Integer Programming I

Modeling with Integer Variables
How the Solver Works
Complexity

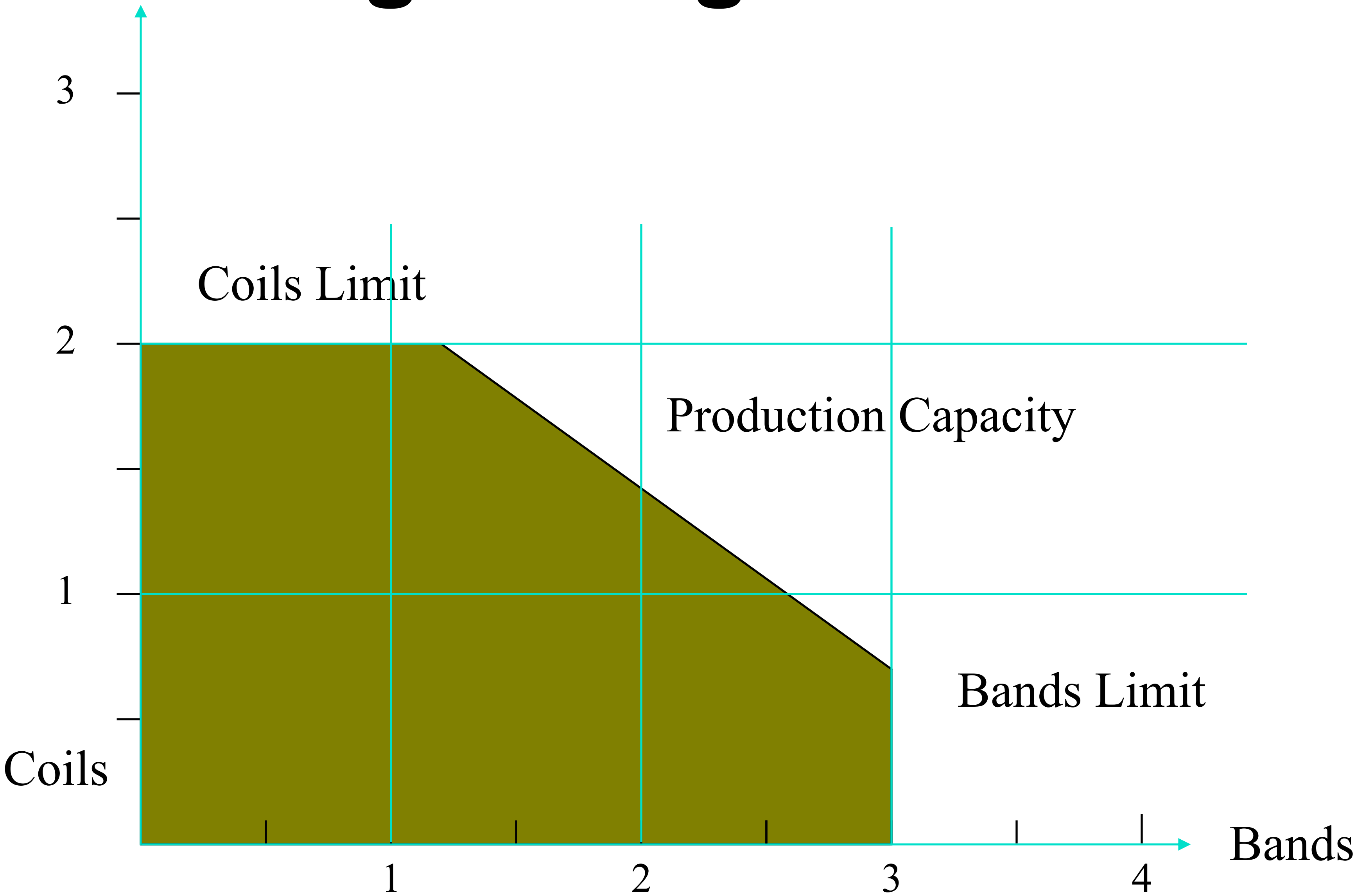
A Simple Example

Steel Cable Company

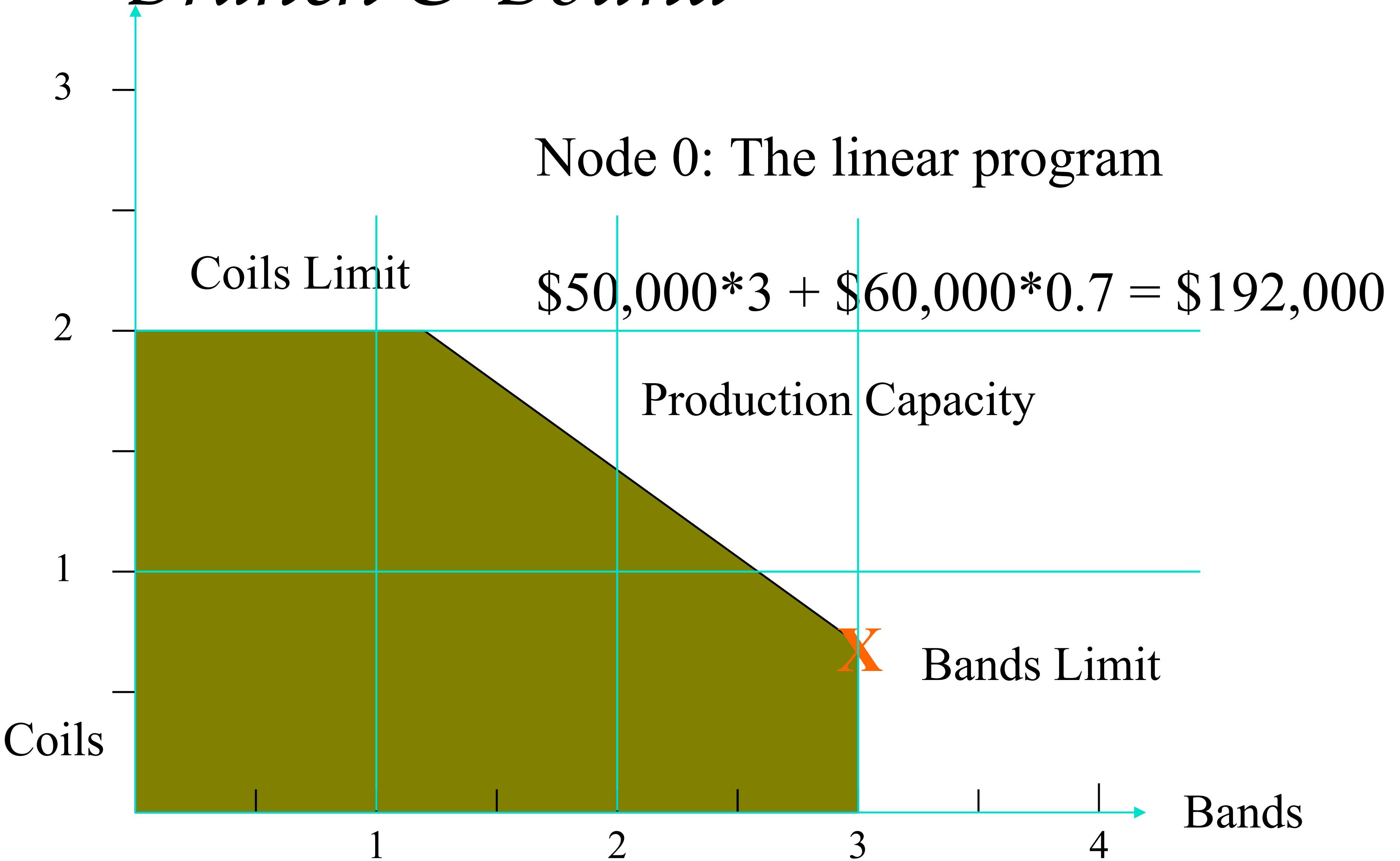
Integer Programming Example

Products	Profit/Ton	Market Limit (Tons)	Hours/Ton
Bands	$50,000	3	3
Coils	$60,000	2	4
		Total Hours	12

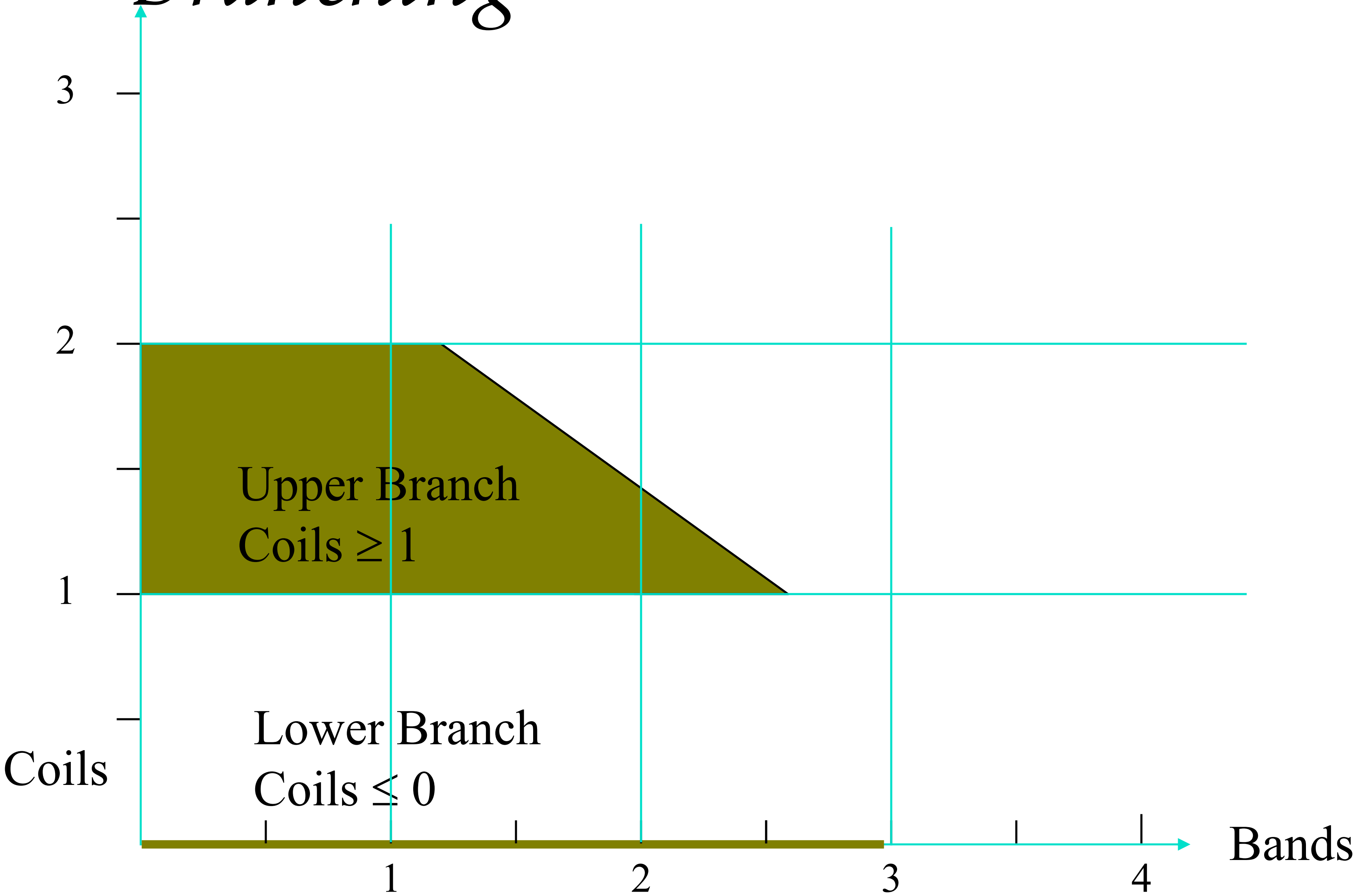
An Integer Program



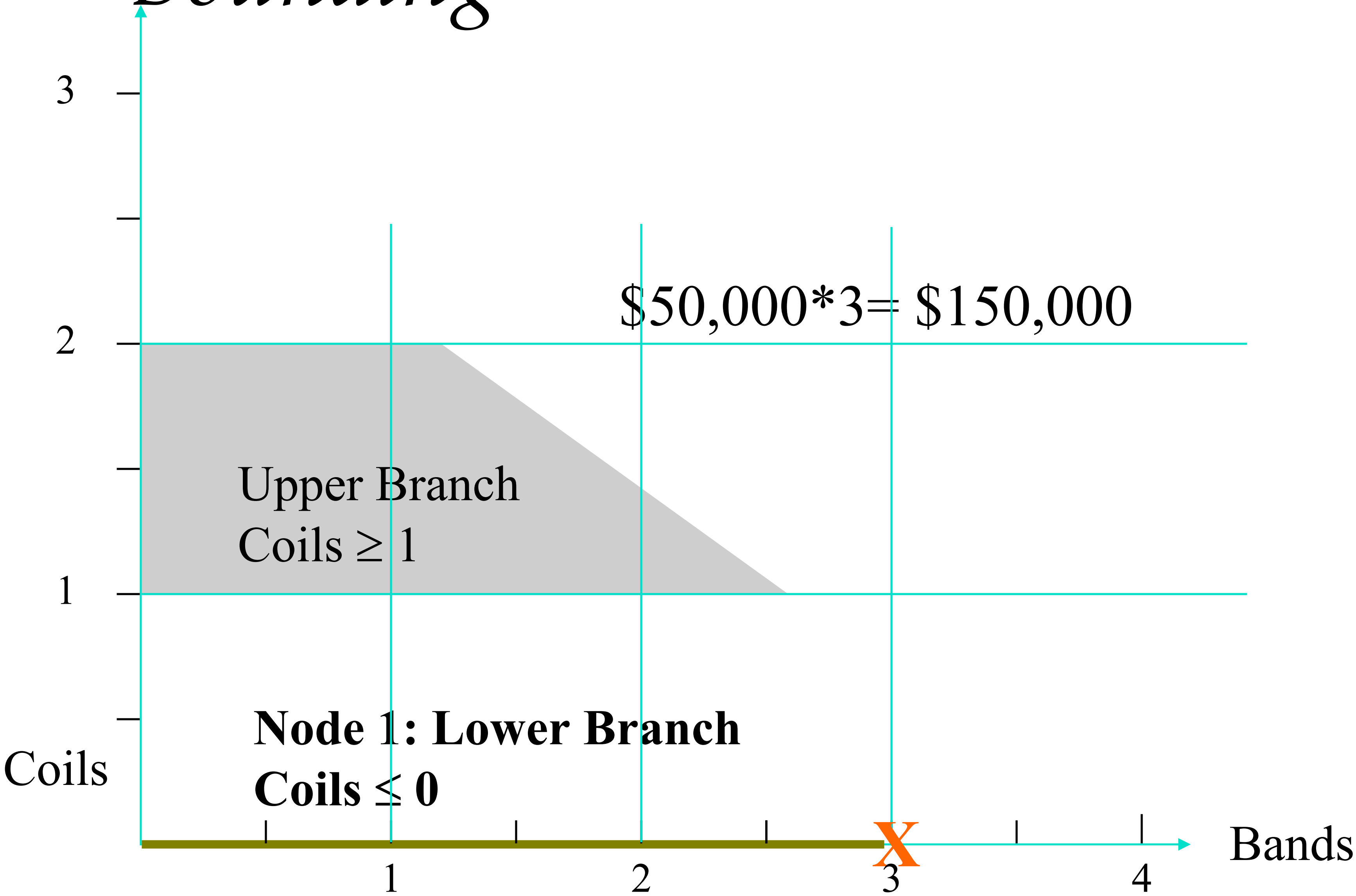
Branch & Bound



Branching

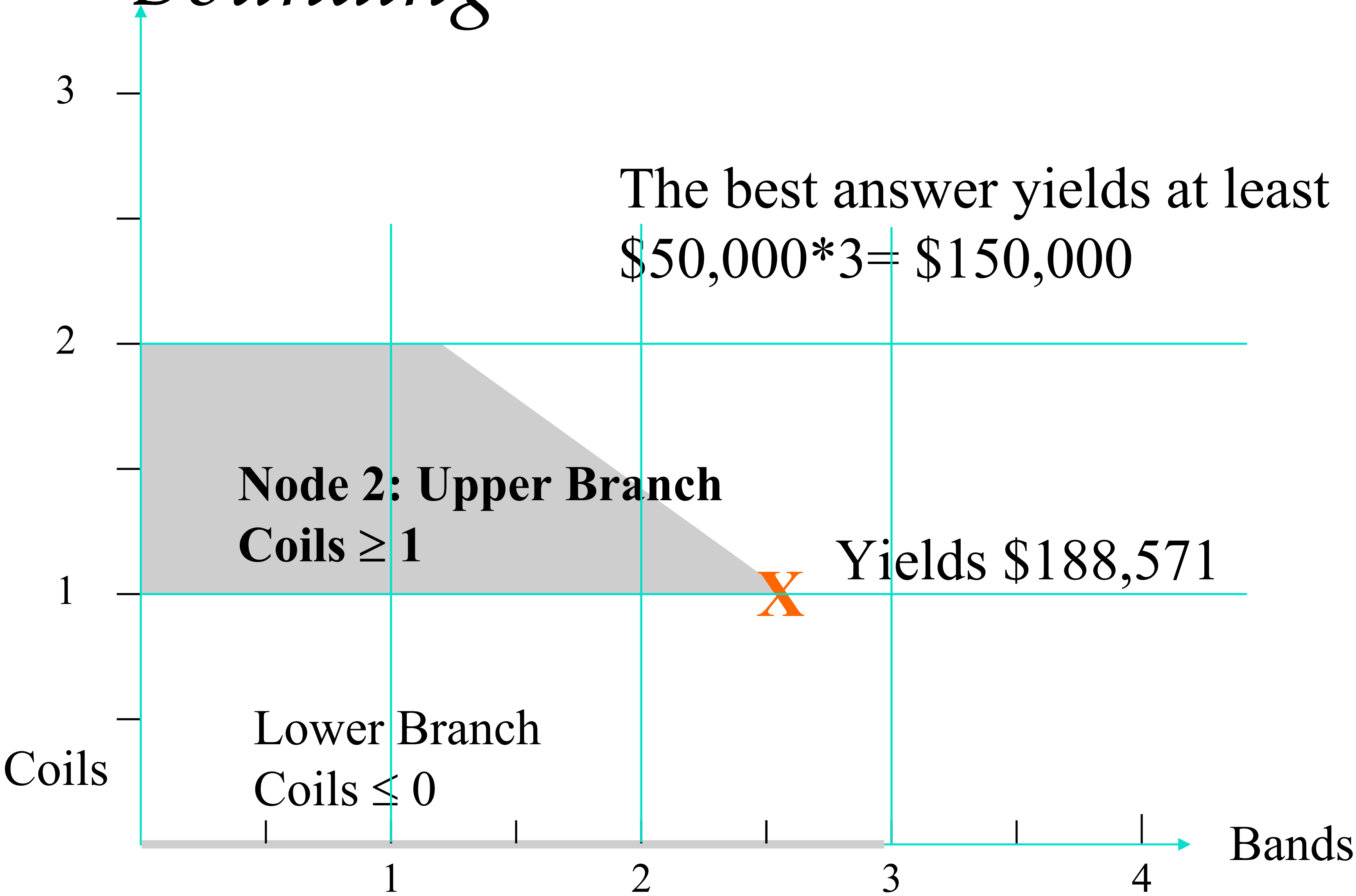

3
2
1
Coils
Upper Branch
Coils ≥ 1
Lower Branch
Coils ≤ 0
1
2
3
4
Bands

Bounding



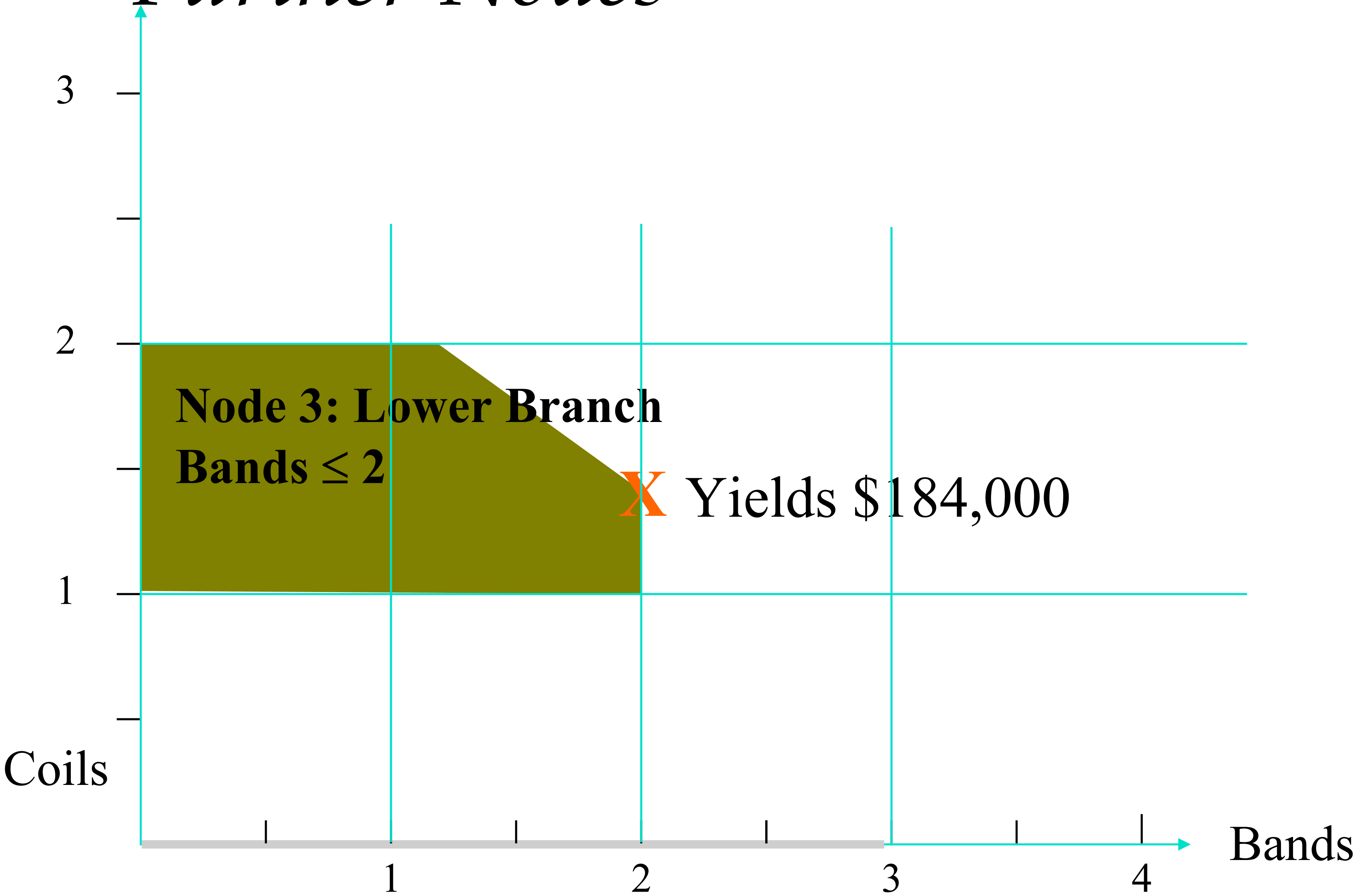
3
2
1
Coils
$50,000*3= $150,000
Upper Branch
Coils ≥ 1
Node 1: Lower Branch
Coils ≤ 0
X
1
2
3
4
Bands

Bounding

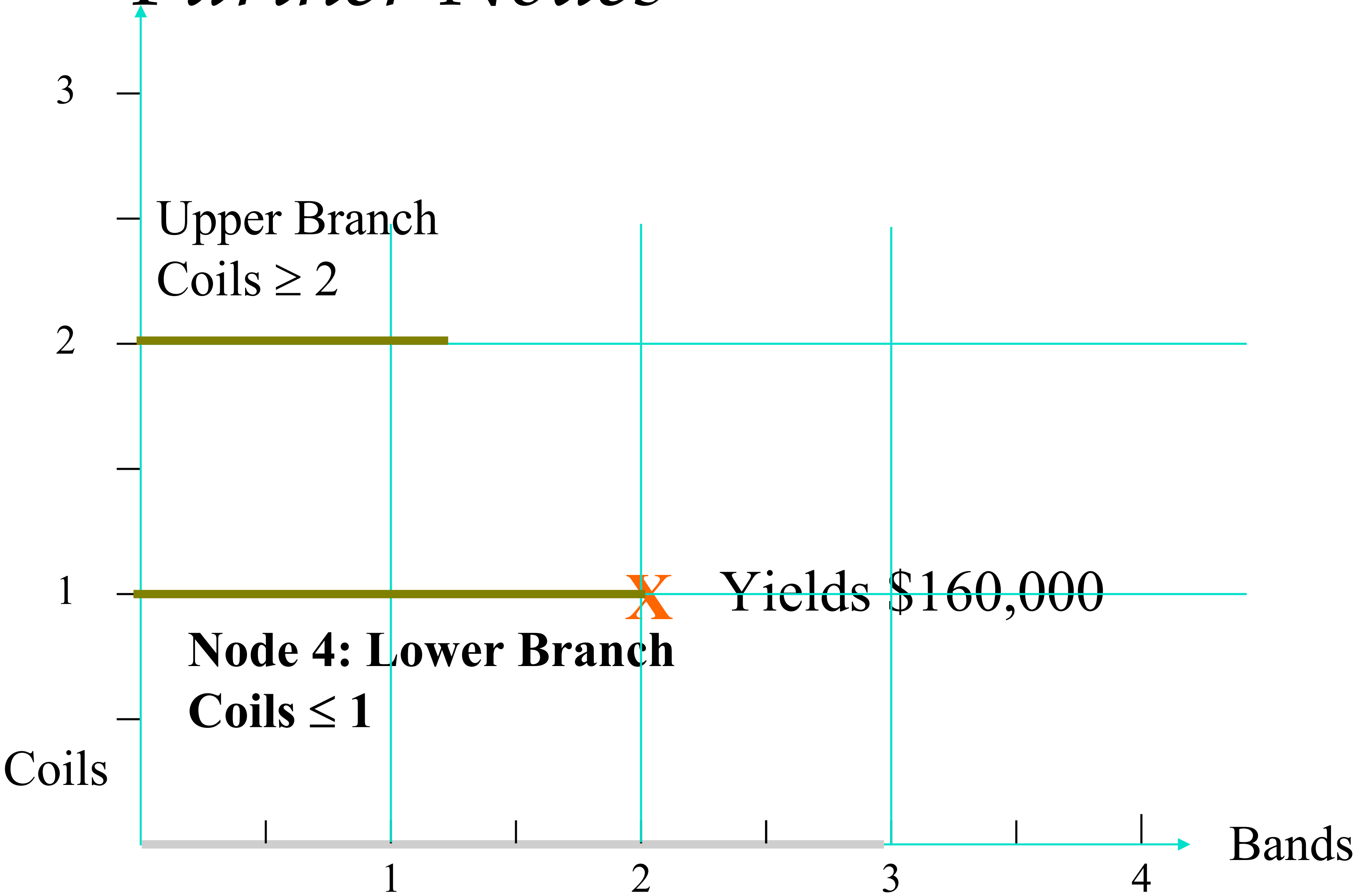


3
2
1
Coils
The best answer yields at least
$50,000*3= $150,000
Node 2: Upper Branch
Coils ≥ 1
X
Yields $188,571
Lower Branch
Coils ≤ 0
1
2
3
4
Bands

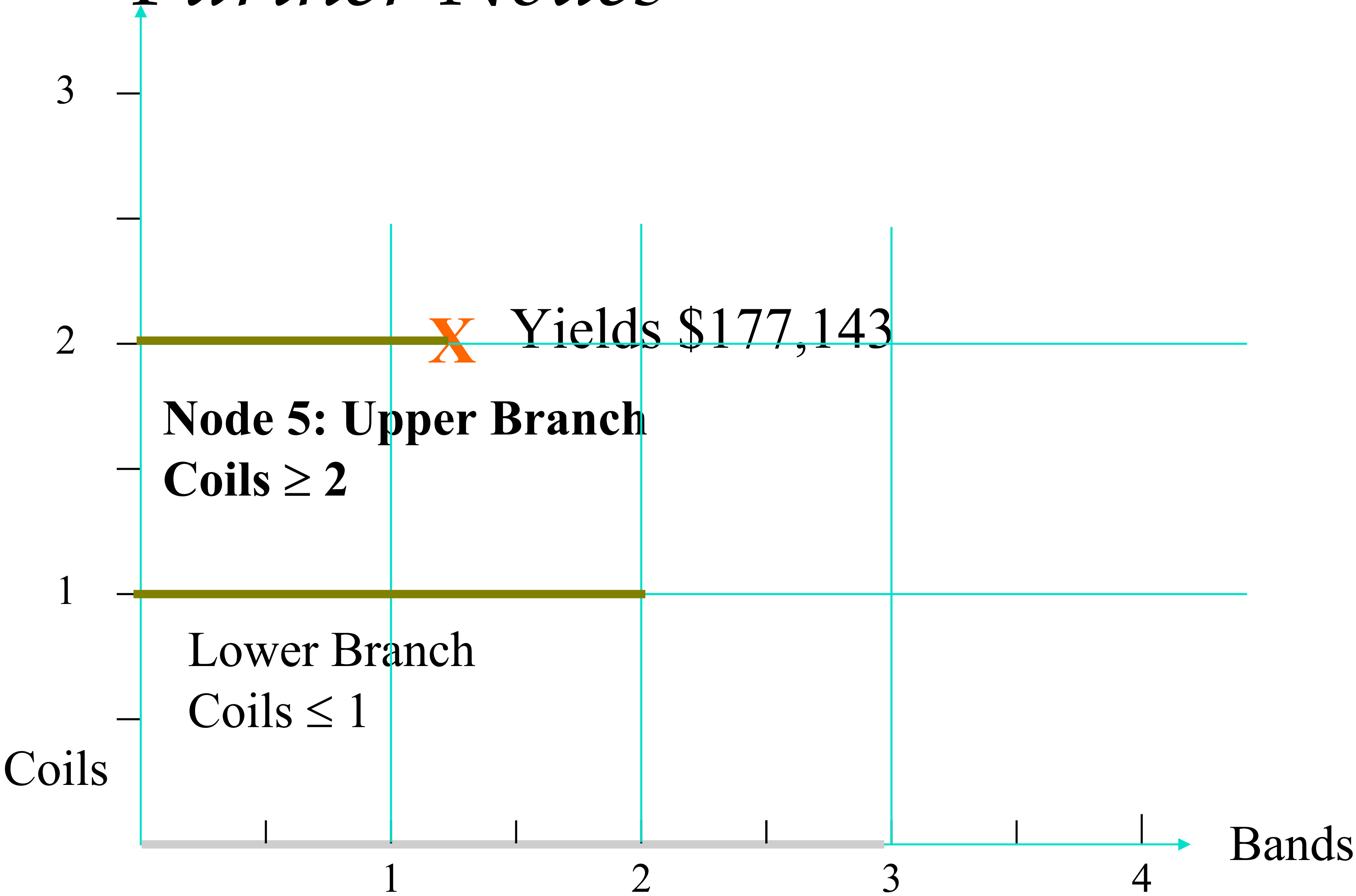
Further Nodes



Further Nodes

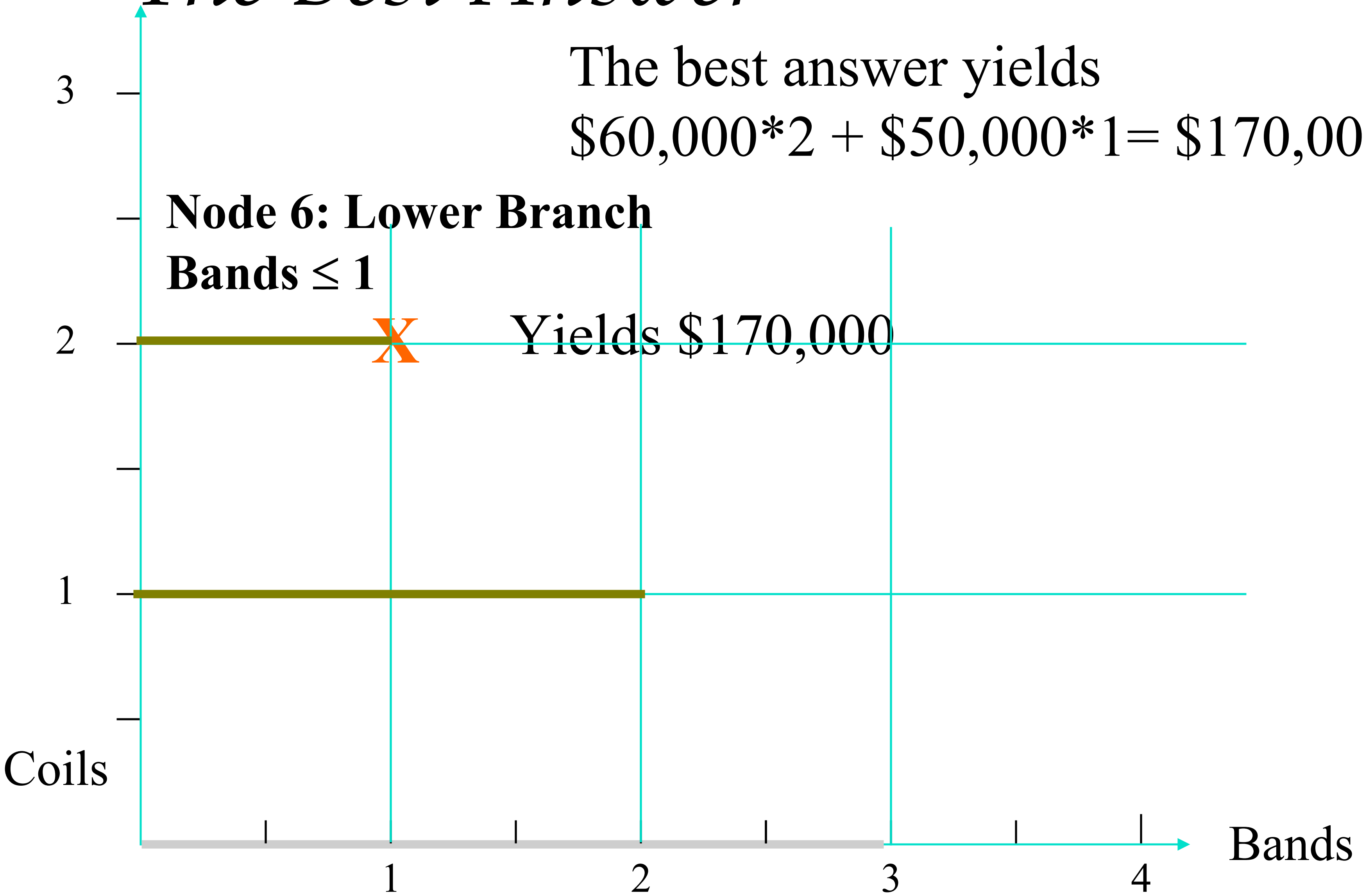


Further Nodes

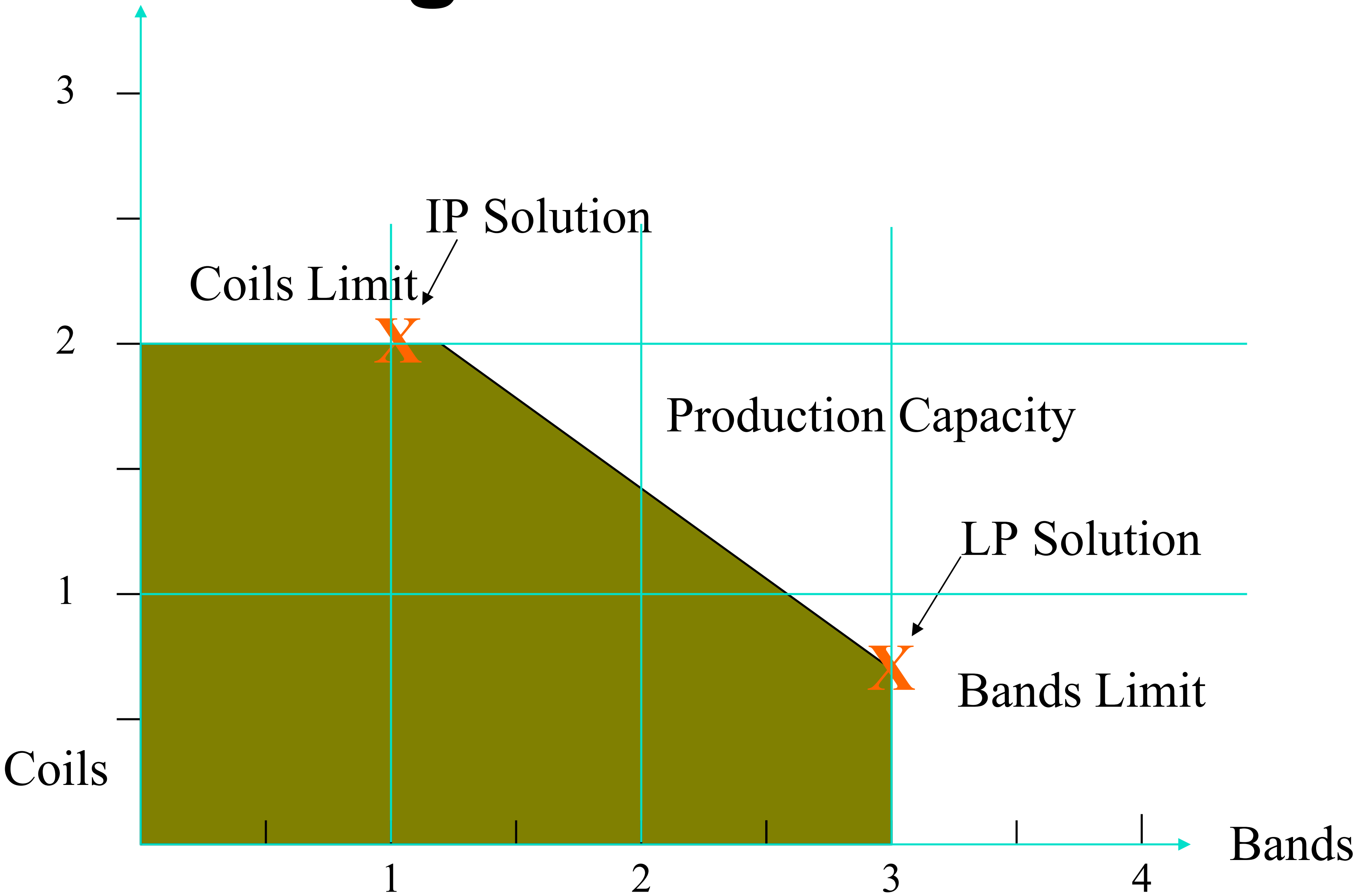


The Best Answer

The best answer yields
\$60,000*2 + \$50,000*1= \$170,000



Rounding Fails



Branch & Bound

- Implicit enumeration
- An Integer Program with 30 binary variables can require over **1 billion nodes**!
- Comes with guarantees
  - The answer is no worse than…
  - And no better than…
- Typically finds a good answer quickly
- Spends a long time guaranteeing it

A Solver Model

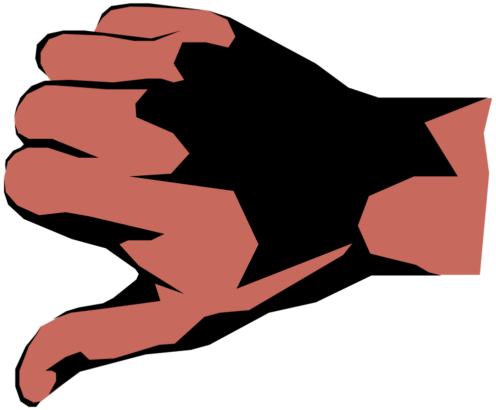
Steel Cable Company

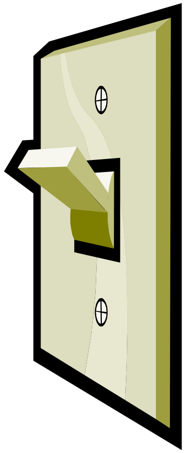
Integer Programming Example

Products	Bands	Coils	
Hours/Ton	**3**	**4**	
Profit/Ton	**$ 50,000**	**$ 60,000**	
Production Plan			
Products	Bands	Coils	
Tons Produced	**0**	**0**	
Maximum Sales	**3**	**2**	
Production Constraint	Hours Used		Hours Available
	0 <=		**12**
Profit Summary	Bands	Coils	Total
	$ -	$ -	$ -

Integer Variables

- Integer: -2, -1, 0, 1, 2, …
- Binary: 0 or 1
- Binary: Yes or No, On or Off

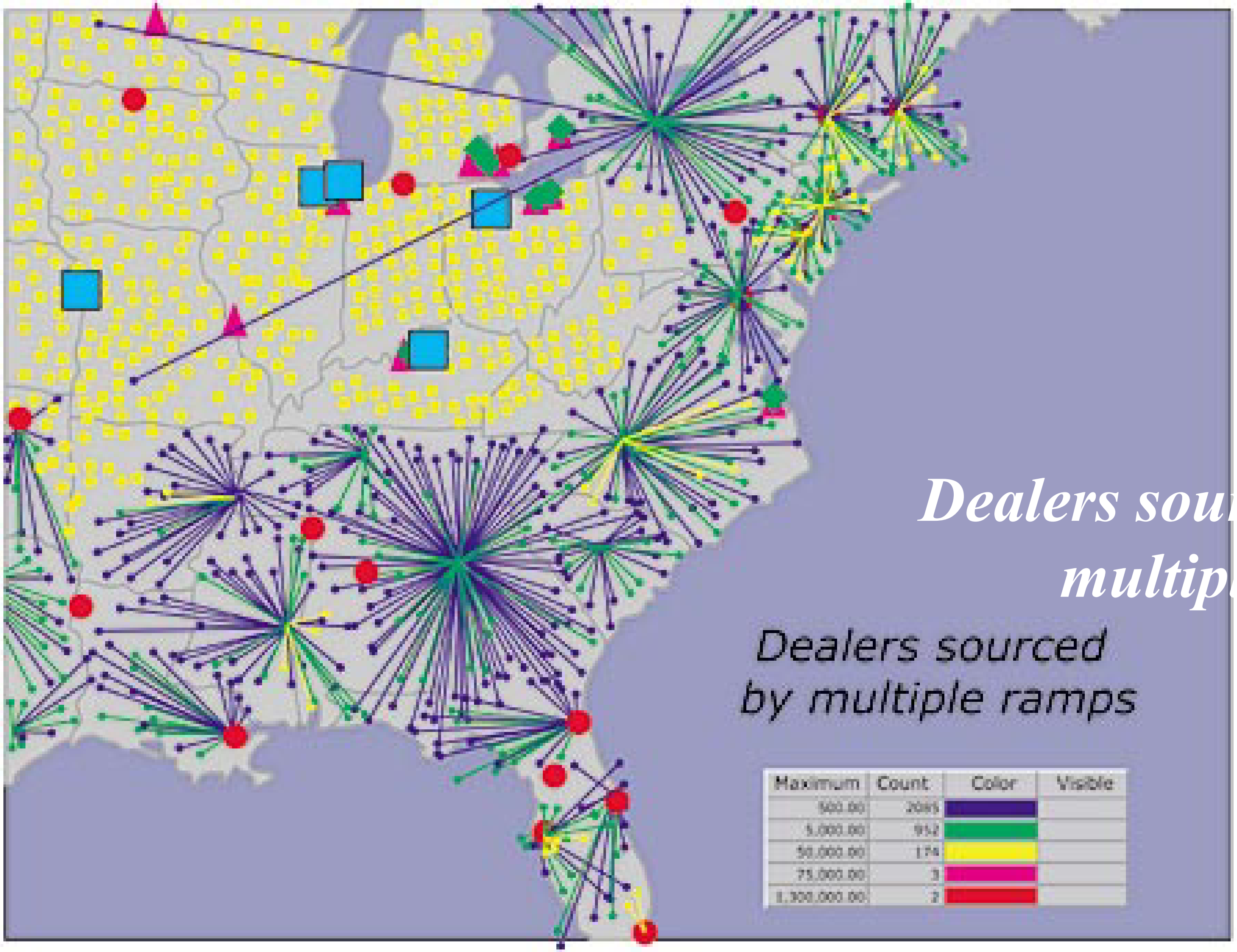




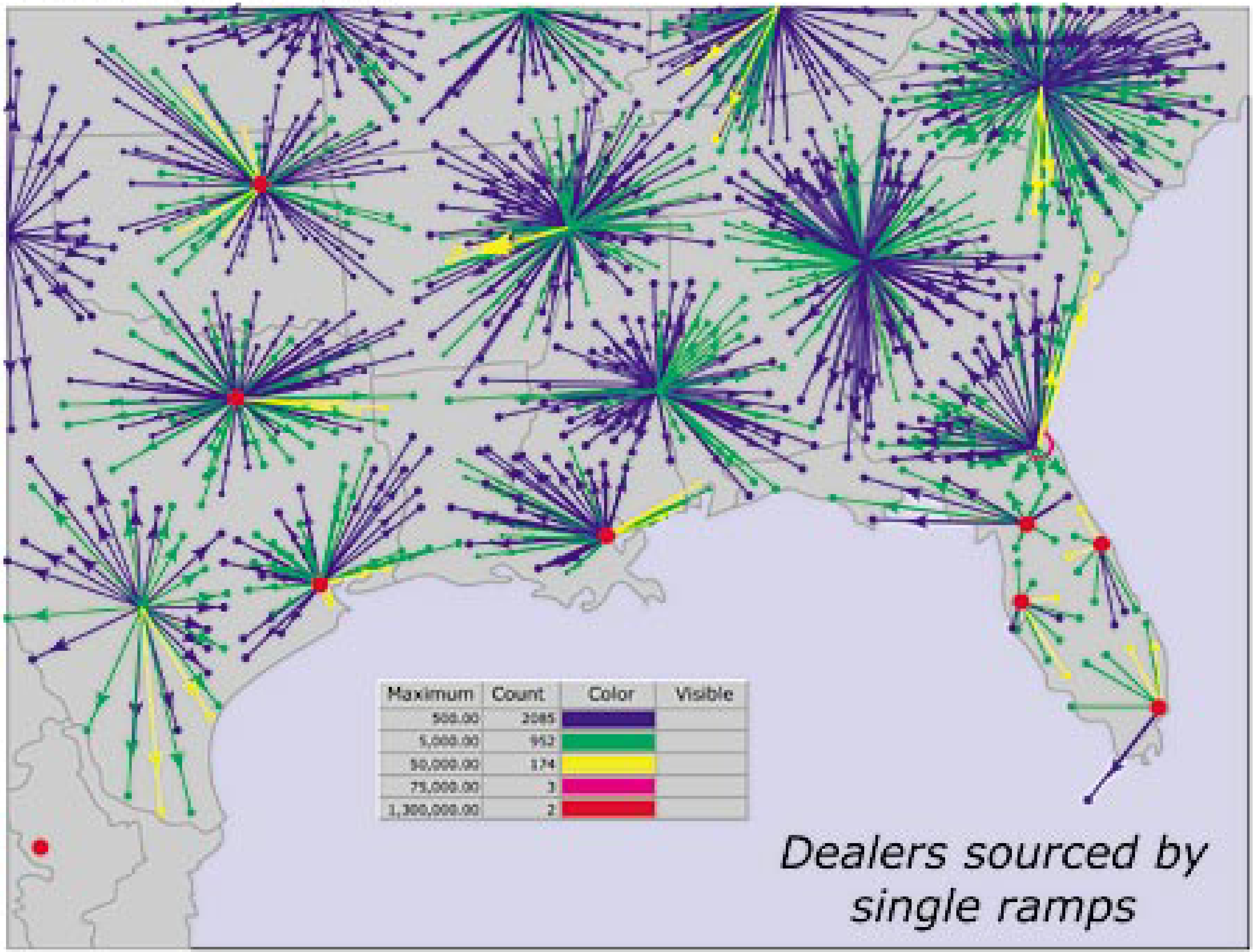
Single Sourcing

- Assign each customer to only one DC
- Simplifies service network

Old Ramp Allocation Southern US



New Ramp Allocation Southern US



Lower Cost?

- Does Single-Sourcing Reduce/Increase distribution cost?

Modeling Single-Sourcing

- Vanilla Approach
  - set DCS;
  - set CUSTOMERS;
  - param Capacity{DCS};
  - param Demand{CUSTOMERS};
  - param Cost{DCS, CUSTOMERS};
  - var Assign{DCS, CUSTOMERS} binary;

```
minimize TotalCost:
    sum{dc in DCS, cust in CUSTOMERS}
        Cost[dc, cust]*Assign[dc, cust] + Other Costs
```

DC gets all the Demand

- s.t. SingleSource{cust in CUSTOMERS}:
  - sum{dc in DCS} Assign[dc, cust] = 1;
- s.t. ObserveCapacity{dc in DCS}:
  - sum{cust in CUSTOMERS} Demand[cust]*Assign[dc,cust] <= Capacity[dc];

Or Part of a Larger Model

- s.t. MeetDemand{dc in DCS}:
  - sum of flows into the dc =
  - sum{cust in CUSTOMERS}
  - Demand[cust]*Assign[dc,cust];

What's the Problem?

- Size!
  - Thousands of customers
  - Scores of DCs
  - Hundreds of thousands of Integer Variables
  - Most useless!
  - Assign a Customer in ME to a DC in CA?

Practical Solution

- Don’t naively include all possible assignments
- Only include those:
  - Within a specified distance
  - Among the N closest
  - ....

Fixed Costs Revisited

Steco's Warehouse Location Model

Unit Costs Warehouse	Lease ($)	Unit Cost/Truck to Sales District 1	2	3	4
A	$ 7,750	$170	$ 40	$ 70	$160
B	$ 4,000	$150	$195	$100	$ 10
C	$ 5,500	$100	$240	$140	$ 60

Monthly Trucks From/To

Decisions	Yes/No	1	2	3	4	Total	Eff. Cap.	Cap.
Lease A	0	0	0	0	0	0	0	200
Lease B	0	0	0	0	0	0	0	250
Lease C	0	0	0	0	0	0	0	300
Total TrucksTo		0	0	0	0			
Demand (Trucks/Mo)		100	90	110	60			

	Lease Cost	To 1	To 2	To 3	To 4	Truck $	Total Cost
A	$ -	$ -	$ -	$ -	$ -	$ -	$ -
B	$ -	$ -	$ -	$ -	$ (0)	$ (0)	$ (0)
C	$ -	$ -	$ -	$ 0	$ -	$ 0	$ 0
Totals	$ -	$ -	$ -	$ 0	$ (0)	$ 0	$ 0

Binary Switches

- Variables
  - Yes/No: Lease warehouse or not (Binary)
  - Shipments from Warehouses to Sales Districts
- Constraints:
  - Meet Demand
  - Don't exceed Leased Capacity
    - Yes/No leased Warehouse *Capacity of Warehouse
    - At Warehouse B: Effective Capacity = 0*250
    - At Warehouse A: Effective Capacity = 1*200
  - Yes/No must be binary

Objective

- Variable Cost for Trucks
  - $100/truck from Warehouse C to District 1
- "Fixed" Costs for Leasing
  - $5,500 if we lease Warehouse C
  - $5,500*Yes/No Lease Warehouse C
- Careful about combining operational costs and capital costs

An AMPL Model

```
set WAREHOUSES;
param Capacity{WAREHOUSES};
param Lease{WAREHOUSES};
set DISTRICTS;
param Demand{DISTRICTS};
param TruckCost{WAREHOUSES, DISTRICTS};
```

```
var Open{WAREHOUSES} binary;
var Ship{WAREHOUSES, DISTRICTS} >= 0;

minimize TotalCost:
  sum{w in WAREHOUSES} Lease[w]*Open[w] +
  sum{w in WAREHOUSES, d in DISTRICTS} TruckCost[w,d]*Ship[w,d];

s.t. MeetDemand{d in DISTRICTS}:
    sum{w in WAREHOUSES} Ship[w,d] >= Demand[d];

s.t. ObserveEffectiveCapacity{w in WAREHOUSES}:
    sum{d in DISTRICTS} Ship[w,d] <= Capacity[w]*Open[w];
```

Challenge: Challenge 2 Revisited

- At least 5 funds
- At most 10 funds
- At least 10% if any

Fund Ratings

Fund Name		T-Bill	Large Value	Large Growth	Small Value	Small Growth	Japan	Pacific	Europe	Emerging Markets	Government	High Yield	International Bonds
Fidelity Adv Equity	0.00	7	71	2	6	7	2	0	0	5	0	0	0
Fidelity Advisor Gro	0.00	0	48	5	26	7	0	0	11	2	0	0	0
Fidelity Equity-Income	0.00	0	60	5	20	0	3	0	0	3	0	9	0
Fidelity Equity Income-II	0.00	0	66	4	16	0	2	1	0	6	0	5	0
Fidelity Growth/Income	0.00	2	47	0	17	11	3	0	5	2	0	12	0
Fidelity Ins Cash Po	0.43	100	0	0	0	0	0	0	0	0	0	0	0
Fidelity Investment	0.16	0	0	0	2	0	0	0	0	4	92	1	0
Fidelity Intermediat	0.00	13	0	0	0	0	0	0	0	0	83	0	3
Fidelity Limited Ter	0.00	5	18	0	0	0	0	4	0	0	45	28	0
Fidelity Mortgage Se	0.00	53	0	0	0	0	2	1	3	0	34	7	0
Fidelity Retirement	0.25	0	8	35	24	16	1	0	3	11	0	0	0
Fidelity Short-Term	0.00	44	0	0	0	0	0	0	0	6	25	23	3
Fidelity Value Fund	0.00	0	50	5	31	1	4	0	8	2	0	0	0
Fidelity Worldwide F	0.16	0	27	0	14	0	11	0	37	11	0	0	0
Totals	1.00	43.00	6.24	8.75	8.52	4.00	1.98	-	6.56	5.13	15.00	0.16	-
Targets		43	3	3	5	4	10	2	5	10	15	0	0
Excess		-	3.24	5.75	3.52	-	-	-	1.56	-	-	0.16	0.00
ShortFall		-	-	-	-	-	8.02	2.00	-	4.87	-	-	-
Adjusted		43.00	3.00	3.00	5.00	4.00	10.00	2.00	5.00	10.00	15.00	(0.00)	(0.00)
Deviation	29.29	-	3.24	5.75	3.52	-	8.02	2.00	1.56	4.87	-	0.16	0.00

Summary

- var Invest{FUNDS} binary;
- var Fraction{FUNDS} >= 0;
- s.t. MaxHoldings;
  - sum{f in FUNDS} Invest[f] <= 10;
- s.t. MinHoldings;
  - sum{f in FUNDS} Invest[f] >= 5;
- s.t. MinimumPercentage{f in FUNDS}:
  - Fraction[f] >= 0.10*Invest[f];
- ...

Don't Forget

- S.t. DefineInvest{f in FUNDS}:
  - Invest[f] >= Fraction[f];
- Can't put money in fund unless you admit to being invested in it.

Set Covering Models

WesternAir

Western Airlines Hub Selection

Mile **1000**

Distance Matrix

	AT	BO	CH	DE	HO	LA	NO	NY	PI	SL	SF	SE
AT	**-**	1,037	**674**	1,398	**789**	2,182	**479**	**841**	**687**	1,878	2,496	2,618
BO	1,037	**-**	1,005	1,949	1,804	2,979	1,507	**222**	**574**	2,343	3,095	2,976
CH	**674**	1,005	**-**	1,008	1,067	2,054	**912**	**802**	**452**	1,390	2,142	2,013
DE	1,398	1,949	1,008	**-**	1,019	1,059	1,273	1,771	1,411	**504**	1,235	1,307
HO	**789**	1,804	1,067	1,019	**-**	1,538	**356**	1,608	1,313	1,438	1,912	2,274
LA	2,182	2,979	2,054	1,059	1,538	**-**	1,883	2,786	2,426	**715**	**379**	1,113
NO	**479**	1,507	**912**	1,273	**356**	1,883	**-**	1,311	1,070	1,738	2,249	2,574
NY	**841**	**222**	**802**	1,771	1,608	2,786	1,311	**-**	**368**	2,182	2,934	2,815
PI	**687**	**574**	**452**	1,411	1,313	2,426	1,070	**368**	**-**	1,826	2,578	2,465
SL	1,878	2,343	1,390	**504**	1,438	**715**	1,738	2,182	1,826	**-**	**752**	**836**
SF	2,496	3,095	2,142	1,235	1,912	**379**	2,249	2,934	2,578	**752**	**-**	**808**
SE	2,618	2,976	2,013	1,307	2,274	1,113	2,574	2,815	2,465	**836**	**808**	**-**

Summary

- Mixed Integer Programming Models
  - Mostly about Binary Variables (Logic)
  - Significantly Harder to solve
- Significantly More Modeling Power
  - Fixed Costs
  - If-Then Constraints
  - Cardinality Constraints
  - Set Covering Models
  - ….