An Algorithm to Select the Optimal Finishing Weight of Swine to Maximize Profit per Day

A Master's Paper in Computer Science
By
Steven Bielski

Submitted in Partial Fulfillment Of the Requirements For the Degree of
Master of Science

May 2002
This paper gives an algorithm to maximize profit per day for a swine finishing facility. The algorithm presented in this paper differs from its peers by focusing on profit as a function that includes time, measured in days, as a variable, while most contemporary algorithms focus on maximizing profit while assuming a fixed time interval. There are two major parts to this report: first the investigation and formulation of the problem as a mathematical model, and second the implementation of this algorithm into a user-friendly web application.

The creation of the mathematical model involves first a collection and interpretation of data, and then the arrangement of this data into a logical algorithm to return the optimal weight at which a pig should be sold, so that profit, per unit of time, is maximized.

The implementation of this data is in the format of a web application where the users have flexibility in defining variables and formulas, and have the ability to share specific results among their peers. There is also an application that matches the format of existing software to calculate this information, that format being Microsoft Excel Spreadsheets.
# Table of Contents

TABLE OF CONTENTS .................................................................................................................. 1  

ACKNOWLEDGEMENTS ............................................................................................................... 3  

STATEMENT OF THE PROBLEM ............................................................................................... 4  
  Overview ...................................................................................................................................... 4  
  Existing Counterparts .................................................................................................................. 5  
  Example Problem ....................................................................................................................... 5  

ASSUMPTIONS ............................................................................................................................... 6  
  Overview ...................................................................................................................................... 6  
  General Assumptions .................................................................................................................. 6  
  Algorithm Assumptions ............................................................................................................ 6  
  Implementation Assumptions ..................................................................................................... 6  

MATHEMATICAL MODEL ............................................................................................................. 7  
  Overview ...................................................................................................................................... 7  
  Input Variable Definitions .......................................................................................................... 7  
  Assumed Constants for the Algorithm ...................................................................................... 7  
  Other Variables .......................................................................................................................... 8  
  Algorithm OptimalWeight ......................................................................................................... 8  
  Function ProfitPerDay .............................................................................................................. 8  
  Function TotalRevenues ............................................................................................................ 9  
  Function TotalExpenses ............................................................................................................ 9  
  Function PreFinishingRangeExpenses ................................................................................... 9  
  Function PostFinishingRangeExpenses .................................................................................. 9  
  Function FinishingRangeExpenses ........................................................................................... 10  
  Conversion Ratios ..................................................................................................................... 10  
  Function CumulativeFeedConsumed [2] .................................................................................. 11  
  Function FeedConsumed ......................................................................................................... 11  
  Function DaysInFinishingRange ............................................................................................... 12  
  Function CycleTime .................................................................................................................. 13  
  Function ProfitPerDay in Terms of the Input Variables ............................................................ 13  

FLEXIBILITY OF THE MODEL .................................................................................................... 14  
  Extensive Variables ................................................................................................................... 14  
  Reducible Variables .................................................................................................................. 14  
  Recursively Dynamic Calls ...................................................................................................... 14  

VALIDATION OF THE MODEL ................................................................................................... 15  
  Overview ...................................................................................................................................... 15  
  Test 1 – Reasonable Data Favoring a Maximal Selling Weight ................................................ 15  
  Test 2 – Reasonable Data Favoring a Light Selling Weight ....................................................... 16  
  Conclusion ................................................................................................................................... 16
IMPLEMENTATION.........................................................................................................................17
  Overview ................................................................................................................................. 17
  Default.asp Screenshot ........................................................................................................ 18
  GetData.asp Page Description .............................................................................................. 19
  GetData.asp Screenshot ........................................................................................................ 20
  CalcPPD.asp Page .................................................................................................................... 21
  CalcPPD.asp Screenshot........................................................................................................ 22
  SaveFile.asp ........................................................................................................................... 22
  FileManager.asp ....................................................................................................................... 23
  FileManager.asp Start Page Screenshot ................................................................................ 24
  FileManager.asp View All Screenshot .................................................................................... 24

CONCLUSIONS ..........................................................................................................................25
  Overview .................................................................................................................................. 25
  Implementation ....................................................................................................................... 25
  Future Work ............................................................................................................................ 25

REFERENCES: .........................................................................................................................26
Acknowledgements

I would like to thank my friends and family who supported me not only for this paper, but also throughout my entire academic career.

I would like to thank the following people from the agriculture departments at various colleges who helped me: Rick Jones, Bob Goodband, Gilbert Hollis, Don Mahan, David H. Baker, and especially Ken Kephart.

I would like to thank the helpful people from Hatfield Meats, including Justina Powell.

I would like to thank the committee that is reviewing this document, including Dr. Linda Null and Dr. Pavel Naumov.

I would like to thank my academic advisor Dr. Thang Bui.

In addition, I would especially like to thank my project advisor, Dr. Jeff Hartzler, without whose help this paper would not exist.
Statement of the Problem

Overview

Pigs are raised for profit in numerous facilities. The most common method of pig farming is a situation as follows:

1. Pigs are purchased at weights between 30 and 50 pounds.

2. A pig farmer raises them until they reach a weight of at least 220 pounds and usually up to 280 pounds. In this interval, they are sold to a packing plant where they are turned into delicious pork products. Although pigs can be sold outside of this interval, the selling price is discounted significantly.

3. While pigs are raised by the farmer, they incur several costs, which can be summarized as follows:
   
   a. The cost of buying the pig while in the 30 – 50 pound interval.
   b. The cost of feeding the pig.
   c. Daily medicine and veterinarian costs.
   d. Overhead costs, such as for the facilities, labor, and transportation.

4. There are also known constants for each cycle, including:
   
   e. The number of days it will take a pig to reach the minimal finishing weight.
   f. The number of days of downtime between pig raising cycles.
   g. The feed conversion value of the feed for the pigs.
   h. The daily average growth rates of the pigs while in the finishing range.

   The question that we are asked to solve is to find an algorithm that can efficiently and correctly determine the optimal selling weight of a pig, dependent upon the input values of these known variables. We are also asked to develop an intuitive interface to implement our algorithm.
Existing Counterparts

Currently there is software, Microsoft Excel Spreadsheets, available that can determine profit as a function of the input variables [3]. This Excel Spreadsheet, however, lacks the ability to search every possible selling weight effectively, and it does not consider profit to be a function of time, but rather considers profit a simple dollar amount.

Example Problem

Let us consider the following example problem:

Let us say a farmer can make $18 per pig by selling them at 280 lbs or $14 per pig by selling at 220 lbs. However, it takes 136 days for the pigs to reach 280 lbs, but only 100 days for the pigs to reach 220 lbs.

Therefore, if we were to calculate the profit achieved per day if we sold the pig at 280 lbs we would get $18 per pig / 136 days = $.132 per pig per day. If we were to calculate, however, the profit achieved per day if we sold the pig at 220 lbs we would get: $14 per pig / 100 days = $.14 per pig per day.

This implies that maximum profit per day achievable in this example is attainable by selling the swine at the minimal finishing range value, 220 lbs. Since the problem we are asked to solve is to create an algorithm that will determine the weight at which the maximum profit per day is achieved, in this case, our algorithm should return 220 lbs as the optimal selling weight.
Assumptions

Overview

We make several assumptions to solve the problem in this section. Some things we must assume for both our algorithm and our implementation, but some may be specific for only our algorithm, or only the implementation.

General Assumptions

1. We must assume that specific pieces of information exist at the time of execution of the algorithm, and that they are known. Specifically, we need to know the cost of feed, the number of days until a pig reaches 220 pounds, the total cost it takes to raise a pig to reach 220 pounds, the daily overhead costs, the selling price that we will receive, the daily cost of medicines, and the number of days that we will spend in downtime. Therefore, this leads to our next assumption.
2. We assume that the algorithm will be run when the pig first enters the finishing range.
3. We assume that specific trends involving feed conversion and swine growth patterns are known. These formulas do not have to be constant for the finishing range, however. If this information is not known, standard default values will be substituted in their place.

Algorithm Assumptions

For our algorithm, we assume that all variables are both known and constant for the duration of the growth cycle. For example, the price of feed will be the same when the pigs enter the finish range as when they exit the finishing range, and the selling price is known at the beginning of the finishing range.

Implementation Assumptions

For our implementation, we will try to make a program that can be used by as wide a range of our target audience as is possible. Since our target audience is pig farmers, who use a variety of computing platforms, a web page format proves to be the most accessible way of creating an implementation. Some users however, can consider this data very secure, so a stand-alone version is also available. We will assume that the user of this program has access to all of the above-described variables when using this program, although for some values, standard defaults should also present themselves as options to the user.
Mathematical Model

Overview

We will define an algorithm OptimalWeight that takes as input all of the variables defined in the problem statement. This algorithm will determine the maximal profit attainable per day and return the weight at which this profit occurs.

As a convention for this paper, we will define functions as a series of words without spaces and the first letter of each word capitalized. We will define constants and variables to be an abbreviation of a series of words, or the first letter of each word, in capitalized form.

It should be noted that all functions in this model are new creations for this algorithm, with the exception of CumulativeFeedConsumed [2].

Input Variable Definitions

a. C220 = Cost of raising a pig to reach 220 pounds, the lightest selling weight, including all overhead costs.
b. D220 = Number of days a pig is held before it enters the finishing range.
c. CF = The cost per pound of pig feed.
d. OC = Overhead costs per day per pig for the finishing facility.
e. MC = The medication costs, per day per pig.
f. SP = The selling price of the swine.
g. DIB = Days between growing cycles required for facility sanitation
h. AI = An array of weight intervals, each having a distinct feed conversion rate and/or a distinct pig growth rate.
i. FCR = An array of feed conversion rates
j. PGR = An array of pig growth rates

Assumed Constants for the Algorithm

a. 220 lbs = The beginning weight of the selling interval
b. 280 lbs = The end weight of the selling interval
Other Variables

a. \( W = \) selling weight of a pig
b. \( W^* = \) optimal selling weight of a pig, maximizing profit per day.

Since we assume the pigs are sold in the finishing range, both \( W \) and \( W^* \) are constrained by \( 220 \leq W, W^* \leq 280 \)

Algorithm OptimalWeight

The algorithm that we are asked to define takes the following parameters:

Input: \( C_{220}, D_{220}, CF, OC, MC, SP, DIB, AI, FCR, PGR \)

Output: \( W^*, 220 \leq W^* \leq 280 \)

We will formally define our algorithm as OptimalWeight:

\[
W^* = \text{OptimalWeight}(C_{220}, CF, OC, MC, SP, DIB, AI, FCR, PGR)
\]

where \( 220 \leq W^* \leq 280 \)

Since \( W^* \) is in a finite range, and represents the weight at which we achieve maximum profit per day, we can also define OptimalWeight in terms of a derivative function ProfitPerDay.

\[
\text{OptimalWeight} (C_{220}, D_{220}, CF, OC, MC, SP, DIB, AI, FCR, PGR) = \max \text{ProfitPerDay} (W, C_{220}, D_{220}, CF, OC, MC, SP, DIB, AI, FCR, PGR) \quad 220 \leq W \leq 280
\]

where we define Max as a simple function that returns the highest value of ProfitPerDay through the range of \( W \)

Function ProfitPerDay

Now what we need to do is to determine the profit per day gained for the set of input variables and a given weight. ProfitPerDay can be concisely defined as the total revenue gained, minus the total expenses accrued, divided by the number of days required to achieve these gains and losses. This means that we are required to find and define three functions: TotalRevenues, TotalExpenses, and CycleTime.

\[
\text{ProfitPerDay} = \frac{\text{TotalRevenues} - \text{TotalExpenses}}{\text{CycleTime}}
\]
Function TotalRevenues

TotalRevenues is the simplest function to define, since we will only receive one source of revenue, and the amount of that revenue is well defined. When a pig is sold, we receive one fixed rate amount per pound, so TotalRevenues will be an increasing function with respect to selling weight. Function TotalRevenues returns the selling weight $W$ multiplied by the selling price $SP$, and it is measured in dollars.

$$\text{TotalRevenues} = W \times SP$$

Function TotalExpenses

TotalExpenses is a more complicated function. First, TotalExpenses can be broken down into three periods: expenses accrued before the pig enters the finishing range, expenses accrued while the pig is in the finishing range, and expenses accrued after the pig has left the finishing range, while the finishing facility is experiencing downtime. We will therefore define three more functions: PreFinishingRangeExpenses, FinishingRangeExpenses, and PostFinishingRangeExpenses.

$$\text{TotalExpenses} = \text{PreFinishingRangeExpenses} + \text{FinishingRangeExpenses} + \text{PostFinishingRangeExpenses}$$

Function PreFinishingRangeExpenses

PreFinishingRangeExpenses can be defined as $C220$, by definition.

$$\text{PreFinishingRangeExpenses} = C220$$

Function PostFinishingRangeExpenses

Post finishing range expenses do not involve any maintenance of swine, but rather the upkeep costs of the facility. They can be defined as the daily overhead costs multiplied by the number of days of downtime.

$$\text{PostFinishingRangeExpenses} = OC \times DIB$$
Function FinishingRangeExpenses

This only leaves the expenses accrued while the pig is in the finishing range. The major complication is that some of these expenses are variable depending on a swine's weight, or the time a swine spends in the finishing range. More precisely, we need to know how much food a pig will consume while in the finishing range, since the total cost of food consumed depends directly on it. We will define a function here that returns the amount of food eaten in the finishing range, FeedConsumed. We will also need to know how many days a pig will stay in the finishing range, since the overhead and medication costs depend on days in the facility. We will define a function DaysInFinishingRange to yield that information.

\[
\text{FinishingRangeExpenses} = (\text{FeedConsumed} (W) \times CF) + (\text{DaysInFinishingRange} (W) \times OC)
\]

The costs associated with feeding a pig can be considered the cost of feed multiplied by the amount of feed necessary to raise a pig from one weight to another. Specifically, we need to know the amount of feed a pig will consume to gain from 220 pounds to \(W\), the selling weight.

Conversion Ratios

When determining how much feed a pig will have consumed at any weight, we must consider the conversion ratio of the feed and swine in a given interval. A conversion ratio is a value that determines how many pounds of feed are required for one pound of swine growth. After research and consultation with various leaders in academic universities, it was discovered that different feeds have different feed conversion ratios associated with them, over entire large intervals of weights associated with pigs [2]. Most conversion ratios are values between 2.5 and 3.1. The conversion ratio is not a constant value, however. When a pig is young and does not weigh much, the conversion ratio for that swine is much lower than when that pig is older and larger. This is because when a pig is larger, it must use more energy to maintain body temperature. The conversion ratio does not increase linearly with weight either. This means that we must somehow approximate the increase of the conversion ratio with weight, and incorporate this into a function to tell us how much feed a pig has consumed at a given weight.

During this research, we discovered two different approaches to this problem, and we will discuss them both. We will demonstrate that the results from both of these methods are comparable, although the results they produce are not identical. We will only include one approach into the final algorithm.
Function CumulativeFeedConsumed [2]

After consulting with Dr. Ken Kephart of Penn State University, and Dr. Rick Jones of the University of Georgia, we were directed to a publication and accompanying table by Kentucky State University [2]. It defined the following formula as approximating the cumulative feed consumed by a swine at any given weight \( W \) and a feed conversion ratio \( CR \). It is notable that with this formula there is only one conversion rate for the entire finishing cycle, so this conversion ratio can be considered an average for the interval. Although there is no mathematical way to prove its general correctness, this formula is based on regressing actual pig consumption data, and is currently used in practice. In this paper we will call this function \( \text{CumulativeFeedConsumed} (W) \), or the Cumulative Feed Consumed at a given weight.

\[
\text{CumulativeFeedConsumed} (W, CR) = \\
(4 \, CR / 11) \times (0.0042 \, W^2 + 1.5394 \, W - 12.899)
\]

After reviewing this formula and the validity of the output it created, we decided that the results were not likely to be entirely correct for the general swine population, particularly when we are only considering them in the finishing range. Therefore, we will chose to use our formula defined below, \( \text{FeedConsumed} (W) \).

Function FeedConsumed

In this approach, we will let conversion ratio values can be described to follow a stepwise function, a linear approximation to the curve of the conversion ratio verses weight. The value of the feed conversion function over the finishing range can be described to have the following values:

General Feed Conversion Rates, or FCR, for swine at various weights within the finishing range can be defined as the following:

\[
\begin{align*}
\text{FCR}(W) = & \quad \{ 2.5 \, \text{lbs feed / lb gain} \} \quad 220 \, \text{lbs} \leq W \leq 240 \, \text{lbs} \\
& \quad \{ 2.75 \, \text{lbs feed / lb gain} \} \quad 240 \, \text{lbs} < W \leq 260 \, \text{lbs} \\
& \quad \{ 3.0 \, \text{lbs feed / lb gain} \} \quad 260 \, \text{lbs} < W \leq 280 \, \text{lbs}
\end{align*}
\]

With this formula, and following the fact that the conversion ratio is the amount of feed consumed divided by the weight of the pig, we can generate the amount of feed a pig will consume while in the finishing range as:
FeedConsumed \( (W) = \)

\[
\begin{align*}
\{ (W-220) \times (2.50) & \quad 220 \text{ lbs} < W < 240 \text{ lbs} \\
\{ (W-240) \times (2.75) + (20 \times 2.50) & \quad 240 \text{ lbs} < W < 260 \text{ lbs} \\
\{ (W-260) \times (3.0) + (20 \times 2.75) + (20 \times 2.50) & \quad 260 \text{ lbs} < W < 280 \text{ lbs}
\end{align*}
\]

We have shown that these results are approximately correct by the definition of a conversion ratio. Specific feed conversion ratios can vary for differing types of feed, and the amount of fat contained in each type of feed.

In addition, these results can be generalized to accept arrays of intervals, \( W \), and array of Feed Conversion Rates, FCR, and that is how they are implemented.

**Function DaysInFinishingRange**

Next, we need to find a formula to relate a pig's weight to its age. Graphs of this function are widely known and follow a general pattern, but the amount of variance between these graphs actually proves to be very large. However, if we observe the graph more closely we can see that during the finishing range the growth rate follows very constant patterns. This pattern can be further broken down into three separate linear models. According to Dr. Ken Kephart, of Pennsylvania State University [3], pig growth versus time rates while in the finishing range can be defined as follows:

- 1.75 lbs/day from 200 – 240 lbs.
- 1.675 lbs/day from 220 – 240 lbs.
- 1.6 lbs/day from 240 – 280 lbs.

From this information, we can likewise break down the time spent in the finishing range as a function separated into three discrete parts, where we can determine the time spent in the finishing range as a function of the weight while in the finishing range. Since these numbers follow a stepwise pattern in nature, we can therefore create a similar function as in the function FeedConsumed.

DaysInFinishingRange \( (W) = \)

\[
\begin{align*}
\{ (W - 220) / 1.75 & \quad 220 \text{ lbs} \leq W \leq 240 \text{ lbs} \\
\{ ((W - 240) / 1.675) + (20 / 1.75) & \quad 240 \text{ lbs} \leq W \leq 260 \text{ lbs} \\
\{ ((W - 260) / 1.6) + (20 / 1.75) + (20 / 1.675) & \quad 260 \text{ lbs} \leq W \leq 280 \text{ lbs}
\end{align*}
\]
For the implementation of this algorithm, the formula is again generalized to accept an array AI for weight intervals and an array PGR of pig growth rates.

Now that we have defined all of the costs associated with raising a pig, all that is left is to determine the number of days in a cycle so that we can divide by that number to achieve profit per day.

**Function CycleTime**

We can use the same DaysInFinishingRange (W) function as described above to determine the number of days a swine spends in the finishing range, and together with the number of days before a pig enters the finishing range and the downtime required after the pig exists the finishing range, we can determine the total time required for a cycle.

\[
\text{CycleTime} = D_{220} + \text{DaysInFinishingRange}(W) + DIB
\]

**Function ProfitPerDay in Terms of the Input Variables**

We can now define our function ProfitPerDay in terms of the given variables.

\[
\text{PPD}(W, C_{220}, D_{220}, CF, OC, MC, SP, DIB, CR) = \\
\frac{(W \times SP) \cdot \left[ C_{220} + (OC \times DIB) + (CF \times \{CFC(W, CR) - CFC(220, CR)\}) + OC \times DFR(W) \right]}{D_{220} + DFR(W) + DIB}
\]

With these formulas now defined, we can go back and apply the original algorithm, and determine the maximum profit per day attainable.
Flexibility of the Model

Extensive Variables

The variables defined in this algorithm are extensive, meaning they provide opportunities for any daily, one time, or interval of costs to be added into the equation correctly. If there is an additional one-time expense per pig, this value can be added to C220 to maintain the integrity of the formula. An additional daily cost can easily be placed in OC, and any other cost of any other duration can be created and compensated for as well.

Reducible Variables

In addition, if the users do not know with certainty the values of any of the parameters, they can easily vary that parameter to do a sensitivity analysis of it, and determine an appropriate value for it. The main variables of concern here are the growth rates and feed conversion rates. With this algorithm users can find predicted feed consumption, and compare that to their own predictions to create values for these variables.

Recursively Dynamic Calls

If, during the finishing range, values of variables change, the user may call the algorithm again with new variable values and different bounds on the finishing range. This allows a user to reconsider whether they should continue to raise the swine based on the new data, or sell them as soon as possibly.

Users may also wish to call the program multiple times to tighten the bounds on their answer. Since the user inputs growth rate values and cumulative feed values based on interval averages, if the algorithm tells the user that the optimal weight is in a specific interval, the user can run the algorithm again, breaking this one interval into an array of subintervals.

This means the algorithm is expandable, so with multiple calls a user can gain increased functionality and accuracy from what is origionally provided. The multiple calls can work together, so that each successive call can provide more detailed information, not just different information of the same caliber.
Validation of the Model

Overview

To determine whether the results of this formula are reasonable or not we can first input some known sample data and see if the output is reasonable. In addition, by changing the value of some variables to various reasonable extremes we should be able to determine whether it is most profitable to sell a pig at the beginning of the finishing range or at the end of the finishing range.

Test 1 – Reasonable Data Favoring a Maximal Selling Weight

First, let us compute a generic example and define whether the results produced are reasonable or unreasonable. We will be aided in this process by an Excel spreadsheet implementation [1]. First, we will define some sample values for the input variables:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Variable Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>C220</td>
<td>100</td>
</tr>
<tr>
<td>D220</td>
<td>100</td>
</tr>
<tr>
<td>CF</td>
<td>0.1</td>
</tr>
<tr>
<td>OC</td>
<td>0.03</td>
</tr>
<tr>
<td>MC</td>
<td>0.01</td>
</tr>
<tr>
<td>SP</td>
<td>0.5</td>
</tr>
<tr>
<td>DIB</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W</th>
<th>FeedConsumed (W, FE)</th>
<th>DaysInFinishingRange (W)</th>
<th>ProfitPerDay (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>220</td>
<td>0</td>
<td>0</td>
<td>0.09037037</td>
</tr>
<tr>
<td>225</td>
<td>12.5</td>
<td>2.857142857</td>
<td>0.098286082</td>
</tr>
<tr>
<td>230</td>
<td>25</td>
<td>5.714285714</td>
<td>0.10580402</td>
</tr>
<tr>
<td>235</td>
<td>37.5</td>
<td>8.571428571</td>
<td>0.112953431</td>
</tr>
<tr>
<td>240</td>
<td>50</td>
<td>11.42857143</td>
<td>0.119760766</td>
</tr>
<tr>
<td>245</td>
<td>63.75</td>
<td>14.41364606</td>
<td>0.125055128</td>
</tr>
<tr>
<td>250</td>
<td>77.5</td>
<td>17.39872068</td>
<td>0.130097429</td>
</tr>
<tr>
<td>255</td>
<td>91.25</td>
<td>20.38379531</td>
<td>0.134905251</td>
</tr>
<tr>
<td>260</td>
<td>105</td>
<td>23.36886994</td>
<td>0.139494579</td>
</tr>
<tr>
<td>265</td>
<td>120</td>
<td>26.49386994</td>
<td>0.142759259</td>
</tr>
<tr>
<td>270</td>
<td>135</td>
<td>29.61886994</td>
<td>0.145875673</td>
</tr>
<tr>
<td>275</td>
<td>150</td>
<td>32.74386994</td>
<td>0.148853696</td>
</tr>
<tr>
<td>280</td>
<td>165</td>
<td>35.86886994</td>
<td>0.151702347</td>
</tr>
</tbody>
</table>

Given these variables, we return a maximum profit of $0.15 per day per pig, achieved at the weight of 280 pounds, or the maximum value at the finishing range. This expected profit is reasonable and the weight that returns the maximum expected profit is the weight that is generally accepted. We also notice that the
profit that is achieved at the minimum weight, 220 pounds, is only $0.09 cents a day.

**Test 2 – Reasonable Data Favoring a Light Selling Weight**

Next, we will alter our data to create a condition where it would be more profitable in the real world to sell the swine as soon as they enter the finishing range. Due to the decrease in feed efficiency as the pig gains more weight, this situation occurs when feed is more expensive. Therefore, our next example will consider the case where all the above variables are the same except for the cost of feed, CF, which will be raised to $0.20 per pound.

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Variable Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>C220</td>
<td>100</td>
</tr>
<tr>
<td>D220</td>
<td>100</td>
</tr>
<tr>
<td>CF</td>
<td>0.2</td>
</tr>
<tr>
<td>OC</td>
<td>0.03</td>
</tr>
<tr>
<td>MC</td>
<td>0.01</td>
</tr>
<tr>
<td>SP</td>
<td>0.5</td>
</tr>
<tr>
<td>DIB</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W</th>
<th>FeedConsumed (W, FE)</th>
<th>DaysInFinishingRange (W)</th>
<th>ProfitPerDay (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>220</td>
<td>0</td>
<td>0</td>
<td>0.09037037</td>
</tr>
<tr>
<td>225</td>
<td>12.5</td>
<td>2.857142857</td>
<td>0.087010309</td>
</tr>
<tr>
<td>230</td>
<td>25</td>
<td>5.714285714</td>
<td>0.083819095</td>
</tr>
<tr>
<td>235</td>
<td>37.5</td>
<td>8.571428571</td>
<td>0.080784314</td>
</tr>
<tr>
<td>240</td>
<td>50</td>
<td>11.42857143</td>
<td>0.077894737</td>
</tr>
<tr>
<td>245</td>
<td>63.75</td>
<td>14.41364606</td>
<td>0.072977601</td>
</tr>
<tr>
<td>250</td>
<td>77.5</td>
<td>17.39872068</td>
<td>0.068294566</td>
</tr>
<tr>
<td>255</td>
<td>91.25</td>
<td>20.38379531</td>
<td>0.063829303</td>
</tr>
<tr>
<td>260</td>
<td>105</td>
<td>23.36886994</td>
<td>0.059566967</td>
</tr>
<tr>
<td>265</td>
<td>120</td>
<td>26.49386994</td>
<td>0.053535862</td>
</tr>
<tr>
<td>270</td>
<td>135</td>
<td>29.61886994</td>
<td>0.04777866</td>
</tr>
<tr>
<td>275</td>
<td>150</td>
<td>32.74386994</td>
<td>0.042277118</td>
</tr>
<tr>
<td>280</td>
<td>165</td>
<td>35.86886994</td>
<td>0.037014576</td>
</tr>
</tbody>
</table>

It can easily be realized that the same profit per day will be achieved at the lightest selling weight, $0.09, but the profit achieved at the maximum selling weight is reduced to less the $0.04 cents per day, thus telling us to sell the swine at the lightest selling weight.

**Conclusion**

With these results, we can therefore conclude that our algorithm is intuitively correct, given the accuracy of our sample data.
Implementation

Overview

In order to implement this algorithm in an easily usable manner that can be accessed by a very geographically dispersed but connected community, that being commercial pig farmers, we decided to implement this algorithm as a web application. The application is based on Active Server Pages. These pages can be placed on any server designed to handle Active Server Pages, including Windows servers and most types of UNIX servers, assuming they have been configured to handle Active Server Pages. The client interface only requires web access and simple JavaScript 1.2 abilities. Therefore, it should be compatible with Netscape versions 4 and above and Microsoft Internet Explorer versions 4 and above. Users have the ability to share their results with one another; however, the results are only shared within the particular implementation on a server. Therefore, if two distinct servers were running the web application, they would not be able to share results between the two.

Running the application can be extremely straightforward, but if a user needs more flexibility, there are some powerful options as well. First, a user would travel to the home page, Default.asp, and then start a new session by clicking the link to GetData.asp. On this page is a form that can be filled out with the required information. After the user completes this form and submits it, the server then sends back another page with the variables inserted specifically into various JavaScript functions, which compute and display the optimal profit per day. Since the actual computation is performed on the client’s browser, the server load is dramatically reduced. After the users view their results, they have the option of publishing them so that other swine professionals can see how various factors affect this process from around the world.

An implementation of this web application is freely available at the following URL: http://www30.brinkster.com/pfinishr. Screenshots of the application are taken from this server.
Welcome To Optimized Pig Finishing Weight Selection Algorithm!

This project is meant to create an interactive estimator to determine the optimal selling weight of a pen of swine based on several variables which are inputted by the user.

This algorithm and site were created to receive credit towards a Masters Degree in Computer Science at Penn State Harrisburg.

To begin you may:

- Read the supporting research or
- Try out the Excel Spreadsheet or
- Create a new Session or
- View a Pre-Existing Session.

Many Thanks go to Dr. Hartzler and Dr. Kephart Enjoy My website!

Figure 1:

This is an image of the starting page for the web application of the algorithm OptimalWeight.
GetData.asp Page Description

After a user first visits the home page, Default.asp, GetData.asp is the first step to using the algorithm. The user simply has to fill in the required information to fulfill the input for the algorithm as follows:

1. CBFR, the cost of raising a pig to reach the minimal value of the finishing range.
2. DBFR, the number of days a pig must be held before it reaches the minimal value of the finishing range.
3. CF, the cost of feed in dollars per pound.
4. OC, the daily overhead costs in daily cost per pig.
5. DIB, the number of days between swine cycles.
6. MC, the daily medicine costs.
7. SAR, which stands for Show All Results, is a checkbox to answer whether they would like results, or profit per day, shown for each weight, or just the optimal results.

If the user wants to customize the form further, the user can input all of the following fields:

8. There is a table accepting up to five weight intervals, and for each interval, the user can specify a conversion ratio, and a growth rate for the swine.

If a user is looking up an already existing profile, then each piece of information is located in the appropriate field already. This means that the page first looks for arguments on the server, and if they exist then they are assigned to Active Server Page variables. Each of these variables is then written to the HTML code at the server. If the variables are not found then each field is left blank, unless a suggested default value is available i.e., in the weight interval table.
Figure 2:

This is an image of the data collection page of the web application of the algorithm OptimalWeight.
CalcPPD.asp Page

This page calculates the results and displays the result. It is separated into several steps. First extensive error checking is done to verify that all of the required variables are assigned. If any required variable is missing or filled in incorrectly, then a descriptive error page is displayed instead of the results of the form.

If all of the appropriate fields are correctly filled in, then a results page is generated. First, the variables are written back to the screen, thus making the page a complete overview of a profile.

Next, the results are generated and displayed. As in the previous page, all of the variables are generated and inserted into the HTML code as ASP variables. All of the actual calculations, however, are done by JavaScript functions. This removes the load of calculating the results from the server, and minimizes the risk of excessive server load caused by inappropriately large interval bounds.

If the user wishes to view all of the results, then each weight and the profit per day are displayed in a table. If the user is only interested in the results, then only the optimal weight and profit per day are displayed.

After this information, there is a form asking the users if they would like to publish their work, as a reference for other users of the program in the future, and to track changes in variables used. If the users select this option, then they furnish their name, a password, email, a session name, the type of pig and the geographic area. The output of the program is submitted automatically. When the user submits the information, it goes to the SaveFile.asp page, which saves the data in a Microsoft Access database.
Figure 3:
This is an image of the display results page of the web application of the algorithm OptimalWeight.
SaveFile.asp

This form accepts valid output from a Profit Per Day application. It also requires profile information from the user, as described above in the CalcPPD.asp section above. It inserts the information into an Access database using Active Server Page database protocols.

FileManager.asp

This is an alternate way to start an application of the algorithm. A user arrives at this page, either to edit his or her sessions or to view all sessions published by all users. There are three main sub pages to FileManager.asp. When a user first comes to FileManager.asp, the start page is the first place the user will arrive.

Here users can log in by entering their name and password, and go the session manager page. From this page, they view all sessions that they have created, seeing the session name they assigned to each profile. They can open any session, and these variables are sent to the GetData.asp page, where the built in reclamation will fill in the form with all of the variables previously specified.

From this screen, a user can also delete a session that he or she has created. If a user wishes to modify a session, he or she can open it, make any necessary changes and save it again under a different session name.

From the start page, a user can also choose an option to view all of the sessions in the database instead. This does not require any type of login, and provides users access to view any existing profiles. The major limitation here is the user does not have the ability to delete any sessions, and although a user can modify and save the session, he or she must first save it under his or her own name.
FileManager.asp Start Page Screenshot

Figure 4:
This is an image of the starting page of the file manager of the web application of the algorithm OptimalWeight.

FileManager.asp View All Screenshot

Figure 5:
This is an image of the file manager page, set to view all entries, for the web application of the algorithm OptimalWeight.
Conclusions

Overview
In this master’s paper, we have demonstrated that there is an algorithm that can determine the optimal selling weight to maximize profit per day. In addition, we have shown there are cases where following the results produced by this algorithm can produce much higher profits than if we attempt to maximize profit without considering time as a factor. Therefore, while it is a generally accepted practice to raise pigs until they reach the maximum weight, regardless of the time it takes to accomplish this task, this paper cites that the rationale for doing so can often be diminished when profit is considered a factor of time. Therefore, we have shown that the algorithm created in this paper can often produce better results that contemporary algorithms that do not consider time a factor.

Implementation
The implementation provides a clear way for people to use the algorithm defined in this document, and demonstrate to their peers the results that they have achieved using this algorithm. It varies from existing spreadsheets to calculate profit in that this program finds the optimal selling weight, instead of accepting the selling weight as an input variable. It also considers profit as a function of time, a feature not available in contemporary software.

Future Work
The implementation of the algorithm OptimalWeight is meant to be used openly by swine industry professionals. In the coming future we hope to publish this document in a trade periodical to attract attention, and acceptance, in the industry.

After this accomplishment, it may be use useful to attempt to include other types of livestock in similar terms. Since there are extensive similarities between swine farming and raising other animals, it naturally follows that a similar algorithm could be effective for these animals as well. Other animals, however, may include other variables that are not addressed here, so application of OptimalWeight should be investigated and researched before applied commercially to livestock other than swine.
References:

[1] Bielski, Steve “Standalone Optimized Finishing Weight Selector”

“Growing-Finishing Pig Recommendations,” Kansas State University
Agriculture Experiment Station and Cooperative Extension Service,
October 1997

[3] Kephart, Ken and Cooper, Joan “Feeder_Pig_Bdgt_AF.xls” Penn State
College of Agriculture, University Park, PA