A DLP Primer

J.L. Nazareth
(May, 2017)

© 2017 John Lawrence Nazareth

1. Introduction

This Primer has two main aims: firstly, to *provide an elementary introduction* to the monograph *DLP and Extensions: An Optimization Model and Decision-Support System* [1] by means of a simple, illustrative application; and secondly, to *promote use of its open-source, prototype implementation DLPEDU* [2] as a platform for further development and application by researchers and entrepreneurs.

We begin in Sec. 2 with a hypothetical and very simple vineyard-planning problem in order to introduce the DLP optimization model and decision-support system and *illustrate* its potential for application to viticulture, in particular, and to other natural and renewable resources, in general. We have devised this problem for convenience of illustration and exposition and we therefore make highly-simplified assumptions on vine-growth characteristics and associated costs and benefits. (Nevertheless, it is our hope that the illustration is not completely unrealistic in the eyes of viticulture experts.) A DLP model for this simple vineyard planning problem is formulated and then optimized using the DLPEDU *prototype implementation* for decision-support presented in the E-Book [2], which is based on DLP techniques developed in [1]. This is followed by a brief discussion of the implications of this illustration for practical viticulture resource planning, which we believe might be an ideal application area for the DLP decision-support system.

The strengths and weaknesses of the DLPEDU prototype implementation are then described in Section 3, along with the current status of DLP and its extensions as presented in [1].

Finally, Section 4 briefly describes usage of DLPEDU for further research, development, and application of the DLP optimization model, algorithm, and decision-support system, and outlines its entrepreneurial potential.

2. A Simple Illustration

Imagine a 20-acre mature vineyard in Washington's wine district that yields 3 tons of grapes per acre during an "average" year, i.e., the total yield or "benefit" each year is 60 tons. The harvested grapes are sold under contract to a neighboring vintner, who accepts a little less or more tonnage when weather and other circumstances differ from the average. The cost of maintaining the vineyard each year is 1 CU/acre, where, for convenience, CU is used to denote a unit of cost (in reality it would be specified in, say, hundreds or thousands of dollars). The total cost of maintaining the vineyard each year is thus 20 CU.

The farmer of the vineyard has obtained financing which will permit him to replace the current, low-quality vines (type-1) with new and much more productive plants (type-2) of the same grape variety. At maturity, these would double the expected yield to 6 tons of fruit per acre in an average year. Meanwhile, the cost of maintaining the replanted vineyard at maturity is expected to remain much the same, i.e., 1 CU per acre. The vintner who currently purchases the fruit is willing to accept the disruption and eager to accept the higher output each year and double her production of wine once the new grapes become available. But she stipulates that her winery must be provided with at least 30 tons, i.e., half the current contract of 60 tons, during each of the first three years of transition, and at least 45 tons during the fourth year. (She will be able to purchase additional supply of grapes as needed elsewhere.) Thus the farmer's vineyard cannot be replanted all at once; and neither does he wish to do so, since he prefers to distribute his costs and receive some income during the transition. After

the four-year transition period, the vintner wants to receive the full production of 120 tons of grapes each year by which time she will have ramped up her own wine production capacity, i.e., the vineyard must be completely replanted and the vines attain maturity by the end of the fourth year.

Let us say that the cost of replanting---refurbishing the trellis, purchasing suitable type-2 plants, maintaining the new vines in the year of planting, and so on---is 4 CU per acre. The normal way to bring type-2 vines to maturity requires two additional years, or growing seasons, during which the cost of maintenance each year is, say, 1.5 CU /acre. There is another option available at the end of a replanting year, namely, during the subsequent year use enhanced fertilizing and watering, which would bring the replanted acreage to maturity at the end of that season; i.e., if desired, a replanted acre of vineyard can be brought to maturity in two years rather than three. The cost of this enhanced maintenance option is 2.5 CU/acre. Prior to coming to maturity, type-2 vines on acres that have been replanted are not permitted to ripen grapes. So the benefit, or yield, of any such replanted acreage prior to attaining maturity is nil. (Note also that harvesting costs are addressed separately, outside this simple model.)

The original vineyard contains type-1, mature vines so each acre exists in only a single state, which we well denote by ST1M. (`ST' is an abbreviation of `STATE' and the last two letters indicate that an original acre contains type-1, Mature vines.) In contrast, a replanted acre can exist in three different states, corresponding to type-2 vines after one, two and three years of growth. (Recall the two options mentioned above.) We will denote these three possibilities by the symbols N, Y and M, meaning New, Young, and Mature, and thus the three possible states of type-2 vines on a replanted acre will be denoted by ST2N, ST2Y, and ST2M, respectively.

On any acre of the vineyard, the actions that are available to manage each of the foregoing states, their associated costs/acre, benefits/acre, and the

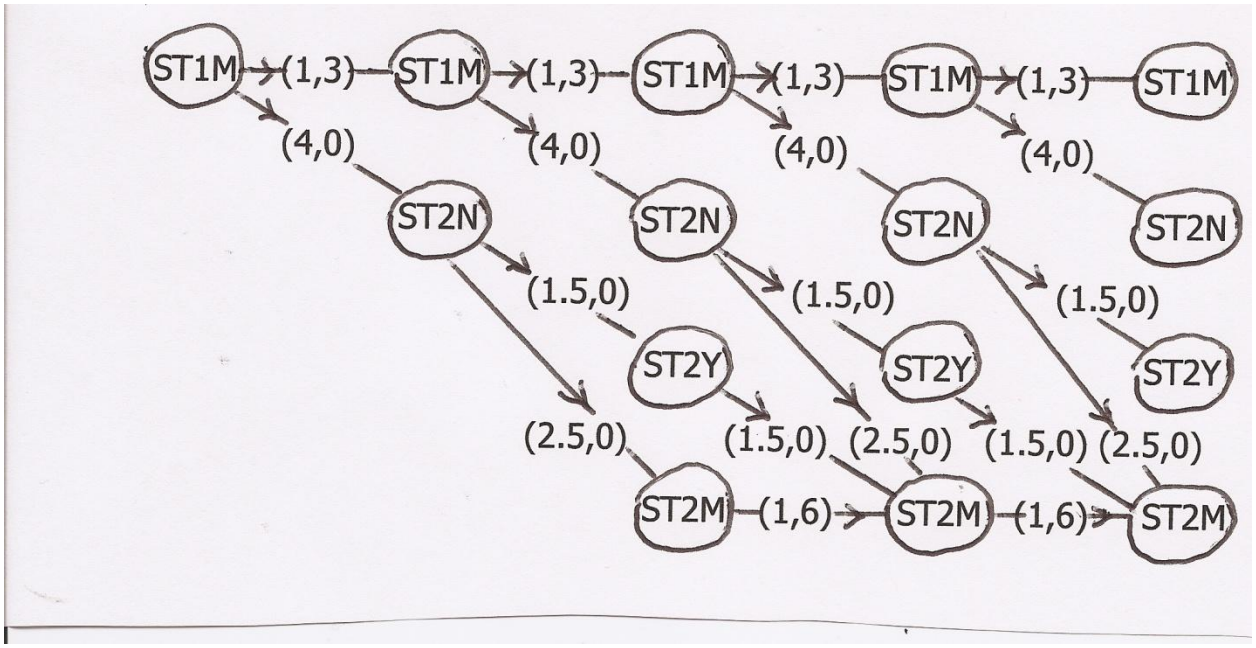
states that result from the actions are given in the following table (see below for further explanation):

State	***Action***	***Cost /acre***	***Benefit/acre***	***New State***
ST1M	MAINT1M	1	3	STIM
ST1M	REPLANT	4	0	ST2N
ST2N	MAINT2N	1.5	0	ST2Y
ST2N	ENHANC	2.5	0	ST2M
ST2Y	MAINT2Y	1.5	0	ST2M
ST2M	MAINT2M	1	6	ST2M

Table 1

Thus, in the first line of the table, any acre of the vineyard that is in state ST1M can be maintained under an action called MAINTM, which costs 1 CU per acre, yields a benefit of 3 tons per acre, and preserves that acre in state ST1M. Again, as specified in the second line of the table, an acre in state ST1M can be replanted at a cost of 4 CU, it yields no benefit, and it results in state ST2N. In the fourth line, an acre in state ST2N can be treated with the enhanced fertilizing and watering action ENHANC, which costs 2.5 CU, yields no benefit, and results in state ST2M (mature type-2 vines). And so on.

Let us now consider a sequence of actions over a planning horizon of four years.



Take any acre of the vineyard, which is in state ST1M at the outset. The sequences of possible actions over the four years for which a management plan needs to be developed and the resulting states at the end of each year form a "decision-maker network", which is depicted above. Each path in the network corresponds to a "management activity" with an associated sequence, or flow, of costs and benefits. For example, an acre in state ST1M could be maintained in the same state during the first year, replanted in the second year which results in state ST2N, enhanced in the third year, which results in the state ST2M (where the type-2 vines are mature), and then maintained in this state during the fourth year. This particular sequence of actions, or management activity, applied to any acre of the vineyard would yield 3 tons of grapes in the first year, nothing in the second and third years, and 6 tons in the final year. It is easily seen that the network contains *eight* different management activities, each of which can be applied to any acre of the vineyard.

The farmer wishes to establish some constraints on any management plan for his 20-acre vineyard. At the end of the first year he requires that at least 5 acres of the vineyard be planted with type-2 vines, at the end of the second year at least 10 acres must contain type-2 vines, at the end of

the third at least 15 acres, and at the end of the fourth year the vineyard must be entirely replanted and mature. As we have noted earlier, the vintner has established her own constraints: she requires that the *total* benefit, or yield, during each of the first three years of transition be at least 30 tons and she needs at least 45 tons in the fourth year. The question then becomes what subset of management activities to choose and how many acres to manage with each selected management activity in order to satisfy all the foregoing constraints and *minimize total cost*. This is the task for which the DLP decision support system (DSS) has been designed.

The foregoing data concerning states, actions with their associated costs and benefits, decision-maker network, constraints and objective, constitute the DLP optimization model of the vineyard resource-decision problem, and this must be input to the DSS. The DLP system provides a user with an *extremely easy to learn* modeling language called DLPFI with which to create the model's input file in a very straightforward manner. Table A-1 of the Appendix shows this input file for the vineyard problem. (Any line in the DLPFI input file that begins with an asterisk is a comment, and we have interspersed comments liberally in order to provide some feeling for the modeling language. A user will usually not need to comment as heavily.) The DLP vineyard model's input data is then fed to the decision support system's optimization algorithm, which produces an output file that contains the optimal management plan. This is shown in Table A-2 of the Appendix. (Again we have inserted comments liberally, i.e., lines that begin with an asterisk.) *This optimal plan for even a vineyard problem as simple as the foregoing is far from obvious*. In summary, it is as follows:

On 5 acres: replant with type-2 vines in the first year, use enhanced fertilizing and watering in the second year to bring them to maturity, and then maintain the mature type-2 vines in the third and fourth years.

On 10 acres: maintain the type-1 vines in the first and second years, replant in the third year, and use enhanced fertilizing in the fourth year to bring the type-2 vines to maturity at the end of the planning period.

On 2.5 acres: maintain the type-1 vines in the first year, replant in the second year, and then maintain the type-2 new and young vines normally in the third and fourth years, respectively, bringing the vines to maturity at the end of the planning period.

On 2.5 acres: replant in the first year, maintain the type-2 new and young vines in the second and third years, bringing them to maturity, and then maintain the mature type-2 vines in the fourth year.

Before deciding on a final management strategy, the farmer would very likely want to run several other scenarios, i.e., variants on the foregoing DLP model. For example, instead of setting constraints on benefit levels in each of the four planning years and minimizing total cost, he might want to explore the converse: set constraints on the cost (CUs that he is willing to spend during each of the four planning years) and *maximize* total benefit. He might seek to vary the constraints on states (the LOCAL constraints in Table A-1), run different cost and benefit estimates for the actions, and so on. All such scenarios require simple modifications to the input file given in Table A-1 and each run of the DLP system would produce an optimal plan akin to Table A-2. A final management strategy for the vineyard that is best suited to the farmer's needs would eventually emerge from such explorations.

The foregoing contrived vineyard planning problem *illustrates* the DLP approach to decision support and the *form* that a DLP model might take within the context of viticulture. It serves to highlight the potential that the DLP decision support system holds out for viticulture resource planning in practice. Practical models could be much more complicated and range from small-scale, through medium-scale, to very large-scale; i.e., models for real-life vineyard resource planning problems could run the gamut from having a single resource class, a few states and actions, a few local and global constraints, and a short planning horizon, as in the illustration, to

the other extreme of having several different resource classes (each modeled by a decision network that has hundreds of states transformed through dozens of actions, resulting in millions, if not billions, of possible management activities), many local and global constraints, and an extended planning horizon.

The development of such models at any scale requires close interaction with vineyard growers and specialists in viticulture, i.e., people who have a solid understanding of the planning needs of small, medium or large-scale vineyards and have access to the data required by a DLP model or the know-how to generate such data. The importance of interacting and developing relationships with those who are intimately involved in the application area---in order to identify planning needs in practice, *evolve* a DLP optimization model from "back-of-the-envelope" to practical, and supply or generate the necessary data---is often underappreciated, as is the fact that a DLP optimization model for resource planning can only *supplement and not replace* expert intuition.

The viticulture resource-decision planning might be the *ideal* application area for the DLP decision support system. (Viticulture is *not* one of the many resource examples discussed in [1].) The Wine Research Center, which has been newly established by Washington State University in Richland, WA, would be a good source for the necessary viticulture expertise and data; see, for example, [4].

3. Strengths and Weaknesses of the DLPEDU Prototype

The *strengths* of the DLP decision-support system and its *prototype implementation* DLPEDU are as follows:

1. It uses very powerful optimization techniques based on linear and dynamic programming coordinated through the optimization strategy known as Dantzig-Wolfe decomposition. However, these techniques

can remain in the background and the casual user of the decision support system need have no knowledge of them.

2. It can handle problems ranging from small to very large scale across a wide range of potential applications: *agriculture*, *forestry*, *viticulture and enology, rangeland management, infrastructure maintenance, land reclamation, fish-farming, and so on.*
3. It provides a simple and very easy-to-learn *modeling language*, or formatting interface, called DLPFI. This serves two purposes. For small-scale models it can be used to formulate a model directly, as in the foregoing illustration. For larger-scale models, DLPFI serves as an interface or target for a user-supplied model-generating preprocessor. Similarly the optimal solution is presented as output in a standardized format and it can be read directly by the user, again as in the foregoing vineyard illustration, or a user-supplied post-processor can be used to present the optimal solution in a more convenient way.
4. The design of the DLP system intentionally encourages the progressive "evolution" and scaling-up of a DLP model from simple to increasingly complex.
5. Finally, the prototype implementation described in [1] [2] was designed to reflect *all* the foregoing strengths; i.e., it is currently capable of running *large-scale* resource decision problems very efficiently. The DLPEDU prototype is a large and complicated implementation that involves several thousand lines of code. It has been carefully written and has extensive, in-line commenting and documentation.

Nevertheless, DLPEDU suffers from the following significant *weaknesses*:

1. It was created using the Fortran-77 computer language, an early Microsoft Fortran-77 compiler, and the linear programming package LPKIT [6]. While Fortran-77 remains, to the present day, a popular

language for scientific computing---for a discussion, see, for example, https://www.arstechnica.com/science/2014/05/scientific-computings-future-can-any-coding-language-top-a-1950s-behemoth/---there is considerable room for refining and updating the implementation; for example, by using a more recent dialect of the Fortran language that facilitates memory management or, alternatively, some version of the C language, and by incorporating a more recent linear programming package. In other words, the prototype DLPEDU can provide the starting point for a progressive refinement of the decision-support system.

2. The *run-time user interface* of the prototype implementation, as contrasted with the *modeling-language interface* DLPFI, is very *primitive* and is based on using a text editor to create the input file of a DLP model and the Command Prompt of Windows. If DLPEDU is to be of practical use to resource planners then this run-time interface to the user is where significant improvement is needed.
3. The focus is on applications with *deterministic* data. Stochastic (probabilistic data) extensions of DLP, which were formulated and illustrated in [1] for infrastructure management problems, in particular, pavement maintenance applications, are not available in the DLPEDU, the prototype DLP implementation. Such extensions could also be very useful in viticulture applications.

The current status of the DLP decision support system and its prototype implementation (DLPEDU) is as follows: the model, algorithm, and implementation have been fully described in [1], which also has an attached CD-ROM containing a *demonstration version* of the prototype implementation, an app called DLPDEMO. (This corresponds to a compiled-and-executable version the prototype DLPEDU, with severe restrictions placed intentionally on the sizes of problems that can be run.) Background on optimization models and algorithms and additional usage instructions for DLPDEMO can also be found in [3]. A simple, illustrative timber-harvesting, i.e., silviculture, application is described, in detail, in [5]

(Chapter 7, Sec. 7.2). A variety of natural and renewable resource-management applications are discussed in [1] for purposes of illustration.

4. Opportunities for Further R&D and Entrepreneurship

The DLPEDU prototype listed in [2] provides the opportunity for developing a robust and easily-usable DLP app (an executable program for PCs and tablets, henceforth called DLPAPP) that could then be used to create and optimize models of natural or renewable resource-management problems matching the standard DLP paradigm developed in [1]. The design of a *good run-time user interface*, which facilitates the creation of a model's DLPFI input file (perhaps even interactively), followed by the optimization of the specified model and the presentation of the optimal solution in a convenient way, will all be of critical importance here. An app along these lines would be especially useful for situations where a DLP model requires a relatively small amount of data and the correspondingly modest-in-size DLPFI file can be created with ease (as in the foregoing vineyard illustration). Such an app could equally well be used for medium- or large-scale DLP models, but in that case its user may need independently to implement suitable pre-processing and post-processing programs to create the (possibly large) DLPFI file and convert the DLP output file containing the optimal solution into a more usable form. DLPAPP would also be useful for educational purposes, for example, in conjunction with the optimization primer [3].

It will often be the case that the standard DLP model as implemented in the foregoing DLPAPP will not exactly fit the needs of a particular resource-management situation. In that case there will necessary to enhance this model and/or optimization algorithm and create the associated executable program (app). Making available the Fortran-77 source code DLPEDU [2] as a well-documented, *open-source* code listing, subject only to the weak restrictions of the GNU General Public License of the Free Software Foundation, will permit others to further develop this source code, re-

implement it if desired, for example, in more recent versions of Fortran or a version of C, and it will facilitate the development of DLP variants and extensions as needed by particular applications. The open-source software listing can also be used for educational purposes on implementation techniques and for further research into the DLP and Extensions approach [1], especially in a university setting.

In this regard it is useful to note that while modification and redistribution of the open-source software by others will be subject to the constraints imposed by the GNU License, any application-specific data together with pre- or post-processing programs, which for medium or large-scale applications with extensive data requirements could well exceed the open-source code in size, and any particular DLP model/algorithm extensions of DLPEDU, for instance, stochastic variants, will *not* be so constrained, and they can remain the property of their developers. While the open-source implementation DLPEDU must remain open and freely available, it nevertheless provides the basis for any entrepreneur world-wide to create a revenue-producing stream addressed to particular natural or renewable resources.

6. References

[1] Nazareth, J.L. (2001), *DLP and Extensions: An Optimization Model and Decision Support System*, Springer, Berlin and Heidelberg, Germany, xvii+207 pgs. & CD-ROM.

[2] Nazareth, J.L. (2017), *DLPEDU: A Fortran-77 Open-Source Prototype Implementation of the DLP Optimization System*, vii+207 pgs. Available as an E-Book in PDF format at www.math.wsu.edu/faculty/nazareth/bio.html (click on the link provided near the bottom of this web-page).

[3] Nazareth, J.L. (2004), *An Optimization Primer: On Models, Algorithms, and Duality*, Springer, New York, 108 pgs.

[4] Komm B. and Moyer, M.M. (2015), "Vineyard yield estimation," Washington State University Extension Bulletin EM086E, 10 pgs.

[5] Nazareth, J.L. (2012), *Numerical Algorithmic Science and Engineering: Foundations and Organization,* PSIPress, Portland, Oregon, USA, 166 pgs. (available at www.math.wsu.edu/faculty/nazareth/NumAlg_19Jul12.pdf).

[6] Nazareth, J.L. (1986), "Implementation aids for optimization algorithms that solve sequences of linear programs," *ACM Transactions on Mathematical Software*, 12, 307-323.

Appendix

This appendix contains two tables, each extending over *two* pages.

The first table is referred to as Table A-1 in the above article and it shows the input file for the vineyard problem. Every line that begins with an `*' is a comment and has been inserted for purposes of explanation. The input file itself would be much briefer when these comments are deleted.

The second table is referred to as Table A-2 and it shows the output file containing the optimal solution. Again comments have been inserted for clarity. This optimal solution is summarized in the body of the article, near the end of Section 2.

```
* The problem is given a name
?PROBLEM   VINE
*
* The number of planning periods is specified, in this case, 4 years
*
?PERIODS
   4
*
* The names of the resource classes are listed, in this case,
* a single resource class called VINEYARD
*
?CLASSES
  VINEYARD
*
* The data needed to specify the decision network for VINEYARD comes next
*
?NAME VINEYARD
*
* First the names of the states for resource class VINEYARD are listed
*
?STATES  VINEYARD
  ST1M ST2N ST2Y ST2M
*
* Then comes the list of management actions
*
?ACTIONS VINEYARD
 MAINT1M REPLANT MAINT2N MAINT2Y ENHANC MAINT2M
*
* Next the transformations as in Table 1, Section 3 of the writeup
*
?NETWORK VINEYARD
  ST1M MAINT1M 1.0 3.0 ST1M
  ST1M REPLANT 4.0 0.0 ST2N
  ST2N MAINT2N 1.5 0.0 ST2Y
  ST2N ENHANC 2.5 0.0 ST2M
  ST2Y MAINT2Y 1.5 0.0 ST2M
  ST2M MAINT2M 1.0 6.0 ST2M
*
* The constraints on states are then specified
*
?LOCAL VINEYARD
*
* First, the size of the vineyard
*
?INITIAL VINEYARD
  ST1M=20.
*
* and then the farmer's constraints of how much acreage
* must contain type-2 vines each year
*
?CONSTRAINTS VINEYARD
  1: ST2N + ST2Y + ST2M >= 5.0
  2: ST2N + ST2Y + ST2M >= 10.
  3: ST2N + ST2Y + ST2M >= 15.0
*
* this final constraint that the entire acreage must be covered
```

```
* with mature, type 2 vines at the end of he planning horizon
* makes allowance for rounding error during optimization
*
  T: ST2M >= 19.999999
*
* Finally the constraints imposed by the vintner on the tonnage
* each year are specified
*
?GLOBAL
  B[VINEYARD:1] >= 30.
  B[VINEYARD:2] >= 30.
  B[VINEYARD:3] >= 30.
  B[VINEYARD:4] >= 45.
*
* the objective is defined
*
  C[:]=ZCOST
*
* and then stated to be optimized
*
?OBJECTIVE
  MINIMIZE ZCOST
*
* This marks the end of the data. Normally comments in lines
* that begin with an asterisk would not be as extensive.
*
?ENDATA
```

```
?PROBLEM   VINE
?NAME  VINEYARD
*
* Five alternatives, or management activities, have been selected
* in the optimal plan
*
?ALTERNATIVES       5
*
* The first is applied to 25% of the 20 acres, i.e., manage 5 acres
* with the actions shown in the first column. Their associated
* costs and benefits and the resulting sequence of states are given in
* subsequent columns.
*
?CLUSTER   2.50000E-01
?INITIAL  ST1M         2.00000E+01     5.00000E+00
 REPLANT      4.00000E+00     0.00000E+00    ST2N
 ENHANC       2.50000E+00     0.00000E+00    ST2M
 MAINT2M      1.00000E+00     6.00000E+00    ST2M
 MAINT2M      1.00000E+00     6.00000E+00    ST2M
*
* The next activity involves only a tiny amount of acreage and can be ignored.
* This is simply a feature of the optimization process.
*
?CLUSTER   5.00000E-08
?INITIAL  ST1M         2.00000E+01     1.00000E-06
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
*
* The second proposed management activity applies to  50% of the
* original acreage, i.e., 10 acres
*
?CLUSTER   5.00000E-01
?INITIAL  ST1M         2.00000E+01     1.00000E+01
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 REPLANT      4.00000E+00     0.00000E+00    ST2N
 ENHANC       2.50000E+00     0.00000E+00    ST2M
*
* The third activity applies to 12.5%, i.e. 2.5 acres
*
?CLUSTER   1.25000E-01
?INITIAL  ST1M         2.00000E+01     2.50000E+00
 MAINT1M      1.00000E+00     3.00000E+00    ST1M
 REPLANT      4.00000E+00     0.00000E+00    ST2N
 MAINT2N      1.50000E+00     0.00000E+00    ST2Y
 MAINT2Y      1.50000E+00     0.00000E+00    ST2M
*
* and the fourth activity also applies to 12.5%, i.e., 2.5 acres
*
?CLUSTER   1.25000E-01
?INITIAL  ST1M         2.00000E+01     2.50000E+00
 REPLANT      4.00000E+00     0.00000E+00    ST2N
 MAINT2N      1.50000E+00     0.00000E+00    ST2Y
 MAINT2Y      1.50000E+00     0.00000E+00    ST2M
```

```
 MAINT2M      1.00000E+00     6.00000E+00    ST2M
*
* This part of the output returns information on the LOCAL constraints
* that were specified in the DLP model input.
*
?LOCAL CONSTRAINTS        4
 -2.50000E+00    G     5.00000E+00
 -2.22045E-16    G     1.00000E+01
 -5.00000E+00    G     1.50000E+01
  0.00000E+00    G     2.00000E+01
*
* This part of the output concerns the GLOBAL constraints
*
?GLOBAL CONSTRAINTS        5
 -7.50000E+00    G            3.00000E+01
  0.00000E+00    G            3.00000E+01
  0.00000E+00    G            3.00000E+01
  0.00000E+00    G            4.50000E+01
 -1.67500E+02    ZCOST        0.00000E+00
*
* This is the optimal objective value (fifth global constraint).
*
?OBJECTIVE        5
  1.67500E+02
?ENDATA
```