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

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

Max
$$\sum_{j} v_{j}X_{j}$$

s.t. $\sum_{j} d_{j}X_{j} \leq W$
 $X_{j} = 0 \text{ or } 1 \text{ for all } j$

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

		Item Volume	Item Value
Variable	Item Name	(Cubic feet)	(\$)
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

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

s.t. $70x_1 + 10x_2 + 20x_3 + 20x_4 + 15x_5 + 5x_6 + 120x_7 + 5x_8 + 20x_9 + 20x_{10} \le 250$
 $x_1 = 0$ or 1, for all j

obj=128

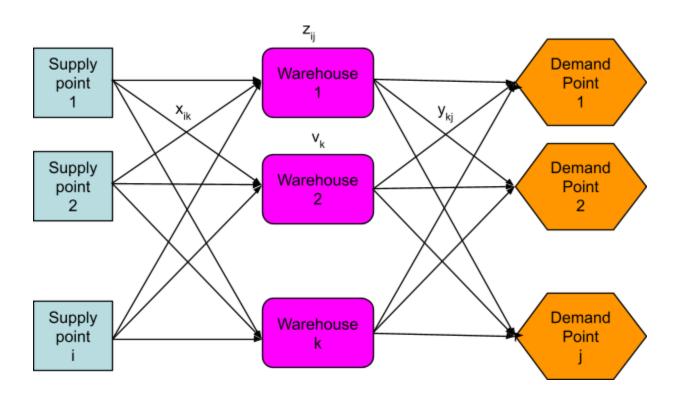
Variable	Value	Reduced Cost
\mathbf{X}_1	1	17
\mathbf{X}_2	1	5
X_3	1	22
X_4	1	12
X_5	1	25
X_6	1	1
X_7	0	15
X_8	1	21
X_9	1	5
X_{10}	1	20
Constraint	Activity	Shadow Price
Space	185	0

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.



Common formulations Warehouse Location (warehous.gms)

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.

Tabl	e 16.6.	For	mulat	tion o	f the	Warel	nouse	Loc	ation	Exar	nple l	Probl	em						
V_{A}	$V_{\rm B}$	V_{C}	X_{1A}	X_{1B}	\mathbf{X}_{1C}	X_{2A}	X_{2B}	X_{2C}	Y _{A1}	Y_{A2}	$Y_{\rm B1}$	$Y_{\rm B2}$	\mathbf{Y}_{C1}	$Y_{\rm C2}$	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		<u> </u>
						1	1	1								1		1	_ 50
									1		1		1		1	1			<u>AA</u> 50
										1		1		1			1	1	<u>AA</u> 75
			-1			-1			1	1									<u> </u>
				-1			-1				1	1							<u> </u>
					-1			-1					1	1					<u> </u>
-9999									1	1									<u> </u>
	-60										1	1							<u> </u>
		-70											1	1					<u> </u>
1	1	1																	<u> </u>
V _A ,	$V_{\rm B}$	V _C																	ε(0,1)
								X _{ik} ,							Y _{ki} ,		Z_{ij}		<u>AA</u> 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
$V_{\scriptscriptstyle B}$	0	2	2	0	0
$V_{\rm C}$	1	0	3	0	7.00
X_{1A}	0	0	4	0	5.00
${ m X_{1B}}$	0	2.00	5	0	-4
X_{1C}	0	10.00	6	0	-3.00
X_{2A}	0	2	7	0	-1.00
$ m X_{2B}$	0	0	8	0	-0.05
$ m 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
                                        WAREHOUSE CONSTRUCTION VARIABLES
          BUILD (WAREHOUSE)
 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
                                     TOTAL COST OF SHIPPING OVER ALL ROUTES;
          TCOST
 EOUATIONS
                                     TOTAL COST ACCOUNTING EQUATION
          TCOSTEO
         SUPPLYEQ (SUPPLYL)

DEMANDEQ (MARKET)

BALANCE (WAREHOUSE)

CAPACITY (WAREHOUSE)

CONFIGURE

LIMIT ON SUPPLY AVAILABLE AT A SUPPLY POINT

MINIMUM REQUIREMENT AT A DEMAND MARKET

WAREHOUSE SUPPLY DEMAND BALANCE

WAREHOUSE CAPACITY

ONLY ONE WAREHOUSE:
          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;
```

Common formulations -- Decreasing Cost deccost.gms

The basic problem in matrix form is

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 ith 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 (Σ G_mX_m). Second balances input

usage (Σ $A_m X_m$) with supply (Z). Third balances resource usage (Σ $H_{im} X_m$) with exogenous supply (b_i).

Common formulations -- Decreasing Cost deccost.gms

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

$$Max \ eY - \sum_{k} f'(Z_{k}^{*})R_{k} \quad s.t. \ Y - \sum_{m} G_{m}X_{m} \leq 0 \quad \sum_{m} A_{m}X_{m} -$$

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.

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.

Common formulations -- Decreasing Cost deccost.gms

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

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

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			

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 per unit shipped
  Cost
  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
                                      2.5
MinimumQ
                0
                          10
                                      100
MaximumQ
                          99
                                     1000
scalar productioncost
                      /3/
       salesprice
                  profit total objective function
variable
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;
                 sum(volumelevs, amount(volumelevs)) = l = production;
                 sales=l=sum(volumelevs, amount(volumelevs));
balancesales..
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 car6-50 cars cost \$4 per car51 to 100 more cars cost \$3 per car

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

The decision variables are

Y_k, the integer number of units of the kth type machinery purchased;

 X_{jm} , the quantity of the jth activity produced using the mth machinery alternative. The parameters of the model are:

F_k, the annualized fixed cost of the kth machinery type;

Cap_{ik}, the annual capacity of the kth machinery type to supply the ith resource;

G_{rk}, usage of rth machinery restriction when purchasing the kth 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 nth type by X_{jm} ;

b_n, the endowment of the nth resource in the year being modeled; and

e_r, the endowment of the rth machinery restriction.

Handling Indivisibilities Machinery Selection

												Machi	nery Use	Continuou	ıs Variables															1
		i										Plow v	with Tract	or l		Plow	with Tracto	or 2		Plant Dis	ic 8			Harvest	with					1
		Machinery Acquisition Integer Variables						and Pl	ow 1	and P	low 2	and P	low 1	and Pl	ow 2	Tractor 1		Tractor 2	2	Tractor	1	Tractor 2	!	Crop Sales	Input Pur-chases	İ				
		Tractor		Plow		Planter		Disc		Harveste	vester		iod	in Per	iod	in Per	in Period		in Period			Planter		Harvest	er	Harvester		i		İ
		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	i		İ
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	1
Tractor 1	1	-160										.2		.1																≤0
Capacity in Period	2	-180										į	.2		.1	į				.1	.0833			į				į	į	İ ≤0
Fractor 2	3	-200	-160									_				+		.05		+				.33	.25					≤0 ≤0
Capacity	2	1	-180									1				1.1	.1	.03	.05	ł		.05	.04167	ł				ł	ļ	≥0
n Period	3	<u>i </u>	-200									<u>i </u>				i	.1		.03	i		.03	.04107	i		.33	.25	i		≤0
Plow 1 Capacity in Period	1			-160								.2				.1														≤0
	2			-180									.2			_	.1			1										≤0
Plow 2 Capacity in Period	1	ļ.			-160							!		.1		ļ		.05		ļ				ļ				ļ	ļ	≤0
Capacity of	1	 			-180	-180						+			.1	+			.05	+		0.05		+						≤0 ≤0
Planter	1	!				-180	-180					-				!				1.1	0.0833	0.03		-				!	!	I ≤0
Capacity of	1	1					-180	-180				+				+				1	0.0833	0.05	.0417	+						≤0
Disc	2	ł							-180			ł				ł				1"	0.0833		.0417	ł				ł	ŀ	≤0
Capacity of Har-	1	 								-200		+				+				1			.0417	.33		.33				≤0
vester	2	i									-200	i				i				i				i	.25		.25	i	i	≤0
Labor	1	1										.24		.12		.12		.06												≤200
Available in Period	2	!										!	.24		.12	!	.12		.06	.12	.11	.06	.055	!				!	!	≤210
Plow-Plant Sequer	3	-										٠,	-1	- 1	-	+.	,	,	-1	1,	1	1		.5	.375	.5	.375			≤250
		-										-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-	1	٠.						≤0
Plant-Harvest Sequ	uencing	1														1				-1	-1	-1	-1	1	I	I	I			≤0
Land Available		+										1	1	1	1	+	1	1	1	+				+						≤600
Planters		1				1	1					╫	•	•		÷	•	•	•	+				+-						≤1
Discs		l				1	1	1	1			ł				1				ł				ł				1	1	
Link Disc-		1				-1		1	1			1				+				1										≤0
Planter		i				•	-1	•	1			i				i				i				i				i	i	_0
Yield Balance																1								-140	-140	-140	-140	1		≤0
Input Balance																1				1	1	1	1						-1	≤0

Handling Indivisibilities Machinery Selection

Table 16.17.Solution for the Machinery Selection Problemobj = 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

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.

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

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.

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

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

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)

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)

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