A SAS/OR® Primer on Using PROC LP

Michael E. Salassi
Department of Agricultural Economics & Agribusiness
Louisiana State University, Baton Rouge, LA

ABSTRACT

Linear programming is a widely used operations research tool in economics and business. It is used to answer such business questions as how to maximize profits from products produced or how to minimize costs of certain business activities given certain constraints. The use of PROC LP in SAS/OR® to answer such questions will be presented including the formulation of linear programming, integer programming and mixed-integer programming models. SAS/OR general syntax will be presented, along with a discussion of dense and sparse data format, reduced cost and dual activity, and sensitivity and range analysis. The use of accounting activities and transfer rows will also be presented, illustrating how to model the SAS/OR program to calculate important values of the problem in addition to the objective function value.

INTRODUCTION

Operations research is simply a scientific approach to decision making that seeks to best design and operate a system, usually under conditions requiring the allocation of scarce resources (Winston, 2004). Sometimes referred to as management science, operations research includes a wide variety of mathematical procedures which can be used to solve a multitude of problems in engineering, transportation, communications, manufacturing, business and economics. With the tremendous advances in the development of computer hardware and software over the past several years, large scale operations research problems can now be solved on laptop computers in a minimal amount of time.

SAS® software includes a wide array of operations research procedures which can be used to solve a multitude of resource allocation and optimization problems. Some of general types of problems which can be solved using procedures in SAS Version 9 include:

- $ Linear, integer, nonlinear and network programming
- $ Project management, resource management and scheduling
- $ Graphical display of Gantt charts and networks
- $ Simulation of queueing systems
- $ Decision analysis and decision tree display
- $ Bill of materials processing

Mathematical programming is the most commonly used operations research procedure. Its simplicity of use and applicability to a wide array of problems are the primary reasons for its widespread appeal. Linear programming is the most popular mathematical programming procedure, although several others are also commonly used. The SAS/OR Version 9 Mathematical Programming package includes several of these procedures:

- $ PROC LP – simplex method linear programming algorithm
- $ PROC NETFLOW – network programming
- $ PROC INTPOINT – interior point linear programming algorithm
- $ PROC NLP – nonlinear programming
- $ PROC QP – quadratic programming
- $ PROC TRANS – transportation models
- $ PROC ASSIGN – assignment models

In using the above mathematical programming procedures, a mathematical model or representation of the problem being solved must first be developed. After that point, a SAS program can then be designed to solve the problem using one of the above procedures. This paper presents a general overview of the use of the PROC LP procedure in SAS/OR to solve linear programming problems in business and economics. The first section of this paper presents the basic concepts of linear programming, including general problem formulation and a simple example problem. The next section discusses the PROC LP procedure, including syntax, data format and displayed output. The following section presents the four types of solutions which may be obtained when solving linear programming problems and illustrates how they can be identified in SAS output. The next section presents several PROC LP solution options which can provide additional information and analysis of the problem, including objective function sensitivity analysis, right-hand-side sensitivity analysis, parametric programming, and goal programming. The final section of the paper presents the use of transfer rows and accounting activities which can be utilized to provide additional summary information within the PROC LP model solution.

LINEAR PROGRAMMING BASICS

In order to utilize PROC LP to solve linear programming problems, the user must have a good working knowledge of the basic concepts of linear programming, what kinds of problems it can be used to solve, and how to formulate a mathematical model or representation of the problem.

GENERAL LP PROBLEM FORMULATION

A general application of linear programming is the process of allocating limited resources to competing activities in an optimal way. An objective function is optimized subject to constraints which limit the feasible choices of variables being evaluated. For example, a model could be developed which would determine what quantities of products produced by a firm would maximize profit over a given time period. Another typical linear programming model might determine what route products should be shipped over from one point to another to minimize transportation costs. *Linear* implies that all functions in the problem are linear functions. *Programming* refers to a sequence of steps that lead to the discovery of the best value of the function. The PROC LP procedure utilized the simplex method to solve linear programming problems. Linear programming models have three basic parts of the problem being evaluated:

- $ Decision variable – quantities under the control of the decision maker for which optimal values are to be determined
- $ Constraints – restrictions on the process that limit the values the decision variables may take
- $ Objective function – a function of the decision variables that is maximized or minimized subject to the constraints

The general mathematical formulation of a linear programming model can be stated as follows:

Max/Min $$Z = c_1 x_1 + \ldots\ldots + c_n x_n \qquad (1)$$

subject to

$$a_{11} x_1 + \ldots\ldots + a_{1n} x_n \quad \leq = \geq \quad b_1 \qquad (2)$$
$$x_i \quad \leq = \geq \quad b_2 \qquad (3)$$

and

$$x_n \geq 0 \qquad (4)$$

where equation (1) is the objective function which is either being maximized or minimized, equations (2) and (3) are typical forms of functional model constraints, x_1 – x_n are the decision variables of the model, c_1 – c_n are the objective function coefficients, a_{11} – a_{1n} are technical coefficients, b_1 and b_2 are right-hand-side values, and equation (4) represents the non-negativity constraint on the decision variables.

Linear programming models have four basic assumptions which serve as a basis for the model formulation. These assumptions influence the formulation of the mathematical model and also impact the solution results. The assumptions of linear programming models include:

- $ Proportionality – all returns and resource usages change proportionally with the variables with no economies of scale (i.e., no exponents other than one)
- $ Additivity – the total effect of any two variables is the sum of their individual effects without synergy or interaction (i.e., no cross-product terms)
- $ Divisibility – variables may assume fractional values
- $ Certainty – all parameters in the model are constants and known with certainty

AN EXAMPLE PROBLEM

In order to illustrate linear programming and the use of PROC LP to solve such problems, a simple example from a popular operations research text will be used (Hillier and Lieberman, 2001). The Wyndor Glass Co. produces two products: windows (product 1) and glass doors (product 2). Aluminum frames and hardware are made at Plant 1, wood frames are made at Plant 2, and Plant 3 produces glass and assembles the products. Product 1 requires production capacity in Plants 1 and 3. Product 2 requires production capacity in Plants 2 and 3. The question to be solved here is: *What are the production rates for the two products which would maximize profit, subject to the limited production capacities at each plant?*

Data necessary to formulate this problem is presented in the table below.

	Production Time per Batch, (Hours)		Production Time Available per Week, (Hours)
Plant	Product 1	Product 2	
1	1	0	4
2	0	2	12
3	3	2	18
Profit	$3,000	$5,000	

This problem will include two decision variables, x1 and x2, representing the quantities of products 1 and 2 produced. The objective function will be formulated as a profit equation which will be maximized. Three constraint equations will be included in the model, each one representing production capacity at each of the three plants. The complete mathematical formulation of the problem can be stated as follows:

$$
\begin{array}{lllll}
\text{Max} & Z = 3x1 + 5x2 & & & \text{(objective function: profit equation)} \\
\text{s.t.} & & & & \\
 & x1 & \leq & 4 & \text{(Plant 1 capacity functional constraint)} \\
 & \quad\quad 2x2 & \leq & 12 & \text{(Plant 2 capacity functional constraint)} \\
 & 3x1 + 2x2 & \leq & 18 & \text{(Plant 3 capacity functional constraint)} \\
\text{and} & & & & \\
 & x1 \geq 0, \quad x2 \geq 0 & & & \text{(non-negativity constraints)}
\end{array}
$$

In solving this problem by linear programming and interpreting the results, it is helpful to understand some basic linear programming terminology. A *feasible solution* is a solution (i.e., values of decision variables) for which all the constraints are satisfied. The *feasible region* is the collection of all feasible solutions. If there are no feasible solutions to the problem, then the feasible region is empty and the problem is infeasible. An *extreme or corner point solution* is a feasible solution at the corner of the feasible region. This point occurs at the intersection of two or more constraints. A *binding constraint* is a constraint that holds as an equality at a particular feasible solution. The *optimal solution* is that feasible solution that optimizes the objective function and the *optimal objective function value* is the value of the objective function at the optimal solution.

Before the simplex method is used to solve a linear programming problem, all functional or structural constraints are converted to equalities through the use of slack or surplus variables. *Slack variables* are variables added to a less than or equal to constraint to convert it to an equality. *Surplus variables* are variables subtracted from a greater than or equal to constraint to convert it to an equality. The resulting problem formulation is referred to as the *augmented model formulation*:

$$
\begin{array}{lllllll}
\text{Max} & Z = 3x1 + 5x2 & & & & & \\
\text{s.t.} & & & & & & \\
 & x1 & + S1 & & & = & 4 \\
 & \quad\quad 2x2 & & + S2 & & = & 12 \\
 & 3x1 + 2x2 & & & + S3 & = & 18 \\
\text{and} & & & & & & \\
 & x1 \geq 0, \quad x2 \geq 0 & & & & &
\end{array}
$$

where S1, S2, and S3 are slack variables which have been added to the three functional constraints. All linear programming software packages, including SAS/OR, automatically add slack and surplus variables to the appropriate constraints. Although the programmer does not have to specify these variables, they will show up in the linear programming output as variables. In addition, linear programming software packages also automatically specify the non-negativity constraint for each decision variable specified in the model.

PROC LP PROCEDURE IN SAS/OR

Linear programming problems can be solved in SAS/OR using the PROC LP procedure. This procedure can solve linear programs, integer programs, or mixed-integer programs. Options with PROC LP also allow for parametric programming and range analysis and reports on solution sensitivity to changes in the right-hand-side constants and objective function coefficients. The LP procedure provides various control options and solution strategies. It also provides the functionality to produce various kinds of intermediate and final solution information. SAS input data for linear programming models solved with the PROC LP procedure can be entered in two different data formats: *dense data format* and *sparse data format*. Both forms of input data format will yield the identical linear programming output in SAS. The following sections of this paper illustrate these two input data formats for the example problem presented above.

DENSE DATA FORMAT

In the *dense data format*, linear programming problem input data is formatted in a manner similar to the mathematical formulation. Input data for each equation is listed on one line of input. Each SAS variable in the input data set corresponds to a model's column. Each SAS observation in the input data set corresponds to a model's row. Input variables specified in the input statement include: user specified decision variables, _ROW_ which identifies the rows in the model, _TYPE_ which indicates how to interpret the observation, and _RHS_ which identifies the right-hand-side values.

The SAS input data set for the above example problem in dense data format would be as follows:

```
DATA NEW:
INPUT _ROW_ $ PRODUCT1 PRODUCT2 _TYPE_ $ _RHS_;
CARDS;
PROFIT    3.0    5.0    MAX     .
PLANT1    1.0      0    LE    4.0
PLANT2      0    2.0    LE    12.0
PLANT3    3.0    2.0    LE    18.0
;
PROC LP;
TITLE ‘LP PROBLEM – DENSE DATA FORMAT’ ;
RUN;
```

Each input data line contains data for five variables, the row name (_ROW_), values for the two decision variables (PRODUCT1 and PRODUCT2), the type of equations (_TYPE_), and the value of the right-hand-side variable (_RHS_). Dense data format is a very concise way to enter a linear programming model in SAS. However, for more realistic problems, it can be somewhat cumbersome. All coefficients in the model, including those with zero values, must be specified in the input data set. For large problems, with hundreds or thousands of coefficients, this can create problems as only a small percentage of those coefficients are nonzero. In addition, as the number of decision variables increases, the input data for one equation can spread over several lines of input, making it difficult to quickly and easily change coefficients in the model when necessary.

SPARSE DATA FORMAT

In the *sparse data format*, linear programming problem input data is formatted in a manner much different than in dense data format. Only the nonzero coefficients of the linear programming problem are entered. Each line of SAS input data identifies a specific nonzero coefficient of the model. Coefficients with zero values do not have to be entered or specified. Input variables specified in the input statement include: _TYPE_ which indicates how to interpret the observation, _ROW_ which identifies the row name, _COL_ which identifies the column name (variable name) and _COEF_ which specifies the value of the nonzero coefficient.

The SAS input data set for the above example problem in sparse data format would be as follows:

```
DATA NEW;
INPUT _TYPE_ $  _ROW_ $ _COL_ $ _COEF_;
CARDS;
MAX       PROFIT       .                 .
.         PROFIT       PRODUCT1       3.0
.         PROFIT       PRODUCT2       5.0
LE        PLANT1       PRODUCT1       1.0
.         PLANT1       _RHS_          4.0
```

```
LE        PLANT2       PRODUCT2       2.0
.         PLANT2       _RHS_         12.0
LE        PLANT3       PRODUCT1       3.0
.         PLANT3       PRODUCT2       2.0
.         PLANT3       _RHS_         18.0
;
PROC LP SPARSEDATA ;
TITLE 'LP PROBLEM – SPARSE DATA FORMAT';
RUN;
```

Since only a small percentage of coefficients in most realistic problems are nonzero, sparse data format provides for a much more efficient way to enter input data into SAS. Each coefficient in the input data set is easily identifiable as each input line includes the row name and column name for the coefficient. To call the sparse data input routine in SAS, the option SPARSEDATA is included just after the PROC LP command. Results from the PROC LP procedure using sparse data format will be identical to the results from using dense data input format.

PROC LP DISPLAYED OUTPUT

Displayed output from the PROC LP procedure consists of four sections:

- $ Problem Summary
- $ Solution Summary
- $ Variable Summary
- $ Constraint Summary

The Problem Summary provides basic statistics concerning the linear programming problem which has been formulated. Information provided includes type of objective function, problem density (the percent of model coefficients which are nonzero), number and type of model variables, and number and type of model equations.

```
  LP PROBLEM - SPARSE DATA FORMAT
         The LP Procedure
         Problem Summary

Objective Function     Max PROFIT
Rhs Variable                _RHS_
Type Variable              _TYPE_
Problem Density (%)         46.67

Variables                  Number

Non-negative                    2
Slack                           3

Total                           5

Constraints                Number

LE                              3
Objective                       1

Total                           4
```

The Solution Summary provides information about the problem solution. The most important information provided here is the value of the objective function for the optimal solution. For this particular problem, the objective function value is 36, or $36,000. Additional information here indicates that it required two Phase 2 iterations to solve this problem, indicating that only two corner-point feasible solutions needed to be evaluated in order to find the optimal solution to this problem.

```
LP PROBLEM - SPARSE DATA FORMAT
      The LP Procedure
      Solution Summary

    Terminated Successfully

Objective Value                               36

Phase 1 Iterations                             0
Phase 2 Iterations                             2
Phase 3 Iterations                             0
Integer Iterations                             0
Integer Solutions                              0
Initial Basic Feasible Variables               5
Time Used (seconds)                            0
Number of Inversions                           3

Epsilon                                     1E-8
Infinity                           1.797693E308
Maximum Phase 1 Iterations                   100
Maximum Phase 2 Iterations                   100
Maximum Phase 3 Iterations              99999999
Maximum Integer Iterations                   100
Time Limit (seconds)                         120
```

The Variable Summary provides information about the variables of the problem for the optimal solution. Five variables are listed for this problem. Two decision variables, PRODUCT1 AND PRODUCT2 were specified by the user. The variables PLANT1, PLANT2 and PLANT3 represent the slack variables added by SAS/OR to the three functional less-than-or-equal-to constraints. SAS/OR gives these slack variables the same name as the row name for which they represent. The same naming convention applies to surplus variables for greater-than-or-equal-to constraints. The 'Status' column identifies which variables are in the optimal solution at nonzero values. Results for this problem indicate that PRODUCT1 and PRODUCT2 are both in the optimal solution, along with the slack variable for the first constraint (PLANT1) indicating that Plant 1 had unused capacity. The 'Type' column identifies the type of each variable, two non-negative decision variables and three slack variables. Objective function coefficients for each variable are listed in the 'Price' column. The slack variables have a zero objective function coefficient since they do not contribute to profit. The optimal solution for this problem is listed in the 'Activity' column. Two batches of Product 1 and six batches of Product 2 result in the maximum obtainable profit for this problem. Plant 1 has two units of labor unused and Plants 2 and 3 have no unused labor. The 'Reduced Cost' column lists how the optimal objective function value would change if one more unit of each variable came into the solution. Non-zero values in this column are an indication of which constraints are binding the optimal solution. Plant 1 has unused capacity. Therefore, having one more unit of labor available will not change the solution, hence the reduced cost value for this variable is zero. All labor at Plants 2 and 3 were utilized. The zero values in the 'Activity' column indicate that the slack variable values are zero. The reduced cost values for these two plants may be interpreted as the reduction in profit if one unit of labor at that plant were unused. For example, if Plant 2 had one less unit of labor to used, 11 instead of 12, total profit would be reduced by 1.5 or $1,500 in the optimal solution. Similarly, one less unit of labor at Plant 3 would reduce profit by 1.0 or $1,000.

```
LP PROBLEM - SPARSE DATA FORMAT
      The LP Procedure
      Variable Summary
```

Col	Variable Name	Status	Type	Price	Activity	Reduced Cost
1	PRODUCT1	BASIC	NON-NEG	3	2	0
2	PRODUCT2	BASIC	NON-NEG	5	6	0
3	PLANT1	BASIC	SLACK	0	2	0
4	PLANT2		SLACK	0	0	-1.5
5	PLANT3		SLACK	0	0	-1

The Constraint Summary provides information about the right-hand-side values of the problem in the optimal solution. In this problem, as in many linear programming problems with the objective of maximizing profit, the right-hand-side values represent quantities of resources available for use. For this problem, four constraints are listed in the constraint summary: the objective function and three functional constraints. The first constraint is the objective function and the optimal objective function value of 36 is listed in the 'Activity' column. Quantities of the labor resources used at each plant are also listed in the 'Activity' column. The 'Dual Activity' column lists, for each right-hand-side value, by how much would the objective function value change if one more unit of that resource were available. Once again, non-zero values in the Dual Activity column indicate constraints which are binding or limiting the solution. Labor at Plants 2 and 3 are fully utilized in the optimal solution and having one more unit of each available would increase the optimal objective function value by 1.5 ($1,500) and 1.0 ($1,000), respectively.

```
                    LP PROBLEM - SPARSE DATA FORMAT
                          The LP Procedure
                         Constraint Summary

           Constraint            S/S                          Dual
       Row Name        Type       Col        Rhs  Activity  Activity

         1 PROFIT      OBJECTVE     .          0        36         .
         2 PLANT1      LE           3          4         2         0
         3 PLANT2      LE           4         12        12       1.5
         4 PLANT3      LE           5         18        18         1
```

INTEGER AND BINARY PROGRAMMING

In the example problem presented here, the optimal solution values for Product 1 and Product 2 happened to be integer values. However, they just as easily could have been non-integer values. In some cases, non-integer values of decision variables in the optimal solution can be rounded off to integer values. One of the dangers of rounding off non-integer values is that the rounded integer decision variable values may not be feasible. SAS/OR does provide the option of defining some or all of the variables in a linear programming model to be integer. If all variables are defined as integer, SAS/OR solves an integer programming model. If some, but not all, of the variables are defined as integer, the problem is referred to as a mixed integer problem.

To define the two decision variables in the example problem as integer, four additional lines of input data need to added to the original problem. The first two lines of input data define the variables PRODUCT1 and PRODUCT2 as integer variables. The next two lines of input data define the upper limits that each variable may take. In the input data listed below, the upper limit of the variables PRODUCT1 and PRODUCT2 are specified as 10.

```
DATA NEW;
INPUT _TYPE_ $  _ROW_ $ _COL_ $ _COEF_;
CARDS;
:             :
:             :
INTEGER   INTVAR      PRODUCT1        1
.         INTVAR      PRODUCT2        2
UPPERBD   PRD1MAX     PRODUCT1       10
UPPERBD   PRD2MAX     PRODUCT2       10
;
PROC LP SPARSEDATA ;
TITLE 'LP PROBLEM – INTEGER VARIABLES';
RUN;
```

In some problems, some or all decision variables need to be specified as binary variables. Binary variables can only take on the integer values of 0 or 1. An example of binary programming is a problem which deals with capital budgeting. In such a problem, several projects are available for funding and only some, but not all, of the projects will likely be chosen for funding. In this case, each project is defined as a binary decision variable taking on the values of 0 or 1 in the optimal solution. A value of 0 implies the project is not funded, while a value of 1 implies that the project is funded. Binary variables are specified in the input data set similar to the manner in which integer variables are specified. Because binary variables can only take on the values of 0 or 1, no upper bound statements are required in this case.

```
DATA NEW;
INPUT _TYPE_ $  _ROW_ $ _COL_ $ _COEF_;
CARDS;
:             :
:             :
BINARY    BINVAR      PRODUCT1        1
.         BINVAR      PRODUCT2        2
;
PROC LP SPARSEDATA ;
TITLE 'LP PROBLEM – BINARY VARIABLES';
RUN;
```

PROC LP SOLUTIONS IN SAS/OR

For any linear programming problem, there are four types of solutions which can be obtained. Two solution types, single optimal solution and multiple optimal solutions, can occur frequently in solving linear programming problems. The other two solution types, infeasible solution and unbounded solution, are indications that the problem has been formulated incorrectly and changes need to be made in the model.

SINGLE OPTIMAL SOLUTION

The solution to the example problem presented above is an illustration of the solution to a linear programming problem with a single optimal solution. There is only one combination of the decision variables which would yield the maximum profit obtainable. For this problem, that combination is Product 1 = 2 and Product 2 = 6 yielding a maximum profit of $36,000. Any other feasible solution combination will yield a profit level less than this value.

MULTIPLE OPTIMAL SOLUTIONS

In many cases, the optimal solution to a linear programming problem may have multiple optimal solutions. In this situation, there are more than one combination of decision variable values which all yield the same optimal objective function value. Such situations exist when one or more of the linear constraints has the same slope as the objective function. For example, if the profit per unit for Product 2 was $2,000 instead of $5,000, the objective function would have the same slope as the constraint for Plant 3. An optimal solution can still be obtained for the problem, although only one of the multiple optimal solutions will be listed in the output. Multiple, or alternate, optimal solutions for linear programming problems solved with the PROC LP procedure can be identified by checking the 'Status' column in the Variable Summary for an indication of alternate optimal solutions as shown below.

```
                    LP PROBLEM - MULTIPLE OPTIMAL SOLUTIONS


                               The LP Procedure

                               Variable Summary

                Variable                                            Reduced
            Col Name     Status Type         Price  Activity           Cost

              1 PRODUCT1 BASIC  NON-NEG          3         4              0
              2 PRODUCT2 BASIC  NON-NEG          2         3              0
              3 PLANT1   ALTER  SLACK            0         0              0
              4 PLANT2   BASIC  SLACK            0         6              0
              5 PLANT3          SLACK            0         0             -1
```

INFEASIBLE SOLUTION

An infeasible solution is an indication that the linear programming problem has been incorrectly specified. The solution is infeasible because there is no combination of the decision variables which can satisfy all constraints as they are correctly formulated. In these cases, the model formulation should be reviewed for possible misspecification. Infeasible solutions are identified in the SAS output by an error message in the Log window and a notification in the Output window. For example, in the example problem illustrated here, if we added a fourth constraint, labeled as PLANT3A, of $x1 \geq 5$, while leaving in constraint 3 of $x1 \leq 4$, we would have an infeasible solution, since no combination of the two decision variables can satisfy both constraints. The error message in the Log window and the Constraint Summary would indicate this infeasible solution as follows.

```
ERROR: Infeasible problem.  Note the constraints in the constraint summary
       that are identified as infeasible.  If none of the constraints are
       flagged then check the implicit bounds on the variables.
```

```
                 LP PROBLEM - INFEASIBLE SOLUTION
                        The LP Procedure

                       Constraint Summary

          Constraint           S/S                           Dual
      Row Name        Type     Col       Rhs  Activity  Activity

        1 PROFIT      OBJECTVE   .         0        12         .
        2 PLANT1      LE         3         4         4         3
        3 PLANT2      LE         4        12         0         0
        4 PLANT3      LE         5        18        12         0
    *INF* PLANT3A     GE         6         5         4         0
```

UNBOUNDED SOLUTION

A solution to a linear programming problem which is specified as unbounded is an indication that a constraint which needs to be in the model formulation has been left out. In these situations, the solution is unbounded because one or more of the decision variables can be increased with no constraint limiting the amount of that increase. Unbounded solutions are identified in the SAS output by an error message in the Log window and a notification in the Output window. For example, in the example problem illustrated here, if constraints two and three (Plants 2 and 3) were eliminated from the problem, the production of Product 2 could be increased indefinitely, resulting in an unbounded solution. The error message in the Log window and the Constraint Summary would indicate this infeasible solution as follows.

```
ERROR: Unbounded objective.  Note the variable in the variable summary
       that is identified as unbounded.
```

```
                  LP PROBLEM - UNBOUNDED SOLUTION
                         The LP Procedure

                         Variable Summary

         Variable                                            Reduced
     Row Name        Status  Type       Price    Activity       Cost

       1 PRODUCT1            NON-NEG        3           0          3
    UBD* PRODUCT2            NON-NEG        5           0          5
       3 PLANT1       BASIC  SLACK          0           4          0
```

PROC LP SOLUTION OPTIONS IN SAS/OR

The PROC LP procedure has several options which can be utilized to provide additional data control, analysis and information. These options include data set options, display control options, interactive control options, preprocessing control options, branch-and-bound algorithm control options, sensitivity/parametric/ranging control options, and simplex algorithm control options. A few of the more important PROC LP options utilized in the analysis of linear programming problems are presented here. These options can be used with either dense data or sparse data input format.

ITERATION SUMMARY

The optimal solution output from the PROC LP procedure includes information and values of variables at the end of the last iteration performed by the simplex algorithm. SAS/OR does provide an option to print out limited information for each simplex method iteration. The FLOW option requests that a journal of pivot information (the iteration log) be displayed after each iteration calculation. This includes the names of the variables entering and leaving the basis, the reduced cost of the entering variable, and the current objective function value.

The program statement calling the FLOW option would be written as follows:

```
PROC LP SPARSEDATA FLOW;
```

The output data log would include an additional section reporting the iteration log. For the example problem presented here, two corner-point feasible solutions were evaluated by the simplex method.

```
                    LP PROBLEM - ITERATION LOG
                         The LP Procedure

                           Iteration Log

                     Entering Leaving                  Objective
      Phase Iteration Variable Variable Reduced Cost       Value

          2         1 PRODUCT2 PLANT2       5.000000          30
          2         2 PRODUCT1 PLANT3       3.000000          36
```

OBJECTIVE FUNCTION SENSITIVITY ANALYSIS

Once an optimal solution for a linear programming problem is obtained, an important next step is to evaluate how sensitive the optimal solution is to changes of the parameters in the problem. The two most important groups of parameters to evaluate are the objective function coefficients and the right-hand-side values. The RANGEPRICE option is used with PROC LP to conduct sensitivity analysis of the objective function coefficients.

The program statement calling the RANGEPRICE option would be written as follows:

```
PROC LP SPARSEDATA RANGEPRICE;
```

Sensitivity analysis of the objective function coefficients provide the range over which each parameter could vary while leaving the optimal solution (values of decision variables) unchanged. Objective function sensitivity analysis results for our example problem is presented below. Results indicate that the profit per unit for Product 1 could vary from $0 to $7,500 and out optimal solution would remain the same. Results for Product 2 indicates and increases in profit per unit for Product 2 would not change the optimal solution. In both cases, the value of the objective function would change with changes in objective function coefficients, but the optimal solution (quantities of Product1 and Product 2) would not change within these ranges.

```
           LP PROBLEM - OBJECTIVE FUNCTION SENSITIVITY ANALYSIS
                           The LP Procedure

                         Price Range Analysis

    Variable ---------Minimum Phi-------- ---------Maximum Phi--------
Col Name         Price Entering Objective     Price Entering Objective

  1 PRODUCT1         0 PLANT3          30       7.5 PLANT2          45
  2 PRODUCT2         2 PLANT2          18  INFINITY .         INFINITY
  3 PLANT1        -4.5 PLANT2          27         3 PLANT3          42
  4 PLANT2   -INFINITY .               36       1.5 PLANT2          36
  5 PLANT3   -INFINITY .               36         1 PLANT3          36
```

RHS SENSITIVITY ANALYSIS

Sensitivity analysis of the right-hand-side values provides information on how sensitive the optimal solution is to changes in those values. For many linear programming problems which deal with production, those right-hand-side values often times represent quantities of inputs and labor available. Since it is possible to obtain more of these quantities, it is important to evaluate the sensitivity of the optimal solution to changes in those values.

Right-hand-side sensitivity analysis is performed using the RANGERHS option, written as follows:

```
PROC LP SPARSEDATA RANGERHS;
```

Output from the right-hand-side sensitivity analysis provides the range over which each right-hand-side value could vary with the optimal solution remaining feasible. For example, in the problem presented here, the quantity of labor at Plant 2 (originally at 12) could vary anywhere between 6 and 18 and the optimal solution would remain feasible. Outside of this range, combinations of Product 1 and Product 2 are not feasible because one or more of the model constraints would be violated.

LP PROBLEM - RHS SENSITIVITY ANALYSIS

The LP Procedure

RHS Range Analysis

	---------Minimum Phi--------			---------Maximum Phi--------		
Row	Rhs	Leaving	Objective	Rhs	Leaving	Objective
PLANT1	2	PLANT1	36	INFINITY	.	.
PLANT2	6	PLANT1	27	18	PRODUCT1	45
PLANT3	12	PRODUCT1	30	24	PLANT1	42

PARAMETRIC PROGRAMMING

Sensitivity analysis provides estimates of the range over which objective function values and right-hand side values can vary with the optimal solution remaining unchanged. However, you may want to vary one of these parameters over some range and see how the optimal solution changes. This is most commonly used when incorporating price or income risk into the analysis. SAS/OR allows for parametric programming of both objective function coefficients and right-hand-side values.

To parametrically vary the objective function value for the example problem, one additional line of input data is required to be added along with an option in the PROC LP statement. As shown below, the PRICESEN statement indicates that the objective function value of Product 2 is to be parametrically varied from a starting value of 1.0. The PRICEPHI option in the PROC LP statement indicates that the objective function coefficient is to be varied by a value of 10. The PARAPRINT option tells SAS to print out the solution each time the basis changes.

```
DATA NEW;
INPUT _TYPE_ $  _ROW_ $ _COL_ $ _COEF_;
CARDS;
:              :            :
PRICESEN  PRD2SEN      PRODUCT2       1.0
;
PROC LP SPARSEDATA PRICEPH1=10 PARAPRINT;
TITLE 'LP PROBLEM – OBJECTIVE FUNCTION PARAMETRIC PROGRAMMING';
RUN;
```

To parametrically vary a right-hand-side value for the example problem, one additional line of input data is required to be added along with an option in the PROC LP statement. As shown below, the RHSSEN statement indicates that the right-hand-side value for Plant 2 is to be parametrically varied from a starting value of 1.0. The RHSPHI option in the PROC LP statement indicates that the right-hand-side coefficient is to be varied by a value of 20. The PARAPRINT option tells SAS to print out the solution each time the basis changes.

```
DATA NEW;
INPUT _TYPE_ $  _ROW_ $ _COL_ $ _COEF_;
CARDS;
:                :             :
RHSSEN    PLANT2     _RHSSEN_       1.0
;
PROC LP SPARSEDATA RHSPH1=20 PARAPRINT;
TITLE 'LP PROBLEM – RHS PARAMETRIC PROGRAMMING';
RUN;
```

GOAL PROGRAMMING

The linear programming examples presented thus far have had a single objective function, that of maximizing profit from the production of two products. In many cases, although maximizing profit or minimizing cost may be the predominant decision to be made, there may also be other goals or objectives which might also be desired. Linear programming problems with more than one objective function are referred to as multiobjective programming or goal programming. These types of problems can be easily solved using PROC LP by converting the multiple objectives to constraints and creating an objective function which includes decision variables representing positive and negative deviations from the stated goals. The same results can also be obtained by using the GOALPROGRAM option with the PROC LP statement.

In our example problem, suppose we have to goals or objectives. The most important goal is to have profit of at least $30,000. A second, less critical goal might be that we would like Plant 1 to be fully utilized with no excess labor. These two goals would be stated as equalities and included in the input data set. Negative and positive variables for each goals would also be included. Since both of these goals desire for profit and Plant 1 labor usage to reach a certain level, the program is set up to let the PROC LP procedure minimize the deviations below these goals. This is accomplished by using the GOALPROGRAM option with the PROC LP statement. An optimal solution is found for the most critical goal first, then a secondary solution is found to achieve the lower tier goal. The SAS input data set for the goal programming example problem in dense data format would be as follows:

```
DATA NEW:
INPUT _ROW_ $ PRODUCT1 PRODUCT2 N1 N2 P1 P2 _TYPE_ $ _RHS_;
CARDS;
PROFIT      .      .    1    .    .    .   MIN     1
PLT1USE     .      .    .    1    .    .   MIN     2
PROFIT    3.0    5.0    1    .   -1    .    EQ  30.0
PLT1USE   1.0      .    .    1    .   -1    EQ   4.0
PLANT1    1.0      0    .    .    .    .    LE   4.0
PLANT2      0    2.0    .    .    .    .    LE  12.0
PLANT3    3.0    2.0    .    .    .    .    LE  18.0
;
PROC LP GOALPROGRAM;
TITLE ‘LP PROBLEM – GOAL PROGRAMMING’ ;
RUN;
```

USE OF ACCOUNTING ACTIVITIES

One extremely useful procedure which can be employed when solving linear programming problems with SAS/OR is the use of accounting activities. These accounting activities are basically variables which are added to the problem for purposes of tabulating additional information about the problem solution. For example, in the sample problem presented here, the only financial information about the optimal solution is the objective function value, total profit. We might like to also know additional information, for example, total revenue, total production cost, etc. These types of calculations can be performed within the PROC LP procedure through the use of transfer rows and accounting activities. This procedure will be illustrated using our two-product profit maximization example.

TRANSFER ROWS

Suppose that we would like to add accounting activities into our problem which would calculate total revenue, total cost, and total profit for us within the PROC LP procedure. In order to do this, we first need to define two variables for each product, one representing quantity produced and the other representing quantity sold. For Product 1, let’s define the variable X1P as quantity of Product 1 produced and X1S as quantity of Product 1 sold. The same variable specification would be conducted for Product 2, X2P representing quantity produced and X2S representing quantity sold. The mathematical formulation of this revised model would be as follows:

$$
\begin{array}{llllllll}
\text{Max} & Z = -3X1P & -5X2P & +6X1S & +10X2S & & & (1) \\
\text{s.t.} & & & & & & & \\
& X1P & & & & \leq & 4 & (2) \\
& & 2X2P & & & \leq & 12 & (3) \\
& 3X1P & +2X2P & & & \leq & 18 & (4) \\
& X1P & & -X1S & & = & 0 & (5) \\
& & X2P & & -X2S & = & 0 & (6)
\end{array}
$$

The objective function values for X1P and X2P represent costs per unit and the objective function values for X1S and X2S represent selling prices. These values were chosen to result in the same profit per unit as in the original problem. Two additional constraints were added, (5) and (6), to enforce the condition that quantity produced equal quantity sold. These constraints are examples of typical transfer rows utilized in linear programming models. There function is to transfer values from one or more variables to one or more other variables in the model. The optimal solution for this model is the same as the original problem, with optimal total profit of $36,000 and quantities of Product 1 and Product 2 produced and sold of 2 and 6, respectively.

ACCOUNTING ACTIVITIES

To add accounting activities to calculate total revenue, total cost and total profit within the PROC LP procedure, three additional constraints are required to be added to the model, one for each accounting activity. Variables and coefficients for these constraints are taken from the objective function as listed below:

$$
\begin{array}{lllll}
6X1S + 10X2S - TOTREV & & = & 0 & (7) \\
3X1P + 5X2P - TOTCOST & & = & 0 & (8) \\
-3X1P - 5X2P + 6X1S + 10X2S - PROFIT & & = & 0 & (9)
\end{array}
$$

The purpose for the addition of these accounting activities is to have total revenue calculated and recorded in the variable TOTREV, total cost calculated and recorded in the variable TOTCOST, and have total profit calculated and recorded in the variable PROFIT. This additional information related to the optimal solution for the problem will be listed in the Variable Summary of the PROC LP output as shown below. Total revenue is $72,000, total cost is $36,000 and total profit is $36,000, with 2 units of Product 1 and 6 units of Product 2 produced and sold.

```
LP PROBLEM – TRANSFER ROWS AND ACCOUNTING ACTIVITIES
                  The LP Procedure

                  Variable Summary
```

Row	Variable Name	Status	Type	Price	Activity	Reduced Cost
1	PROFIT	BASIC	NON-NEG	0	36	0
2	TOTCOST	BASIC	NON-NEG	0	36	0
3	TOTREV	BASIC	NON-NEG	0	72	0
4	X1P	BASIC	NON-NEG	-3	2	0
5	X1S	BASIC	NON-NEG	6	2	0
6	X2P	BASIC	NON-NEG	-5	6	0
7	X2S	BASIC	NON-NEG	10	6	0
8	PLANT1	BASIC	SLACK	0	2	0
9	PLANT2		SLACK	0	0	-1.5
10	PLANT3		SLACK	0	0	-1

CONCLUSIONS

Linear programming is a widely used and extremely useful tool for evaluating a wide variety of business problems. Objective functions and structural constraints can be easily formulated to address very specific types of business decisions and applications. The PROC LP procedure in SAS/OR has the capability of solving several types problems which can be formulated as linear, integer, or mixed integer programs. Options are available which can provide additional information and analysis on problems being solved, including objective function and right-hand-side sensitivity analysis, parametric programming and goal programming. The SAS/OR software package also includes other SAS procedures which are designed to solve other types of typical operations research problems.

REFERENCES

Hillier, Frederick S., and Gerald J. Lieberman (2001), *Introduction to Operations Research*, 7th edition, New York, NY: McGraw-Hill Co., Inc.

SAS Institute (2004), *SAS/OR® 9.1 User's Guide: Mathematical Programming*, Cary, NC: SAS Institute, Inc.

Winston, Wayne L. (2004), *Operations Research: Applications and Algorithms*, 4th edition, Belmont, CA: Brooks/Cole Thomson Learning.

CONTACT INFORMATION

Michael E. Salassi
Department of Agricultural Economics & Agribusiness
Louisiana State University
101 Agricultural Admin. Bldg.
Baton Rouge, LA 70803
Phone: (225) 578-2713
Fax: (225) 578-2716
Email: msalassi@agcenter.lsu.edu