COST OPTIMIZATION OF BLEND PREPARATION WITH THE USE OF THE FLEXSIM ENVIRONMENT

Ireneusz Kaczmar*

Institute of Mechatronics at the East European State Higher School in Przemyśl
*Contact details: ul. Żołnierzy I Armii Wojska Polskiego 1E, 37-700 Przemyśl; e-mail: irek286@gmail.com

ARTICLE INFO

ABSTRACT

This work presents a new approach to solving the problem of optimization of selection of the blend components quantity. The problem of blending refers to provision of the required amounts of nutrients, at minimal cost of purchasing the products needed for production of a compound feed. The problem of diet is essential for humans, animals and agricultural production. In the presented case the linear programming method in the Flexsim environment was applied. Firstly a simulation model was built that shows the technology of preparing the mixture, while maintaining the minimum required amounts of nutrients. Secondly, the optimizer tool was used to determine the structure of the nutrients, so that the cost of purchase was the cheapest. The results are presented in a numerical and graphical form.

Introduction

Within 1930-1945 George Stigler considered the problem of determination of an adequate diet for an individual at a minimal cost. Very quickly, it became clear that a feasible method for solving linear programming was needed. In the same time a concept of duality was developed by John von Neumann, the mathematical thread uniting linear programming and the game theory. Then in 1951 George Dantzig developed the simplex method, which was very boring and time-consuming. The actual techniques of linear programming have been successfully applied in the petroleum industry, food processing, steel industry and more (Thie and Keough, 2011). In the 50's and 60's, the interest in the Operational Research was still growing and, due to its application in the field of commerce and industry. Over the time new tools of simulation and optimization were developed. FlexSim was released in 2003 as general purpose software, it is especially suitable for production and logistic tasks. It offers a graphical user an interface with a 2D or 3D view of the model and several standard library objects that can be included into the model by drag & drop.

Product blending is a difficult problem, and has a very high impact on the costs of all manufacturing processes (Dantzig, 1998). We are focusing on the single blend optimization. The main objective is to find the least expensive combination of components, which satisfies all the diet specifications and is available in sufficient amounts in order to produce...
the desired product quantity. On the basis of this example we will be able to build more complex blending models. Solution is the most classic problem of operation research. We should look for solutions to this type of problem in the methods of linear programming. Many cases are described in the world’s literature (Diwekar, 2008; Rardin, 1998; Sen & Higle, 1999). Graphical methods may be used to solve the presented problem, but these are inaccurate. We can also use simpler software for example: Excel sheet with a built-in optimizer Solver. But Excel does not allow visualization of production, there is no possibility of introducing the dynamics of the process, such as: machining time, frequency of providing nutrients, statistical distributions, etc. Flexsim environment can do it all. But the most important step is the idea of how to develop a model that correctly maps the real production process, as shown in this example. So the work is an important addition to the existing information gap in this area.

A linear programming problem may be defined as the problem of maximizing or minimizing a linear function subject to linear constraints. The constraints may be equalities or inequalities. In general, the issue of the optimal composition of the mixture, the decision maker wants to determine how many resources you need to purchase to complete the mix. The decision maker should get a mix of desired chemical composition, at the lowest cost to purchase raw materials. The standard minimum problem is to find a m-vector, \( y=(y_1,\ldots,y_m) \), to minimize (Ferguson and Sargent, 1958):

\[
Y^T b = y_1b_1 + \ldots + y_mb_m ;
\]

subject to the constraints:

\[
\begin{align*}
    y_1a_{11} + y_2a_{21} + \ldots + y_ma_{m1} & \geq c_1 \\
    y_1a_{12} + y_2a_{22} + \ldots + y_ma_{m2} & \geq c_2 \\
    \vdots & \quad \vdots \\
    y_1a_{1n} + y_2a_{2n} + \ldots + y_ma_{mn} & \geq c_n
\end{align*}
\]

(or \( y^T A \geq c^T \));

and \( y_1 \geq 0, y_2 \geq 0, \ldots, y_m \geq 0, \) (or \( y \geq 0 \));

In this case, we have to prepare a mixture of nutrients. The mixture should be composed of various nutrients (e.g. proteins, fats, vitamins, minerals, etc.), in different amounts. We assume that we have \((n)\) different types of food, which contain nutrients \((r)\). Parameters in this task are:

- \( a_{ij} \) – amount of \( i\)-th nutrient, contained in one unit \( j\)-th types of food;
- \( b_i \) – min. amount \( i\)-th of nutrient, which has to be delivered for an organism contained in one unit of a defined type of food;
- \( c_j \) – price per unit for a defined type of food;

In some situations, additional requirements are provided for example, to vary the diet:

- \( d_j \) – minimum number of the \( j\)-th type of food that should be consumed;
- \( g_j \) – maximum number of the \( j\)-th type of food that the body can receive;

The task consists in determination of the amount of purchase of various types of food that will ensure the proper development of the body. In addition, the cost of purchasing food should be as low as possible. Decision variables in this case are the numbers of productsto
Cost optimization...

be purchased: \( x_j \) – amount of defined types of food, where \( j = 1, 2, \ldots, n \) types of food. If we want to find an answer to this question, we need to solve the following optimization tasks:

\[
\begin{align*}
 a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \geq b_1; \\
 \vdots & \\
 a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & \geq b_m;
\end{align*}
\]

\( d_j \leq x_j \leq g_j \) for some \( j \);

\[ x_1, \ldots, x_n \geq 0; \]

\[ c_1x_1 + c_2x_2 + \ldots + c_nx_n \rightarrow \min. \]

In the next section the above equation will be solved using a simulation model, which shows the implementation of the manufacturing process. For the execution of the experiment FlexSim ver. 7.3.6 software has been used, with OptQuest optimizer on board.

**Description of the problem and solution**

Let us take the following diet example on the farm (Kukula, 1999). In the real case the farmer has to prepare a feed mixture for chickens. Unfortunately on the market there are only two types of food products to produce this formula. The feed mixture consists of only two products QA and QB. A compound with which chickens are fed must be provided with required nutrients \( (N_1, N_2, N_3) \) in a defined minimum amount. The content of nutrients in the unit of defined product, prices of products, as well as minimum amounts of ingredients are given in Table 1.

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Content of nutrients in the one unit of defined product</th>
<th>Min. amount of nutrient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product A ( Q_A )</td>
<td>Product B ( Q_B )</td>
</tr>
<tr>
<td>( N_1 )</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>( N_2 )</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>( N_3 )</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Price ($)</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Source: Kukula (1999)

A farmer needs to buy each product \( (Q_A, Q_B) \) in proper amounts, to provide chickens with nutrients required for normal development. A minimal number of nutrients \( (N_1, N_2, N_3) \) to ensure normal development is presented in Table 1. Unfortunately, a farmer has to blend those products \( (Q_A, Q_B) \), but he does not know how many products he has to buy, so that the overall cost of purchasing is the lowest. To determine the structure of purchase of \( Q_A \) and \( Q_B \) products, first we have to build a mathematical model. In this case the following decision variables were introduced:

\( x_1 \) – quantity of \( Q_A \) product to purchasing;

\( x_2 \) – quantity of \( Q_B \) product to purchasing;
We also have the following constraints:

1. The total quantity of the nutrient $N_1$ in the products $Q_A$ and $Q_B$ should be larger or equal to 27 units: $3x_1 + 9x_2 \geq 27$;

2. The total quantity of the nutrient $N_2$ in the products $Q_A$ and $Q_B$ should be larger or equal to 32 units: $8x_1 + 4x_2 \geq 32$;

3. The total quantity of the nutrient $N_3$ in the products $Q_A$ and $Q_B$ should be larger or equal to 36 units: $12x_1 + 3x_2 \geq 36$;

4. Addition the decision variables $x_1, x_2$ must be positive, because the quantity of the purchased products should be greater than zero:

$$x_1 \geq 0, \ x_2 \geq 0;$$

5. The costs of purchasing are the criteria for minimization:

$$F(x_1, x_2) = 6x_1 + 9x_2 \rightarrow \text{min.}$$

The solution of this task is a pair of numbers $(x_1, x_2)$ that fulfills the conditions: 1), 2), 3), 4) and for which the function $F$ takes the minimum value 5). Only then the farmer will get information about the structure of the purchase, which is the cheapest.

To find a solution we should build a simulation model in the first step. A detailed description about the software interface and functions performed by the individual objects can be found in the literature (Beaverstock, Greenwood, Lavery, Nordgren, 2011). This work is focused on the implementation of the problem above. You can see the simulation model used for the experiment in Figure 1. Here we will need the following resources from the object library:

- two sources - (Quantity_A and Quantity_B);
- three sources - (Nutrient_1, Nutrient_2, Nutrient_3);
- two combiners - (Product_A, Product_B);
- one sink - (Ready_fodder);

A combiner can mix the flow items together permanently, or it can pack them so that they can be separated later. In this case, combiners combine the nutrients in previously defined Product A ($Q_A$) or Product B ($Q_B$). Fig. 1. The nutrients delivered from sources (Nutrient_1, Nutrient_2, Nutrient_3) are combined in proper proportions in the machines called combiners. Thus the production process of two types of food, consisting of three types of components has been shown. The sources have been used to create the flow of items that travel through a model. In this case there two types of sources were used. Each type of a source has a different task. The first type was used to create nutrients, the second one has been used as a decision variable (Quantity A, Quantity B). A ready-mixed feed is released into the sink (Ready_fodder).
For storage of auxiliary variables two global tables have been used. Global Tables are accessed from the Toolbox (View menu > Toolbox > Add > Global Table). Global Tables can store numerical or string data. This data can be accessed by any object in the model using the various table commands. A model may have any number of Global Tables. In this case data from global tables has been used by combiners and an optimizer.
In the second step to find the best solution, it is necessary to use OptQuest optimizer. It is one of the tools built into FlexSim. It gives us the possibility to optimize simulation experiments. Another interesting application of the optimizer can be found in the literature (Pawlewski and Hoffa, 2014; Cempel, Dąbal and Nogły, 2014) and a general description of an optimizer (Law and McComas, 2002). In an optimizer we have to define the decision variables. In this case we have two variables: \( x_1 \) and \( x_2 \) – quantity of product \( Q_A \) and \( Q_B \). Defining decision constraints in an optimization model is possible in the optimize design tab. Decision constraints have been presented in equations 1), 2), 3), 4) on the acceptable values for decision variables. We have to input these restrictions in an optimizer. They are checked at the beginning of a simulation, when a new set of decision variables is being tested. For example, a constraint might be that the minimum stock held is three units, or that the maximum number of people not receiving an appropriate level of service is ten, or that 100% of the manufactured product is distributed between shipments. In this case the constraints refer to amount of nutrients in the mix of the feed.

In this blending model, an optimizer has been tested to minimize a single objective. The best solutions found in the time given have been returned to csv file (Tab. 2). The table below lists the best twenty replications, performed by an optimizer. The optimal solution has been returned in the second replication of the experiment result. The best solution has ID no 2, rank 1, structure of purchasing: quantity \( Product\ A \) – 3 units, quantity \( Product\ B \) – 2 units, total cost 36 monetary units.

Table 2.
Results of the top twenty replications of the experiment

<table>
<thead>
<tr>
<th>Solution ID</th>
<th>Rank</th>
<th>Single Objective</th>
<th>Best Iteration</th>
<th>Goal</th>
<th>Quantity A</th>
<th>Quantity B</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>Total value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>36</td>
<td>2</td>
<td>36</td>
<td>3</td>
<td>2</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>36</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>42</td>
<td>2</td>
<td>42</td>
<td>4</td>
<td>2</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>42</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>45</td>
<td>2</td>
<td>45</td>
<td>6</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>45</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>45</td>
<td>2</td>
<td>45</td>
<td>3</td>
<td>3</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>45</td>
</tr>
<tr>
<td>19</td>
<td>5</td>
<td>48</td>
<td>2</td>
<td>48</td>
<td>5</td>
<td>2</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>48</td>
</tr>
<tr>
<td>28</td>
<td>6</td>
<td>48</td>
<td>2</td>
<td>48</td>
<td>2</td>
<td>4</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>48</td>
</tr>
<tr>
<td>22</td>
<td>7</td>
<td>51</td>
<td>2</td>
<td>51</td>
<td>7</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>51</td>
</tr>
<tr>
<td>43</td>
<td>8</td>
<td>51</td>
<td>2</td>
<td>51</td>
<td>4</td>
<td>3</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>9</td>
<td>54</td>
<td>2</td>
<td>54</td>
<td>3</td>
<td>4</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>54</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>57</td>
<td>2</td>
<td>57</td>
<td>2</td>
<td>5</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>57</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>57</td>
<td>2</td>
<td>57</td>
<td>5</td>
<td>3</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>57</td>
</tr>
<tr>
<td>27</td>
<td>12</td>
<td>57</td>
<td>2</td>
<td>57</td>
<td>8</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>57</td>
</tr>
<tr>
<td>23</td>
<td>13</td>
<td>60</td>
<td>2</td>
<td>60</td>
<td>7</td>
<td>2</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>60</td>
</tr>
<tr>
<td>40</td>
<td>14</td>
<td>60</td>
<td>2</td>
<td>60</td>
<td>4</td>
<td>4</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>63</td>
<td>2</td>
<td>63</td>
<td>9</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>63</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>63</td>
<td>2</td>
<td>63</td>
<td>3</td>
<td>5</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>63</td>
</tr>
<tr>
<td>29</td>
<td>17</td>
<td>63</td>
<td>2</td>
<td>63</td>
<td>6</td>
<td>3</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>63</td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td>66</td>
<td>2</td>
<td>66</td>
<td>5</td>
<td>4</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>66</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>66</td>
<td>2</td>
<td>66</td>
<td>2</td>
<td>6</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>66</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>69</td>
<td>2</td>
<td>69</td>
<td>10</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>150</td>
<td>69</td>
</tr>
</tbody>
</table>
We assumed fifty replications of the simulation experiment. The result for all replications can be seen in figure 2. In fact the optimal solution was very quickly found by an optimizer, it is Solution no. 2 (marked with a red larger circle).

![Optimizer Results](chart.png)

**Figure 3. Results of fifty replications of experiment performed by an optimizer**

The figure 4 shows the optimal structure of purchases. The blend should consist of substances that provide nourishment essential for growth and maintenance of life of a chicken. Thus, in the result of the calculation, we have the following data: three units of product A and two units of product B. Thus the optimal structure of purchases for preparing of the blend is lowest, because $6 \cdot 3 + 9 \cdot 2 = 36$. On the X-axis you can see the amount of component B, on the Y-axis the amount of component A determined by the software. The time of experiment was about 5.71 seconds, the experiment was limited to fifty solutions. For calculations standard PC with a processor Intel Core 2 Quad CPU Q8400 2.67 GHz, 4GB RAM and 32-bit operation system has been used.
Conclusions

Why Flexsim environment was chosen, if such type of a combinatorial task can be solved in Excel? The reason is that the model presented is the basis for launching a dynamic simulation. After finding answers referring to an optimum ratio of purchased nutrients, you can enter the dynamic parameters of model and run a simulation of the real process of manufacturing compound feed. After defining parameters, such as processing time, the distance between the machines, road transport, energy consumption can be determined:

- bottleneck in the manufacturing process;
- bottleneck in the storage;
- the costs of electricity, depending on the speed of a machine;
- the time of compounding depending on the settings of the machine;
- performance of machines, operators and more.

The amount of a feed material on the farm is estimated usually through an empirical observation. This method may lead to the increase of storage costs. Using simulation is a better way, because we can get exact amount of needed components. The main objective of this study was to find optimal structure of purchasing and to show possibilities of an optimizer. Often in real life there are more difficult situations, but all restrictions can be introduced into the simulation model. If maximization or minimization of one goal does not suffice, the optimizer can to try to maximize the cumulative value of all given objectives. In this particular case, it is possible to optimize more than one objective. For example, optimi-
Cost optimization may maximize the profit and minimize the cost. A description of the solutions to such problems can be found in other publications of the same author. A generally presented simulation model can be adopted on many fields, whenever the cost optimization is important, inter alia: selection of diet components, a manufacturing process and time optimization, agricultural production, refining industry.

Due to the limitations, the paper does not show the behavior of the model in non-linear problems. It does not introduce dynamic parameters, because a different study is required in this scope. The work in this field will be continued and presented in other publications by the author.

References


OPTYMALIZACJA KOSZTÓW SPORZĄDZANIA MIESZANKI Z WYKORZYSTANIEM ŚRODOWISKA SYMULACYJNEGO FLEXSIM


Słowa kluczowe: optymalizacja, problem diety, problem mieszanek, programowanie liniowe, badania operacyjne, flexsim, symulacja komputerowa