generally, binding constraints are understood to have zero slack. Per Appendix B2, we note the constraints with zero slack all concern raw material availability. However, in the context of question 3, we further identify that raw material quantity constraints with negative slack values are also binding—although we allow violation of the constraints through relaxation, the raw materials are entirely consumed, meaning they in actuality limit the production of shoes and the associated constraints are binding. Additionally, no other types of constraints have negative slack values. Hence, there are 29 zero slack constraints, and 50 negative slack constraints, amounting to a total of 79 binding constraints, all corresponding to the available quantity of raw materials.

Question 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?

Warehouse capacity is not binding (per Appendix B2, `WarehouseCapacity.slack = 4413`.). That is, under the current conditions of the formulation, there is extra warehouse storage space that cannot be filled in any further—there are simply not enough shoes produced. Assuming that no other parts of the formulation are changed, the extra space bought will go unused. Given that each space for shoe boxes costs a positive amount at $10/space, buying additional space is not economical.

Question 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?

Changing the machine operation limit to 8 hours per day results in a loss of 4 hours per day. Converting to seconds per month gives:

$$Decrease\ in\ operation\ limit\ =\ (\frac{60\ s}{min})(\frac{60\ min}{h})(\frac{4\ h}{d})(\frac{28\ d}{m})\ =\ 403\ 200\frac{s}{m}\ lost$$

For the optimal solution and objective value to change, the slack must be less than 403 200 seconds/month, which would cause the associated constraint to become binding. However, none of the machine operation time limit slacks (`MachineOperationTimeLimit.slack`) are less than 403200 (the closest is for machine #63, which has a reported slack of 432015 seconds; per Appendix B2.). Therefore, assuming nothing else in the formulation is changed, the operation time has not been decreased enough such as to affect the optimal production plan; the binding constraints do not change.

Question 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 again.

Adding an additional $7 000 000 would change the raw material budget constraint:

$$\sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij}c_j x_i \leq 10\ 000\ 000 \Rightarrow \sum_{i=1}^{557}\sum_{j=1}^{165} q_{ij}c_j x_i \leq 17\ 000\ 000$$

Adding $7 000 000 to the raw material budget may increase its slack depending on if the budget constraint is binding. However, it is not, and has plenty of slack (`MaterialBudget.slack = 5394230`, see Appendix B2.); that is, the initial $10 000 000 is not used completely. Adding $7 000 000 would just add more slack, and not change the optimal production solution.

6.0 Conclusion

As requested by the WARP Shoes Company, an optimal production plan was investigated through the creation of a mathematical program, ultimately determining a maximum achievable profit of $11 837 500 alongside several key insights concerning shoe production and storage. The program adheres to constraints on raw material budget and availability, as well as machine operation time and warehouse storage capacity. Examination of binding constraints—which actively limit profitability—and sensitivity analysis on non-binding constraints provide further direction on potential reallocation or adjustment of assets and resources for February 2006. Overall, consideration of the optimization of production and resources by WARP Shoes is recommended in order to meet the changes in demand within current production and resource constraints, while also improving profit, enabling the company to further solidify its position in the shoe manufacturing market.

7.0 Appendices

Appendix A: AMPL source files

A1. Basic model file (.mod)

WarpShoes.mod:

```
### Constraint parameters ###
param material_stock{MATERIAL};            # sj  ; Stock of raw material j (material unit)
param warehouse_capacity{WAREHOUSE};       # wl  ; Capacity of warehouse l (pair of shoes)
### Decision variables ###
#var shoes_produced{SHOE} integer >= 0;     # xi ; Quantity of shoe i to produce
var warehouse_used{WAREHOUSE} binary;      # yl ; 1 if warehouse l is used, 0 otherwise
### Relaxed decision variables ###
# (uncomment and comment corresponding constrained decision variable to use)
var shoes_produced{SHOE} >= 0;             # Relax shoe to allow for fractional production
#var warehouse_used{WAREHOUSE} >= 0, <= 1    # Relax warehouse to allow for fractional use
### Objective function ###
# Function of profit = revenue - costs
# => want to maximize profit
maximize Profit:                           # Profit =
   sum {i in SHOE}                        #   revenue from shoes
       sale_price[i]                      #   ...
     * shoes_produced[i]                  #   ...
 - sum {i in SHOE, j in MATERIAL}         # - cost from raw material used
       material_quantity[i,j]             #   ...
     * material_cost[j]                   #   ...
     * shoes_produced[i]                  #   ...
 - sum {i in SHOE, k in MACHINE}          # - cost from operating machines
       (1/60)                             #   <conversion from seconds to minutes>
     * machine_cost[k]                    #   ...
     * operation_time[i,k]                #   ...
     * shoes_produced[i]                  #   ...
 - sum {i in SHOE, k in MACHINE}          # - cost from operator wages
       (1/144)                            #   <$25/1hr * 1hr/3600s = 25/3600 = 1/144>
     * machine_cost[k]                    #   ...
```

```
      * operation_time[i,k]                    #    ...
      * shoes_produced[i]                      #    ...
 - sum {l in WAREHOUSE}                        # - cost from operating warehouses
        warehouse_cost[l]                      #    ...
      * warehouse_used[l]                      #    ...
 - sum {i in SHOE}                             # - cost from missing demand penalty
        10                                     #   <$10 penalty for missed demand>
      * max(0, shoe_demand[i] - shoes_produced[i]); # <do not apply penalty if demand is met>
 - 10 *
### Constraints ###
subject to MaterialBudget:                      # Material budget
   sum {i in SHOE, j in MATERIAL}              # ...
        material_quantity[i,j]                 # ...
      * material_cost[j]                       # ...
      * shoes_produced[i]                      # ...
   <= 10000000;                                # <material costs <= $10000000>
subject to MaterialAvailability {j in MATERIAL}: # Material availability (across all 165 materials)
   sum {i in SHOE}                             # ...
        material_quantity[i,j]                 # ...
      * shoes_produced[i]                      # ...
   <= material_stock[j];                       # ...
subject to MachineOperationTimeLimit {k in MACHINE}: # Machine operation time limit over the month (across all
72 machines)
   sum {i in SHOE}                             # ...
        operation_time[i,k]                    # ...
      * shoes_produced[i]                      # ...
    <= 1209600;                                    # <60 sec/1 min * 60 min/1 hr * 12 hr/1 day * 28 days = 1209600
sec>
subject to WarehouseCapacity:                   # Warehouse capacity (total across all 8 warehouses)
   sum {i in SHOE}                             # Rewrite inequality to have all terms on LHS and 0 on RHS
        shoes_produced[i]                      # ...
 - sum {l in WAREHOUSE}                        # ...
        warehouse_capacity[l]                  # ...
      * warehouse_used[l]                      # ...
   <= 0;                                       # ...
```

A2. Data file (.dat)

WarpShoes.dat:

```
table Products IN "ODBC" ".\WARP2011W.mdb" "Product_Master":
   SHOE <- [Product_Num], sale_price ~ Sales_Price;
read table Products;
table Demand IN "ODBC" ".\WARP2011W.mdb" "SQL=SELECT Product_Num, 20*(AVG(Demand)) \
   AS [Projected_Feb_Demand] \
   FROM Product_Demand \
   WHERE Month=2 \
   GROUP BY Product_Num \
   ORDER BY Product_Num":
   [Product_Num], shoe_demand ~ Projected_Feb_Demand;
read table Demand;
table RawMaterials IN "ODBC" ".\WARP2011W.mdb" "RM_Master":
   MATERIAL <- [RM_Num], material_cost ~ Cost, material_stock ~ S_Quantity;
read table RawMaterials;
table BOM IN "ODBC" ".\WARP2011W.mdb" "BOM":
   [Product_Num, RM_Num], material_quantity ~ Quantity;
read table BOM;
table Machines IN "ODBC" ".\WARP2011W.mdb" "Machine_Master":
   MACHINE <- [Machine_Num], machine_cost ~ OpCost_per_min;
read table Machines;
table MachineAssign IN "ODBC" ".\WARP2011W.mdb" "Machine_Assign":
   [Product_Num, Machine_Num], operation_time ~ Avg_Duration;
read table MachineAssign;
table Warehouses IN "ODBC" ".\WARP2011W.mdb" "Warehouse_Master":
   WAREHOUSE <- [Warehouse_Num], warehouse_capacity ~ Capacity, warehouse_cost ~ Op_Cost;
read table Warehouses;
```

A3. Run file (.run)

WarpShoes.run:

```
reset;                                       # Reset environment
model WarpShoes.mod;                         # Load model
data WarpShoes.dat;                          # Load data
option solver gurobi;                        # Set solver to Gurobi
solve;                                       # Solve model

# Display profit value and vars solution to "WarpShoes.out"
display Profit, shoes_produced, warehouse_used > WarpShoes.out;

close WarpShoes.out;                         # Close WarpShoes.out
```

A4. Run file add-ins

WarpShoes.run:

```
# Round solutions (add-in)
for {i in SHOE} {
   let shoes_produced[i] := round(shoes_produced[i]);
}
# Display slack (add-in)
display MaterialBudget.slack > WarpShoes.out;
display MaterialAvailability.slack > WarpShoes.out;
display MachineOperationTimeLimit.slack > WarpShoes.out;
display WarehouseCapacity.slack > WarpShoes.out;
```

Appendix B: AMPL/Gurobi Output

B1. Basic solution report

Console output:

```
Gurobi 12.0.3: Gurobi 12.0.3: optimal solution; objective 11838914.93
2504 simplex iterations
1 branching node
absmipgap=1.86265e-09, relmipgap=1.57332e-16
```

WarpShoes.out (decision values formatted in two columns):

```
Profit = 11837500
:     shoes_produced warehouse_used    :=

1              .              0              SH029      375              .
2              .              1              SH030        0              .
3              .              1              SH031      309              .
4              .              1              SH032      491              .
5              .              0              SH033        0              .
6              .              1              SH034      373              .
7              .              1              SH035        0              .
8              .              0              SH036        0              .
SH001          0              .              SH037        0              .
SH002        449              .              SH038      479              .
SH003          0              .              SH039      389              .
SH004          0              .              SH040      433              .
SH005         33              .              SH041      355              .
SH006          0              .              SH042      214              .
SH007          0              .              SH043        0              .
SH008          0              .              SH044      442              .
SH009         35              .              SH045      305              .
SH010          0              .              SH046        0              .
SH011        433              .              SH047      400              .
SH012        348              .              SH048      271              .
SH013        425              .              SH049        0              .
SH014          0              .              SH050      437              .
SH015          0              .              SH051        0              .
SH016          0              .              SH052        0              .
SH017          0              .              SH053        0              .
SH018        307              .              SH054        0              .
SH019        255              .              SH055        0              .
SH020          0              .              SH056        0              .
SH021          0              .              SH057      152              .
SH022          0              .              SH058      432              .
SH023          0              .              SH059        0              .
SH024        400              .              SH060      237              .
SH025        401              .              SH061      191              .
SH026          0              .              SH062      458              .
SH027        434              .              SH063      446              .
SH028          0              .              SH064        7              .
```

SH065	293	.	SH114	0	.
SH066	0	.	SH115	0	.
SH067	0	.	SH116	214	.
SH068	0	.	SH117	129	.
SH069	0	.	SH118	260	.
SH070	0	.	SH119	132	.
SH071	422	.	SH120	0	.
SH072	0	.	SH121	508	.
SH073	250	.	SH122	0	.
SH074	0	.	SH123	0	.
SH075	0	.	SH124	0	.
SH076	166	.	SH125	0	.
SH077	336	.	SH126	41	.
SH078	407	.	SH127	358	.
SH079	9	.	SH128	0	.
SH080	409	.	SH129	0	.
SH081	175	.	SH130	58	.
SH082	212	.	SH131	0	.
SH083	55	.	SH132	0	.
SH084	0	.	SH133	92	.
SH085	0	.	SH134	490	.
SH086	0	.	SH135	0	.
SH087	430	.	SH136	0	.
SH088	423	.	SH137	0	.
SH089	0	.	SH138	0	.
SH090	475	.	SH139	0	.
SH091	0	.	SH140	0	.
SH092	0	.	SH141	0	.
SH093	236	.	SH142	399	.
SH094	361	.	SH143	0	.
SH095	422	.	SH144	449	.
SH096	453	.	SH145	0	.
SH097	0	.	SH146	489	.
SH098	0	.	SH147	23	.
SH099	0	.	SH148	0	.
SH100	0	.	SH149	166	.
SH101	0	.	SH150	425	.
SH102	0	.	SH151	0	.
SH103	0	.	SH152	0	.
SH104	0	.	SH153	157	.
SH105	418	.	SH154	437	.
SH106	144	.	SH155	0	.
SH107	0	.	SH156	0	.
SH108	0	.	SH157	0	.
SH109	0	.	SH158	0	.
SH110	417	.	SH159	0	.
SH111	0	.	SH160	0	.
SH112	0	.	SH161	0	.
SH113	454	.	SH162	0	.

SH163	0	.
SH164	0	.
SH165	460	.
SH166	0	.
SH167	0	.
SH168	320	.
SH169	0	.
SH170	0	.
SH171	0	.
SH172	196	.
SH173	210	.
SH174	0	.
SH175	0	.
SH176	450	.
SH177	0	.
SH178	206	.
SH179	184	.
SH180	0	.
SH181	0	.
SH182	0	.
SH183	0	.
SH184	74	.
SH185	293	.
SH186	187	.
SH187	116	.
SH188	262	.
SH189	0	.
SH190	0	.
SH191	84	.
SH192	0	.
SH193	0	.
SH194	227	.
SH195	219	.
SH196	0	.
SH197	437	.
SH198	0	.
SH199	0	.
SH200	0	.
SH201	476	.
SH202	0	.
SH203	501	.
SH204	0	.
SH205	263	.
SH206	0	.
SH207	17	.
SH208	0	.
SH209	437	.
SH210	5	.
SH211	0	.

SH212	0	.
SH213	437	.
SH214	0	.
SH215	0	.
SH216	0	.
SH217	249	.
SH218	0	.
SH219	0	.
SH220	0	.
SH221	468	.
SH222	0	.
SH223	409	.
SH224	443	.
SH225	327	.
SH226	419	.
SH227	265	.
SH228	0	.
SH229	0	.
SH230	115	.
SH231	0	.
SH232	384	.
SH233	449	.
SH234	405	.
SH235	471	.
SH236	497	.
SH237	0	.
SH238	451	.
SH239	28	.
SH240	136	.
SH241	405	.
SH242	0	.
SH243	0	.
SH244	285	.
SH245	0	.
SH246	0	.
SH247	422	.
SH248	129	.
SH249	405	.
SH250	0	.
SH251	0	.
SH252	0	.
SH253	381	.
SH254	438	.
SH255	210	.
SH256	434	.
SH257	0	.
SH258	0	.
SH259	0	.
SH260	0	.

SH261	409	.	SH310	0	.
SH262	411	.	SH311	0	.
SH263	0	.	SH312	477	.
SH264	0	.	SH313	483	.
SH265	393	.	SH314	391	.
SH266	0	.	SH315	0	.
SH267	0	.	SH316	238	.
SH268	0	.	SH317	292	.
SH269	429	.	SH318	0	.
SH270	94	.	SH319	0	.
SH271	0	.	SH320	462	.
SH272	0	.	SH321	0	.
SH273	284	.	SH322	124	.
SH274	42	.	SH323	0	.
SH275	0	.	SH324	0	.
SH276	256	.	SH325	475	.
SH277	423	.	SH326	147	.
SH278	0	.	SH327	0	.
SH279	77	.	SH328	203	.
SH280	0	.	SH329	400	.
SH281	324	.	SH330	0	.
SH282	415	.	SH331	66	.
SH283	0	.	SH332	0	.
SH284	460	.	SH333	0	.
SH285	0	.	SH334	387	.
SH286	451	.	SH335	0	.
SH287	410	.	SH336	0	.
SH288	171	.	SH337	0	.
SH289	430	.	SH338	176	.
SH290	75	.	SH339	0	.
SH291	0	.	SH340	428	.
SH292	148	.	SH341	429	.
SH293	0	.	SH342	332	.
SH294	0	.	SH343	0	.
SH295	403	.	SH344	0	.
SH296	0	.	SH345	379	.
SH297	0	.	SH346	0	.
SH298	50	.	SH347	231	.
SH299	388	.	SH348	489	.
SH300	427	.	SH349	40	.
SH301	0	.	SH350	0	.
SH302	0	.	SH351	0	.
SH303	0	.	SH352	397	.
SH304	0	.	SH353	0	.
SH305	0	.	SH354	0	.
SH306	0	.	SH355	0	.
SH307	182	.	SH356	186	.
SH308	319	.	SH357	0	.
SH309	0	.	SH358	0	.

SH359	0	.	SH408	0	.
SH360	0	.	SH409	0	.
SH361	325	.	SH410	319	.
SH362	419	.	SH411	409	.
SH363	0	.	SH412	95	.
SH364	0	.	SH413	0	.
SH365	352	.	SH414	0	.
SH366	153	.	SH415	438	.
SH367	424	.	SH416	0	.
SH368	0	.	SH417	482	.
SH369	378	.	SH418	345	.
SH370	0	.	SH419	192	.
SH371	0	.	SH420	58	.
SH372	332	.	SH421	417	.
SH373	417	.	SH422	0	.
SH374	0	.	SH423	466	.
SH375	0	.	SH424	0	.
SH376	0	.	SH425	459	.
SH377	109	.	SH426	0	.
SH378	423	.	SH427	477	.
SH379	232	.	SH428	0	.
SH380	0	.	SH429	157	.
SH381	95	.	SH430	430	.
SH382	0	.	SH431	0	.
SH383	277	.	SH432	9	.
SH384	505	.	SH433	0	.
SH385	0	.	SH434	0	.
SH386	0	.	SH435	93	.
SH387	0	.	SH436	0	.
SH388	0	.	SH437	0	.
SH389	0	.	SH438	68	.
SH390	0	.	SH439	0	.
SH391	87	.	SH440	0	.
SH392	0	.	SH441	0	.
SH393	0	.	SH442	0	.
SH394	0	.	SH443	0	.
SH395	348	.	SH444	105	.
SH396	0	.	SH445	463	.
SH397	0	.	SH446	0	.
SH398	0	.	SH447	426	.
SH399	443	.	SH448	0	.
SH400	0	.	SH449	314	.
SH401	0	.	SH450	0	.
SH402	181	.	SH451	387	.
SH403	0	.	SH452	0	.
SH404	0	.	SH453	471	.
SH405	0	.	SH454	0	.
SH406	0	.	SH455	22	.
SH407	437	.	SH456	430	.

SH457	227	.	SH506	0	.
SH458	230	.	SH507	77	.
SH459	231	.	SH508	427	.
SH460	0	.	SH509	0	.
SH461	0	.	SH510	0	.
SH462	156	.	SH511	398	.
SH463	0	.	SH512	120	.
SH464	481	.	SH513	272	.
SH465	282	.	SH514	397	.
SH466	441	.	SH515	0	.
SH467	218	.	SH516	425	.
SH468	14	.	SH517	0	.
SH469	21	.	SH518	0	.
SH470	306	.	SH519	0	.
SH471	421	.	SH520	122	.
SH472	238	.	SH521	248	.
SH473	357	.	SH522	0	.
SH474	261	.	SH523	50	.
SH475	0	.	SH524	92	.
SH476	417	.	SH525	191	.
SH477	322	.	SH526	0	.
SH478	353	.	SH527	138	.
SH479	488	.	SH528	0	.
SH480	0	.	SH529	0	.
SH481	0	.	SH530	0	.
SH482	97	.	SH531	0	.
SH483	435	.	SH532	475	.
SH484	0	.	SH533	0	.
SH485	0	.	SH534	406	.
SH486	0	.	SH535	441	.
SH487	18	.	SH536	59	.
SH488	0	.	SH537	135	.
SH489	441	.	SH538	0	.
SH490	537	.	SH539	0	.
SH491	303	.	SH540	37	.
SH492	114	.	SH541	0	.
SH493	0	.	SH542	0	.
SH494	65	.	SH543	259	.
SH495	0	.	SH544	0	.
SH496	0	.	SH545	0	.
SH497	450	.	SH546	219	.
SH498	252	.	SH547	53	.
SH499	66	.	SH548	0	.
SH500	425	.	SH549	46	.
SH501	421	.	SH550	0	.
SH502	0	.	SH551	0	.
SH503	0	.	SH552	134	.
SH504	0	.	SH553	17	.
SH505	0	.	SH554	0	.

```
SH555         0              .             SH557       470              .
SH556       423              .             ;
```

B2. Slack output

Printed using the slack .run file add-in; see Appendix A4.

WarpShoes.out:

```
MaterialBudget.slack = 5394230
MaterialAvailability.slack [*] :=
 1  0    20 -2    39  2    58  1    77  0     96 -2    115 -3    134 -1    153 -1
 2  5    21  7    40  3    59  2    78  4     97 -1    116  0    135 -2    154 -1
 3 13    22  6    41  2    60  2    79  1     98  2    117  0    136  3    155  1
 4 -4    23  3    42 -2    61 -5    80  0     99  1    118  0    137  1    156  0
 5  0    24  5    43  1    62  4    81  0    100  5    119  2    138  1    157  3
 6  3    25  1    44  0    63  7    82 -2    101  0    120 -3    139 -1    158  1
 7 -1    26 -5    45 -3    64 -1    83 -3    102  0    121  0    140  2    159  0
 8 -5    27  7    46  0    65  4    84 -4    103 -2    122 -6    141  0    160  0
 9  3    28  2    47 -1    66  1    85 -2    104 -3    123  5    142 -1    161  0
10 -1    29  0    48  1    67  2    86  3    105  2    124  1    143  1    162  2
11 -5    30 -1    49  2    68  3    87  0    106 -1    125  3    144  2    163 -1
12  0    31  2    50  3    69  4    88  1    107  3    126  5    145  1    164  7
13 -8    32 -2    51  2    70  3    89 -1    108 -2    127 -1    146  1    165  0
14  8    33  1    52 -1    71  4    90  5    109 -4    128  2    147 -3
15  2    34 -3    53  4    72  0    91  1    110  6    129  0    148  4
16 -4    35 -4    54 -1    73 -4    92 -2    111  3    130  3    149 -4
17  6    36  1    55  1    74  0    93  0    112  2    131  2    150  1
18  3    37  1    56  0    75  3    94 -1    113  2    132 -1    151  0
19  6    38  4    57 -1    76  1    95  0    114  1    133  2    152  2
;
MachineOperationTimeLimit.slack [*] :=
1 1135990   13 1014750   25 1166190   37 1093700   49  696235   61  822822
2  884544   14 1037800   26 1161540   38 1092210   50  851572   62  662038
3  895625   15  991600   27 1141610   39 1156470   51 1157990   63  432015
4 1003460   16 1091170   28 1161000   40 1183320   52 1071150   64  658576
5  780404   17 1102320   29 1166040   41  793446   53 1040820   65  787684
6  849584   18 1082760   30 1186770   42  754920   54 1023660   66  638808
7  881510   19 1091520   31 1173270   43  737871   55  995078   67  684753
8  919830   20 1092540   32 1105080   44  710634   56 1108870   68  473395
9  869492   21 1187670   33 1131030   45  887358   57 1084900   69  576320
10 1040100   22 1186000   34 1134850   46  723354   58 1094610   70  778390
11 1106780   23 1160910   35 1109870   47  859997   59 1112770   71  873091
12 1044830   24 1176410   36 1122170   48  592161   60 1113330   72  885624
;
WarehouseCapacity.slack = 4413
```