

Handling Indivisibilities

Notes for AGE 622

Bruce McCarl

Regents Professor of Agricultural Economics

Texas A&M University

Handling Indivisibilities All or None -- Integer Programming McCarl and Spreen Chapter 15

Many investment and problems involve cases where one has to take all or none of an item. We cannot build $\frac{1}{2}$ of a plant or buy $\frac{3}{4}$ of a machine. We build 1 or 2 or 3 or none but not a fractional part This leads to integer programming

$$\begin{array}{llll}
 \max & C_1 W & + C_2 X & + C_3 Y \\
 \text{s.t.} & A_1 W & + A_2 X & + A_3 Y \leq b \\
 & W & & \geq 0 \\
 & & X & \geq 0 \text{ and integer} \\
 & & & Y \text{ equals } 0 \text{ or } 1
 \end{array}$$

W is a normal, continuous LP variable,

X is an integer variable,

Y is a zero one variable

When problems have **only X they are called pure integer**
only Y they are called pure zero one
W and X they are called mixed integer

other variants exist

Handling Indivisibilities

Logical Conditions -- Integer Programming

Integer programming also allows logical conditions to be imposed.

Suppose I am modeling a bottling plant that runs white milk but can run chocolate. If they run any chocolate they encounter cleaning cost of F . Let X be amount of chocolate milk processed. Then add model component

$$\begin{array}{ll} \max & C X - F Y \\ \text{s.t.} & X - M Y \leq 0 \\ & X \geq 0 \\ & Y \text{ equals } 0 \text{ or } 1 \end{array}$$

Note in this component M is a big number (10 billion) and for $X > 0$ this implies $Y = 1$ while if $F > 0$ $Y = 0$ if $X = 0$.

So if we run any chocolate milk we clean whether it be 1 gallon or one million.
 Y is an indicator variable.

Handling Indivisibilities Integer Programming

Suppose we can buy from k different types of machines and get from them capacity for the i th time period

$$\begin{aligned}
 \max \quad & \sum_m C_m X_m - \sum_k F_k Y_k \\
 \text{s.t.} \quad & \sum_m a_{im} X_m - \sum_k CAP_{ik} Y_k \leq 0 \text{ for all } i \\
 & X_m \geq 0 \text{ for all } m \\
 & Y_k \text{ equals } 0 \text{ or } 1 \text{ for all } m
 \end{aligned}$$

In this case if they were mutually exclusive we could also add

$$\sum_k Y_k = 1$$

or if buying one meant we must buy another

$$Y_1 - Y_2 = 0$$

or if a machine can only be purchased if we have a minimum volume

$$-\sum_m a_{im} X_m + \sum_k LL_{ik} Y_k \leq 0 \text{ for all } i$$

Handling Indivisibilities

Integer Programming Solution Difficulty

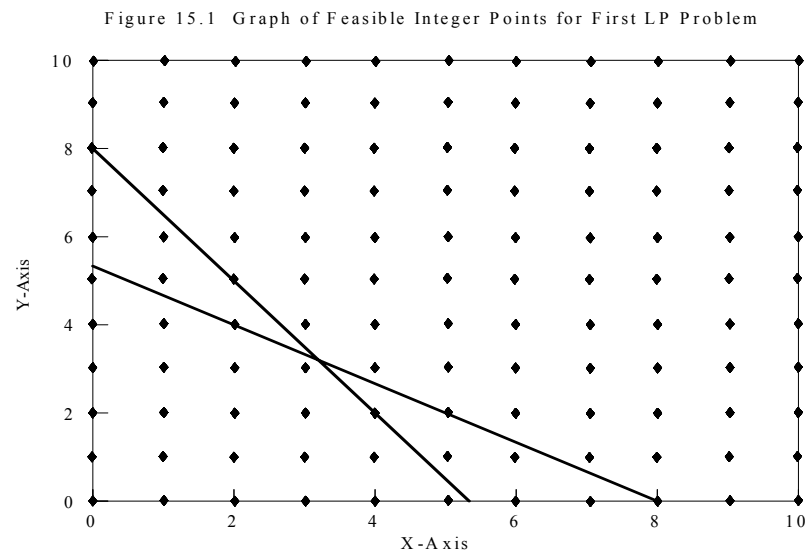
All sounds good but problems are hard. Let's explore why
Calculus is basis of all continuous optimization but not here because there is no neighborhood around a point in which a derivative can be defined

$$2X_1 + 3X_2 \leq 16$$

$$3X_1 + 2X_2 \leq 16$$

$$X_1, X_2 \geq 0 \text{ \& integer}$$

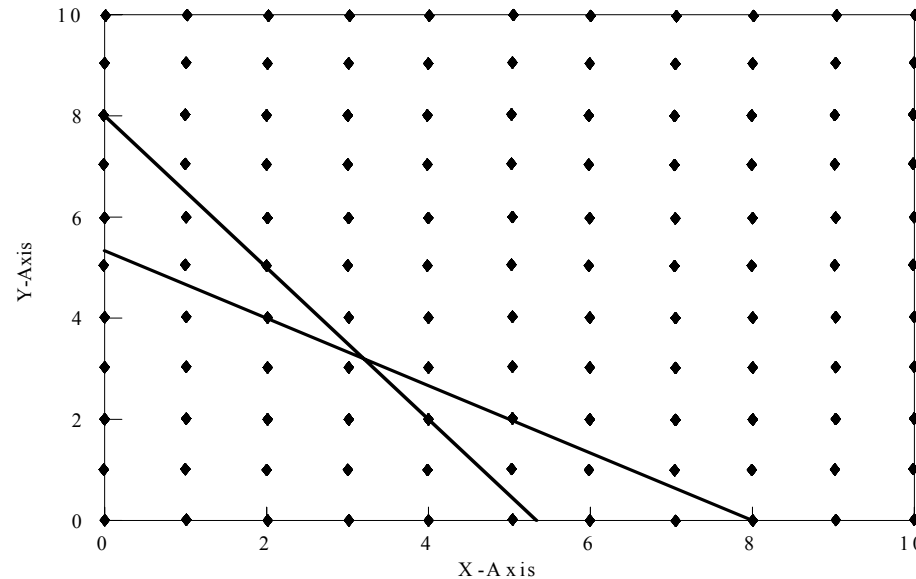
Feasible Region for X,Y nonnegative integers



Handling Indivisibilities

Integer Programming Solution Difficulty

Figure 15.1 Graph of Feasible Integer Points for First LP Problem



Note

1. Solutions are finite
2. A line between 2 feasible points does not contain all feasible points
3. Moving between points is not always easy
4. Points are on boundary, interior and not in general at corners
5. Rounding of LP point may not be bad

Handling Indivisibilities

Integer Programming Solution Difficulty

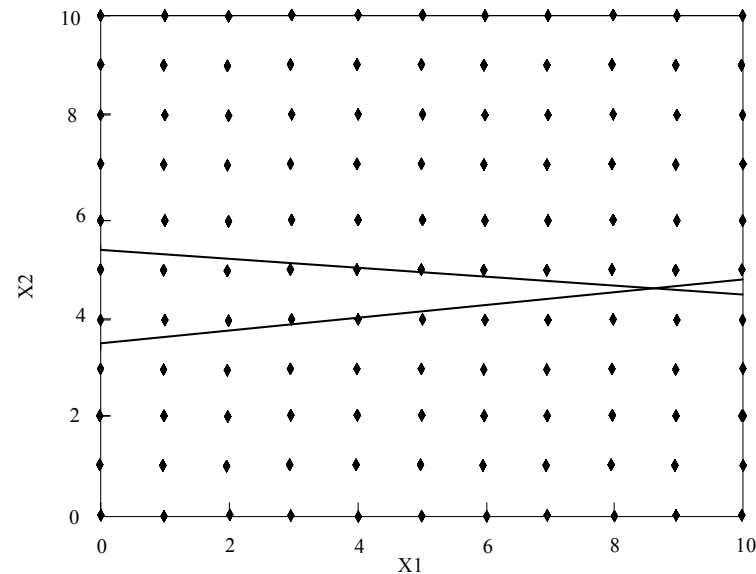
Consider

$$-X_1 + 7X_2 \geq 23$$

$$X_1 + 10X_2 \leq 54$$

$$X_1, X_2 \geq 0 \text{ \&int eger}$$

Figure 15.2 Graph of Feasible Integer Points for Second Integer Problem



Here Rounding not good and Movement between points hard

Handling Indivisibilities

Integer Programming Solution Difficulty

Mixed Integer Programming Feasible Region X_1 integer, X_2 continuous

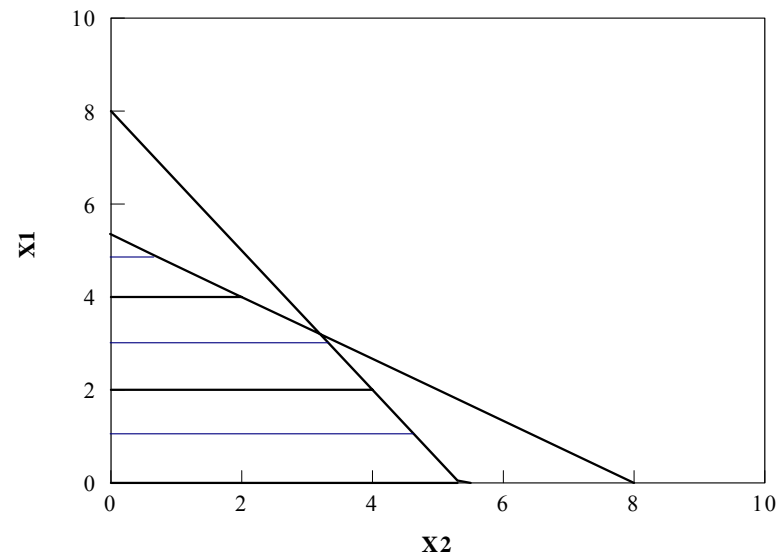
$$3X_1 + 2X_2 \leq 16$$

$$2X_1 + 3X_2 \leq 16$$

$$X_1 \geq 0 \text{ \& integer}$$

$$, X_2 \geq 0$$

Figure 15.3 Mixed Integer Feasible Region



Handling Indivisibilities

Integer Programming Solution – Rounding

Solving the problem

$$\begin{array}{ll} \text{Max} & 1.4 X_1 + X_2 \\ \text{s.t.} & 2 X_1 + 3 X_2 \leq 16 \\ & 3 X_1 + 2 X_2 \leq 16 \\ & X_1, X_2 \geq 0 \text{ \&int eger} \end{array}$$

as an LP yields $X_1=X_2=3.2$ $\text{Obj}=7.68$

which can be rounded to $X_1=X_2=3$, $\text{Obj}=7.2$

But this may not always be feasible or optimal

In this case an objective of 7.6 arises at $X_1=4, X_2=2$ ([solint.gms](#))

Rounding only works well if variable values are large

Handling Indivisibilities

Integer Programming Solution -- Branch and Bound

Solving

$$\begin{aligned}
 \text{Max} \quad & 1.4 X_1 + X_2 \\
 \text{s.t.} \quad & 2 X_1 + 3 X_2 \leq 16 \\
 & 3 X_1 + 2 X_2 \leq 16 \\
 & X_1, X_2 \geq 0 \text{ \& int eger}
 \end{aligned}$$

as an LP yields $X_1=X_2=3.2$ $\text{obj}=7.68$

We can generate 2 related problems that collectively do not exclude integer variables as follows

$$\text{Prob A} \left[\begin{array}{l} \text{max} \quad 1.4X_1 + X_2 \\ 2X_1 + 3X_2 \leq 16 \\ 3X_1 + 2X_2 \leq 16 \\ X_1 \leq 3 \\ X_1, X_2 \geq 0 \text{ \& int eger} \end{array} \right] \text{ or Prob B} \left[\begin{array}{l} \text{max} \quad 1.4X_1 + X_2 \\ 2X_1 + 3X_2 \leq 16 \\ 3X_1 + 2X_2 \leq 16 \\ X_1 \geq 4 \\ X_1, X_2 \geq 0 \text{ \& int eger} \end{array} \right]$$

Suppose we solve Prob A we get $x_1=3, x_2=3.33$ and again generate 2 more problems the first with $x_2 \leq 3$ and the other with $x_2 \geq 4$. Solving these yields an integer solution at $x_1=x_2=3$ $\text{obj}=7.2$ and another at $x_1=2, x_2=4$ $\text{obj}=6.8$.

But optimal solution is $x_1=4, x_2=2$, $\text{obj}=7.6$ from problem B above

Handling Indivisibilities

Integer Programming Solution -- Branch and Bound

Now since we are maximizing we choose the 7.2 as the best solution and call it the incumbent. But it is not necessarily optimal (in fact it is not at all). To verify its optimality we need to go back and investigate the problems we have not yet solved which still have the potential of having an objective function above our current best (7.2). We would then go back to the right hand problem from the first setup and eventually find $X_1=4, X_2=2, \text{Obj}=7.6$.

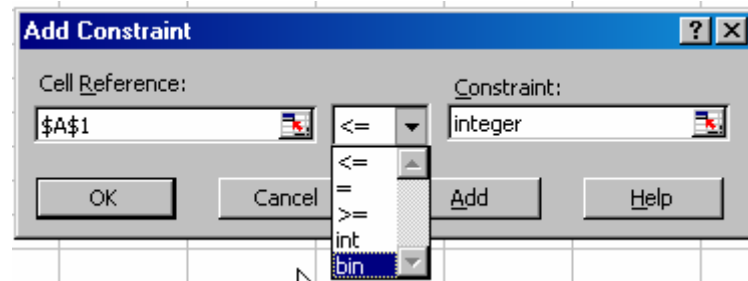
The above reveals the basic nature of branch and bound. It begins by solving an LP then finds a variable that is not integer and generates 2 problems (creating a branch). It then solves one of these and continues until it finds an integer solution which establishes a bound. We then backtrack and try to eliminate all other possible branches either by finding they cannot have a better objective than the incumbent (the bounding step) or are infeasible.

This means intermediate solutions are found and we converge to within a tolerance

Handling Indivisibilities

Integer Programming Solution in EXCEL

Integer programming problems can be solved with Excel. To do this one adds a constraint restricting the variables to take on integer values. This constraint is imposed as one adds any other constraints. Namely a variable is chosen with the inequality imposed as one of the last two choices. This is portrayed in the following screenshot.



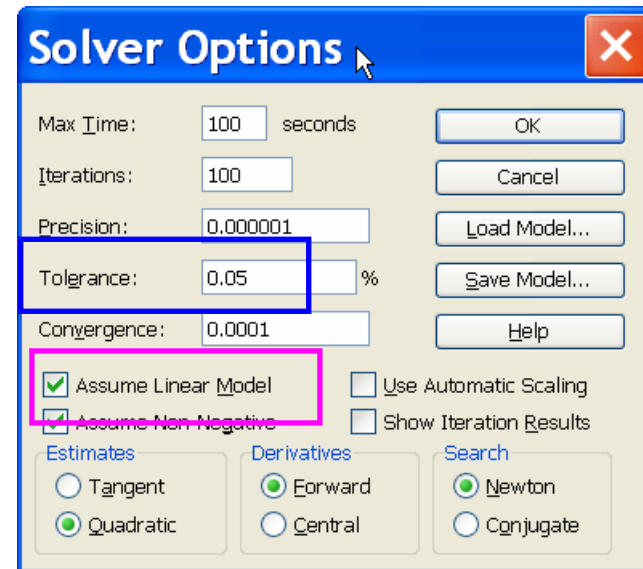
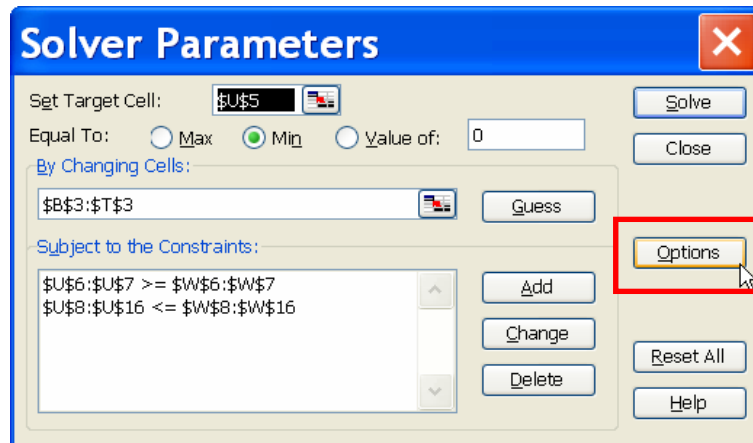
Namely choosing int restricts the variables to integer values and Bin is chosen to restrict variables to binary or 0-1 values.

Handling Indivisibilities

Integer Programming Solution in EXCEL

Unfortunately the default Excel solver does not reliably get the proper solution for integer programming problems unless you set some options

Namely when you call up the solver go into the **options box**. Then check that you are to **assume a linear model** and **set the tolerance smaller**.



Handling Indivisibilities

Integer Programming Solution in EXCEL

In addition, students should realize that EXCEL is not an extraordinarily good choice for the solution of large linear, integer, or nonlinear programming problems and software like GAMS may be better in a professional setting.

Excel was taught in this class so as to not require students to learn multiple and initially unfamiliar software packages

One can buy improved plug-ins by looking on the Internet and purchasing an upgraded solver which has this capability. For example, a Google search for the words excel in solver integer turns up a number of choices.

Handling Indivisibilities

McCarl and Spreen Chapter 16

Integer Programming Solution -- Knapsack

The knapsack problem is a famous IP formulation. Suppose a hiker selects the most valuable items to carry, subject to a weight or capacity limit. Partial items are not allowed, thus choices are depicted by zero-one variables.

The general problem formulation assuming only one of each item is available is

$$\begin{aligned} \text{Max} \quad & \sum_j v_j X_j \\ \text{s.t.} \quad & \sum_j d_j X_j \leq W \\ & X_j = 0 \text{ or } 1 \quad \text{for all } j \end{aligned}$$

The decision variables indicate whether the j th alternative item is chosen. Each item is worth v_j . The objective function gives total value of items chosen. The capacity used by each X_j is d_j . The constraint requires total capacity use to be less than or equal to the capacity limit (W).

Handling Indivisibilities

Integer Programming Solution -- Knapsack

Suppose an individual is preparing to move. Assume a truck is available that can hold at most 250 cubic feet of items. Suppose there are 10 items which can be taken and that their names, volumes and values implicit below. The resultant formulation is

$$\begin{array}{ll} \text{Max} & 17x_1 + 5x_2 + 22x_3 + 12x_4 + 25x_5 + x_6 + 15x_7 + 21x_8 + 5x_9 + 20x_{10} \\ \text{s.t.} & 70x_1 + 10x_2 + 20x_3 + 20x_4 + 15x_5 + 5x_6 + 120x_7 + 5x_8 + 20x_9 + 20x_{10} \leq 250 \\ & x_j = 0 \text{ or } 1, \text{ for all } j \end{array}$$

The optimal objective function value equals 128. The solution indicates that all items except furniture, X7, should be taken.

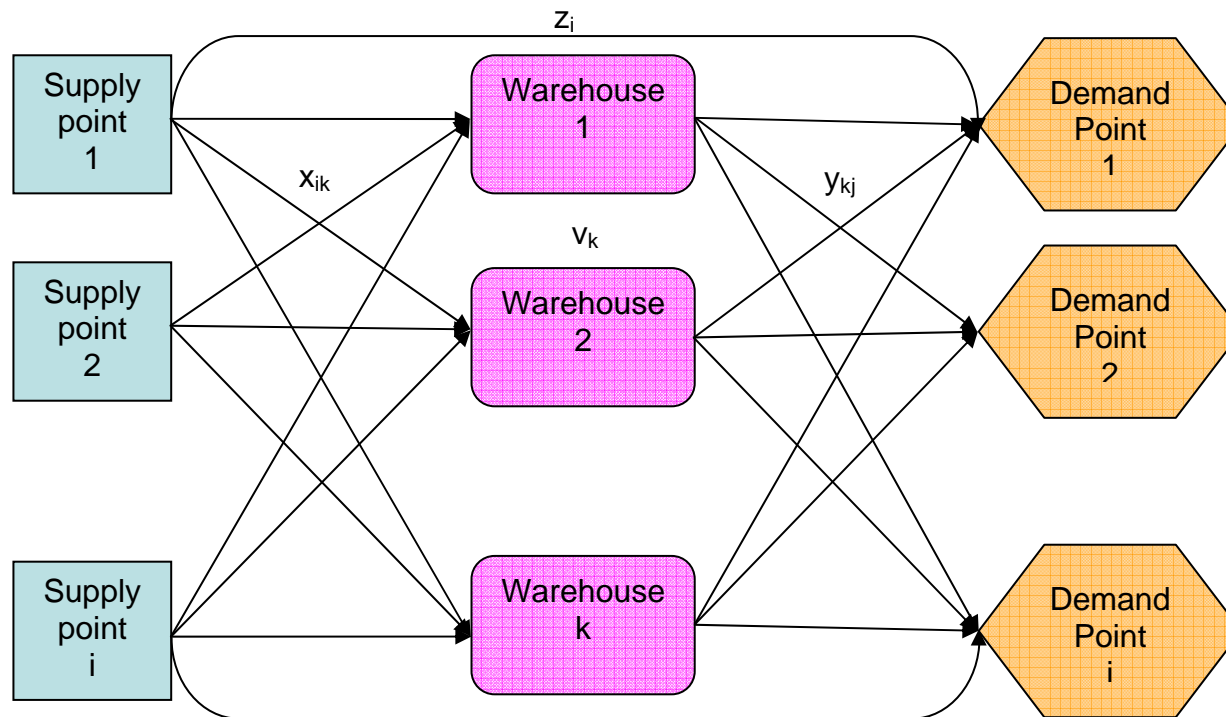
There are peculiarities which should be noted. First, the constraint has 65 units in slack (250-185) and no shadow price. However, for practical purposes the constraint does have a shadow price as the X7 variable would come into the solution if there were 120 more units of capacity, but slack is only 65. Further, note that each of the variables has a non-zero reduced cost. This is because bounds have been added. Shadow price and red cost are misleading.

Handling Indivisibilities

Warehouse Location

McCarl and Spreen Chapter 16

Warehouse location problems involve location of warehouses within a transportation system so as to minimize overall costs. Basic decision involves tradeoffs between fixed warehouse construction costs and transportation costs.



Handling Indivisibilities Warehouse Location

$$\begin{aligned}
 \text{Min} \quad & \sum_k F_k V_k + \sum_i \sum_k C_{ik} X_{ik} + \sum_k \sum_j D_{kj} Y_{kj} + \sum_i \sum_j E_{ij} Z_{ij} \\
 \text{s.t.} \quad & \sum_k X_{ik} + \sum_j Z_{ij} \leq S_i \quad \text{for all } i \\
 & \sum_k Y_{kj} + \sum_i Z_{ij} \geq D_j \quad \text{for all } j \\
 & -\sum_i X_{ik} + \sum_j Y_{kj} \leq 0 \quad \text{for all } k \\
 & -CAP_k V_k + \sum_j Y_{kj} \leq 0 \quad \text{for all } k \\
 & \sum_k A_{mk} V_k \leq b_m \quad \text{for all } m \\
 & V_k = 0 \text{ or } 1, \quad X_{ik}, \quad Y_{kj}, \quad Z_{ij} \geq 0 \quad \text{for all } i, j, k
 \end{aligned}$$

Merges Fixed Charge - Capacity and Transportation Problem with transshipments. We consider moving goods from supply i to demand j or from i to warehouse k and then on to demand j .

Handling Indivisibilities Warehouse Location

V_A	V_B	V_C	X_{1A}	X_{1B}	X_{1C}	X_{2A}	X_{2B}	X_{2C}	Y_{A1}	Y_{A2}	Y_{B1}	Y_{B2}	Y_{C1}	Y_{C2}	Z_{11}	Z_{12}	Z_{21}	Z_{22}	
50	60	68	1	2	8	6	3	1	4	6	3	4	5	3	4	8	7	6	Min
									1		1		1		1		1		≥ 75
										1		1		1		1		1	≥ 50
			1	1	1										1	1			≤ 50
						1	1	1									1	1	≤ 75
			-1			-1			1	1									≤ 0
				-1			-1				1	1							≤ 0
					-1			-1					1	1					≤ 0
-9999									1	1									≤ 0
	-60										1	1							≤ 0
		-70											1	1					≤ 0
1	1	1																	≤ 1
$V_A, V_B, V_C \in (0,1)$			$X_{ik},$			$Y_{kj},$			$Z_{ij} \geq 0$			for all i, j, k							

Handling Indivisibilities Warehouse Location

Variable	Value	Reduced Cost	Row	Activity	Shadow Price
V_A	0	0	1	50	-3.00
V_B	0	2	2	75	0
V_C	1	0	3	75	7.00
X_{1A}	0	0	4	50	5.00
X_{1B}	0	2.00	5	0	-4
X_{1C}	0	10.00	6	0	-3.00
X_{2A}	0	2	7	0	-1.00
X_{2B}	0	0	8	0	-0.05
X_{2C}	70	0	9	0	-1.00
Y_{A1}	0	1.052	10	0	-1.00
Y_{A2}	0	5.052	11	1	-2
Y_{B1}	0	0			
Y_{B2}	0	3.00			
Y_{C1}	20	0			
Y_{C2}	50	0			
Y_{11}	50	0			
Y_{12}	0	6.00			
Y_{21}	5	0			
Y_{22}	0	1.00			

Handling Indivisibilities

Machinery Selection

McCarl and Spreen Chapter 16

The machinery selection problem is a common investment problem. In this problem one maximizes profits, trading off the annual costs of machinery purchase with the extra profits obtained by having that machinery. A general formulation of this problem is

$$\begin{aligned}
 \text{Max} \quad & - \sum_k F_k Y_k + \sum_j \sum_m C_{jm} X_{jm} \\
 \text{s.t.} \quad & - \text{Cap}_{ik} Y_k + \sum_j \sum_m A_{ijkm} X_{jm} \leq 0 \quad \text{for all } i \text{ and } k \\
 & \sum_j \sum_m D_{njm} X_{jm} \leq b_n \quad \text{for all } n \\
 & \sum_k G_{rk} Y_k \leq e_r \quad \text{for all } r \\
 & Y_k \text{ is a nonnegative integer, } X_{jm} \geq 0 \quad \text{for all } j, k, \text{ and } m
 \end{aligned}$$

The decision variables are Y_k , the integer number of units of the k^{th} type machinery purchased; X_{jm} , the quantity of the j^{th} activity produced using the m^{th} machinery alternative. The parameters of the model are: F_k , the annualized fixed cost of the k^{th} machinery type; Cap_{ik} , the annual capacity of the k^{th} machinery type to supply the i^{th} resource; G_{rk} , the usage of the r^{th} machinery restriction when purchasing the k^{th} machinery type; C_{jm} , the per unit net profit of X_{jm} ; A_{ijkm} , the per unit use by X_{jm} of the i^{th} capacity resource supplied by purchasing machine k ; D_{njm} , the per unit usage of fixed resources of the n^{th} type by X_{jm} ; b_n , the endowment of the n^{th} resource in the year being modeled; and e_r , the endowment of the r^{th} machinery restriction.

Handling Indivisibilities Machinery Selection

		Machinery Acquisition Integer Variables										Machinery Use Continuous Variables														Crop Sales	Input Purchases			
												Plow with Tractor 1 and Plow 1				Plow with Tractor 2 and Plow 2				Plant Disc 8 Tractor 1		Tractor 2		Harvest with Tractor 1				Tractor 2		
		Tractor 1	2	Plow 1	2	Planter 1	2	Disc 1	2	Harvester 1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2			1	2	
Objectives (max)		-5000	-9000	-1000	-1200	-2000	-2100	-1000	-1200	-10000	-12000	-2.4	-2.4	-1.2	-1.2	-1.2	-1.2	-0.6	-0.6	-1.46	-1.22	-0.73	-0.61	-9.33	-8.35	-9.33	-8.25	2.5	-110	
Tractor 1 Capacity in Period	1	-160										.2		.1															.0	
	2	-180											.2		.1				.1	.0833									.0	
	3	-200																					.33	.25					.0	
Tractor 2 Capacity in Period	1		-160												.1		.05												.0	
	2		-180												.1		.05												.0	
	3		-200																						.33	.25			.0	
Plow 1 Capacity in Period	1		-160									.2			.1														.0	
	2		-180										.2			.1													.0	
Plow 2 Capacity in Period	1			-160										.1			.05												.0	
	2			-180													.05												.0	
Capacity of Planter	1				-180														.1		0.05								.0	
	2				-180																0.0833		.0417						.0	
Capacity of Disc	1					-180													.1		0.05								.0	
	2					-180															0.0833		.0417						.0	
Capacity of Harvester	1							-200															.33		.33				.0	
	2							-200																.25		.25			.0	
Labor Available in Period	1									.24		.12			.12		.06												.200	
	2										.24		.12			.12		.06		.12	.11	.06	.055						.210	
	3																						.5	.375	.5	.375			.250	
Plow-Plant Sequencing											-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1							.0	
Plant-Harvest Sequencing																			-1	-1	-1	-1	1	1	1	1			.0	
Land Available											1	1	1	1	1	1	1	1											.600	
Planters						1	1																						.1	
Discs								1	1																				.1	
Link Disc-Planter						-1		1																					.0	
								-1		1																			.0	
Yield Balance																													.0	
Input Balance																			1	1	1	1							-1	.0

Table 16.17. Solution for the Machinery Selection Problem
obj = 116,100

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Buy Tractor 1	1	-5,000	Tractor 1 capacity in Period 1	100	0
Buy Tractor 2	0	0	Tractor 1 capacity in Period 2	130	0
Buy Plow 1	0	0	Tractor 1 capacity in Period 3	50	0
Buy Plow 2	1	-1,200	Tractor 2 capacity in Period 1	0	12
Buy Planter 1	0	0	Tractor 2 capacity in Period 2	0	14.6
Buy Planter 2	1	-3300	Tractor 2 capacity in Period 3	0	22.26
Buy Disc 1	0	0	Plow 1 capacity in Period 1	0	6.25
Buy Disc 2	1	0	Plow 1 capacity in Period 2	0	0
Buy Harvester 1	0	0	Plow 2 capacity in Period 1	100	0
Buy Harvester 2	1	0	Plow 2 capacity in Period 2	180	0
Plow with Tractor 1 and Plow 1 in Period 1	0	-2.45	Planter 1 capacity	0	0
Plow with Tractor 1 and Plow 1 in Period 2	0	-1.20	Planter 2 capacity	130	0
Plow with Tractor 1 and Plow 2 in Period 1	600	0	Disc 1	0	0
Plow with Tractor 1 and Plow 2 in Period 2	0	0	Disc 2	130	0
Plow with Tractor 2 and Plow 1 in Period 1	0	-1.825	Harvester 1	0	50
Plow with Tractor 2 and Plow 1 in Period 2	0	-1.46	Harvester 2	50	0
Plow with Tractor 2 and Plow 2 in Period 1	0	0	Labor available in Period 1	128	0
Plow with Tractor 2 and Plow 2 in Period 2	0	0.13	Labor available in Period 2	144	0
Plant with Tractor 1 and Planter 1	0	-1.91	Labor available in Period 3	25	0
Plant with Tractor 1 and Planter 2	600	0	Plow Plant	0	230.533
Plant with Tractor 2 and Planter 1	0	-1.077	Plant Harvester	0	341.75
Plant with Tractor 2 and Planter 2	0	0	Land	0	229.333
Harvest with Tractor 1 and Harvester 1	0	-17.75	One Planter	0	0
Harvest with Tractor 1 and Harvester 2	600	0	One Disc	0	0
Harvest with Tractor 2 and Harvester 1	0	-25.17	Planter 1 to Disc 1	0	0
Harvest with Tractor 2 and Harvester 2	0	-5.565	Planter 2 to Disc 2	0	0
Sell Crop	84,000	0	Yield Balance	0	2.5
Purchase Inputs	600	0	Input Balance	0	110