More Linear Programming Models

The Assembly Problem - Primal Algebra

$$
\begin{array}{llllll}
\text{Max} & \sum_j c_j X_j & - & \sum_k d_k Q_k & & \\
& \sum_j a_{kj} X_j & - & w_k Q_k & \leq & h_k & \text{for all } k \\
& \sum_j e_{ij} X_j & + & \sum_k f_{ik} Q_k & \leq & b_i & \text{for all } i \\
& X_j & & & \geq & g_j & \text{for all } j \\
& X_j, & & Q_k & \geq & 0 & \text{for all } k,j
\end{array}
$$

Objective: Maximize the return summed over all the final products produced less the cost of the component parts purchased.

Constraints: The first constraint equation is a supply-demand balance and constrains the usage of the component parts to be less than or equal to inventory plus purchases.

The second constraint limits the resources used in manufacturing final products and purchasing component parts to the exogenous resource endowment.

The last constraint imposes a minimum sales requirement on final product production

$$
\begin{array}{llllll}
\text{Max} & \sum_j c_j X_j & - \sum_k d_k Q_k & & & \\
 & \sum_j a_{kj} X_j & - \quad w_k Q_k & \leq & h_k & \text{for all k} \\
 & \sum_j e_{ij} X_j & + \sum_k f_{ik} Q_k & \leq & b_i & \text{for all i} \\
 & X_j & & \geq & g_j & \text{for all j} \\
 & X_j, & Q_k & \geq & 0 & \text{for all k,j}
\end{array}
$$

The dual problem is not very much different from those before, thus, suppose we only look at the dual constraint associated with Q_k. That constraint

$$
- w_k U_k + \sum_i f_{ik} Z_i \geq - d_k
$$

where U_k is the return to one unit of component part k; and Z_i is the return to one more unit of limited resource i.

This constraint is more easily interpreted if it is rewritten as follows

$$
\sum_i f_{ik} Z_i + d_k \geq w_k U_k
$$

or, equivalently,

$$
\frac{\sum_i f_{ik} Z_i + d_k}{w_k} \geq U_k
$$

This inequality says that the internal value of a component part unit is less than or equal to its purchase price plus the cost of the resources used in its acquisition. Therefore, the internal value of a component part can be greater than the amount paid externally.

More Linear Programming Models

The Assembly Problem – An Example

Table 8-6: Components, Resources and cost Required to Assemble a Cake

	Vanilla Wedding Cake	French Vanilla Cakes	Boston Cream Cake	Lemon Cake
Flour in cups	6.00	2.67	1.00	2.50
Eggs in amount	12.00	3.00	2.00	3.00
Sugar in cups	4.50	1.67	1.67	4.00
Butter in pounds	3.00	1.00	0.50	1.00
Milk in cups	4.50	0.50	2.50	1.00
Labor in hours	10.00	1.00	0.91	0.60
Refrigerator Space	7.00	0.00	2.00	1.00
Oven Time in hours	2.50	0.60	0.60	0.60
Other Cost in $	22.50	2.50	7.00	4.50
Sale Price in $	330.00	35.00	45.00	38.00
Max Sale Potential	18	100	100	95
Min Sale Requirement	12	70	12	14

Table 8-7: Component Part Acquisition Information

	Unit of Purchase	Inventory	Cost to Purchase in $	Labor Use in Hours	Use of Refrig Space	Parts in Purchase
Flour	50 lb sack	22 cups	28.50	0.10		167 Cups
Eggs	Box containing 15 dozen	72 eggs	29.00	0.20	7.00	180 Eggs
Sugar	Skid containing 50 sacks each weighing 50 lbs	55 cups	2029.00	2.00		5000 Cups
Butter	44-pound pail	12 pounds	133.00	0.33	3.50	44 lbs
Milk	100 lbs	55 cups	26.50	0.30	8.00	185 cups

Table 8-8: Resources Available

Resource	Available
Labor in hours	340
Refrigerator Space in sq ft	500
Oven Time in hours	180

More Linear Programming Models

The Assembly Problem – An Example

Table 8-9: Tableau Setup of Cary's Cake Emporium Problem

	--------- Assemble Cakes ----------				-------------- Buy Component Parts --------------					
	Vanilla Wed Cake	French Vanilla Cake	Boston Cream Cake	Lemon Cake	50 lb sack of Flour	15 dozen Eggs	50 Sack Skid of Sugar	44 Pounds of Butter	100 Pounds Of Milk	
	x_{vwc}	x_{fvc}	x_{bcc}	x_{lc}	q_f	q_e	q_s	q_b	q_m	
Profit	307.50	32.50	38.00	33.50	-28.50	-29.00	-2029.00	-133.00	-26.50	Maximize
Flour SD Balance	6.00	2.67	1.00	2.50	-167					≤ 22
Eggs SD Balance	12.00	3.00	2.00	3.00		-180				≤ 72
Sugar SD Balance	4.50	1.67	1.67	4.00			-5000			≤ 55
Butter SD Balance	3.00	1.00	0.50	1.00				-44		≤ 12
Milk SD Balance	4.50	0.50	2.50	1.00					-185	≤ 55
Labor Available	10.00	1.00	0.91	0.60	0.10	0.20	2.00	0.33	0.30	≤ 340
Refrigerator Space	7.00		2.00	1.00		0.70		3.50	8.00	≤ 500
Oven Time in hours	2.50	0.60	0.60	0.60						≤ 180
Max Assembly	18	100	100	95						
Min Assembly	12	70	12	14						

More Linear Programming Models

The Assembly Problem – An Example

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Assemble Vanilla Wedding Cake	13.925	0	Flour SD Balance	0	0.185
Assemble French Vanilla Cake	70.000	0	Eggs SD Balance	0	0.187
Assemble Boston Cream Cake	76.981	0	Sugar SD Balance	0	0.415
Assemble Lemon Cake	95.000	0	Butter SD Balance	0	3.201
Buy Flour 50 pound sack	3.371	0	Milk SD Balance	0	0.182
Buy Eggs 15 dozen	4.134	0	Labor Available	0	23.727
Buy Sugar 50 bag skid	0.127	0	Refrigerator Space	91.8	0.
Buy Butter 44 pound tub	5.301	0	Oven Time	0	21.835
Buy Milk 100 pounds	1.784	0	Max Vanilla Wedding	4.075	0.
			Max French Vanilla	30.000	0.
			Max Boston Crean	23.019	0.
			Max Lemon	0 .	0.095
			Min Vanilla Wedding	1.925	0.
			Min French Vanilla	.	-9.368
			Min Boston Crean	64.981	0.
			Min Lemon	81.000	0.

More Linear Programming Models

The Disassembly Problem – Primal Algebra

$$Max \; -\sum_j c_j X_j + \sum_k d_k Q_k \quad -\sum_j a_{kj} X_j + \quad Q_k \le 0 \; for\, all\, k \; \sum_j e_{rj} X_j + \sum_k f_{rk} Q_k \le b_r \; for\, all\, r \quad X_j \le g_j \; for\, all\, j \quad Q_k \le h_k \; for\, all\, k \quad Q_k \ge$$

Objective: The objective function maximizes operating profit, which is the sum over all final products sold (Q_K) of the total revenue earned by sales less the costs of all purchased inputs.

Constraints: The first constraint is a product balance -limiting the quantity sold to be no greater than the quantity supplied when the raw product is disassembled.

The next constraint is a resource limitation constraint on raw product disassembly and product sale.

This is followed by an upper bound on disassembly as well as upper and lower bounds on sales.

More Linear Programming Models

The Disassembly Problem – An Example

Table: Proportional component parts (%) and resources required

	Recover Metal, Junk the Rest	Recover as much as you can
METAL (%)	50	53
SEATS (%)	0	8
OTHER (%)	0	12
JUNK (%)	50	27
Disassemble cost ($)	100	120
Labor (hour)	10	20
Shop Capacity	1	1.2

Table : Part Data

Part Data	Max Sales (US ton)	Min Sales (US ton)	PRICE ($/US ton)	Inventory on hand (US ton)	LABOR (hours/US ton)
METAL	20	2	700	1	2
SEATS	4	1	1100	2	4
OTHER	7		950	4	1
JUNK			-15	10	0.5

Other Information

Car Information		**Resources Available**	
Car Weight	3000 lb EA	Labor	500 hours
Car Price	$225 EA	Shop Capacity	28 cars
		Maximum Car Purchase Allowance	25 cars

More Linear Programming Models

The Disassembly Problem – An Example

	Metal	Seats	Others	Junk	Recover Metal, Junk the Rest	Recover as much as you can		
Obj	700	1100	950	**-15**	-325	-345		
Metal	1				-0.75	-0.795	<=	1
Seats		1			0	-0.12	<=	2
Other			1		0	-0.18	<=	4
Junk				1	-0.75	-0.405	**= or >=**	10
Labor	2	4	1	0.5	10	20	<=	500
Shop Capacity					1	1.2	<=	28
Car Max					1	1	<=	25
Max Sales	20	4	7					
Min Sales	2	1						
Non-negativity	1,	1,	1,	1,	1,	1,	>=	0

More Linear Programming Models

The Disassembly Problem – An Example

Solution

Objective = 17441.99

Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
Metal	20	0	Metal	18	0
Seats	4	0	Seats	3	0
Others	7	0	Other	0	456.37
Junk	21.83	0	Junk	0.21	0
Recover Metal, Junk the Rest	5.85	0	Labor	0.31	0
Recover as much as you can	18.38	0	Shop Capacity	0	-15.29
			Car Max	0	0.58
			Metal Max	0.09	0
			Seats Max	0.77	0
			Other Max	0	242.46
			Metal Min	0	1097.68
			Seats Min	0	949.42

More Linear Programming Models

The Assembly-Disassembly Problem

Primal Algebra

$$Max \; - \sum_j c_j X_j \; + \; \sum_k d_k Q_k \; + \; \sum_i s_i T_i \; - \; \sum_i p_i Z_i \quad - \; \sum_j a_{ij} X_j \; + \; \sum_k b_{ik} Q_k \; +$$

Objective: The objective function maximizes the revenue from final products and component parts sold less the costs of the raw products and component parts purchased.

Constraints: The first constraint is a supply-demand balance, and balances the use of component parts through their assembly into final products and direct sale, with the supply of component parts from either the disassembly operation or purchases.

The remaining equations impose resource limitation constraints and upper bounds.

More Linear Programming Models

The Assembly-Disassembly Problem

An Example

Table 7.7. Data for Chicken Example Yields from Cutting

	Parts	Halves	Quarters	Meat	Leg-Breast -Thigh
Wings	2				
Legs	2				2
Thighs	2				2
Back	1				
Breasts	2				2
Necks	1				1
Gizzards	1	1	1	1	
Meat		0.05	0.07	1	0.2
Breast Quarter			2		
Leg Quarter			2		
Halves		2			

Selling Price and Labor Use for Chicken Packs

Pack	Labor	Price
A	2	$2.05
B	1.3	2.00
C	1.2	1.45
D	1.1	1.95
E	1.25	1.25
Gizzard	1.0	0.90

Individual Selling Prices for Parts

Part	Price	Part	Price
Wings	0.10	Gizzards	0.07

Legs	0.20	Meat	2.00/lb.
Thighs	0.25	Breast Quarters	0.45
Backs	0.12	Leg Quarter	0.40
Breasts	0.33	Halves	0.90
Necks	0.05		

More Linear Programming Models

The Assembly-Disassembly Problem

Table 7.8. Primal Formulation of Charles Chicken Company Problem

	Disassemble					Assemble						Sell											Buy			RHS
	X_p	X_h	X_q	X_m	X_L	X_a	X_b	X_c	X_d	X_e	X_g	Wings	Leg	Thigh	Back	Breast	Neck	Gizzard	Meat	Breast Qtr	Leg Qtr	Halves	Wings	Legs	Thighs	
Object	-1	-1	-1	-1	-1	2.05	2.00	1.45	1.95	1.25	.90	.10	.20	.25	.12	.33	.05	.07	2.0	.45	.40	.90	-.12	-.22	-.27	Max
Wings	-2					2						1											-1			≤ 0
Legs	-2				-2	2				2			1											-1		≤ 0
Thighs	-2				-2	2				2				1											-1	≤ 0
Backs	-1					1									1											≤ 0
Breasts	-2				-2	2										1										≤ 0
Necks	-1				-1	1											1									≤ 0
Gizzards	-1	-1	-1	-1							10							1								≤ 0
Meat		-.05	-.07	-1	-.2														1							≤ 0
Breast Qtr.			-2				4													1						≤ 0
Leg Qtr.			-2					4													1					≤ 0
Halves		-2							2													1				≤ 0
Chickens	1	1	1	1	1																					≤ 1000
Labor						2	1.3	1.2	1.1	1.25	1															≤ 3000
Wing																							1			≤ 20
Leg																								1		≤ 20
Thigh																									1	≤ 20

More Linear Programming Models

The Assembly-Disassembly Problem

Solution

Table 7.9. Solution to the Charles Chicken Co. Problem

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Objective function = 1362.7					
X_p	0	-0.22	Wings	0	0.120
X_h	0	0	Legs	0	0.355
X_q	0	-0.33	Thighs	0	0.270
X_m	0	-0.27	Backs	0	0.180
X_L	1000	0	Breasts	0	0.330
X_a	0	0	Necks	0	0.050
X_b	0	0	Gizzards	0	0.090
X_c	0	-0.15	Meat	0	2.000
X_d	0	-0.22	Breast Qtr.	0	0.500
X_e	1010	0	Leg Qtr.	0	0.400
Gizzards	0	0	Halves	0	1.085
Wings	0	-0.02	Chickens	0	1.36
Legs	0	-0.02	Labor	1737.5	0
Thighs	0	-0.155			
Backs	0	-0.06			
Breasts	2000	0			
Necks	1000	0			
Gizzards	0	-0.02			
Meat	200	0			
Breast Qtr.	0	-0.05			
Leg Qtr.	0	0			
Halves	0	-0.185			
Wings	0	0			
Legs	20	0			
Thighs	20	0.135			

More Linear Programming Models

The Assembly-Disassembly Problem

Violation of Separability Assumption

The Blending Problem:

$$
\begin{array}{lcccccccccccc}
\text{Max} & 3A & + & 2B & & & & & & & \\
 & -A & - & 2B & + & 2G_1 & + & G_2 & \leq & 0 \\
 & -A & - & 2B & + & G_1 & + & 2G_2 & \leq & 0 \\
 & A & + & B & - & G_1 & - & G_2 & = & 0 \\
 & & & & & G_1 & & & \leq & 20 \\
 & & & & & & & G_2 & \leq & 20 \\
 & A, & & B, & & G_1, & & G_2 & \geq & 0
\end{array}
$$

Table 7.10. Data for the Grain Blending Example

	Grade		Characteristics	
	Maximums		Grain Batch 1	Grain Batch 2
	A	B		
Moisture	1	2	2	1
Foreign Matter	1	2	1	2

Table 7.11. Solution of the First Formulation of the Grain Blending Problem

Objective = 100

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
A	20	0	Moisture	0	1
B	20	0	Foreign Matter	0	0
G_1	20	2	Weight	0	4
G_2	20	3			

There is a problem with this solution. It is impossible, given the data above, to make a mix containing 20 units each of grade A and grade B grain.

More Linear Programming Models

The Assembly-Disassembly Problem

Violation of Separability Assumption

The proper formulation of the blending problem is

$$
\begin{array}{llcccccccccccl}
\text{Max} & 3A & + & 2B & & & & & & & & & & \\
\text{s.t.} & -A & & & + & 2G_{11} & + & G_{21} & & & & & \leq & 0 \\
 & -A & & & + & G_{11} & + & 2G_{21} & & & & & \leq & 0 \\
 & A & & & - & G_{11} & - & G_{21} & & & & & = & 0 \\
 & & - & 2B & & & & & + & 2G_{12} & + & G_{22} & \leq & 0 \\
 & & - & 2B & & & & & + & G_{12} & + & 2G_{22} & \leq & 0 \\
 & & & B & & & & & - & G_{12} & - & G_{22} & = & 0 \\
 & & & & & G_{11} & & & + & G_{12} & & & \leq & 20 \\
 & & & & & & & G_{21} & & & + & G_{22} & \leq & 20 \\
 & A, & & B, & & G_{11}, & & G_{21}, & & G_{12}, & & G_{22} & \geq & 0
\end{array}
$$

Table 7.12. Optimal Solution to the Correct Formulation of the Grain Blending Problem

Objective = 80

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
A	0	0	1	0	1
B	40	0	2	0	1
G_{11}	0	0	3	0	5
G_{12}	20	0	4	20	0
G_{21}	0	0	5	20	0
G_{22}	20	0	6	0	2
			7	0	2
			8	0	2

More Linear Programming Models

Sequencing Problems

Sequencing Constraints:

Assuming that returns and resource usage are independent of activity timing we have:

Week1	$-X_1$					+	Y_1						$\leq$	0
Week2	$-X_1$	-	X_2			+	Y_1	+	Y_2				$\leq$	0
Week3	$-X_1$	-	X_2	-	X_3	+	Y_1	+	Y_2	+	Y_3		$\leq$	0
Week1	aX_1					+	dY_1						$\leq$	T_1
Week2			bX_2					+	eY_2				$\leq$	T_2
Week3					cX_3					+	fY_3		$\leq$	T_3

When returns to the successor activities depend on the timing of the preceding activities we have:

Predecessor date	Wk 1		Wk 1		Wk 1		Wk 2		Wk 2		Wk 3		
Successor date	Wk 1		Wk 2		Wk 3		Wk 2		Wk 3		Wk 3		
Wk 1	aZ_{11}	+	bZ_{12}	+	dZ_{13}							$\leq$	T_1
Wk 2			cZ_{12}			+	fZ_{22}	+	gZ_{23}			$\leq$	T_2
Wk 3					eZ_{13}			+	hZ_{23}	+	iZ_{33}	$\leq$	T_3

More Linear Programming Models

Sequencing Problems

General Formulation

$$\text{Max} \quad -\sum_{j}\sum_{t_1} c_j X_{jt_1} \; - \; \sum_{k}\sum_{t_2} d_k Y_{kt_2} \; + \; \sum_{s}\sum_{t_3} e_s Z_{st_3}$$

$$\text{s.t.} \quad -\sum_{j}\sum_{t_1 \le t} X_{jt_1} + \sum_{k}\sum_{t_2 \le t} Y_{kt_2} \le 0 \qquad \text{for } t \in t_2$$

$$-\sum_{k}\sum_{t_2 \le t} Y_{kt_2} + \sum_{s}\sum_{t_3 \le t} Z_{st_3} \le 0 \qquad \text{for } t \in t_3$$

$$+\sum_{j} a_j X_{jt} + \sum_{k} b_j Y_{kt} + \sum_{s} f_s Z_{st} \le g_{mt} \qquad \text{for all } m, t$$

$$X_{jt}, \quad Y_{kt_2}, \quad Z_{st_2} \ge 0 \qquad \text{for all } j, k, s, t_1, t_2, t_3$$

More Linear Programming Models

Sequencing Problems- Example 1

Table 7.13. LP Formulation of Sequencing Example 1

		Plow - X			Disc - Y			Plant etc. - Z			RHS	
		April	May	June	May	June	July	May	June	July		
Obj		-100	-100	-100	-20	-20	-20	400	400	400		max
X – Y	May	-1	-1		1						≤	0
link	June	-1	-1	-1	1	1					≤	0
	July	-1	-1	-1	1	1	1				≤	0
Y – Z	May				-1			1			≤	0
link	June				-1	-1		1	1		≤	0
	July				-1	-1	-1	1	1	1	≤	0
Labor	April	0.2									≤	160
	May		0.2		0.3			0.3			≤	160
	June			0.2		0.3		0.1	0.3		≤	160
	July						0.3	0.1	0.1	0.3	≤	160
	Aug.							0.1	0.1	0.1	≤	160
	Sept.							0.5	0.1	0.1	≤	160
	Oct.								0.5	0.1	≤	160
	Nov.									0.5	≤	160
Land		1	1	1							≤	600

More Linear Programming Models

Sequencing Problems-Example 1 Solution

Table 7.14. Solution to Sequencing Example 1

Objective function = 168,000							
Variable		Value	Reduced Cost	Equation		Slack	Shadow Price
Plow	April	600	0	Plow-Disc	May	-192.59	0
	May	0	0 (alt)		June	200.00	0
	June	0	0 (alt)		July	0	380
Disc	May	407.41	0	Disc-Plant	May	88.89	0
	June	0	0		June	0	0
	July	192.59	0		July	0	400
Plant	May	125.93	0	Labor	April	97.78	0
	June	281.48	0		May	0	0
	July	192.59	0		June	0	0
					July	0	0
					Aug.	100	0
					Sept.	11.11	0
					Oct.	51.11	0
					Nov.	60	0
				Land		0	280

More Linear Programming Models

Sequencing Problems-Example 2

This example reflects a farm planning situation and illustrates what needs to be done when planting and harvesting dates influence yield

Table 7.15. Yields for Crops 1 and 2 by Crop Planting and Harvest Dates

	Planting Date					
Harvest Date	Crop 1			Crop 2		
	April	May	June	April	May	June
September	110	105	90	38	40	35
October	125	120	118	35	38	40

More Linear Programming Models

Sequencing Problems-Example 2

Rows																Mar	April	May	Mar	April	May										
		Mar	Apr	May	Jun	Apr	May	Jun	Apr	May	Jun	Apr	May	Jun	Apr	May	Jun	Apr	May	Jun	Mar	Apr	May	Jun	Sep	Oct	Nov	Crop 1	Crop 2	
									Sep	Sep	Sep	Oct	Oct	Oct	Sep	Sep	Sep	Oct	Oct	Oct										
Objective		-5	-5	-5	-3	-3	-3	-60	-60	-60	-60	-60	-60	-43	-43	-43	-43	-43	-43	-10	-10	-10	-10	-10	-10	-10	-10	3	8.7	Max
Land Balance		1	1	1	1																									≤ 1500
Plowed Land Balan	Mar	-1													1			1												≤ 0
	Apr	-1	-1			1									1	1		1	1											≤ 0
	May	-1	-1	-1		1	1								1	1	1	1	1	1										≤ 0
	Jun	-1	-1	-1	-1	1	1	1							1	1	1	1	1	1										≤ 0
Disced Land Balan	Apr					-1			1			1																		≤ 0
	May					-1	-1		1	1		1	1																	≤ 0
	Jun					-1	-1	-1	1	1	1	1	1	1																≤ 0
Labor Avail-Ability	Mar	0.3													0.2			0.2			-1									≤ 300
	Apr		0.3			0.2			0.22			0.22			0.22	0.2		0.22	0.2			-1								≤ 300
	May			0.3			0.2		0.1	0.22		0.1	0.22		0.1	0.22	0.2	0.1	0.22	0.2			-1							≤ 300
	Jun				0.3			0.2		0.1	0.22		0.1	0.22		0.1	0.22		0.1	0.22				-1						≤ 300
	Jul										0.1			0.1			0.1			0.1					-1					≤ 300
	Sep								0.7	0.7	0.7				0.6	0.6	0.6									-1				≤ 300
	Oct											0.7	0.7	0.7				0.6	0.6	0.6							-1			≤ 300
Yield	Crop 1								-110	-105	-90	-125	-120	-118														1		≤ 0
	Crop 2														-38	-40	-35	-35	-38	-40									1	≤ 0

More Linear Programming Models

Sequencing Problems-Example 2 Solution

Table 7.17. Solution for Sequencing Example 2

Objective function = 449,570

Variable		Value	Reduced Cost
Acreage Plowed in:	March	1275	0
	April	0	0
	May	225	0
	June	0	0
Acreage Disced for Crop 1 in:	April	775	0
	May	0	0
	June	0	0
Acreage of Crop 1 planted/harvested in:	Sept./April	0	-40.15
	Sept./May	0	-49.81
	Sept./June	0	-92.65
	Oct./April	775	0
	Oct./May	0	-9.66
	Oct./June	0	-13.5
Acreage of Crop 2 planted/harvested in:	Sept./April	0	-19.24
	Sept./May	500	0
	Sept./June	0	-39.34
	Oct./April	0	-49.5
	Oct./May	0	-21.56
	Oct./June	225	0
Labor hired in:	March	82.5	0
	April	125.5	0
	May	0	-7
	June	0	-10
	July	0	-10
	Sept.	0	-6.93
	Oct.	377.5	0
Crop 1 Sales		96875	0
Crop 2 Sales		29000	0

Equation		Slack	Shadow Price
Land		0	292.5
Plowed Land:	March	1275	0
	April	0	2.10
	May	0	14.4
	June	0	284.0
Disced Land:	April	0	13.16
	May	0	5.34
	June	0	287.0
Labor:	March	0	10
	April	0	10
	May	0	3
	June	200.5	0
	July	277.5	0
	Sept.	0	3.067
	Oct.	0	10
Yield:	Crop 1	0	3
	Crop 2	0	8.7

More Linear Programming Models

The Storage Problem

Primal Algebra

$$Max \sum_t c_t X_t - \sum_{t\, t \neq T} cs_t H_t \quad s.t. \quad X_1 \quad + \; H_1 \leq s_0 \quad X_t \; -$$

Objective: It involves summation across all the periods of the revenues from the sales of the good less the costs of storage of the good. We only include storage from the time periods 1 through T-1, assuming that everything must be sold in the last time period.

Constraints: The first constraint limits the quantity sold in the first period plus the quantity stored into the second period to be less than or equal to the initial inventory available.

The next constraints are active in all time periods excepting 1 and T. This limits the amount sold in each period plus the amount stored into the next period to not exceed the amount held over from the period before.

The third constraint gives the inventory condition for the last time period requiring that sales not exceed inventory carried over from the time period before.

The next two constraints impose upper and lower limits on the amount that can be sold during any time period.

The last constraint imposes an upper limit on storage in the first period.

More Linear Programming Models

The Storage Problem – An Example

Table 7.18. Formulation of Storage Example

		Sell				Store			
Objective		$2.3X_1$	$+ 2.5X_2$	$+ 2.7X_3$	$+ 2.9X_4$	$- .1h_1$	$- .2h_2$	$- .3h_3$	
Grain Inventory	1	X_1				$+ h_1$			≤ 100
	2		X_2			$- h_1$	$+ h_2$		≤ 0
	3			X_3			$- h_2$	$+ h_3$	≤ 0
	4				X_4			$- h_3$	≤ 0
	1	X_1							≤ 50
Max	2		X_2						≤ 50
Sales	3			X_3					≤ 50
	4				X_4				≤ 50
Min	1	X_1							≥ 15
Sales	2		X_2						≥ 5
Max Store						h_1			≤ 75

More Linear Programming Models

The Storage Problem – Example Solution

Table 7.19. Primal Solution to the Storage Problem Example

Objective = 237.5

Variable	Value	Reduced Cost	Constraint	Slack	Shadow Price
X_1	25	0	Pd1 Inventory	0	2.3
X_2	50	0	Pd2 Inventory	0	2.5
X_3	25	0	Pd3 Inventory	0	2.7
X_4	0	0	Pd4 Inventory	0	2.9
h_1	75	0	Max sale Pd1	25	0
h_2	25	0	Max sale Pd2	0	0
h_3	0	-0.1	Max sale Pd3	25	0
			Max sale Pd4	50	0
			Capacity	0	0.1
			Min sale Pd1	10	0
			Min sale Pd2	45	0
			Min sale Pd3	25	0
			Min sale Pd4	0	0

More Linear Programming Models

Block Diagonal

- This model depicts production in several different locations and/or time periods.
- The blocks arise when individual production units utilize immobile resources.
- The problem also depicts some usage of unifying resources at the overall firm level.

$$\begin{array}{llcllll}
\text{Max} & \sum_k c_k X_k & + & \sum_j \sum_L d_{jL} Y_{jL} & & & \\
\text{s.t.} & \sum_k a_{ik} X_k & + & \sum_j \sum_L g_{ijL} Y_{jL} & \leq & b_i & \text{for all I} \\
 & & & \sum_j e_{jLM} Y_{jL} & \leq & f_{LM} & \text{for all L and M} \\
 & X_k & , & Y_{jL} & \geq & 0 & \text{for all k, j and L}
\end{array}$$

Objective: The problem maximizes profit summed over the global and sub-unit activities subject to an overall linking constraint and individual sub-unit constraints.

A Closer Look

$$
\begin{array}{ccccccccccccc}
cX & + & d_1 Y_1 & + & d_2 Y_2 & . & . & . & + & d_n Y_n & & \max \\
AX & + & g_1 Y_1 & + & g_2 Y_2 & . & . & . & + & g_n Y_n & \leq & b \\
 & & & & e_1 Y_1 & & & & & & \leq & f_1 \\
 & & & & & e_2 Y_2 & & & & & \leq & f_2 \\
 & & & & & & . & & & & & . \\
 & & & & & & & . & & & & . \\
 & & & & & & & & . & & & . \\
 & & & & & & & & & e_n Y_n & \leq & f_n
\end{array}
$$

More Linear Programming Models Block Diagonal - Example

Table 7.24. Matrix Formulation of Block Diagonal Problem

		PLANT 1			PLANT 2				PLANT 3							
		Sell Sets FC FY	Make Table FC FY	Sell Table	Transport Chair FC FY	Sell Chair FC FY	Make Functional Chairs Norm MxSm MxLg	Make Fancy Chairs Norm MxSm MxLg	Transport Table FC FY	Transport Chair FC FY	Sell Table FC FY	Sell Chair FC FY	Make Table FC FY	Make Functional Chairs Norm MxSm MxLg	Make Fancy Chairs Norm MxSm MxLg	RHS
Objective		600 100	-80 -100	200 300	-5 -5	82 105	-15 -16 -15.7	-25 -26 -26.6	-20 -20	-7 -7	200 300	82 105	-80 -100	-15 -16 -15.7	-25 -26.5 -26.5	Max
PLANT 1	Table FC	1	-1	1					-1							≤ 0
	Inventory FY	1	-1	1					-1							≤ 0
	Chair FC	4			-1					-1						≤ 0
	Inventory FY	6			-1					-1						≤ 0
	Labor		3 5													≤ 175
	Top Capacity		1 1													≤ 50
PLANT 2	Chair FC				1	1	-1 -1 -1									≤ 0
	Inventory FY				1	1		-1 -1 -1								≤ 0
	Small Lathe						0.8 1.3 0.2	1.2 1.7 0.5								≤ 140
	Large Lathe						0.5 0.2 1.3	0.7 0.3 1.5								≤ 90
	Chair Bottom Carver						0.4 0.4 0.4	1 1 1								≤ 120
	Labor						1 1.05 1.1	0.8 0.82 0.84								≤ 125
PLN 3	Table FC								1		1		-1			≤ 0
	Inventory FY								1		1		-1			≤ 0
	Chair FC									1		1		-1 -1 -1		≤ 0
	Inventory FY									1		1			-1 -1 -1	≤ 0
	Small Lathe													0.8 1.3 0.2	1.2 1.7 0.5	≤ 130
	Large Lathe													0.5 0.2 1.3	0.7 0.3 1.5	≤ 100
	Chair Bottom Carver													0.4 0.4 0.4	1 1 1	≤ 110
	Labor												3 5	1 1.05 1.1	0.80 0.82 0.84	≤ 210
	Top Capacity												1 1			≤ 40

More Linear Programming Models
Block Diagonal – Example Solution

Table 7.25. Primal Solution to the Block Diagonal Problem

Objective = 36206.9

Variable		Value	Reduced Cost	Equation		Slack	Shadow Price
Plant1	Sell FC set	24.40	0	Plant1	FC Tables	0	212
	Sell FY set	29.01	0		FY Tables	0	320
	Make FC Table	24.40	0		FC Chairs	0	97
	Make FY Table	20.36	0		FY Chairs	0	130
	Sell FC Table	0	-12		Labor	0	44
	Sell FY Table	0	-20		Top Cap	5.240	0
Plant2	Trans FC Chair	62.23	0	Plant2	FC Chair	0	92
	Trans FY Chair	78.2	0		FY Chair	0	125
	Sell FC Chair	0	-10		Sm Lathe	0	47.77
	Sell FY Chair	0	-20		Lrg Lathe	0	38.83
	Make FC Table	0	-58.11		Chair Bot	16.907	0
	Make FY Table	0	-96.85		Labor	0	19.37
	Make FC Chair N	62.23	0	Plant3	FC Table	0	200
	Make FC Chair MS	0	-14.2		FY Table	0	300
	Make FC Chair ML	0	-5.04		FC Chair	0	90
	Make FY Chair N	73.02	0		FY Chair	0	123
	Make FY Chair MS	0	-10.24		Sm Lathe	0	18.50
	Make FY Chair ML	5.18	0		Lrg Lathe	0	12.19
Plant3	Trans FC Table	0	-8		Chair Bot	0	35.27
	Trans FY Table	8.649	0		Labor	0	40.00
	Trans FC Chair	35.37	0		Top Cap	20.562	0
	Trans FY Chair	95.85	0				
	Sell FC Table	0	0				
	Sell FY Table	10.79	0				
	Sell FC Chair	0	-8				
	Sell FY Chair	0	-18				
	Make FC Table	0	0				
	Make FY Table	19.44	0				
	Make FC Chair N	35.37	0				
	Make FC Chair MS	0	-8.59				
	Make FC Chair ML	0	-3.35				
	Make FY Chair N	76.83	0				
	Make FY Chair MS	0	-6.68				
	Make FY Chair ML	19.02	0				