Sensitivity Analysis and LINGO

Before we start, we need to set an option in LINGO. After that, we'll proceed with a simple example.

Setting Options for Ranges in LINGO

There is a particular tab that needs to be changed in order to get a range report in LINGO.

Under *Solver*, go to *Options*. Find a tab at the top of the dialog box that says *General Solver*, then find the box titled **Dual Computations**. Set that box so compute **Prices and Ranges** (the default is prices only).

You should select *Apply* and *Save* (so that you don't have to set the options every time). Here's the dialog box from the Linux version:

Interface General Solver	Model Generator Global S						
Multithreading:	Runtime Limits:						
	Iterations:						
Threads: 1	None						
Mode:							
Solver Decides 🔻							
Debugger:	None						
Output Level:	@SOLVE Time Limit:						
15 - Max	Time (seconds):						
Cold Start Solver:	None						
Solver Decides 🔻	Kill scripts on limit						
Warm Start Solver:	Scaling Warning Threshold:						
Solver Decides 💌							
Solver Decides *	1e+12						
Dual Computations:	Default Starting Point:						
Prices & Ranges 🔻	1.2345678						
Variables as	sumed non-negative						
	nat for names for MPS I/O						
✓ Favor reprod							

Example

Here is an LP that we could solve using a graphical technique (It is exercise 1, section 3.1, where x_1 is the acres of corn, x_2 is the acres of wheat):

$$\begin{array}{rll} \max z &= 30x_1 &+ 100x_2 \\ \mathrm{s.t.} & x_1 &+ x_2 &\leq 7 \quad (\mathrm{Land\ Constraint}) \\ & 4x_1 &+ 10x_2 &\leq 40 \quad (\mathrm{Labor\ Constraint}) \\ & 10x_1 &\geq 30 \quad (\mathrm{Govt.\ Constraint}) \\ & x_1 \geq 0 \quad x_2 \geq 0 \end{array}$$

We input the linear program into the "Lingo Model" window:

```
max=30*x1+100*x2;
x1+x2<=7;
4*x1+10*x2<=40;
10*x1>=30;
```

Now solve the model using either the *Solver* tab or the dartboard icon. We get the following output in the "Solution Report":

Global optin Objective	370.0000		
Variable	Value	Reduced Cost	
X1	3.000000	0.00000	
X2	2.800000	0.00000	
Row	Slack or Surplus	Dual Price	
	Slack or Surplus		
1	370.0000	1.000000	
2	1.200000	0.00000	
3	0.00000	10.00000	
4	0.000000	-1.000000	

We haven't discusses "Reduced Cost" or "Dual Price" yet- Let's try an experiment. If you increase the RHS of constraint 2 so that:

$4x_1 + 10x_2 \le 41;$

How does that change z? If you solve it, z = 380 now- An increase of 10. This is the *shadow* price, or *dual price*. We notice that the shadow price of the first constraint (Row 2) is zero-That's because, since that constraint isn't currently binding, it doesn't matter if we increase the RHS. On the other hand, increasing the RHS of the last constraint by 1 unit will decrease the value of z by 1.

NOTE: Our text says that the **shadow price** is the amount by which z is *improved* when a constraint (RHS) increases by 1. Careful here- For the maximization problem, positive shadow prices increase the value of z, but for a minimization problem, a positive shadow price will *decrease* the value of z.

To see the range for each variable, we have to click (or select) the model window, then select *Range* under the *Solver* tab as shown below, which should result in the range report (right image, below).

e Edit S	g solution	Ctrl+U	Solution Report - LingoZ.ing X			File Edit Solver)	うぐ 🗹 🖻 🍋	🔍 (+) 🏾 🐌 🔀 Solution Report - LingoZ		E 🖪 🦷 ?	• ×)	
-	Range		Soution Report - Lingozing X			Cingo Produi - 1	Lingo Model - Lingo2.lng*			ange report on journ	, × (
ax=30*3	Options	Ctrl+I				1 pax=30*x1+100*x						
(1+x2<=1 -	Generate	Ctrl+K				2 x1+x2<=7;						
*x1+10+C	Picture		Solution Repo	rt - Lingo2.lng	- 0 X	3 4*x1+104x2c=40+ 4 10*x1>= 14		Range Report - Ling	o2 log		_ = × ngo2.ing	
	Debug Model Statistics	Ctrl+D Ctrl+E	Global optimal solution found. Objective value: Infeasibilities:	370.000			in which the basis is un					370.0000
			Total solver iterations: Elapsed runtime seconds:	0.0				Objective C	oefficient Ranges:			0.04
			Model Class: Total variables: Nonlinear variables: Integer variables:	2 0 0	P t		Variable X1 X2	Current Coefficient 30.00000 100.0000 Righthm	Allowable Increase 10.00000 INFINITY nd Side Ranges:	Allowable Decrease INFINITY 25.00000	2 0 0	LP
			Total constraints: Nonlinear constraints: Total nonzeros: Nonlinear nonzeros:	4 0 7 0	U t f s		Row 2 3 4	Current RHS 7.000000 40.00000 30.00000	Allowable Increase INFINITY 12.00000 20.00000	Allowable Decrease 1.200000 28.00000 30.00000	4 0 7 0	
					Y hr	h					ď	
	_	_	MOD	Ln 4, Col 11	2:08 pm						La	L, Col 1 2:09 p

In this case, we get the following report:

Ranges in which the basis is unchanged:

	Current	Allowable	Allowable
Variable	Coefficient	Increase	Decrease
X1	30.00000	10.00000	INFINITY
X2	100.0000	INFINITY	25.00000
	Righthand	l Side Ranges:	
	Current	Allowable	Allowable
Row	RHS	Increase	Decrease
2	7.000000	INFINITY	1.200000
3	40.00000	12.00000	28.00000
4	30.00000	20.00000	30.00000

Objective Coefficient Ranges:

Note that these are the allowable *changes*, not the actual values, by which we can change the current values (of the coefficients shown or the RHS of the constraints) and maintain the current set of basic variables.

For example, the 30 for x_1 can be increased to 40, and we would still have the same set of basic variables comprise the BFS at which the optimum occurs.

By the way, we assume these changes are done *one at a time*. There is an analysis one can perform for multiple changes, but we will not look at that this semester (Section 6.4).