Least cost feed formulation by linear programming: Limitations and Extensions

A. G. Munford, Ultramix Feed Formulation Systems, 31 via Papa Giovanni XXIII, 33040 Pradamano, Udine, Italy. E-mail: agm@ultramix.co.uk

Abstract

This paper reviews how linear programming has been used in least cost feed formulation since its invention over 50 years ago and describes the great advances that have taken place due to the developments in computer hardware facilities. The human diet formulation problem played an important part in the development of linear programming as it was the first serious problem to be solved by the simplex method after its invention by Dantzig in 1948. Various types of feed formulation problems are described, as are extensions to these problems which are sometimes beyond the scope of linear programming. Some of these extended problems can be solved by solving a sequence of linear programs, thus avoiding the necessity of writing special packages. The associated problems of modelling nutrient requirements and of report writing are also addressed.

Introduction

Linear programming (LP) is arguably one of the most widely-used mathematical tools in management science. Besides its use in feed formulation, it has widespread uses in many diverse applications, such as logistics, distribution, scheduling, timetabling, resource allocation and oil refinery management. The name is unfortunate; linear optimisation is a more accurate description. LP is a methodology for solving problems in which the objective is to maximize or minimize a linear function of many variables subject to a number of linear inequality or equality constraints on those variables. The methodology itself has nothing to do with computer programming, although the solution of practical LP problems is almost impossible without the use of a computer. By mathematical standards, LP is very new. Fourier and Gauss were aware of its potential in the early part of the 19th century, but were unable to put forward a practical solution procedure. The major breakthrough came in 1947 when Dantzig (1948) proposed the famous "simplex method" for the solution of LP problems.

Interestingly, there were many well-documented LP problems already formulated before the simplex method was invented, mostly in the fields of economics and game theory, but it was a feed formulation problem that was used as the first serious test of the new method. In 1945 the Nobel prize-winning economist Stigler published a paper in which he attempted to minimise the cost of feeding a man for a day. From a list of 77 commonly available foods he attempted to find the cheapest way of providing the 9 nutrients listed in Table 1.

Stigler cleverly created a feed matrix not in terms of nutrient density as we would today, but in terms of nutrients supplied for one dollar's worth of each food. He then reduced the list of 77 foods to 15 by eliminating those foods which contained no more of each nutrient than did some other food. But his diet problem was published unsolved.

By modern standards, a feed formulation problem with 77 feeds and 9 nutrients is small, and would take a small fraction of a second on a modern personal computer to solve. However in 1947, with only hand operated desk calculators available, Stigler's problem took approximately 120 man-days to solve. This problem is very useful for teaching purposes, as the solution is completely unreasonable, requiring a 70kg male to eat 470g of dried beans, 370g of flour, 100g of cabbage, 23g of spinach and 3g liver daily, and it oversupplies protein by a factor of more than 2. The deficiency in Stigler's model is that he omitted to consider food inclusion constraints, i.e. he should have placed daily limits on the individual foods as well as the nutrients.

Table 1. Daily nutrient allowances in Stigler's diet problem

Nutrient	*Daily allowance*	*Units*
Energy	3,000	kCal
Protein	70	g
Calcium	0.8	g
Iron	12	mg
Vitamin A	5,000	IU
Thiamine	1.8	mg
Riboflavin	2.7	mg
Niacin	18	mg
Vitamin C	75	mg

Impact of the Personal Computer

The popular IBM 1130 "mini" computer introduced in 1965 which was used for some of the early feed formulation problems came with just 8 kilobytes of memory. This memory had to be used for the programme instructions as well as data storage. Bearing in mind that each number requires 4 bytes of storage, special techniques had to be invented to minimise the amount of storage space required. Even Stigler's diet problem required 77 x 9 x 4 =2772 bytes just for the feed matrix, so in those days it could not be regarded as a small problem. In fact data were not stored in the computer's memory as they would be today; data and results of intermediate calculations would be stored on magnetic tape and repeatedly be written and read back into memory. A typical modern notebook computer with 1 gigabyte of memory has over 130,000 times as much memory as the IBM 1130 of 40 years ago. Huge advances in hardware capabilities have been made at a fraction of the cost.

Feed formulation problems

The classical feed formulation problem Munford (1996,1) can be of one of two types: batch mix, or complete diet (per head). In the batch mix formulation, the idea is to make a fixed quantity of feed subject to constraints on minimum and maximum nutrient concentration levels as well as minimum and maximum feed inclusion constraints. The units of the nutrients in the batch are the same as those of the constituent feeds. In order that the constraints are met within the batch size, there is always a dummy nutrient "Bulk" which takes the value 1 for all nutrients and must be 1 in the finished batch. Sometimes nutritionists allow the bulk to float between two numbers on either side of 1 which would necessitate a change in the feed rate, but this is a clumsy approach and the desired effect can be achieved in more elegant ways.

Complete diet formulations are commonly used for example in the case of ruminants. Here, the idea is to provide absolute quantities of nutrients, rather than nutrient concentrations, so MJ/kg becomes MJ, g/kg becomes g and percent becomes g etc. Structurally, batch mix formulation and complete diet formulation are the same; all that is needed is a signal to the software system that nutrient unit conversion is to take place in the case of complete diets, and also that percentage values must be multiplied by 10 to convert them to g/kg. The early feed formulation systems which ran on mini computers such as the IBM 1130 were designed only for batch mixes. Perhaps because this was the only readily available tool, historically, pig and poultry rations have always been formulated on a batch basis as it is assumed that the energy density of the optimum diet is known in advance. But recently, Whittemore *et al* (2003) have published new standards in which pig diets are specified on a per head basis.

In addition to nutrient and feed inclusion constraints, there may be constraints on groups of feeds. For example if there are 4 types of molasses on offer and each has an upper inclusion rate of 2.5% then it is quite possible that the least cost diet will call for all 4 at the upper level. What is required is an overall group constraint which constrains the total molasses level to be no more than 2.5%. This can be achieved by inventing a molasses group "nutrient" which is 1 for all members of the molasses group and 0 otherwise. Thus group inclusion constraints can be dealt with within the linear programming framework. Most serious feed formulation packages do this in the background and there is no need for user intervention other than to specify the group membership and the group inclusion limits.

There is one golden rule in least cost formulation and that is that each extra constraint can only increase the minimum cost. At best it will be redundant, but if it is active then the cost will go up. A common error is to try and drive the formulation by adding unnecessary feed inclusion constraints. It is rather like having a satellite navigation system for finding the optimum route between two cities, but forcing it to pass through a list of places that the user feels ought to be on the optimum route.

Another common error is to over-constrain minerals and micro minerals which can tend to have more effect on the final solution than energy and protein. Mineral constraints should be avoided unless there is a readily available source of the mineral available in a relatively cheap form. What can happen is that a least cost formulation can be forced to satisfy some mineral constraint by taking a large quantity of the only raw material containing a sufficiently high concentration of it, thereby forcing energy and protein constraints to be satisfied by including expensive concentrates. An experienced formulator will always look hard at the marginal costs associated with mineral constraints, as this is the indicator of a stable formulation.

Multi-mix problems

A feed mill will typically produce many tens of products, each of which will have inclusion limits on individual raw materials. A typical situation is that supplies of one or more raw materials are limited, and that the combined requirement of all the products exceeds the available supply. The problem here is that the products cannot be formulated individually, but must be formulated simultaneously, with extra constraints which specify the maximum (and possibly the minimum) available tonnage of each raw material. Minimum limits will apply in those cases where it is required to use up certain raw materials, which may well have been ordered many months previously. Fundamentally, a linear programming problem is characterized by a matrix. In the case of a feed formulation problem, it is a feed matrix, with rows representing nutrients and columns representing feeds. The difficulty of a problem is related in a major part by the number of rows, and to a lesser extent to the number of columns. In a standard problem, there may be say 50 rows (nutrients) and 100 columns (feeds). In a multimix problem with R raw materials, N nutrients and P products, it can be shown that the related multimix formulation has RxP columns and NxP+R rows. In the previous example, suppose there are P=80 products, then the problem would have 8,000 columns and 4,100 rows and the corresponding matrix would have 32,800,000 elements, of which at least 32,392,000 would be zero. Special versions on the simplex method have been written to exploit the special structure of multi-mix problems, but even so, problems on this scale would have been very difficult on an early mini computer. On a modern portable computer they are readily solvable. The levels of complexity can increase even more because time can be taken into account, so that we have multi-period formulations, and larger feed companies will have several mills, resulting in a multi-mill multi-period multi-mix problem with huge numbers of rows and columns.

Multi-mix not only applies to batch mix problems; it can also be used for per-head formulations at the farm level. This is particularly useful for dairy and beef enterprises where stocks of home-produced forage are limited, and of varying quality. The idea would be to model the activity of the farm over the entire year, with constraints on the weekly nutrient requirements of the livestock during its various stages of growth and production, subject to constraints on the availability, in terms of quantity and timing, of home-produced feed. The model would then suggest which groups of livestock would be fed which home-produced forage and when, and which feeds would be bought-in. It is also possible to build even more elaborate models which take into account costs of labour, accommodation, veterinary costs and borrowing, but these models require a huge amount of data input and their use is restricted to strategic planning, rather than day to day management.

Nonlinearity

Often it is required to control a ratio of nutrients in a formulation. For example a simple ratio such as that of calcium to phosphorus, or a more complicated one such as forage dry matter to concentrate dry matter in a ruminant complete diet. Also controlling the dry matter percentage *as-fed* in a diet involving wet feeds is a ratio constraint. Ratio constraints are non-linear and fall outside the scope of the linear programming framework but can easily be transformed into linear constraints. For further details and how to calculate the associated marginal costs, see Munford (1989-1) and (1989-2).

A least cost ration is often regarded as a starting point and the formulation is often finished off by hand. This usually involves omitting ingredients with an inclusion rate below some fixed threshold and rounding other ingredients to realistic weighing quantities. It is possible to incorporate these constraints directly into the model but the problem remains no longer linear and becomes what is known as a (mixed) integer programming problem. The algorithm is based on the simplex method but is an order of magnitude more difficult. For details of the method, see Williams (1993).

Other constraints which can be modelled using integer programming are those when there is a limit to the actual number of raw materials that can be used in a formulation (including multi-mix). This situation is very common because the number of available raw material storage bins is always limited.

Controlling variability

It is widely accepted that the density of certain nutrients vary considerably not only from batch to batch, but also within batches; a good example of this is crude protein density in soya. In formulating a batch mix, it is usually specified that a minimum level of the nutrient in question will be achieved with a fixed probability. For example it can be stated that the crude protein level is 18% with probability 95%. In order to solve such a problem, it is necessary to have some model for the variance of the amount of each nutrient as a function of the inclusion rate. It is quite common to assume a quadratic relationship (it is arguable that this assumption is fundamentally flawed, but it is not important to the following discussion), and therefore the formulation problem becomes linear but with one quadratic constraint. Special algorithms have been written for this problem but it is quite possible to solve the problem by solving a series of linear programming problems; effectively updating the feed matrix after each pass. This is called recursive linear programming and is described in Munford (1996-1).

Modelling nutrient variability is an extremely difficult problem because it is essential to model also the spatial variability of the nutrient content throughout a batch. For instance if raw materials are added from a bin which releases its content at a constant rate, then it is necessary to know the mathematical form of the correlation between nutrient density for every pair of times t_1 and t_2 at which releases take place. Intuitively, the closer are the two points in time, the higher should be the correlation. As the two points become further apart, we would expect the correlation to reduce. The usual quadratic variance model assumes perfect correlation, for all temporal separations and is therefore of limited practical value. With more realistic variance models, parameter estimation becomes notoriously difficult. The concept of a variance model is not simple, but it is helpful to consider the problem of how the variance of the crude protein of 2 kg of a raw material is related to the variance of 1 kg. There are two properties of random variables which can be found in every basic statistics textbook

Model 1: $\text{var}(cX) = c^2\text{var}(X)$
Model 2: $\text{var}(X+Y)=\text{var}(X)+\text{var}(Y)$

where X and Y are random variables and c is a constant. The first model gives rise to the widely used quadratic model. It assumes that every kg is variable but that all kg are identical. Therefore in this case it would be sufficient to analyse the first kg and the content of all the others is known, so that the variability problem is no more. The second model assumes that every kg is uncorrelated with every other. This is more appealing, but makes less sense if the same model is applied at the gram or milligram level. This reasoning highlights the fact that more sophisticated spatial/temporal models are required. This problem must be addressed before linear programming can be used to control nutrient variance.

Variable efficiency factors

A common problem in ruminant complete diets is that the efficiency of nutrient utilization depends on the diet being fed. In broad terms, the higher the energy concentration of the diet, the better the animal can make use of the energy and the overall limit is reduced. Put another way, lower energy concentration diets call for a higher overall level of energy. For example in the UK metabolizable energy (ME) system (AFRC, 1993), the efficiency of utilization of ME depend on a quantity q, the ratio of ME to gross energy (GE) in the diet. Similarly in the NRC system (NRC,2001), there is a so-called diet dependent "discount factor" which reduces the efficiency of utilization of total digestible nutrients (TDN). In these

Table 2. Convergence of metabolizability in a dairy ration

Cycle	*ME*	*GE*	*Q*
0	-	-	0.70000
1	275.64	435.15	0.63343
2	285.24	445.30	0.64055
3	284.36	446.17	0.63734
4	284.85	445.80	0.63896
5	284.60	446.11	0.63796
6	284.75	445.92	0.63858
7	284.66	446.04	0.63820
8	284.72	445.96	0.63843
9	284.68	446.01	0.63829
10	284.70	445.98	0.63838

cases it is impossible to specify the nutrient requirements in advance before the diet is calculated, and it is obviously impossible to calculate the diet before the nutrient requirements are specified. There needs to be some way of breaking the circle and this again can be achieved by recursive linear programming. The idea is that an initial value of the unknown parameter is assigned, the requirements are calculated and the diet calculated. From this the parameter is re-evaluated, the requirements recalculated and so on. This process is continued until some degree of convergence is obtained. This process has been successfully applied to the calculation of least cost diets in the UK ME system, the NRC (2001) system, and more recently to the UK Feed into Milk System. Table 2 shows how the recursive procedure leads to convergence in a typical dairy ration with a starting value of q=0.7.

Modelling

With the massive computing power so readily available in recent times, feed formulation systems can now do much more than simply manage feed data and calculate least cost diets. For example the Ultramix system (Munford, 2005) there is an integrated modeller and report writer. The modeller is a spreadsheet-like environment in which nutrient requirements can be calculated. Also results of previous formulations can be referenced in calculations so that the recursive linear programming technique described above can be performed. Table 3 shows part of a model for the UK ME system. Note how the formula for *q* references the values of ME and GE in the previous ration.

Table 3. Part of an Ultramix model for the UK ME system

Symbol	*Formula*	*Value*
q	if([IsOne],rat(ME)/rat(GE),0.7)	0.612
km	0.35*[q]+0.503	0.717
kl	0.35*[q]+0.420	0.634
MEM	(0.53*[BW]/1.08)^0.67+0.0095*[BW])/[km]	58.947
MELWC	[LWC]*if([LWC]>0,34,28)	-17.000
MELact	(0.384*[Fat%]+0.223*[Prot%]+0.199*4.7-0.108)/[kl]	4.886
MEReqt	[MEM]+[MELWC]+[MELact]*[Y]	212.970

LP formulations driven my models are extremely useful for on-farm extension work. An adviser can specify his model in terms of easy to understand parameters such as liveweight and liveweight gain, rather than megaJoules of energy or grams of protein. Unfortunately, batch formulations do not lend themselves so readily to being model-driven, so most pig and poultry models are static.

Report writing

The raw output from feed formulation system, while containing useful information for the livestock adviser, will often be too detailed for the farmer. Some systems have reports at two levels; a detailed adviser report, and a farmer report containing only a summary. But what is required is the ability for the adviser to create his own reports, at various levels of detail to suit his client base, and in his own house style. This last point is important to most advisers who will not want to present their clients with reports in the identical form to those of their competitors.

A report writer is a module of a formulation system that produces reports according to a template. A report template is a document containing all the static text and formatting of the desired report, as well as special codes which refer to characteristics of the formulation or formulations to be printed out. In the Ultramix system, these special codes all begin with the "@" symbol. For example @date() returns the current date, @rat(*x*) returns the level of nutrient *x* in the current ration and @ratdm(*x*) returns the same thing but on a dry matter basis. The Ultramix report writer has its own programming language for performing calculations, executing loops and logical statements. For example the following code would produce a single line in the report according to the level of crude protein in the dry matter.

```
@do(ratdm(CP)<120)
        The crude protein concentration is very low
@elsedo(ratdm(CP)<140)
        The crude protein concentration is quite low
@elsedo(ratdm(CP)<160)
        The crude protein concentration is medium
@elsedo(ratdm(CP)<180)
        The crude protein concentration is quite high
@else()
        The crude protein concentration is high
@enddo()
```

Since the report templates themselves are rich text documents, text formatting such as bold and italic, as well as colour can be added to various lines of the report for emphasis. Once composed, text can be added or deleted, and the final document printed, or alternatively saved as a document in rich text format (rtf), which can then be emailed to the client who can open the document in a word processor of his choice.

A big advantage of an integrated report writer is that it removes the possibility of errors arising from transcription of data from the optimiser to an external software package. Of secondary importance is the saving in time and effort which is not inconsiderable. There is no limit to the type of report that can be generated. For example, the report writer can be used to print labels using the analysis from the formulation. It can also be used to produce text files that can interface directly with mixing and weighing machines.

Using the Ultramix report writer programming language, templates can also be created to perform complex post-optimization calculations. These can involve results of the formulation and also reference the modeller. So for instance if a least cost ration is performed leaving the vitamins and minerals to take free values, it is quite possible to use a report to calculate the specification of a mineral vitamin supplement to balance the ration. This type of report finds great favour among field representatives who find that this is a useful sales tool.

Conclusions

Linear programming has been used with great success over the last 50 years to solve feed formulation problems at both the farm and feed mill level. There is an enormous number of variations on the basic problem which make the associated linear programs very large, but fortunately the advances in available computing facilities have more than kept up with the modelling developments. Feed formulation involves a combination of nutritional skills and mathematical skills. While most of the mathematics is taken care of in the formulation software packages, it is important that formulators have a good understanding of mathematics in order to do their job well. In the future, we expect to see more formulation modelling linked to growth models, but this will require pig and poultry formulations to be done on a per-head rather than a batch basis as is current practice. There is also much further research needed to propose models for nutrient variance functions and the associated statistical estimation problems.

References

AFRC (1993) Energy and Protein Requirements of Ruminants. *CAB International, Wallingford, U.K.*

DANTZIG, G.B. (1948) Programming in a Linear Structure. *Comptroller, USAF, Washington D.C.*

STIGLER, G.J. (1945) The Cost of Subsistence. *Journal of Farm Economics* **27**: 303-314

MUNFORD, A.G. (1992) An Integrated Computer System for Feed Formulation, Livestock Modelling and Feed Database Management. *in Relationship of Feed Composition to Animal Production*, Edited by J E Knipfel, International Network of Feed Information Centres

MUNFORD, A.G. (1993) Using the Ultramix Livestock Rationing System in Agricultural Extension. *Proceedings of the XXV CIOSTA-CIGR V Congress, Wageningen*

MUNFORD, A.G. (1996) The Use of Mathematical Programming in Practical Applications of Animal Diet Formulation. *Mathematics and Computers in Simulation* **42**: 255-261.

MUNFORD, A.G. (1996) Successive Linear Programming: A Powerful Tool for Livestock Ration Formulation. *Proceedings of the 6th International Congress for Computer Technology in Agriculture, Wageningen*

MUNFORD, A.G. (1989) A Microcomputer System for Formulating Animal Diets which may Involve Liquid Raw Materials. *European Journal of Operational Research* **41**: 270-276.

MUNFORD, A.G. (1989) Evaluating Marginal Costs Associated with Ratio and Other Constraints in Linear Programmes. *Journal of the Operational Research Society* **40**: 933-936.

MUNFORD, A.G. (2005) The Ultramix-Professional Feed Formulation and Livestock Modelling System. *AGM Systems Ltd, Udine, Italy*

NRC (2001) Nutrient requirements of dairy cattle. *National Academy Press, Washington D.C.*

WHITTEMORE, C.T., HAZZLEDINE, M.J. and CLOSE, W.H. (2003) Nutrient Requirement Standards for Pigs. *British Society of Animal Science, Penicuik*

WILLIAMS, H.P. (1993) Model Solving in Mathematical Programming. *John Wiley & Sons, London*