

The first two sections of this chapter provides a very brief introduction to the terminology of linear programming, how a problem is represented and solved by the computer, and how the techniques of sensitivity analysis can provide useful information in addition to the solution of the problem. The last section deals with how to formulate linear programming models to deal with practical problems.

Tableaus and Solutions

We will use the example problem from Chapter 1:

$$\begin{aligned} &\text{maximize} && 20x_1 + 30x_2 + 15x_3 \\ &\text{subject to} && \\ & && 1.5x_1 + x_2 + x_3 \leq 100 \\ & && x_1 + 2x_2 + x_3 \leq 100 \\ & && x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

This will go through a number of transformations. First of all, the inequalities are transformed into equalities by the use of **slack variables**: for each inequality constraint there is a slack variable, which expresses the difference between the two sides. We will use the names m_1 and m_2 for the slack variables. LINEAR will call them ‘SLACK M1’ and ‘SLACK M2’, corresponding to the names of the constraints. When you refer to them from the keyboard, you may if you wish omit ‘SLACK’ and just use the name of the constraint. The objective (call it p) can also be written as an equality:

$$\begin{aligned} p &= && 20x_1 + 30x_2 + 15x_3 \\ m_1 &= && 100 - 1.5x_1 - x_2 - x_3 \\ m_2 &= && 100 - x_1 - 2x_2 - x_3 \end{aligned}$$

The objective p in some ways plays the role of a variable, although it is not generally referred to as a variable. Note that the equations above express the objective and the two variables m_1 and m_2 in terms of the other variables x_1 , x_2 and x_3 . The variables m_1 and m_2 , which are expressed in terms of the other variables, are called **basic variables**. The variables x_1 , x_2 and x_3 are called **nonbasic variables**. A set of equations such as these is called a **dictionary**.

A **tableau** represents these equations using an array of numbers. First put all the variables on the left side of the equations:

$$\begin{aligned} p - 20x_1 - 30x_2 - 15x_3 &= 0 \\ m_1 + 1.5x_1 + x_2 + x_3 &= 100 \\ m_2 + x_1 + 2x_2 + x_3 &= 100 \end{aligned}$$

Now, instead of writing down the names of the variables in each equation, just write down the numbers in an array, and label the columns of the array by the variables. There are two ways to do this. In the **explicit basis** format, every variable, basic or nonbasic, has its column:

x_1	x_2	x_3	m_1	m_2	p	
-20	-30	-15	0	0	1	0
1.5	1	1	1	0	0	100
1	2	1	0	1	0	100

The p column is almost always omitted when a tableau is written down, because it will never change. The column of constant terms is usually not labelled, and is separated by a vertical line from the rest of the tableau. This manual will refer to it as the “right-hand-side”, or **RHS**, column.

In the **implicit basis** format, only the nonbasic variables label columns, while the objective and basic variables label rows:

	x_1	x_2	x_3	
p	-20	-30	-15	0
m_1	1.5	1	1	100
m_2	1	2	1	100

The implicit basis format has the advantages of being more compact, and making it more obvious which are the basic variables. This is the format that `LINEAR` uses.

- To see `LINEAR`'s version of the tableau, load the problem `EXAMPLE1.PRB` and use the commands `'Tableau Show tableau All columns All rows'`.

Our example involves maximizing the objective. If p was to be minimized instead, the signs in the objective row would all be changed: in effect we would be minimizing p by maximizing $-p$. The objective row would still be labelled by p , although it would really represent $-p$.

Conventions differ: some people put the RHS column on the left instead of the right, and some people put the objective row on the bottom instead of the top. This is unlikely to cause confusion, because of the lines separating the RHS column and the objective row from the rest of the tableau. Some people also use the opposite signs in the objective row.

A set of values for the variables and objective that satisfies the equations of the tableau is called a **solution**. Solutions can be obtained by arbitrarily prescribing values for the nonbasic variables, and letting the tableau equations give the values of the objective and basic variables. For example, if you took

$$x_1 = 50, x_2 = 100 \text{ and } x_3 = -100$$

you would get

$$p = 2500, m_1 = 25 \text{ and } m_2 = -50$$

But even though we call it a solution, this does not satisfy all the requirements of the problem. Two of these requirements are violated: x_3 is negative (violating the requirement that the variables all be ≥ 0), and also m_2 is negative (which means that the corresponding inequality is violated). A **feasible solution** of the problem must have all the variables (decision and slack) ≥ 0 : it would represent a possible production plan for the factory, satisfying all the constraints. For example, you could take

$$x_1 = x_2 = x_3 = 0$$

and get a feasible solution with

$$p = 0 \text{ and } m_1 = m_2 = 100$$

This solution would have the factory producing nothing! It is feasible because none of the constraints force the factory to produce anything. But the management would hope it is not **optimal**, i.e. there should be a feasible solution with a larger value of p . An **optimal** solution is a feasible solution with the best possible value of the objective among all feasible solutions. There might be many optimal solutions (all with the same value of the objective), or no optimal solutions. There are two ways that a problem could have no optimal solutions.

- There may be no feasible solutions. We say the problem is **infeasible**;
- There may be feasible solutions with arbitrarily large values (positive for a maximization problem, or negative for minimization) of the objective. We say the problem is **unbounded**.

In any tableau, the solution obtained by setting all the nonbasic variables to 0 is called a **basic solution**. The values of the basic variables and objective are then found in the RHS column of the tableau. In the example above we have a **basic feasible solution**, but basic solutions are not always feasible.

- To see the values of the basic variables and objective in the basic solution for the current tableau, use the command `Val V`

$$x_1 + 2x_2 + x_3 \geq 100$$

the equation for m_2 in the dictionary would have been

$$m_2 = -100 + x_1 + 2x_2 + x_3$$

A bit more of a change would be necessary to allow an '=' constraint such as

$$x_1 + 2x_2 + x_3 = 100$$

We would still write

$$m_2 = 100 - x_1 - 2x_2 - x_3$$

but this time, in order for a solution to be feasible, we would need not just $m_2 \geq 0$ but $m_2 = 0$. The variable corresponding to an '=' constraint is called an **artificial variable** (LINEAR would call this artificial variable 'ARTIF M2'). In order for a solution to be feasible, all the artificial variables must be 0 while all the other variables must be ≥ 0 .

- In the `Show tableau` command, LINEAR designates artificial variables with a *.

The choice of which variables are basic and which are nonbasic is to some extent arbitrary: we could perhaps express some other set of variables in terms of the variables not in the set, getting a new but equivalent tableau. For example, suppose we wanted to make x_2 into a basic variable. By turning the third equation of the tableau around, we could express x_2 in terms of x_1 , m_2 and x_3 :

$$x_2 = 50 - 0.5x_1 - 0.5m_2 - 0.5x_3$$

This expression for x_2 could then be substituted into the other equations:

$$p - 20x_1 - 30(50 - 0.5x_1 - 0.5m_2 - 0.5x_3) - 15x_3 = 0$$

$$m_1 + 1.5x_1 + (50 - 0.5x_1 - 0.5m_2 - 0.5x_3) + x_3 = 100$$

Simplifying these, and including the expression for x_2 , we get a new tableau, in which m_1 and x_2 are the basic variables.

	x_1	m_2	x_3	
p	-5	15	0	1500
m_1	1	-0.5	0.5	50
x_2	0.5	0.5	0.5	50

We say that x_2 has **entered the basis**, and m_2 has **left the basis**. The operation that we performed is called **pivoting**. The new tableau is equivalent to the old one, in the sense that any solution of one tableau is also a solution of the other. However, the **basic solutions** obtained from these tableaus by setting the nonbasic variables to 0 are different: the new basic solution is $x_1 = 0$, $x_2 = 50$, $x_3 = 0$, $m_1 = 50$, $m_2 = 50$, $p = 1500$. This happens to be a feasible solution again, and better than the first (since it has a larger p).

- To perform this pivot with LINEAR, use the commands ‘**Tableau Do pivot**’, enter ‘x2’ as “entering variable”, and enter ‘m2’ as “leaving variable.” You can use ‘**Tableau Show tableau All columns All rows**’ again to see the new tableau. Note that LINEAR’s tableau is rearranged a bit from the one above: LINEAR shows the rows and columns in their original order, with decision variables before slack variables.

However, this is still not the optimal solution. According to the new tableau, if you increased x_1 , keeping the other nonbasic variables at 0, p would increase. This is because the entry in the p row and x_1 column is negative. On the other hand, if all the entries in the p row for columns labelled by non-artificial variables were ≥ 0 , there would be no way to increase the value of p by changing the nonbasic variables. This is how an optimal solution is recognized.

From a conceptual point of view it is useful to think of pivoting in terms of substitution in the tableau equations. In practice, it is more efficient to work directly with the numbers in the tableau. See Appendix A for the procedure actually used.

The **Simplex Method** of linear programming involves repeatedly performing pivot operations, each time “improving” the basic solution, until either an optimal solution is obtained or it is clear that there is none. It can be proven that if there is an optimal solution, there is one that is basic, and the Simplex Method can find it.

LINEAR’s version of the Simplex Method involves three phases:

Phase 0 removes artificial variables from the basis.

Phase 1 finds a basic feasible solution.

Phase 2 finds an optimal solution.

The details of the method are presented in Appendix A.

- To have LINEAR solve the problem, use the command ‘**Solve**’. If you prefer to experiment with different pivots, ‘**Tableau Get candidates**’ will display some possible choices for each pivot, and allow you to choose among them.

For detailed information on the ‘**Tableau**’, ‘**Values**’ and ‘**Solve**’ commands, see Chapter 5.

Sensitivity analysis

Sensitivity analysis tells what would happen to the optimal solution if small changes were made in the problem. It can help you answer questions such as:

- what would be the effect of changes in prices?
- what should we be willing to pay for additional amounts of resources?
- how could we increase the production of a certain item?

The most important sources of such information are the **shadow prices**, sometimes called **reduced costs** or **dual variables**. These are the numbers in the objective row of the tableau for the optimal solution.

- To see the shadow prices for the current tableau, use the command ‘**Values Shadow prices**’.

THE SHADOW PRICE OF A NONBASIC VARIABLE REPRESENTS

- (1) THE COST PER UNIT OF SMALL INCREASES IN THAT VARIABLE
- (2) IN THE CASE OF A SLACK OR ARTIFICIAL VARIABLE, THE MAXIMUM AMOUNT PER UNIT THAT ONE WOULD PAY FOR RELAXING THE CONSTRAINT BY A SMALL AMOUNT, OR THE PENALTY PER UNIT FOR TIGHTENING THE CONSTRAINT BY A SMALL AMOUNT
- (3) THE CHANGE IN THE CONTRIBUTION OF THAT VARIABLE TO THE OBJECTIVE THAT WOULD MAKE IT ENTER THE OPTIMAL SOLUTION

I will illustrate these points using the following tableau, which gives the optimal solution of our example: $x_1 = 50$, $x_2 = 25$, $x_3 = 0$, $m_1 = 0$, $m_2 = 0$, $p = 1750$.

- To follow the discussion using LINEAR, load the problem EXAMPLE1.PRB and use the command ‘**Solve**’.

	m_1	m_2	x_3	
p	5	12.5	2.5	1750
x_1	1	-0.5	0.5	50
x_2	-0.5	0.75	0.25	25

- (1) According to the tableau, the shadow price of the nonbasic variable x_3 is 2.5. If instead of following the optimal solution you insisted on having x_3 at some positive value t , the best possible value for p would be $1750 - 2.5t$. The tableau also says how this could be arranged, because it expresses the basic variables in terms of the nonbasic ones. Keeping m_1 and m_2 at 0 and letting $x_3 = t$, you would have $x_1 = 50 - 0.5t$ and $x_2 = 25 - 0.25t$. Of course this could not go on forever — it would only be feasible as long as $t \leq 100$. That is why I referred to the cost of “small” increases in the variable.
- In LINEAR, the ‘**Analyze**’ command will give you this information. Enter ‘**x3**’ in response to ‘Name of variable or constraint’, and then choose ‘**As parameter**’. The results are as follows:

```

Analysis of parameter X3
X3 is nonbasic: shadow price      2.5000
parameter interval:  -infinity    100.0000
PROFIT values:      +infinity    1500.0000
    
```

Thus the shadow price is 2.5, which works for t up to 100. It would also work for negative values of t all the way to $-\infty$ (ignoring the requirement that x_3 itself must not be negative). At $t = 100$ the profit value would be \$1500. To see how the basic variables would depend on t in this interval, choose ‘**Values**’. This shows their values at the endpoints of the interval and the coefficient of t (-0.5 for x_1 and -0.25 for x_2). Use **Esc** to return to the main menu.

- (2) Consider m_2 , which has a shadow price of 12.5 in our tableau. The variable m_2 represents unused time on machine M_2 , in hours per week. This variable is nonbasic, so all the available time on machine M_2 is being used. By (1), the cost per unit of increasing m_2 is 12.5. Thus if you were unable to use the full 100 hours of M_2 time in a week, the loss would be \$12.50 per hour. On the other hand, if more than 100 hours were available, you would gain \$12.50 for each extra hour. This would correspond to allowing m_2 to become negative (for one of the decision variables x_i , this would not be a realistic possibility). According to the tableau, for each extra hour of M_2 time you would produce 0.5 less units of P_1 and 0.75 more units of P_2 . This would hold true for values of m_2 between -100 and $33 \frac{1}{3}$, i.e. if there were anywhere between $66 \frac{2}{3}$ and 200 hours of M_2 time available. Thus you should be willing to pay up to \$12.50 per hour for up to 100 hours of extra time on machine M_2 .

- Use ‘**Analyze**’ with ‘m2’ and ‘**Rhs**’. You are shown

```

RHS analysis for M2
RHS is      100.0000
M2 is binding: shadow price      12.5000
      RHS interval:      66.6667      200.0000
      PROFIT values:      1333.3333      3000.0000

```

‘**Values**’ will show how the values of x_1 and x_2 would depend on the amount of M_2 time available in this interval. Use Esc to return to the main menu.

One question often causes confusion: is the shadow price the maximum price to pay for additional amounts of a resource (e.g. M_2 time), or the maximum **additional price** beyond what is paid now? The answer depends on how the objective function was set up. The shadow price represents a price beyond what is included in the coefficients of the objective function. If the price now paid per unit of the resource was included in the objective function, then the shadow price represents an additional price beyond this amount. If the cost of the resource (at current available levels) was considered as a “fixed cost” and not included in the objective function coefficients, then the shadow price represents a total price.

- (3) In the current situation, small increases in x_3 cost 2.5 per unit. If the coefficient of x_3 in the objective increased, this might compensate for that cost. It would take an additional contribution of 2.5 per unit of x_3 to compensate for the cost of increasing x_3 . Thus if the contribution of product P_3 to profit were to increase by less than \$2.50 per unit, the optimal solution would still be unchanged. But if the contribution were to increase by more than \$2.50 per unit, it would become profitable to increase x_3 .

- Use ‘**Analyze**’, ‘x3’ and ‘**Objective**’. You are shown

```

Objective analysis for X3
PROFIT coefficient is      15.0000
X3 is nonbasic: shadow price      2.5000
Objective coefficient interval:      -infinity      17.5000

```

The coefficient of x_3 in p is 15 in the original problem. This would have to increase by at least 2.5 to 17.5 to have any effect on the solution.

We have seen how the shadow prices provide information about nonbasic variables. We might ask for analogous information about a basic variable.

- (1) HOW CAN WE PRODUCE SMALL CHANGES IN THE VALUE OF A BASIC VARIABLE, AND WHAT WOULD BE THE COST PER UNIT OF DOING SO?

Since the tableau expresses the basic variables in terms of the nonbasic variables, the way to change a basic variable is to increase one or more nonbasic ones. In fact, the best way is to increase one nonbasic variable at a time.

Suppose in our example we wanted to increase the basic variable x_2 . According to the tableau, increasing m_1 by one unit will increase x_2 by 0.5 units, while increasing m_2 or x_3 will decrease x_2 . So it is m_1 that should be increased. Now, according to the shadow price, one unit of increase of m_1 costs 5. To increase x_2 by one unit would require increasing m_1 by two units. Thus the cost per unit of increasing x_2 is $10 = 5/0.5$. This method of increasing x_2 could not go on forever, because increasing m_1 also decreases x_1 . Since x_1 can not be negative, x_2 can only be increased in this way by at most 25 units.

Next suppose we wanted instead to decrease x_2 . This time there is a choice of m_2 or x_3 to increase. Increasing m_2 would cost $12.5/0.75 = 16.667$ per unit decrease of x_2 , while increasing x_3 would cost $2.5/0.25 = 10$ per unit decrease of x_2 . Thus the cheapest way to decrease x_2 is to increase x_3 .

In general, then, the best way to increase or decrease a basic variable by a small amount is by increasing a nonbasic variable, chosen so that the change is in the correct direction and the cost per unit change of the basic variable is least. The amount of change possible by this method is usually limited, because a basic variable might hit 0. It could happen that no nonbasic variable would produce a change in the right direction, in which case the desired change would be impossible.

- From the main menu, use ‘**Analyze**’, ‘x2’, ‘**As parameter**’.

```

Analysis of parameter X2
X2 is basic: value      25.0000
Shadow price for increase:    10.0000
Shadow price for decrease:    10.0000

```

To see how much x_2 can be increased or decreased by these methods, and the effects on the other variables, use ‘**values Up**’ or ‘**values Down**’. Note that this lists x_2 first, followed by the objective, the nonbasic variable that would enter (here m_1 or x_3), and the other basic variables.

- (2) WHAT HAPPENS WHEN A CONSTRAINT WITH A BASIC SLACK VARIABLE IS TIGHTENED OR RELAXED BY A SMALL AMOUNT?

This is a very easy question to answer. Tightening the constraint simply reduces the value of the slack variable, and relaxing the constraint increases it. The constraint may be relaxed by any amount, or tightened by up to the current value of the slack variable, without changing anything else.

Note that we are talking only of inequalities and slack variables, not equalities and artificial variables. This is because normally the artificial variables will all be nonbasic in the optimal solution.

- If the right-hand-side of the second constraint were 250, ‘**Analyze**’, ‘m2’, ‘**Rhs**’ would produce

RHS analysis for M2		
RHS is	250.0000	
M2 is not binding		
RHS interval:	200.0000	+infinity

Here the slack variable m_2 is basic, with a value of 50. This means that 50 hours of available M_2 time are unused. The amount of M_2 time available could drop by up to 50 (to 200 from 250), with no changes except in the value of m_2 . Since the other variables are not changing, in this situation ‘**Values**’ will show only one set of numbers.

(3) WHAT HAPPENS WHEN THE COEFFICIENT OF A BASIC VARIABLE IN THE OBJECTIVE CHANGES?

The same general principle that applied in the nonbasic case applies here. In order to make increasing or decreasing a variable profitable, the variable’s coefficient in the objective must be increased or decreased by the per-unit cost of such a change as found in (1). A change in coefficient by less than these amounts will leave the optimal solution unchanged. The value of the objective changes, because the same amount of the basic variable contributes a different amount to the objective.

Another way to look at this situation is to consider the changes in the shadow prices of the nonbasic variables. According to the tableau, $x_2 = 25 + 0.5m_1 - 0.75m_2 - 0.25x_3$. By adding t to the coefficient of x_2 in the profit, i.e. adding tx_2 to $p = 1750 - 5m_1 - 12.5m_2 - 2.5x_3$, the profit becomes

$$p' = (1750 + 25t) - (5 - 0.5t)m_1 - (12.5 + 0.75t)m_2 - (2.5 + 0.25t)x_3$$

Thus the shadow prices of m_1 , m_2 and x_3 are changed to $5 - 0.5t$, $12.5 + 0.75t$ and $2.5 + 0.25t$ respectively. The solution will still be optimal if none of these shadow prices are negative, i.e. if $-10 \leq t \leq 10$. Note that this agrees with the costs found in (1) for increasing and decreasing x_2 . Since the contribution of x_2 to the profit in the original problem is 30, that contribution can vary between 20 and 40 without changing the optimal solution.

- From the main menu, use ‘**Analyze**’, ‘x2’, ‘**Objective**’.

Objective analysis for X2			
PROFIT coefficient is	30.0000		
X2 is basic: value	25.0000		
Objective coefficient interval:	20.0000	40.0000	
PROFIT values:	1500.0000	2000.0000	

Now ‘**Shadow prices**’ will show how the shadow prices of the nonbasic variables would depend on the contribution of x_2 to the profit, within this interval.

Dependence of shadow prices on contribution of X2			
	At contribution:		
	20.0000	40.0000	Coefficient
X3	0.0	5.0000	0.2500
M1	10.0000	0.0	-0.5000
M2	5.0000	20.0000	0.7500

Parametric programming studies how the optimal solution depends on the value of an adjustable parameter in the problem. The results of this section so far can be considered as parametric programming. In (1) the variable we want to increase or decrease has really become a parameter (a variable's value is found by solving the linear programming problem, while a parameter's value is part of the statement of the problem). In (2) the right-hand-side of the constraint, and in (3) the coefficient of the variable in the objective, can be considered as parameters.

Some other types of parameters can be dealt with using the techniques of sensitivity analysis. First, consider a parameter that affects more than one entry in the RHS column of the tableau. We might modify our example with such a parameter t as follows:

$$\begin{aligned} &\text{maximize} && 20x_1 + 30x_2 + 15x_3 \\ &&& \text{subject to} \\ &1.5x_1 + x_2 + x_3 &\leq & 100 + 2t \\ &x_1 + 2x_2 + x_3 &\leq & 100 + t \\ &x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

One way of dealing with this is to consider t as if it were a variable. However, since we don't want to optimize its value, make it an artificial variable, i.e. don't let it enter the basis. The optimal tableau is like the previous one, but with a column for t inserted. You could get the t column by adding -2 times the m_1 column and -1 times the m_2 column.

	m_1	m_2	x_3	t	
p	5	12.5	2.5	-22.5	1750
x_1	1	-0.5	0.5	-1.5	50
x_2	-0.5	0.75	0.25	0.25	25

According to this tableau, the profit is $1750 + 22.5t$ as long as the basic solution is feasible, which is from $t = -33.333$ to $t = 100$.

- Use the command '**Change Add Variable**' to add a new variable to the problem. Enter '***t**' as the name of the new variable (the asterisk indicates that this is to be an artificial variable). Enter 0 as the contribution to PROFIT. Enter '**m1**' as a '**Constraint involving T**', and -2 as the '**Coefficient in M1.**' Again you are asked for '**Constraint involving T**'; enter '**m2**' and -1. Since these are all the constraints involving T, just press the third time. '**Tableau Show tableau**' will now show the tableau above. '**Analyze**', '**t**', '**As parameter**' presents the following results:

Analysis of parameter T			
T is nonbasic(artificial):	shadow price	-22.5000	
parameter interval:	-33.3333	100.0000	
PROFIT values:	1000.0000	4000.0000	

Another type of parameter affects several coefficients in the objective. We might modify our example with such a parameter s as follows:

$$\begin{aligned} &\text{maximize} && (20 - s)x_1 + (30 - 2s)x_2 + (15 + s)x_3 \\ &&& \text{subject to} \\ &1.5x_1 + x_2 + x_3 &\leq & 100 \\ &x_1 + 2x_2 + x_3 &\leq & 100 \\ &x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

The method of (3) could be used to deal with this, if we introduced a new variable defined as $z = -x_1 - 2x_2 + x_3$. Then the objective could be written as $20x_1 + 30x_2 + 15x_3 + sz$. We insert a new row in the tableau, expressing z in terms of the nonbasic variables:

	m_1	m_2	x_3	
p	5	12.5	2.5	1750
x_1	1	-0.5	0.5	50
x_2	-0.5	0.75	0.25	25
z	-2	0	-1	-100

We find that the cost of small increases in z is 1.25, while z can not be decreased at all. Thus the current solution is optimal for s in the range $-\infty$ to 1.25.

- There is a complication in doing this with LINEAR: unlike ordinary variables, z must be allowed to be negative. One way to deal with this is to add a sufficiently large constant to z so that it will always be positive. In this case 1000 should do the trick. Now if you added a new variable z as in the last example, you would also have to add the constraint defining z in terms of the other variables. Instead, it is easier to treat z as the slack variable for an inequality constraint. So use ‘**Change Add Constraint**’ and enter the new constraint as

`-x1 -2 x2 + x3 > -1000 [z];`

Note that this defines the slack variable z as $-x_1 - 2x_2 + x_3 + 1000$. ‘**Tableau Show tableau**’ will show the tableau as above, but with -100 replaced by 900 (because of the extra 1000 in z). You may also have a column for t if you added it, but that is harmless. Now ‘**Analyze**’, ‘**z**’, ‘**Objective**’ will produce the following:

Objective analysis for SLACK Z			
PROFIT coefficient is		0.0	
Z is basic: value		900.0000	
Objective coefficient interval:		-infinity	1.2500
	PROFIT values:	-infinity	2875.0000

The profit values must be adjusted for the extra 1000 in z . Thus the profit at $s = 1.25$ is actually $2875 - 1.25 \times 1000 = 1625$.

In each case so far, sensitivity analysis has provided information valid for parameter values within certain intervals. It is also possible to see what happens outside these intervals. Usually what happens is a pivoting operation. After the pivot, the new tableau describes the optimal solution in a new interval adjacent to the old one. It sometimes happens, however, that there is no new optimal solution: after leaving the interval, the problem becomes infeasible or unbounded. It should also be noted that sometimes an “interval” can have zero length, and several pivots may be necessary to get any real change. In any case, by moving from interval to interval we can eventually work out the effects of all possible values of the parameter.

In the case of (1) or (2), a pivot is necessary when going outside the interval because otherwise the basic solution would become infeasible, i.e. some variable would become negative. That variable will leave the basis on this pivot. The variable entering the basis is chosen by the **dual simplex** method (see Appendix A). In the case of (3), the variable entering the basis is the one whose shadow price becomes negative, and the variable leaving the basis is chosen by the **primal simplex** method.

- In any of the examples discussed above, the ‘pivot **H**igher’ and ‘pivot **L**ower’ commands perform the pivot and go to the next interval. For example, in the ‘**O**bjective’ analysis of x_2 , where the objective coefficient interval was from 20 to 40, ‘pivot **H**igher’ will have the following result:

Pivot: SLACK M1	enters, X1	leaves	
X2 is basic:	value 50.0000		
Objective coefficient interval:		40.0000	+infinity
PROFIT values:		2000.0000	+infinity

Note that when you return to the main menu after having performed a pivot in ‘**A**nalyze’, the (not solved) indicator comes on, to remind you that the current tableau may no longer be optimal for the original problem. You should then use ‘**S**olve’ before doing anything else.

See Chapter 5 for more detailed information on the ‘**A**nalyze’ command. Also described there are the ‘**V**alues **R**anging’ commands, which find the objective coefficient intervals for all decision variables or the RHS intervals for all constraints.

Modelling

The most difficult and time-consuming aspect of linear programming lies in formulating a linear programming model for a real-life situation. This is the one step that can not be automated, and which can never have a complete set of instructions. This section can only provide a rough guide and a few tips.

The first step in the modelling process usually consists of deciding the scope of the model — what activities and time periods should be covered, and what level of detail should be used. It is usually impossible to deal with all aspects of a real-life situation, so only those that are most significant should be considered. Adding more details increases the size of the problem and the difficulty of gathering the required data. It is often best to start with a simpler, less detailed model, and use the insights gained from it in formulating and interpreting a more complicated model. For example, before setting up a model involving many time periods, it may be useful to see what happens in a model for a single time period.

The next steps consist of deciding on the variables, the objective and the constraints. There is no single “best” order for doing these — it depends on the particular problem, and often these steps are intertwined. Choose variables that make it easy to express the objective and constraints. You should have a clear idea of the meaning of each variable and constraint. Give them names that are easy to remember and recognize.

You should *not* put much effort into keeping the number of variables small, unless the problem gets so big that it can’t fit in the computer’s memory. Your first priority should be human convenience, not machine convenience. More variables can often make the problem easier to formulate and modify, and the results easier to interpret. Suppose that some quantity Q that plays a significant role in the problem can be expressed in terms of other variables. You might be tempted to substitute this expression for Q wherever it appears. This temptation should probably be resisted, especially if the expression is complicated. Instead, use Q as a variable, with its expression in terms of other variables as a constraint. This will make it easier for a human reader to understand your formulation. Moreover, since Q is a significant quantity in the formulation, its value may be useful information to find in the solution. You may even want to perform sensitivity analysis on it.

In problems involving a flow of materials between places or processes, it is often helpful to start by drawing a diagram of the situation, with arrows representing the flows. In general, each arrow in the diagram may be represented by a variable.

Here are some common types of constraints:

- **Resource availability.** There is a limited amount of a certain resource which is required by some activities. The usual form of this constraint is $\sum_j a_j x_j \leq b$, where the variables x_j represent the amounts of activities using the resource, a_j are the amounts of resource required per unit of activity j , and b is the available amount of the resource. The constraints of the example from Chapter 1 are of this type.

You may want to use a variable rather than a constant for the amount of the resource used, if that quantity occurs elsewhere in the problem, or simply if you want to know the amount. Then you would have $\sum_j a_j x_j = R$ where R is the variable for the resource, and $R \leq b$. In some cases you might want to replace the '=' by ' \leq ', in a situation where it is possible to have a surplus of the resource (and, presumably, to dispose of that surplus somehow).

- **Process yields.** The outputs of some process are expressed in terms of the amounts of inputs. For example, suppose each ton of raw material A yields 0.1 tons of product C and 0.2 tons of product D, while each ton of raw material B yields 0.3 tons of C and 0.4 of D. The appropriate constraints are

$$C = 0.1 A + 0.3 B$$

$$D = 0.2 A + 0.4 B$$

A very common error in this situation is to write the constraints as

$$A = 0.1 C + 0.2 D$$

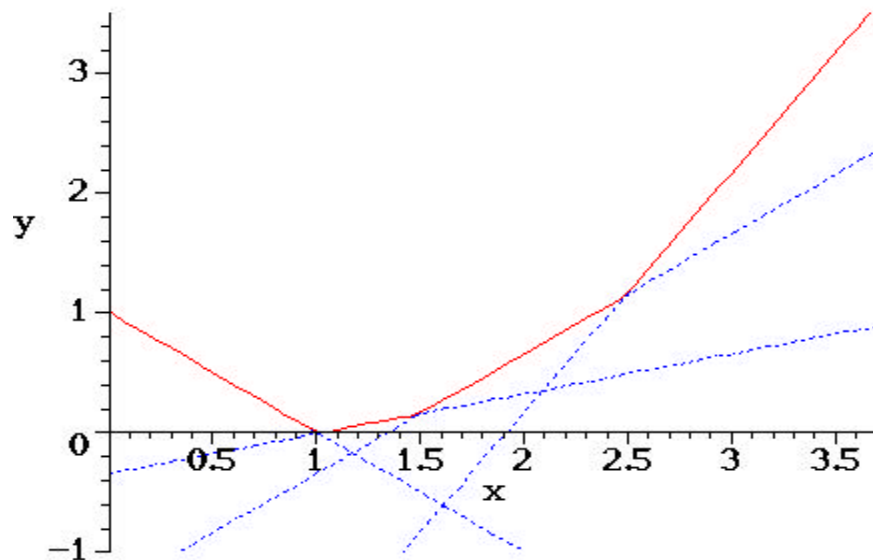
$$B = 0.3 C + 0.4 D$$

(WRONG!)

by analogy with the resource availability constraints. It is important to understand the distinction: the resource availability constraints show how the amounts of resources used depend on the products produced, while the process yield constraints show how the amounts of products produced depend on the raw materials used. The nature of the process determines which type of constraint is appropriate. Thus in manufacturing chairs, each chair of a certain type requires certain amounts of various materials (wood, fabric, etc.). This leads to resource availability constraints. In refining ores, on the other hand, the metal produced may come from various ores, and the amounts of the different types of ore used determine the quantities of various metals produced. This leads to process yield constraints. In general, you will have resource availability constraints when each output requires specific quantities of inputs (independent of other outputs), and process yield constraints when each input produces specific quantities of outputs (independent of other inputs).

- **Material balance.** This often arises in situations where some material is being produced by some activities and consumed by others. The material balance constraint states that the total amount produced is equal to the total amount consumed. If you draw a diagram representing the flow of the material, there will be a material balance constraint whenever several arrows come together. Sometimes the material balance constraint may use ' \geq ' rather than '=', if it is possible to produce a surplus of the material (and dispose of it for free).
- **Upper and lower bounds** on variables or combinations of variables can come from various sources: limited markets for products, contracts that must be fulfilled, capacities of processes, etc.

- **Blending.** When two or more products are blended to produce a new product, in addition to a process yield or material balance constraint there are often constraints involving the quality of the blended product. For example, suppose product C is produced by blending products A and B , and must be at most 50% (by weight) water, while A and B are respectively 30% and 60% water. If A, B and C represent quantities measured by weight, the material balance constraint will be $A + B = C$. Since water is not added or subtracted in the blending process, the amount of water in the product will be $.3A + .6B$, so that the quality constraint will be $.3A + .6B \leq .5C$.
- **Secondary objectives.** Instead of a single objective, you may have several conflicting goals. In this situation you may make one goal into the objective, and treat the others as constraints (setting up minimum acceptable levels for each). Sensitivity analysis may be used to see how the solution depends on those levels.
- **Soft constraints.** While some constraints are impossible to violate, others are not so absolute — it is possible to get around them, but at some cost. For example, resource availabilities are often of this type: only a certain amount is available under current conditions, but more could be obtained if we were willing to pay enough. Thus it might be possible to replace a “hard” constraint $X \leq 5$ by a “soft” constraint $X \leq 5 + E$. The new variable E , representing the amount by which the original constraint is violated, will appear in the objective with a suitable coefficient, representing the cost per unit of violation.
- **Convex and concave functions.** In general, linear programming can only accommodate linear functions, but there are special circumstances in which nonlinear functions may be modelled. An example of this is provided by income tax, in which the amount of tax T is a nonlinear function of taxable income I . This function is **convex**: the marginal tax rate, which is the slope of the graph of T as a function of I , never decreases as I increases. Moreover, we wish to pay as little tax as possible (other things being equal). The dependence of T on I may be modelled by a collection of inequalities of the form $T \geq aI + b$, one for each tax bracket. In looking for the optimal solution, LINEAR will choose the least value of T allowed by these inequalities, and that is the correct amount of tax.



A piecewise-linear convex function

Another method that could be used if bounds for I were known is the following: Let (I_i, T_i) be the corners of the $I - T$ graph in order of increasing I , where $I_0 \leq I \leq I_n$. Introduce new variables u_0, \dots, u_n with the three constraints

$$\begin{aligned}u_0 + \dots + u_n &= 1 \\I_0 u_0 + \dots + I_n u_n &= I \\T_0 u_0 + \dots + T_n u_n &= T\end{aligned}$$

These constraints say that (I, T) is a convex combination of the corner points (I_i, T_i) ; again, LINEAR will choose the least T for a given I . If $I_0 = T_0 = 0$ (as is usual with taxes), we can omit u_0 and write the constraints as

$$\begin{aligned}u_1 + \dots + u_n &\leq 1 \\I_1 u_1 + \dots + I_n u_n &= I \\T_1 u_1 + \dots + T_n u_n &= T\end{aligned}$$

In general, these methods can be used to model a convex function that is to be minimized, or a concave function that is to be maximized. If the graph of the function does not consist of line segments, it may be approximated by one that does.

Multiperiod models. Many problems involve activities taking place over some length of time. The modeller must choose some definite span of time, and divide it into a number of periods. Usually there is a similar set of variables and constraints for each period, describing what happens during the period. The different periods are typically tied together by “inventory” variables describing amounts carried over from one period to the next. There are “material balance” constraints stating that the inventory at the start of a period, plus or minus amounts added and subtracted during the period, equals the inventory at the start of the next period. If the periods are not tied together in this or some other way, it would be simpler to treat them as separate problems.

In problems involving money and time spans of a year or more, it is a good idea to use **discounting**. This means that money today is worth more than the same dollar amount at some time in the future. At a discount rate of $r\%$ per year, a cash transaction t years in the future should be divided by $(1 + r/100)^t$ to obtain its true worth. A good choice for r is the going interest rate. This applies to transactions that are not subject to inflation. In the case of amounts that can be expected to inflate (e.g. sales or purchases where the price may rise over time), you should instead use the “real” interest rate, i.e. the difference between the interest rate and the inflation rate.

Careful attention should be paid to what happens to the inventory variables at the end of the last period. In most cases the end of the last period is only a “planning horizon”, not the actual end of the activities being modelled. Perhaps the worst choice is not to allow any final inventory at all. This can cause quite unrealistic behavior in the last few periods, as inventories are forced to decrease toward zero. If final inventories are allowed, but the objective function takes no account of their worth, similar distortions can occur — there is no point in keeping costly items that will soon be worthless. The best method might be to have the objective function include a contribution from the final inventories. However, there may be some difficulty in choosing numbers for these contributions. Another method would be to assign reasonable “targets” as constraints for the final inventories. In either case, it is a good idea to use sensitivity analysis to see how the choice of these objective contributions or targets affects the behavior of the model.

Within a single period, activities are treated as if they all occurred at the same time, or perhaps were evenly spread out through the period. These assumptions, if too far from reality, may sometimes lead to serious inaccuracies in the model. For example, if expenditures occur near the beginning of a month while income arrives at the end, reliance on a model that lumps together all transactions during the month could create short-term cash flow problems. The remedy in this case could be to consider the income as belonging to the next month rather than the current one. In other situations we might stipulate that only a certain percentage of a quantity produced in a period is available for other activities during that period.

The inaccuracies caused by lumping together all activities during the same period, and the possibility of improving the solution by making more frequent changes, are incentives to make the periods as short as possible. Together with the desire to plan as far ahead as possible, this increases the number of periods in the model. Unfortunately, the numbers of constraints and variables are essentially proportional to the number of periods. This could lead to a linear programming problem too large for LINEAR to handle (or even too large, or too expensive, for a mainframe computer). You must try to obtain adequate accuracy while not allowing the problem to get too large. Moreover, as the number of periods increases the accuracy of the data often becomes suspect, so that little real advantage is obtained. This is particularly true when trying to extend the planning horizon.

In many multi-period models, only the results for the first few periods are really important, as these reflect decisions that will be made now. This is fortunate, because the uncertainties connected with the treatment of final inventories and inaccurate predictions about data for later periods may have little effect on the first few periods of the model. Before the decisions involving later periods are taken, the problem will be run again with updated data. In such circumstances, it may be worthwhile to pay more attention to the details of the first few periods than those of later periods. In particular, there is often no reason for all the periods to be of the same length. By using shorter periods at the beginning and longer periods toward the end, it may be possible to increase the accuracy of the early periods of the model while keeping the problem small.