More on Indivisibilities

Bruce A. McCarl

Specialist in Applied Optimization
Regents Professor of Agricultural Economics,
Texas A& M University
Principal, McCarl and Associates

bruceamccarl@cox-internet.com
agecon.tamu.edu/faculty/mccarl

979-693-5694
979-845-1706

More on Indivisibilities

Common formulations

Knapsack Problem

The knapsack problem, also known as the capital budgeting or cargo loading problem, is a famous IP formulation.

The knapsack context refers to a hiker selecting 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

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

More on Indivisibilities

Common formulations -- Knapsack Problem

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 that can be taken and that their names, volumes, and values are

Table 16.1. Items for the Knapsack Example

Variable	Item Name	Item Volume (Cubic feet)	Item Value ($)
X_1	Bed and mattress	70	17
X_2	TV set	10	5
X_3	Turntable and records	20	22
X_4	Armchairs	20	12
X_5	Air conditioner	15	25
X_6	Garden tools and fencing	5	1
X_7	Furniture	120	15
X_8	Books	5	21
X_9	Cooking utensils	20	5
X_{10}	Appliances	20	20

More on Indivisibilities Common formulations -- Knapsack Problem

$$\text{Max } 17x_1 + 5x_2 + 22x_3 + 12x_4 + 25x_5 + x_6 + 15x_7 + 21x_8 + 5x_9 + 20x_{10}$$

$$\text{s.t. } 70x_1 + 10x_2 + 20x_3 + 20x_4 + 15x_5 + 5x_6 + 120x_7 + 5x_8 + 20x_9 + 20x_{10} \leq 250$$

$$x_j = 0 \text{ or } 1, \text{ for all } j$$

obj=128

Variable	Value	Reduced Cost
X_1	1	17
X_2	1	5
X_3	1	22
X_4	1	12
X_5	1	25
X_6	1	1
X_7	0	15
X_8	1	21
X_9	1	5
X_{10}	1	20

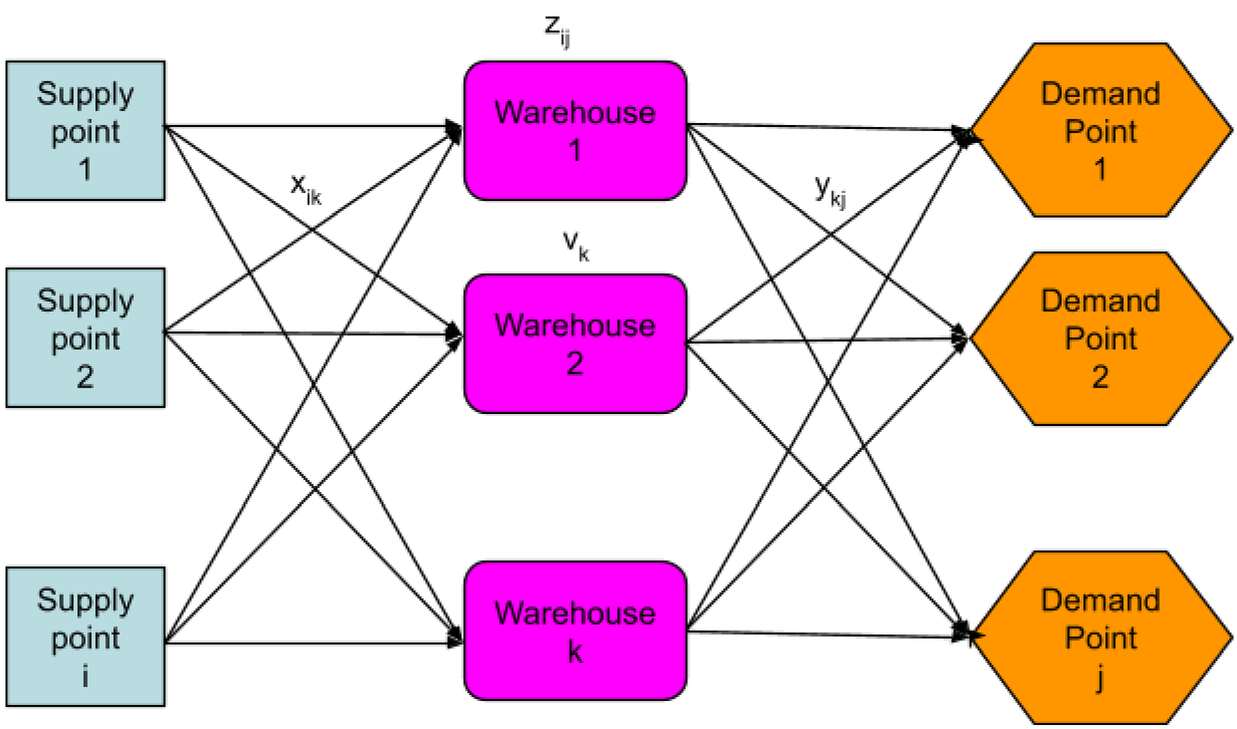
Constraint	Activity	Shadow Price
Space	185	0

Handling Indivisibilities

Warehouse Location

McCarl and Spreen Chapter 16

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



More on Indivisibilities

Common formulations

Warehouse Location (warehous.gms)

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

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.

Table 16.6. Formulation of the Warehouse Location Example Problem

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_{21}	Z_{12}	Z_{22}	RHS	
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			AA	50
										1		1		1			1	1	AA	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																	ε (0,1)	
								x_{ik},							Y_{kj},		Z_{ij}		AA	0

-1 1 =l= 0

Table 16.7. Solution Results for the Warehouse Location Example Obj = 623

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
V_A	0	0	1	0	-3.00
V_B	0	2	2	0	0
V_C	1	0	3	0	7.00
X_{1A}	0	0	4	0	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	0	-2
Y_{B1}	0	0			
Y_{B2}	0	3.00			
Y_{C1}	20	0			
Y_{C2}	50	0			
Z_{11}	50	0			
Z_{12}	0	6.00			
Z_{21}	5	0			
Z_{22}	0	1.00			

More on Indivisibilities
Common formulations

Warehouse Location In GAMS (warehous.gms)

```
 BINARY VARIABLES
         BUILD(WAREHOUSE)           WAREHOUSE CONSTRUCTION VARIABLES
 POSITIVE VARIABLES
         SHIPSupWar(SUPPLYL,WAREHOUSE) AMOUNT SHIPPED TO WAREHOUSE
         SHIPWarMkt(WAREHOUSE,MARKET)  AMOUNT SHIPPED FROM WAREHOUSE
         SHIPSupMkt(SUPPLYL,MARKET)    AMOUNT SHIPPED DIRECT TO DEMAND;
 VARIABLES
         TCOST                       TOTAL COST OF SHIPPING OVER ALL ROUTES;
 EQUATIONS
         TCOSTEQ                     TOTAL COST ACCOUNTING EQUATION
         SUPPLYEQ(SUPPLYL)           LIMIT ON SUPPLY AVAILABLE AT A SUPPLY POINT
         DEMANDEQ(MARKET)            MINIMUM REQUIREMENT AT A DEMAND MARKET
         BALANCE(WAREHOUSE)          WAREHOUSE SUPPLY DEMAND BALANCE
         CAPACITY(WAREHOUSE)         WAREHOUSE CAPACITY
         CONFIGURE                   ONLY ONE WAREHOUSE;
 TCoSTEQ..   TCOST =E=     SUM(WAREHOUSE,
         DATAWar(WAREHOUSE,"COST")/DATAWar(WAREHOUSE,"LIFE")*BUILD(WAREHOUSE))
   +SUM((SUPPLYL,MARKET)
,SHIPSupMkt(SUPPLYL,MARKET)*COSTSupMkt(SUPPLYL,MARKET))

+SUM((SUPPLYL,WAREHOUSE),SHIPSupWar(SUPPLYL,WAREHOUSE)*COSTSupWar(SUPPLYL,WAREH
OUSE))
   +SUM((WAREHOUSE,MARKET) ,SHIPWarMkt(WAREHOUSE,MARKET)
*COSTWarMkt(WAREHOUSE,MARKET));
 SUPPLYEQ(SUPPLYL)..     SUM(MARKET, SHIPSupMkt(SUPPLYL, MARKET))
                       + SUM(WAREHOUSE,SHIPSupWar(SUPPLYL,WAREHOUSE))
                        =L= SUPPLY(SUPPLYL);
 DEMANDEQ(MARKET)..     SUM(SUPPLYL, SHIPSupMkt(SUPPLYL, MARKET))
                     +  SUM(WAREHOUSE, SHIPWarMkt(WAREHOUSE, MARKET))
                        =G= DEMAND(MARKET);
 BALANCE(WAREHOUSE)..     SUM(MARKET, SHIPWarMkt(WAREHOUSE, MARKET))
                        - SUM(SUPPLYL,SHIPSupWar(SUPPLYL,WAREHOUSE)) =L= 0;
 CAPACITY(WAREHOUSE)..   SUM(MARKET, SHIPWarMkt(WAREHOUSE, MARKET))
                        -BUILD(WAREHOUSE)*DATAWar(WAREHOUSE,"CAPACITY")  =L= 0 ;
 CONFIGURE..             SUM(WAREHOUSE,BUILD(WAREHOUSE)) =L= 1;
```

More on Indivisibilities

Common formulations -- Decreasing Cost deccost.gms

The basic problem in matrix form is

$$
\begin{array}{llllllll}
\text{Max} & eY & & & - & f(Z) & & \\
\text{s.t.} & Y & - & \sum_m G_m X_m & & & \leq & 0 \\
 & & & \sum_m A_m X_m & - & Z & \leq & 0 \\
 & & & \sum_m H_{im} X_m & & & \leq & b_i \quad \text{for all } i \\
 & Y, & & X_m, & & Z & \geq & 0
\end{array}
$$

where

- Z is the quantity of input used,
- $f(Z)$ total cost of Z and exhibits diminishing marginal cost (per unit cost falls as more purchased);
- e is the sale price for a unit of output (Y);
- G_m is the quantity of output produced per unit of production activity X_m;
- A_m is the amount of resource used per unit of X_m; and
- H_{im} is the number of units of the i^{th} fixed resource which is used per unit of X_m.

Objective function maximizes total revenue (eY) less total costs ($f(Z)$). The first constraint balances products sold (Y) with production ($\Sigma\, G_m X_m$). Second balances input usage ($\Sigma\ A_m X_m$) with supply (Z). Third balances resource usage ($\Sigma\ H_{im} X_m$) with exogenous supply (b_i).

More on Indivisibilities

Common formulations -- Decreasing Cost deccost.gms

This problem may be reformulated as an IP problem by following an approximation point approach.

$$Max\ eY \quad - \quad \sum_k f^{'}(Z_k^{*})R_k \qquad s.t.\ Y \ - \ \sum_m G_m X_m \quad \leq \ 0 \quad \sum_m A_m X_m \ -$$

The variables are Y and X_m, as above, but the Z variable has been replaced with two sets of variables: R_k and D_k. The variables R_k which are the number of units purchased at cost $f'(Z_k^*)$; Z_k^* are a set of approximation points for Z where $Z_0^* = 0$; where $f'(Z_k^*)$ is the first derivative of the f(Z) function evaluated at the approximation point Z_k^*. While simultaneously the data for D_k is a zero-one indicator variable indicating whether the k^{th} step has been fully used.

This problem may be reformulated as an IP problem by following an approximation point approach.

$$
\begin{array}{llllllll}
\text{Max} & eY & & - & \sum_k f'(Z_k^*) R_k & & & \\
\text{s.t.} & Y & - \sum_m G_m X_m & & & & \leq & 0 \\
 & & \sum_m A_m X_m & - & \sum_k R_k & & \leq & 0 \\
 & & \sum_m H_{im} X_m & & & & \leq & b_i \\
 & & & & R_k & - (Z_k^* - Z_{k-1}^*) D_k & \leq & 0 \\
 & & & & - R_k & + (Z_k^* - Z_{k-1}^*) D_{k+1} & \leq & 0 \\
 & Y, & X_m, & & R_k & & \geq & 0 \\
 & & & & & D_k = 0 \text{ or } 1 & &
\end{array}
$$

The variables are Y and X_m, as above, but the Z variable has been replaced with two sets of variables: R_k and D_k. The variables R_k which are the number of units purchased at cost $f'(Z_k^*)$; Z_k^* are a set of approximation points for Z where $Z_0^* = 0$; where $f'(Z_k^*)$ is the first derivative of the $f(Z)$ function evaluated at the approximation point Z_k^*. While simultaneously the data for D_k is a zero-one indicator variable indicating whether the k^{th} step has been fully used.

More on Indivisibilities

Common formulations -- Decreasing Cost deccost.gms

$Max\, 4Y \quad - \quad (3\; -.125Z)\, Z \quad Y \; - \; 2X \quad \leq \; 0 \quad X \; - \quad Z \; \leq \; 0 \quad X$

Suppose we approximate Z at 2, 4, 6, 8 and 10. The formulation becomes

$$
\begin{array}{lcccccccccccccccccc}
\text{Max} & 4Y & & & - & 2.50R_1 & - & 2.00R_2 & - & 1.50R_3 & - & 1.00R_4 & - & 0.50R_5 & & & \\
 & Y & - & 2X & & & & & & & & & & & & \leq & 0 \\
 & & & X & - & R_1 & - & R_2 & - & R_3 & - & R_4 & - & R_5 & & \leq & 0 \\
 & & & X & & & & & & & & & & & & \leq & 5 \\
 & & & & & R_1 & & & & & & & & & -2D_1 & \leq & 0 \\
 & & & & & & & R_2 & & & & & & & -2D_2 & \leq & 0 \\
 & & & & & & & & & R_3 & & & & & -2D_3 & \leq & 0 \\
 & & & & & & & & & & & R_4 & & & -2D_4 & \leq & 0 \\
 & & & & & & & & & & & & & R_5 & -2D_5 & \leq & 0 \\
 & & & & - & R_1 & & & & & & & & & +2D_2 & \leq & 0 \\
 & & & & & & - & R_2 & & & & & & & +2D_3 & \leq & 0 \\
 & & & & & & & & - & R_3 & & & & & +2D_4 & \leq & 0 \\
 & & & & & & & & & & - & R_4 & & & +2D_5 & \leq & 0
\end{array}
$$

More on Indivisibilities

Common formulations -- Decreasing Cost deccost.gms

Table 16.11. Solution to the Decreasing Costs Example

Objective function = 29.50

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
Y	10	0	Y balance	0	4.0
X	5	6.5	Z balance	0	1.5
R_1	2	0	R_1D_1	0	0
R_2	2	0	R_2D_2	0	0
R_3	1	0	R_3D_3	1	0
R_4	0	0	R_4D_4	0	0.5
R_5	0	0	R_5D_5	0	1.0
D_1	1	0	R_1D_2	0	1.0
D_2	1	-2	R_2D_3	0	0.5
D_3	1	-1	R_3D_4	1	0
D_4	0	1	R_4D_5	0	0
D_5	0	2			

More on Indivisibilities

Common formulations -- Decreasing Cost deccost.gms

Suppose cost of shipping decreases as volume shipped increases. A formulation is

```
sets volumelevs /Low,Medium,High/
sets volterms /
  Cost       cost per unit shipped
  MinimumQ   minimum volume that must be shipped to get this rate
  MaximumQ   maximum volume that gets this rate/
table volumedata(volterms,volumelevs) data on cost and min shipping
                Low    Medium        High
Cost            5         4           2.5
MinimumQ        0        10           100
MaximumQ        9        99          1000
scalar productioncost  /3/
       salesprice      /4/
variable            profit total objective function
binary variables    volumeuse(volumelevs)   volume level used
variables           production  level of production
                    amount(volumelevs) shipped at this volume level
                    sales amount sold
equations           obj   objective function
                    balanceprod   product balance
                    balancesales balance of shipped products
                    minlimitship(volumelevs) lower limits on shipping
                    maxlimitship(volumelevs) upper limits on shipping
                    mutexclusiv  can only use one volume level   ;
obj.. productioncost*production
 +sum(volumelevs,amount(volumelevs)*volumedata("cost",volumelevs))
     +sales*salesprice  =e= profit;
balanceprod..    sum(volumelevs,amount(volumelevs))=l=production;
balancesales..   sales=l=sum(volumelevs,amount(volumelevs));
minlimitship(volumelevs)..
    amount(volumelevs)
        =g=volumedata("minimumq",volumelevs)*volumeuse(volumelevs);
maxlimitship(volumelevs)..
    amount(volumelevs)
        =l= volumedata("maximumq",volumelevs)*volumeuse(volumelevs);
mutexclusiv.. sum(volumelevs,volumeuse(volumelevs))=l=1;
production.up=95;
model deccost /all/
solve deccost using mip maximizing profit;
```

Solution second integer variable is chosen

1-5 cars cost \$5 per car
6-50 cars cost \$4 per car
51 to 100 more cars cost \$3 per car

$$
\begin{array}{llllllll}
\textbf{Min} & 5x1 & +4x2 & +3x3 & & & & \\
 & X1 & +x2 & +x3 & & & & >= 11 \\
 & X1 & & & -5D1 & & & \leq 0 \\
 & & X2 & & & -50D2 & & \leq 0 \\
 & & & X3 & & & -100D3 & \leq 0 \\
 & & X2 & & & -6D2 & & \geq 0 \\
 & & & X3 & & & -51D3 & \geq 0 \\
 & & & & D1 & +D2 & +D3 & \leq 1 \\
 & & & & D1 & ,D2, & D3 & \text{is } (0,1)
\end{array}
$$

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

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

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}, usage of r^{th} machinery restriction when purchasing the k^{th} machinery type;

C_{jm}, the per unit net profit of X_{jm};

A_{ijkm}, per unit use by X_{jm} of i^{th} cap. 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																		
		Tractor		Plow		Planter		Disc		Harvester		Plow with Tractor 1 and Plow 1 in Period		Plow with Tractor 1 and Plow 2 in Period		Plow with Tractor 2 and Plow 1 in Period		Plow with Tractor 2 and Plow 2 in Period		Plant Disc 8 Tractor 1 Planter		Plant Disc 8 Tractor 2 Planter		Harvest with Tractor 1 Harvester		Harvest with Tractor 2 Harvester		Crop Sales	Input Pur-chases	
		1	2	1	2	1	2	1	2	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			.05	.04167							≤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										.1				.05											≤0
Capacity of Planter in period 2	1					-180														.1		0.05								≤0
	2						-180														0.0833		.0417							≤0
Capacity of Disc in period 2	1							-180												.1		0.05								≤0
	2								-180												0.0833		.0417							≤0
Capacity of Har-vester in period 3	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																								-140	-140	-140	-140	1		≤0
Input Balance																				1	1	1	1						-1	≤0

Handling Indivisibilities Machinery Selection

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 capactiy 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 Harvestor 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

More on Indivisibilities
Practical Side of Solving

Sounds good but integer problems can be hard to solve due to search nature of solution process

Three approaches can help

1. Reformulate

    a. to better tie integer variables together
    b. to better tie integer and continuous variables together
    c. to eliminate "unnecessary" cases of integer variables

2. Use MIP solver features through options and GAMS

3. Start with a good solution

I have more faith in the first strategy but sometimes latter two help

Also, these solves are generally slower so one must be patient

More on Indivisibilities
Practical Side of Solving
Tie integer variables better together

Often model formulations contain interrelated variables which the formulation and economics tie together. For example, in LPs that choose plant size and hire labor, one could count on the solution to hire enough people.

However, in the MIP world if one has 2 sets of integer variables one for plant size and one for labor force, I would recommend tying them together with constraints requiring the large plant to have a large labor force or that hiring certain sizes of labor force requires certain plant sizes

Consider a trash recycling problem. A formulation was set up to choose the size of a recycling effort including the size of the truck fleet, ferrous metals separator, glass separator, unsorted trash compactor, etc. The MIP was very slow, and the solutions were not good enough. I suggested including constraints so that a given number of tons of truck capacity implied a minimum size for the materials separator etc.
(BUYCOM(size)-BUYTRK(SIZE)=0;). In turn, the formulation yielded improved solutions faster

Why does this work? This eliminates irrelevant cases and shrinks the number of solutions that need to be searched.

More on Indivisibilities
Practical Side of Solving
Tie integer and LP variables better together

Often model formulations require that the integer variables take on certain values so that the continuous part of the problem is feasible. For example, in a warehouse location problems, a given volume of goods may need to go through some warehouse somewhere so that the problem be feasible.

In such cases, I would recommend that one require that the capacity of the warehouses built to be subjected to a lower bound constraint so that the capacity constructed exceeds the volume required.

In a problem I was solving for locating grain handling facilities I discovered that new facilities were needed for about 1/3 of the crop. By requiring a minimum volume of such facilities, I cut the required solution time by more than 75%.

Why did this work? Again I eliminated irrelevant cases and shrank the number of solutions which needed to be searched

More on Indivisibilities

Practical Side of Solving

Tie integer and LP variables better together

Better tying things together also works in terms of tying the integer variables to the continuous variables. For example, it is common in formulations to have constraints such as the following

$$Y_1 + Y_2 - Md \le 0$$

where the Y's are continuous, M is a capacity and d is a zero one indicator or facility construction variable.

In such cases, computational experiments have found that the solution to the problem with the addition of the two constraints below yields faster solutions.

$$\begin{aligned} Y_1 \quad\quad - Md &\le 0 \\ Y_2 - Md &\le 0 \end{aligned}$$

Why does this work. Provides a more direct link between the individual variables and the integer variable, not just an aggregate link. Also provides better signals when looking for variables on which to branch. Note the better solver may reformulate for this automatically but see if it works.

More on Indivisibilities
Practical Side of Solving
Limit Feasible region in terms of Integer Solution Space

In integer programming one should endeavor to add as many constraints as possible to limit the feasible solution space to the relevant solution space. Lets look at some reasons and approaches

Consider a problem with N zero-one variables. In such a case there are 2^N possible solutions. But suppose we know no more than 1 of the integer variables will be employed. If we enter a constraint requiring the sum of these N variables to be less than or equal to one, then the number of possible solutions falls to N+1 (one for all zeros plus N possibilities in which each of the integer variables is equal to one).

So impose whatever problem knowledge you can on the situation to limit the feasible integer space as this greatly reduces the size of the branch and bound search tree.

One can also go further with this topic by solving related problems which can be used to formulate constraints that limit the feasible space as we discuss on the following pages

More on Indivisibilities
Practical Side of Solving
Limit Feasible region in terms of Integer Solution Space
Insight from the RMIP

Yet another strategy that can be used to narrow the feasible integer space is to examine the LP solution and see if some insight about minimum and maximum values of integer variables can be gleaned.

Consider a machinery selection problem (a simple version of which is in machsel.gms). One can set up a MIP but solve it as an RMIP which treats the integer variables as if they were continuous. In turn, one might observe the values of the machinery purchase variables and use those values to formulate maximum and minimum limits for classes of variables.

In one case I did that and solved a model where the variables for purchase of tractors came out to be 3.4 and 4.1. In turn, I added constraints to the model that the integer variables be greater than or equal to 2 and less than or equal to 6. In a small model, this reduced solution time by 90%, and in the resultant large model we were in fact able to solve it and otherwise never would have

Again we reduced the number of possible solutions that needed to be considered in the search tree

More on Indivisibilities

Practical Side of Solving
Limit Feasible region in terms of Integer Solution Space
Insight from Auxiliary models

One may be able to gain insight into a problem by solving problems that are subsets and gain insight into overall problem feasible region restrictions.

Recently I did a model which was designed to locate personnel at a number of facilities. These facilities were spread across the nation and the question was do we locate a person at this place or serve this place from a nearby location or hire temporaries? This involved a large MIP with hundreds of integer location variables and over a million continuous variables meeting demands at the service locations (since the problem service aspect involved a monthly dimension). Initial attempts to solve the whole problem showed the solvers were very slow and probably would not converge. So we employed a strategy involving regional solutions. In particular, we solved for the number of people in a reduced service area like a 200 mile radius around San Francisco, and did this for 15 or so service sub areas. The resultant solutions revealed a set of possible repair man locations which could be dropped (those in the inner part of city radii which were not used) and also provided upper and lower bounds on the number of people to be hired. It also provided a feasible starting solution and an initial bound.

More on Indivisibilities
Practical Side of Solving
GAMS Options

While I advocate one try to solve integer problems faster by tightening the formulation, one can also employ GAMS and solver features to try to speed up solution processes.

Within GAMS there are two parameters that can be set

modelname.cheat=k; requires that subsequent solutions have an objective function which is at least k units (an absolute amount) better than the current solution (works in OSL and CPLEX)

option optcr=k; allows the solver to stop when the theoretically best possible integer solution is within k percent of the current best found integer solution. There is also the command option optca=k2; where k2 is an absolute amount.

Both of these options cause the solvers to give a solution that can be suboptimal falling only within the criteria specified of the best possible optimal solution. However, they reduce search time substantially and often the optimal solution is found or is much closer to the solution found than the bound.

Practical Side of Solving
Solver Options

Solver options can also be used

CPLEX permits one to use an options file (cplex.opt) which

- impose a trial solution specified for the integer variable levels as a starting solution (mipstart)
- impose priorities for variables to deal with first (mipordind). Note modelname.prioropt also permits management of this)
- alter the way problems are selected from the branch and bound tree (varsel,nodelsel)
- manage the memory used for the branch and bound tree

These and many other options are discussed in the CPLEX solver manual

OSL also permits options to be used that alter branch and bound strategies (particularly strategy 48 and bbpreproc)

More on Indivisibilities
Risk and Integer

Modelers may wish to impose integer restrictions on nonlinear formulations which for example treat risk.

GAMS contains the DICOPT and SBB solvers which permit this. They tie together other solvers. For example SBB can use CPLEX to solve IP sub-problems and CONOPT to solve nonlinear problems.

For example, suppose we impose restrictions in our portfolio problem that a minimum of 10 shares be bought if any and that we buy integer numbers of shares (INTEV.gms)

```
Integer VARIABLES   INVEST(STOCKS)       MONEY INVESTED IN EACH STOCK
binary variables    mininv(stocks)       at least 10 shares bought
VARIABLE               OBJ               NUMBER TO BE MAXIMIZED ;
EQUATIONS              OBJJ              OBJECTIVE FUNCTION
                       INVESTAV          INVESTMENT FUNDS AVAILABLE
                       minstock(stocks)  at least 10 units to be bought
                       maxstock(stocks)  Set up indicator variable ;
OBJJ.. OBJ =E=    SUM(STOCKS, MEAN(STOCKS) * INVEST(STOCKS))
               - RAP*(SUM(STOCK, SUM(STOCKS,
                       INVEST(STOCK)* COVAR(STOCK,STOCKS) * INVEST(STOCKS))));
INVESTAV..     SUM(STOCKS, PRICES(STOCKS) * INVEST(STOCKS)) =L= FUNDS;
minstock(stocks)..     invest(stocks) =g= 10*mininv(stocks);
maxstock(stocks)..     invest(stocks)=l=1000*mininv(stocks);
MODEL EVPORTFOL /ALL/ ;
SOLVE EVPORTFOL USING MINLP MAXIMIZING OBJ ;
```

When using DICOPT and SBB it is very important to tighten the link between continuous and integer variables