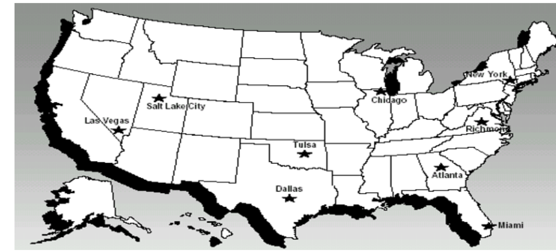
Management Science

Non-Linear & Goal Programming

Dan Brandon, Ph.D., PMP

Traveling Salesman Problem (TSP)

- Problems involving **ordering or permutations** of choices are very difficult to model using conventional constraints, even with integer variables
- An example is the famous **Traveling Salesman Problem (TSP)**, where a salesman must choose the order of cities to visit so as to minimize travel time (or distance), and each city must be visited exactly once

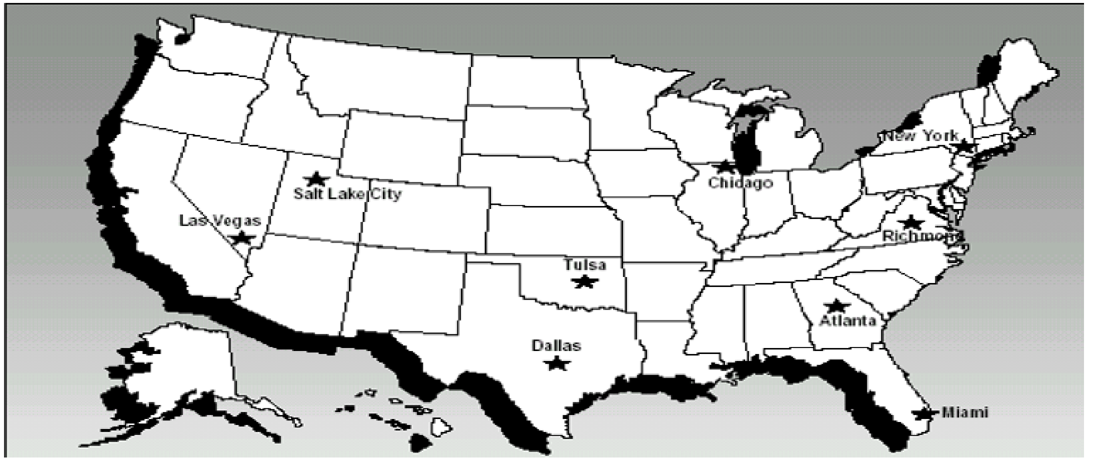


- Starting in New York, visit each city only once and return to New York:
  - What is the order of your visits ?
  - What is your total mileage ?

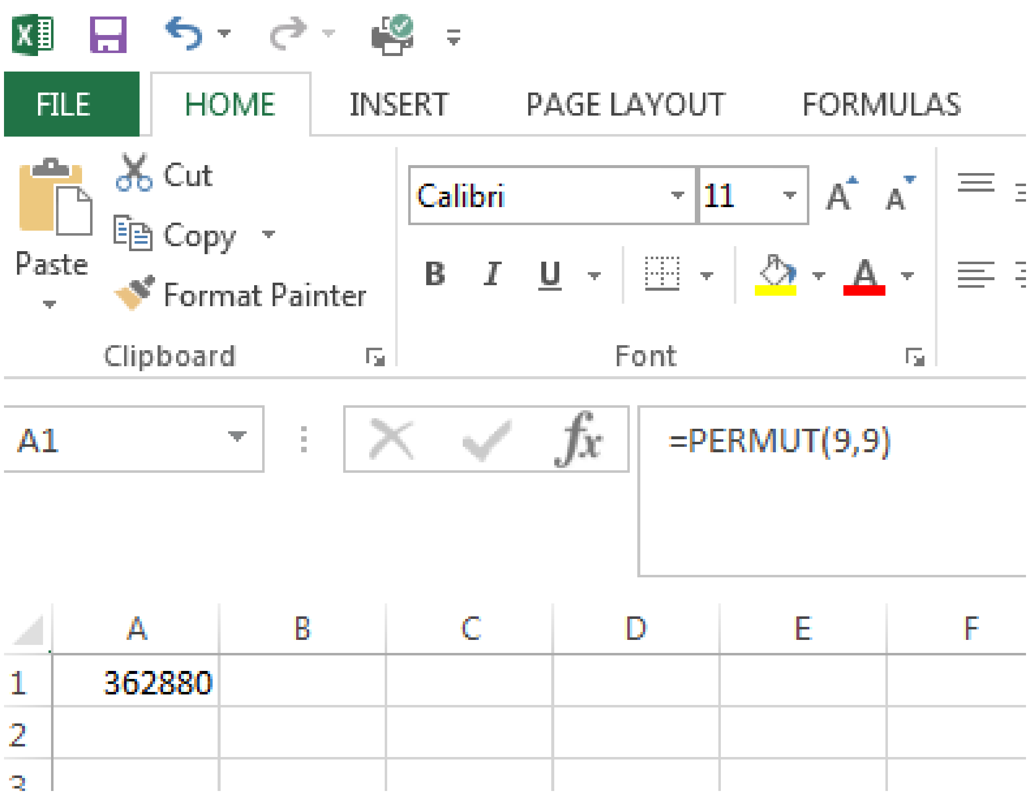
	A	B
1	City	Mileage to Next
2	NYC	--------
3	...	
4	...	
5	...	
6	...	
7	...	
8	...	
9	...	
10	...	
11	Back to NYC	
12	Total	

	NYC	RMD	ATL	MIA	DAL	TUL	LVG	CHI	SLC
New York City	-	339	841	1308	1552	1344	2548	802	2182
Richmond	339	-	510	1165	1266	1211	2430	748	2110
Atlanta	841	510	-	655	795	772	1964	674	1878
Miami	1308	1165	655	-	1329	1398	2526	1300	2532
Dallas	1552	1266	795	1329	-	257	1221	917	1242
Tulsa	1344	1211	772	1398	257	-	1228	683	1172
Las Vegas	2548	2430	1964	2526	1221	1228	-	1772	433
Chicago	802	748	674	1300	917	683	1772	-	1390
Salt Lake City	2182	2110	1878	2532	1242	1172	433	1390	-

- How many possible routes are there ?

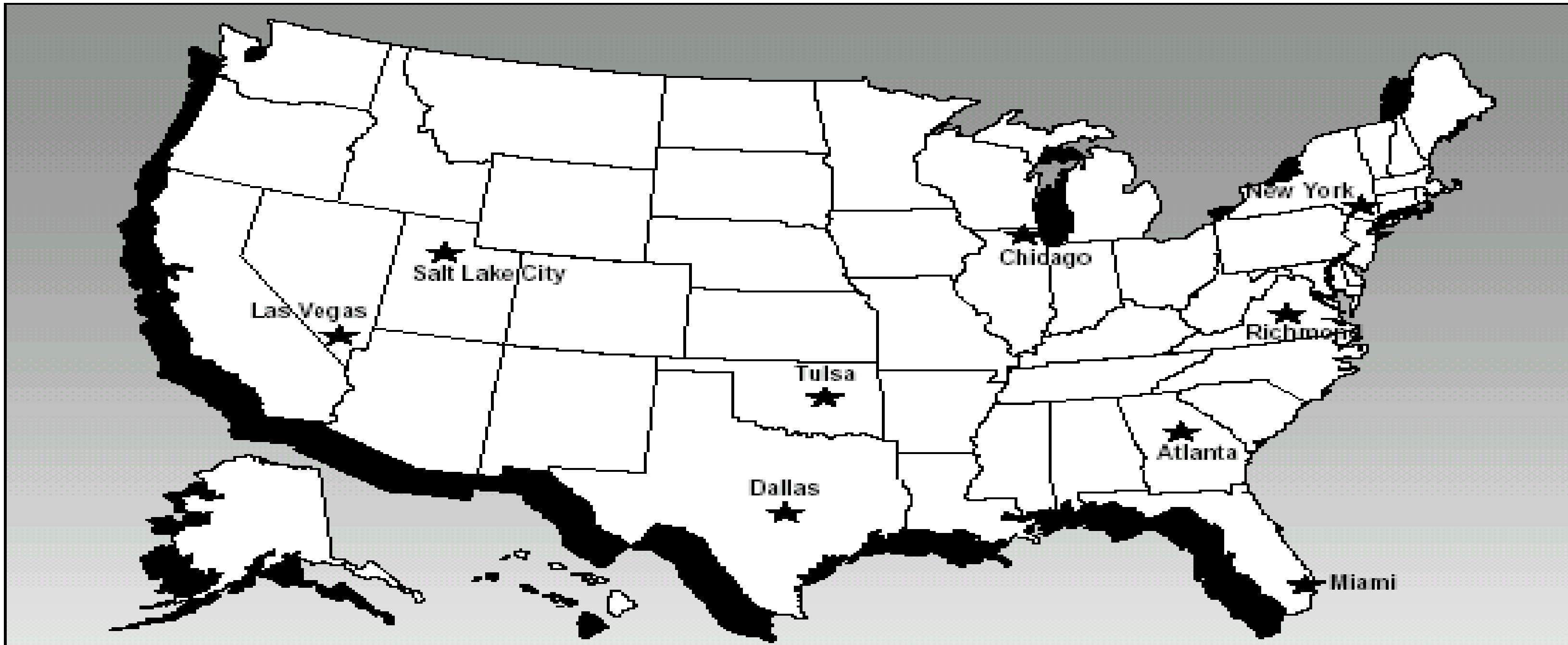


Permutations (order counts) of 9 Cities



- Do not look ahead !

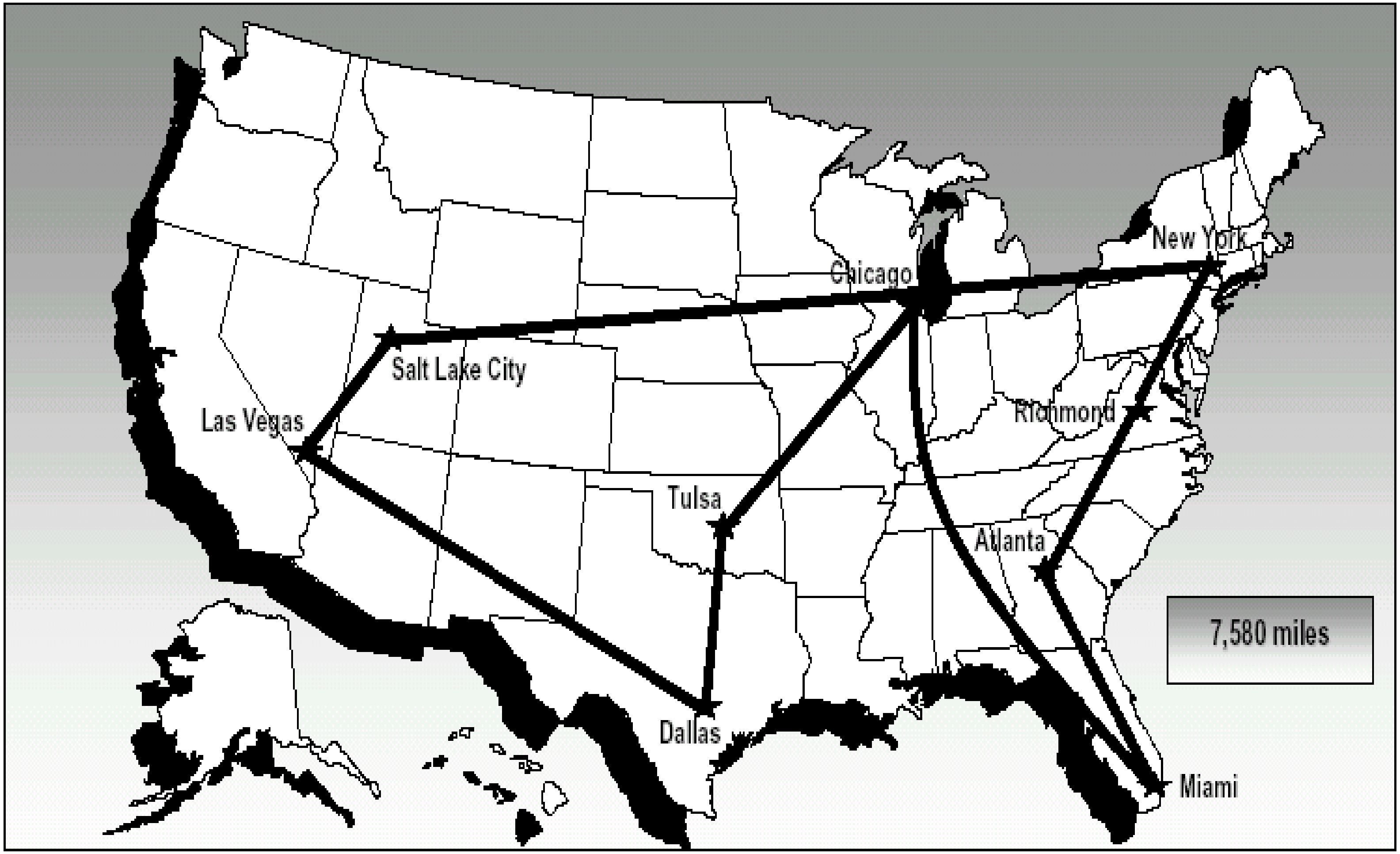
Traveling Salesman Problem (con't)



	NYC	RMD	ATL	MIA	DAL	TUL	LVG	CHI	SLC
New York City	-	339	841	1308	1552	1344	2548	802	2182
Richmond	339	-	510	1165	1266	1211	2439	748	2110
Atlanta	841	510	-	655	795	772	1964	674	1878
Miami	1308	1165	655	-	1329	1398	2526	1300	2532
Dallas	1552	1266	795	1329	-	257	1221	917	1242
Tulsa	1344	1211	772	1398	257	-	1228	683	1172
Las Vegas	2548	2439	1964	2526	1221	1228	-	1772	433
Chicago	802	748	674	1300	917	683	1772	-	1390
Salt Lake City	2182	2110	1878	2532	1242	1172	433	1390	-

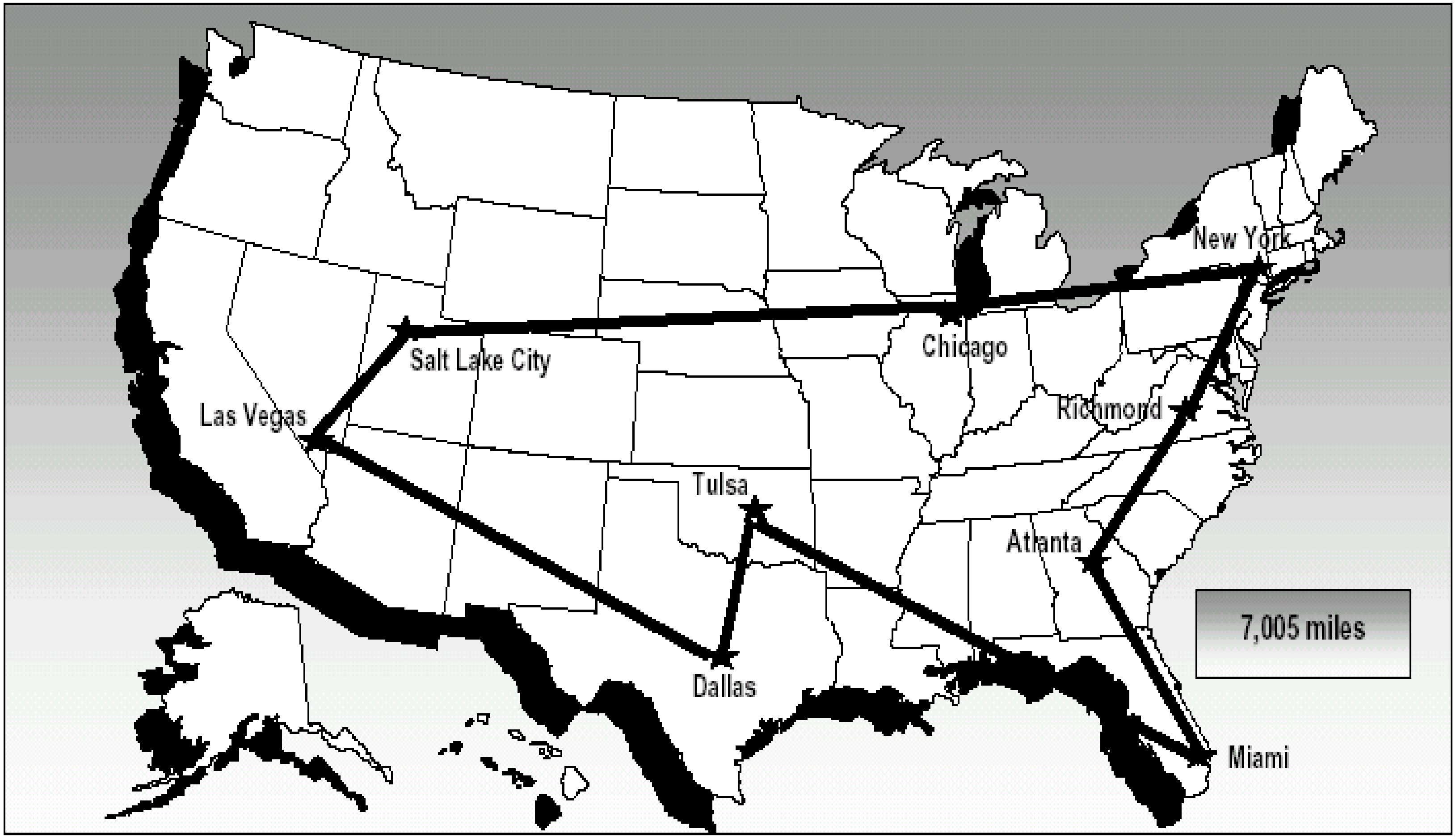
Initial Heuristic Solution

Rule: start at home (NYC) , go to closest city



Modified Heuristic Solution

Rule: no crossing any connection, no backtracking



TSP set up as LP

Let x_{ij} equal 1 if the path goes from city *i* to city *j*, and 0 otherwise, for cities 0, ..., *n*. Let u_i for *i* = 1, ..., *n* be the sequence that the nodes are being visited, and c_{ij} let be the distance from city *i* to city *j*

$$\min \sum_{i=0}^{n} \sum_{j \neq i, j=0}^{n} c_{ij} x_{ij}$$

$$0 \leq x_{ij} \leq 1 \qquad i, j = 0, \ldots n$$

$$x_{ij} \text{ integer} \qquad i, j = 0, \ldots n$$

$$\sum_{i=0, i \neq j}^{n} x_{ij} = 1 \qquad j = 0, \ldots, n$$

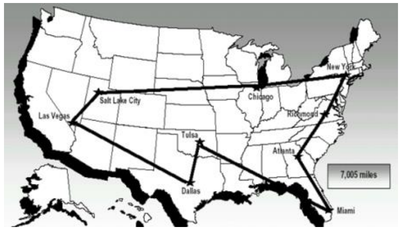
$$\sum_{j=0, j \neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n$$

$$u_i - u_j + n x_{ij} \leq n - 1 \quad 1 \leq i \neq j \leq n.$$

See next slide →

There is a new “all different” constraint in Excel to facilitate this last constraint.

Sum Of Every Column And Row Is 1



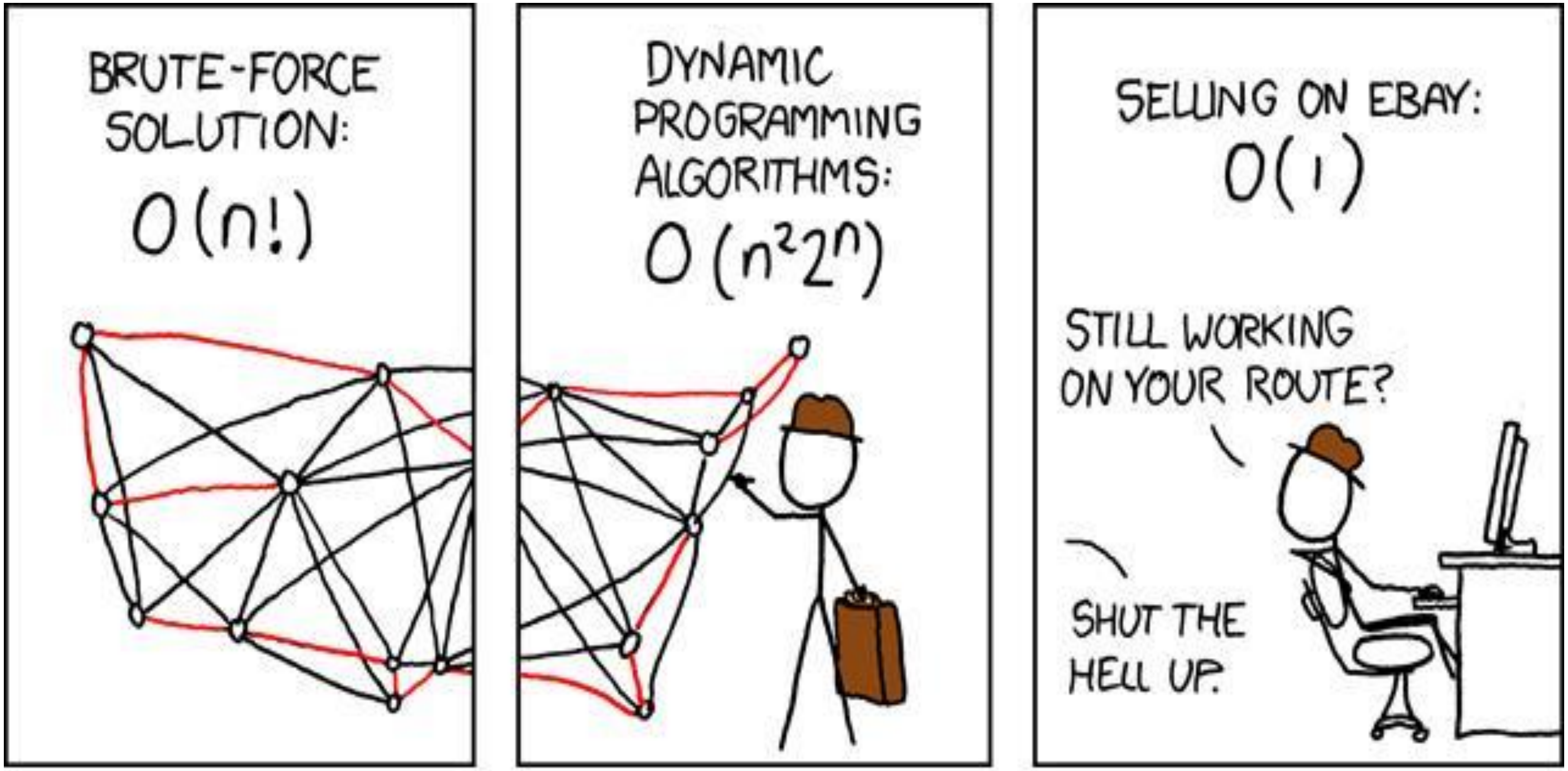
To

	NYC	Richmond	Atlanta	Miami	Dallas	Tulsa	Las Vegas	Chicago	SLC
NYC		1							
Rich.			1						
Atlanta				1					
Miami						1			
Dallas							1		
Tulsa					1				
Las Vegas									1
Chicago	1								
SLC								1	

Excel iteger, binary, all different Constraints

- A constraint such as **A1:A5 = integer**, where A1:A5 are decision variable cells, requires that the solution values for A1 through A5 must be integers or whole numbers, such as -1, 0 or 2, to within a small tolerance (determined by the **Constraint Precision** option).
- A constraint such as **A1 = binary** is equivalent to specifying A1 = integer, A1 >= 0 and A1 <= 1. This implies that A1 must be *either 0 or 1* at the solution; hence A1 can be used to represent a "yes/no" decision, such as whether or not to build a new manufacturing plant.
- A constraint such as **A1:A5 = alldifferent**, where A1:A5 are decision variable cells, requires that these cells must be integers in the range 1 to N (N = 5 in this example), with each variable *different* from all the others at the solution. Hence, A1:A5 will contain a *permutation* of integers, such as 1,2,3,4,5 or 1,3,5,2,4.

TSP vs Ebay



BRUTE-FORCE SOLUTION:
O(n!)
DYNAMIC PROGRAMMING ALGORITHMS:
O(n²2ⁿ)
SELLING ON EBAY:
O(1)
STILL WORKING ON YOUR ROUTE?
SHUT THE HELL UP.

Further LP Extensions

- Revised Simplex Method
- Sparse Matrices
  - The constraint matrix A (in $AX \leq D$) is mostly zeros
  - Computational techniques for only storing and operating on the non-zero elements
- Interior Point Methods
- Quadratic Programming
  - The objective function can have quadratic functions of the variables, but the constraints are still linear
- Non-linear Problems
  - The constraints and/or object function is non-linear
- Goal Programming

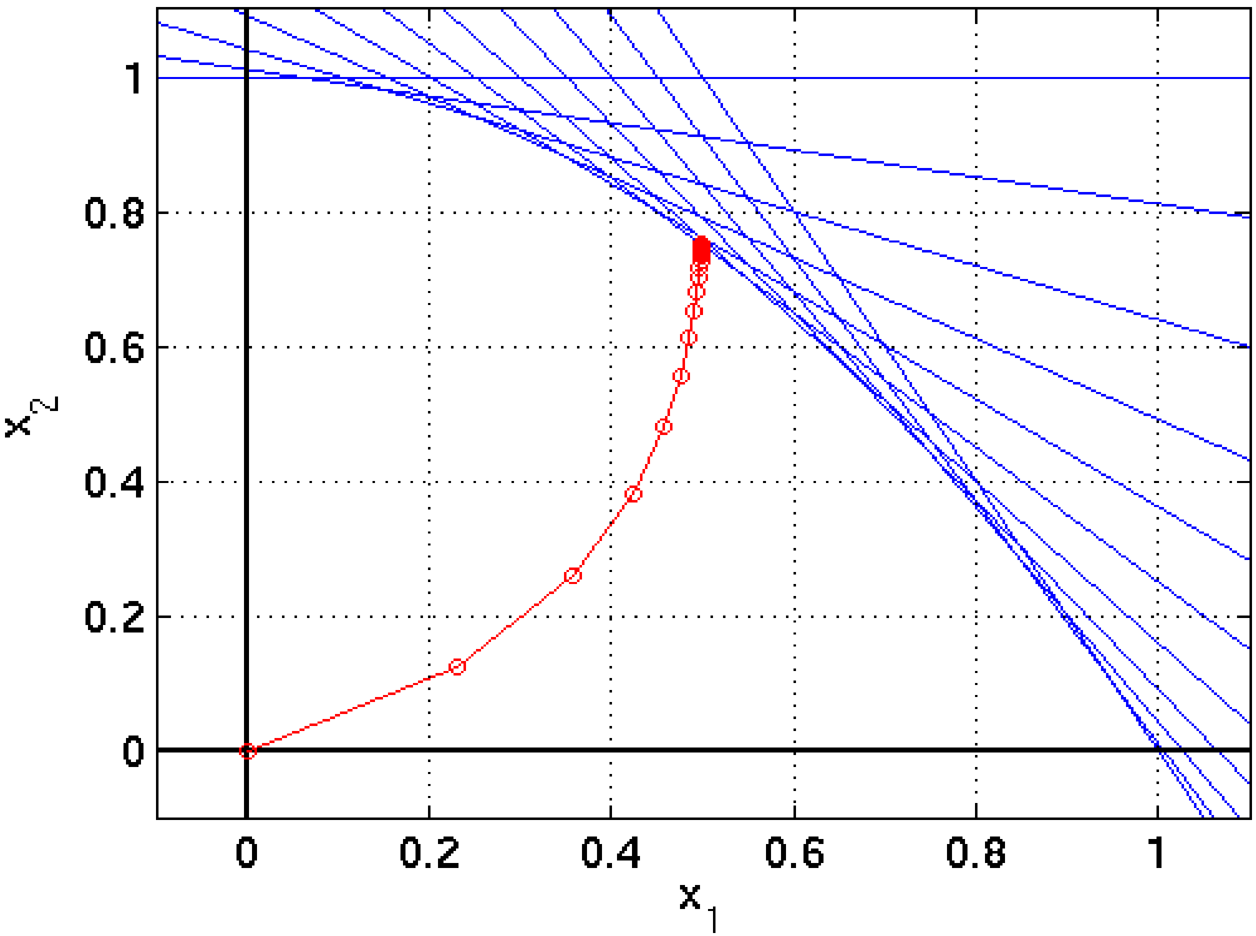
Revised Simplex Method

- The standard version of the simplex method updates the entire simplex tableau at each iteration
- However, not all of the numbers in the tableau are actually needed in each iteration
- What is needed is enough of the tableau to choose a variable to enter the current basis and a variable to leave the basis
- Choosing the entering basic variable requires (in terms of the tableau) only the numbers in the row of objective function coefficients
- Choosing the leaving basic variable then requires (again, in terms of the tableau) the numbers in the column corresponding to that entering basic variable and the numbers in the right-hand side of the tableau
- That's only one row and two columns out of the entire $m \times n$ tableau
- The revised simplex method uses the right matrix implementation and rules for avoiding inverting the basis matrix at every iteration to calculate only the numbers in this row and two columns
- By doing so it can speed up the performance of the simplex method considerably

Interior Point Methods

- **Interior point methods** are a relatively new class of algorithms to solve linear and nonlinear convex optimization problems
- These algorithms have been inspired by Karmarkar's algorithm, developed by Narendra Karmarkar in 1984 for linear programming
- Contrary to the simplex method, it reaches an optimal solution by traversing the interior of the feasible region instead of examining the vertices
- Karmarkar's breakthrough revitalized earlier studies of interior point methods and barrier problems, showing that it was possible to create an algorithm for linear programming characterized by polynomial complexity and, moreover, that was competitive with the simplex method
- The class of primal-dual path-following interior point methods is now considered the most successful; Mehrotra's predictor-corrector algorithm provides the basis for most implementations of this modern class of methods

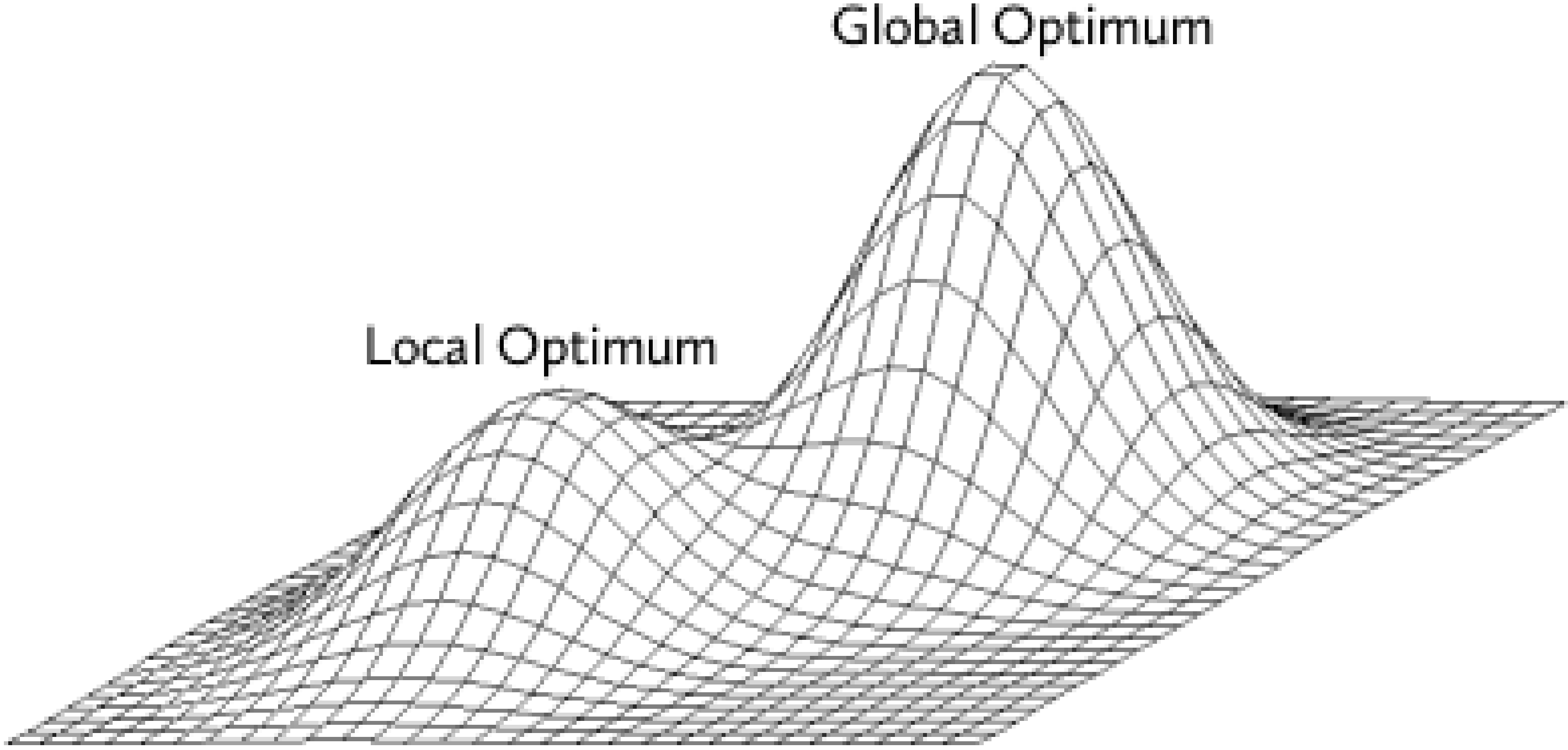
Interior Point Methods (con't)



Nonlinear Programming

- The methods seen so far have assumed that the objective function and constraints are linear
- Terms such as X_1^3, $1/X_2$, $\log X_3$, or $5X_1X_2$ are not allowed
- But there are many nonlinear relationships in the real world that would require the objective function, constraint equations, or both to be nonlinear
- Excel can be also used to solve these ***nonlinear programming*** (*NLP*) problems
- One disadvantage of NLP is that the solution yielded may only be a local optimum, rather than a global optimum
  - In other words, it may be an optimum over a particular range, but not overall

Local and Global Optimums



Quadratic Programming Example

- The Great Western Appliance Company sells two models of toaster ovens, the Microtoaster (X1) and the Self-Clean Toaster Oven (X2)
- They earn a profit of $28 for each Microtoaster no matter the number of units sold
- For the Self-Clean oven, profits increase as more units are sold
- The profit function for the Self-Clean oven may be expressed as:

$$21X_2 + 0.25X_2^2$$

Quadratic Programming Example (con't)

The objective function is nonlinear and there are two linear constraints on production capacity and sales time available.

$$\text{Maximize profit} = 28X_1 + 21X_2 + 0.25X_2^2$$

$$\text{subject to} \quad X_1 + X_2 \leq 1{,}000 \quad \text{(units of production capacity)}$$

$$0.5X_1 + 0.4X_2 \leq 500 \quad \text{(hours of sales time available)}$$

$$X_1, X_2 \geq 0$$

When an objective function contains a squared term and the problem constraints are linear, it is called a *quadratic programming* problem

Quadratic Programming Example (con't)

Excel Solver Solution for Quadratic Problem

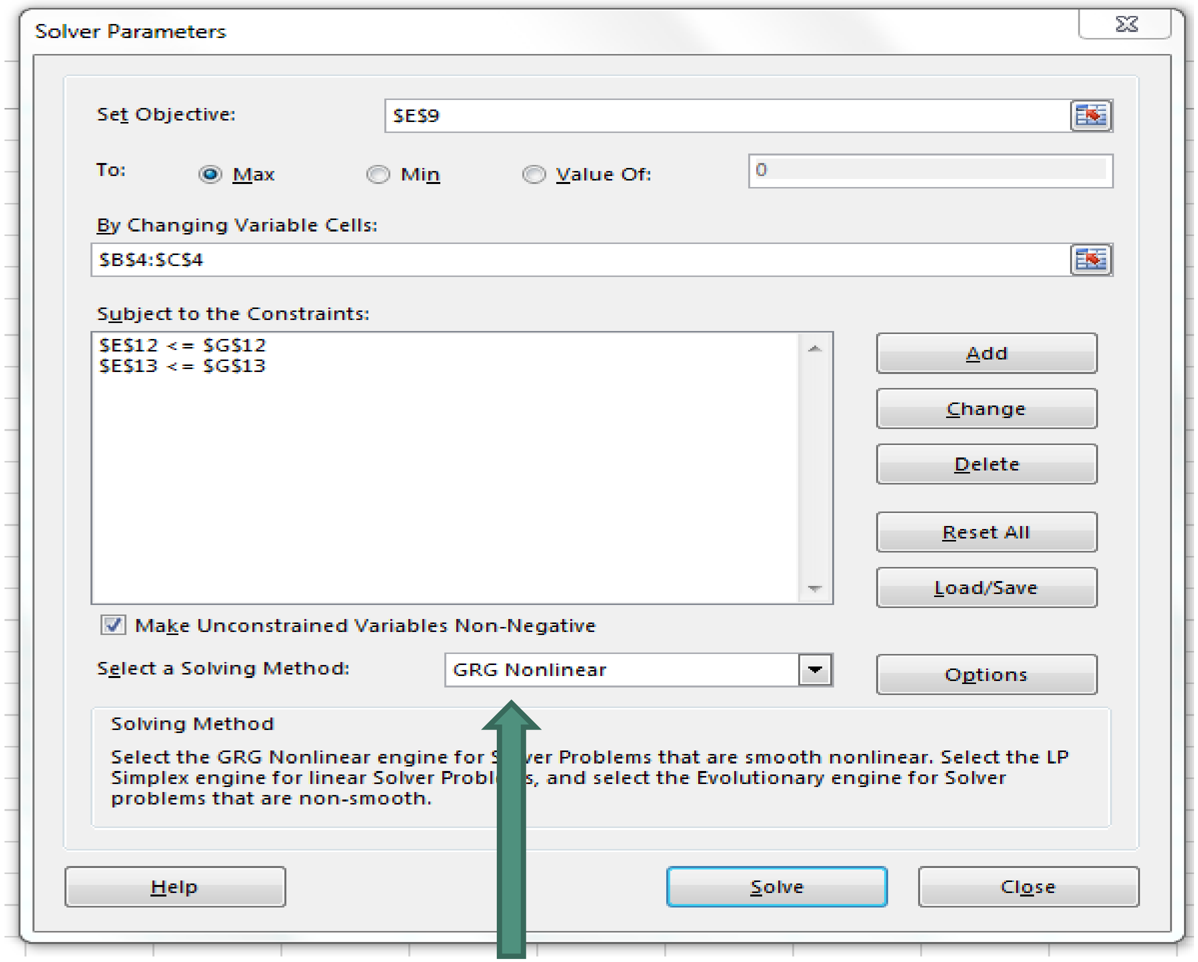
	A	B	C	D	E	F	G
1	**Great Western Appliance**						
2		**Micro**	**Self-Clean**				
3	**Variables**	X1	X2				
4	**Values**	0	0				
5							
6							
7	**Terms**	X1	X2	$X2^2$			
8	**Calc Values**	0	0	0	Profit		
9	**Profit**	28	21	0.25	0		
10							
11	**Constraints**				LHS	Sign	RHS
12	**Capacity**	1	1		0	<	1000
13	**Hours**	0.5	0.4		0	<	500
14							

Note that $X2^2$ is just another variable

Quadratic Programming Example (con't)

	A	B	C	D	E	F	G
1	Great Western Appliance						
2		Micro	Self-Clean				
3	Variables	X1	X2				
4	Values	0	0				
5							
6							
7	Terms	X1	X2	$X2^2$			
8	Calc Values	=B4	=C4	=C4^2	Profit		
9	Profit	28	21	0.25	=B8*B9+C8*C9+D8*D9		
10							
11	Constraints				LHS	Sign	RHS
12	Capacity	1	1		= B4*B12+C4*C12	<	1000
13	Hours	0.5	0.4		= B4*B13+C4*C13	<	500
14							

Quadratic Programming Example (con’t)



Great Western Appliance						
	Micro	**Self-Clean**				
Variables	X1	X2				
Values	0	1000				
Terms	X1	X2	$X2^2$			
Calc Values	0	1000	1000000	Profit		
Profit	28	21	0.25	271000		
Constraints				LHS	Sign	RHS
Capacity	1	1		1000	<	1000
Hours	0.5	0.4		400	<	500

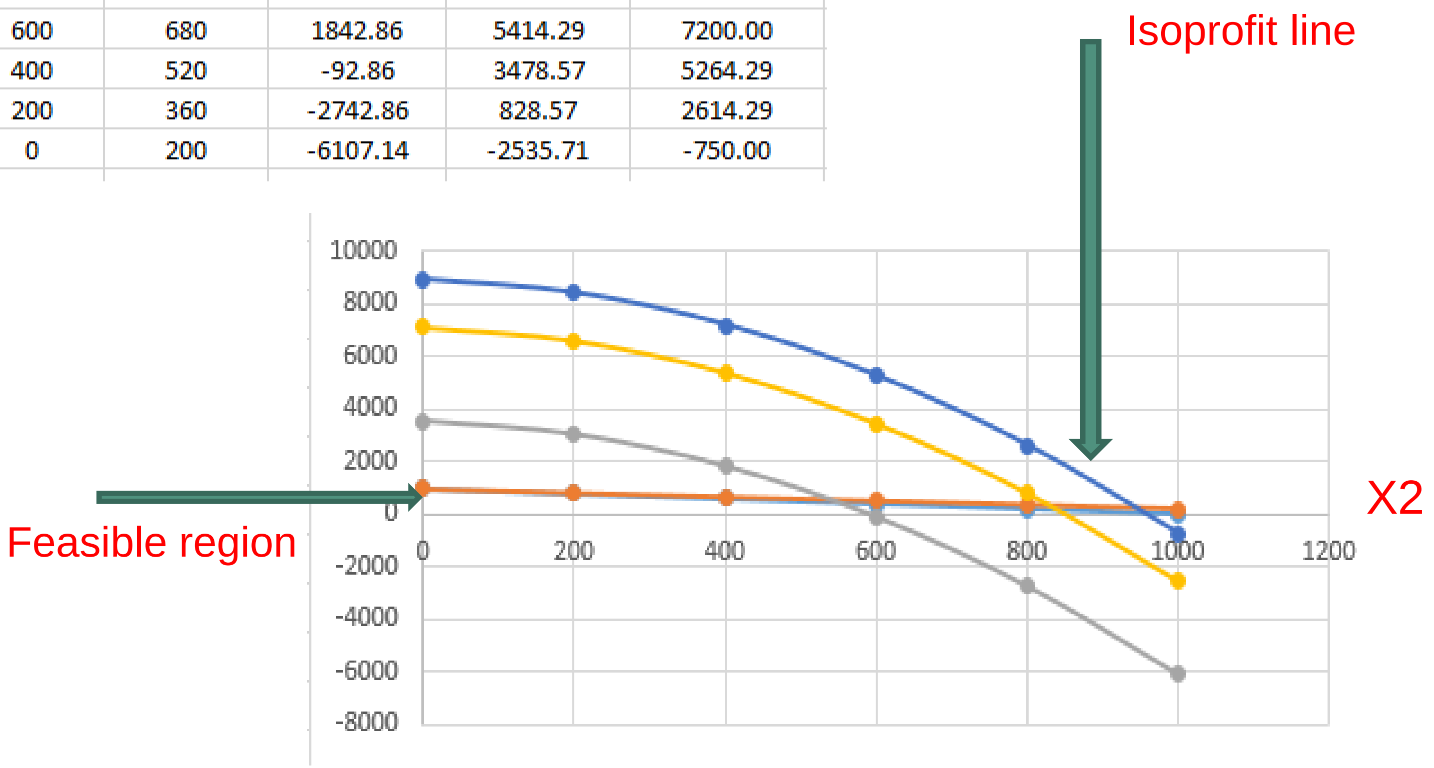
Build all self-cleaning ovens

Max profit

Hours constraint is non-binding

Max profit is 271000; at x1=0 & x2=1000

	Constraints				
	Prod	Sales	Isoprofit X1's		
X2	X1	X1	100000	200000	250000
0	1000	1000	3571.43	7142.86	8928.57
200	800	840	3064.29	6635.71	8421.43
400	600	680	1842.86	5414.29	7200.00
600	400	520	-92.86	3478.57	5264.29
800	200	360	-2742.86	828.57	2614.29
1000	0	200	-6107.14	-2535.71	-750.00



Nonlinear Objective Function and Nonlinear Constraints

- The annual profit at a medium-sized (200-400 beds) corporation hospital depends on the number of medical patients admitted (X_1) and the number of surgical patients admitted (X_2)
- The objective function for the hospital is nonlinear
- They have identified three constraints, two of which are nonlinear
  - Nursing capacity - nonlinear
  - X-ray capacity - nonlinear
  - Marketing budget required

Nonlinear Objective Function and Nonlinear Constraints (con't)

The objective function and constraint equations for this problem are:

Maximize profit = $\$13X_1 + \$6X_1X_2 + \$5X_2 + \$1/X_2$

subject to $2X_1^2 + 4X_2 \leq 90$ (nursing capacity in thousands of labor-days)

$X_1 + X_2^3 \leq 75$ (x-ray capacity in thousands)

$8X_1 - 2X_2 \leq 61$ (marketing budget required in thousands of \$)

Nonlinear Objective Function and Nonlinear Constraints (con't)

Excel Solution for Hospicare's NLP Problem

	A	B	C	D	E	F	G	H	I	J
1	Hospicare Corp									
2										
3	Variables	X1	X2							
4	Values	6.0663	4.1003							
5										
6	Terms	X1	$X1^2$	X1*X2	X2	$X2^3$	1/X2			
7	Calculated Values	6.0663	36.7995	24.8732	4.1003	68.9337	0.2439	Total Profit		
8	Profit	13	0	6	5		1	248.8457		
9										
10	Constraints							LHS	Sign	RHS
11	Nursing		2		4			90.00	≤	90
12	X-Ray	1				1		75.00	≤	75
13	Budget	8			-2			40.33	≤	61

	H
8	=SUMPRODUCT($B$7:$G$7,B8:G8)

	B	C	D	E	F	G
7	=B4	=B4^2	=B4*C4	=C4	=C4^3	=1/C4

Linear Objective Function and Nonlinear Constraints

- Thermlock Corp. produces massive rubber washers and gaskets like the type used to seal joints on the NASA Space Shuttles
- It combines two ingredients, rubber (X_1) and oil (X_2)
- The cost of the industrial quality rubber is $5 per pound and the cost of high viscosity oil is $7 per pound
- Two of the three constraints are nonlinear

Linear Objective Function and Nonlinear Constraints (con't)

The firm's objective function and constraints are:

Minimize costs = $\$5X_1 + \$7X_2$

subject to $3X_1 + 0.25X_1^2 + 4X_2 + 0.3X_2^2 \geq 125$ (hardness constraint)

$13X_1 + X_1^3 \geq 80$ (tensile strength)

$0.7X_1 + X_2 \geq 17$ (elasticity)

Linear Objective Function and Nonlinear Constraints (con't)

Excel Solution for Thermlock NLP Problem

	A	B	C	D	E	F	G	H	I
1	**Thermlock Gaskets**								
2									
3	**Variables**	**X1**	**X2**						
4	**Values**	**3.325**	**14.672**	**Total Cost**					
5	**Cost**	5	7	**119.333**					
6									
7		**X1**	**$X1^2$**	**$X1^3$**	**X2**	**$X2^2$**			
8	**Value**	3.325	11.058	36.771	14.672	215.276			
9	**Constraints**						**LHS**	**Sign**	**RHS**
10	**Hardness**	3	0.25		4	0.3	136.012	≥	125
11	**Tensile Strengt**	13		1			80	≥	80
12	**Elasticity**	0.7			1		17	≥	17

	D
5	**=SUMPRODUCT($B$4:$C$4,B5:C5)**

	G
10	=SUMPRODUCT($B$8:$F$8,B10:F10)

	B	C	D	E	F
8	=B4	=B4^2	=B4^3	=C4	=C4^2

Exercise

- Woodchuck's Carpentry Inc. produces tables and chairs
- The profit from selling tables (x_1) is $100\ X_1 - 2\ X_1^2$ and the profit from selling chairs (x_2) is $200\ X_2 - X_2^2$
- Woodchuck faces one constraint, as he only has 1000 units of mahogany wood available, and he needs 2 units for each table, and 3 units for each chair
- **How many tables and chairs (as integers) should be produce ? Formulate objective function and constraints first.**

- Do not look ahead !

Variables & Constraint

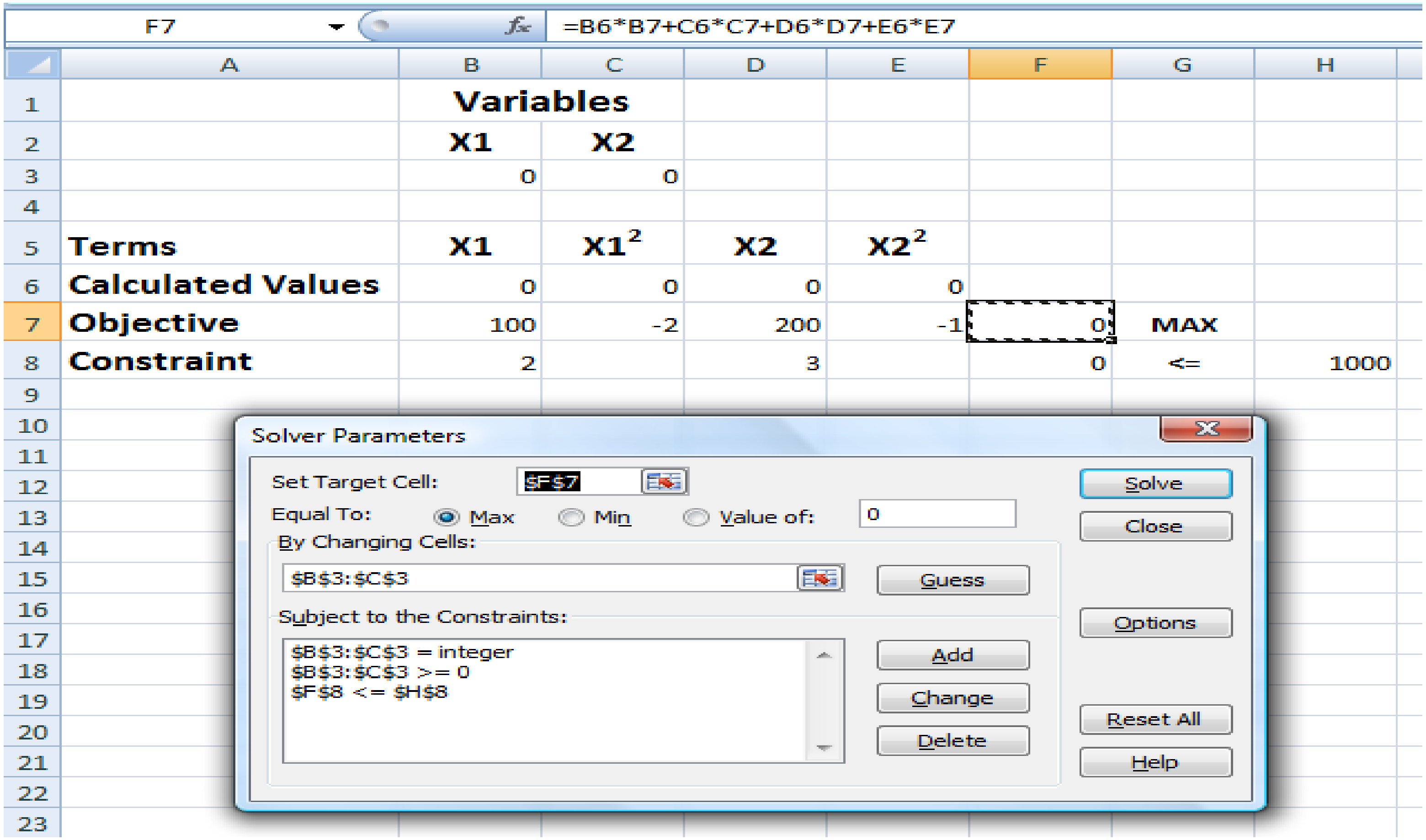
- Max: $Z=100 X_1 - 2 X_1^2 + 200 X_2 - X_2^2$
- Subject to:
  - $2 X_1 + 3 X_2 \leq 1000$
- Now solve in Excel

- Do not look ahead !

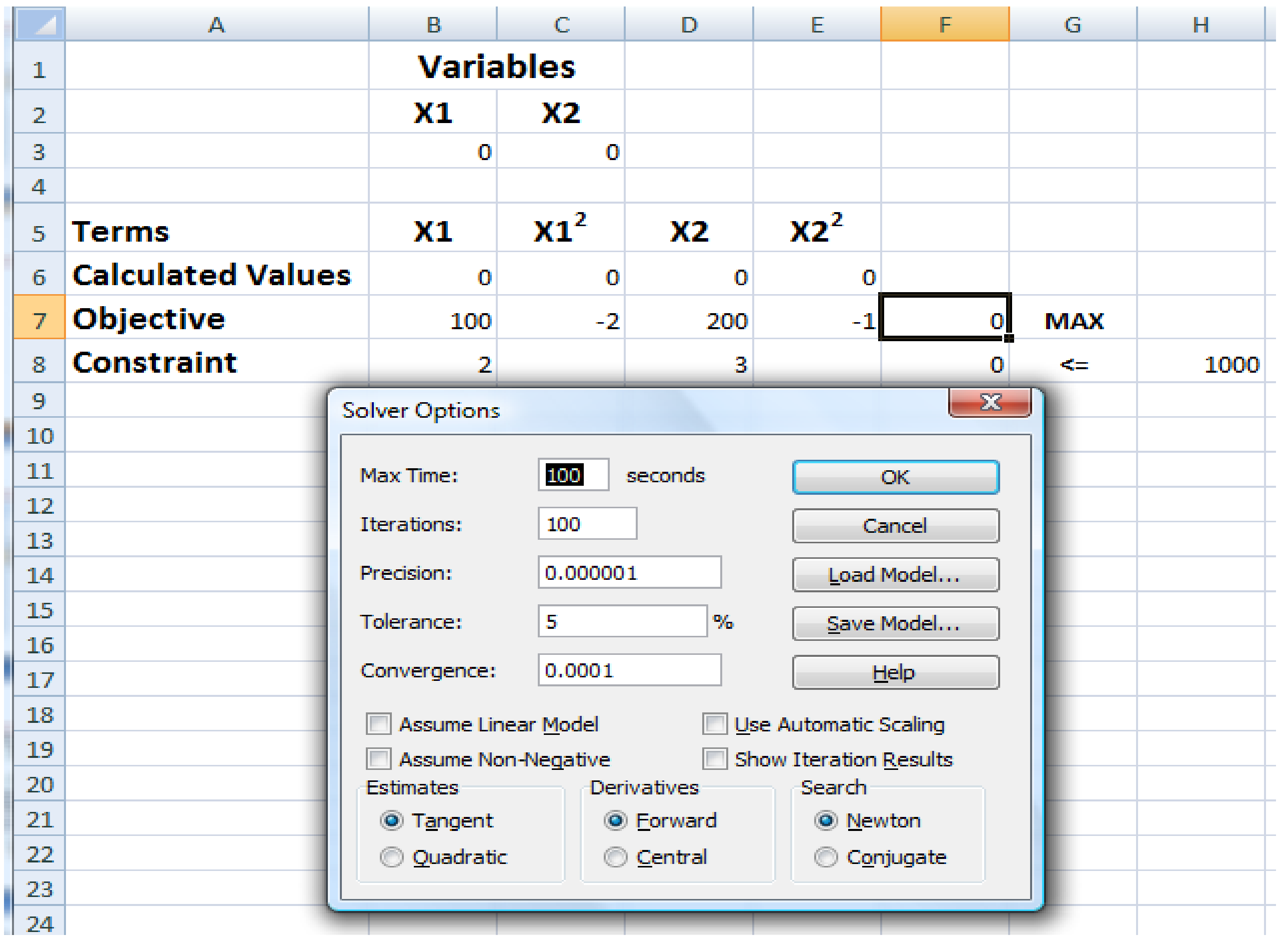
Excel Setup

	A	B	C	D	E	F	G	H
1		Variables						
2		X1	X2					
3		0	0					
4								
5	Terms	X1	$X1^2$	X2	$X2^2$			
6	Calculated Values	=B3	=B3^2	=C3	=C3^2			
7	Objective	100	-2	200	-1	=B6*B7+C6*C7+D6*D7+E6*E7	MAX	
8	Constraint	2		3		=B6*B8+D6*D8	<=	1000
9								

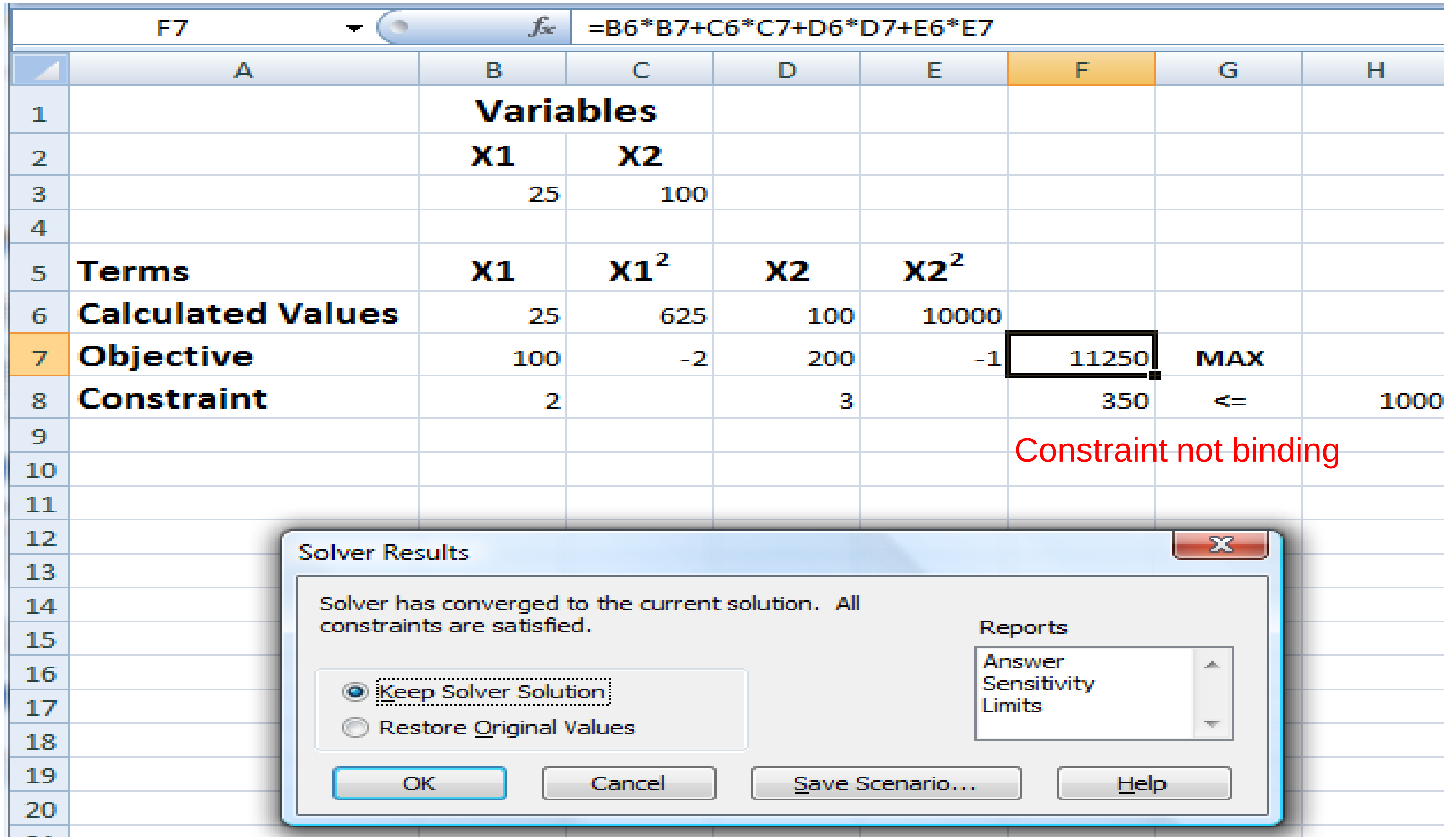
Solver


F7
=B6*B7+C6*C7+D6*D7+E6*E7
Variables
X1 X2
0 0
Terms X1 X1^2 X2 X2^2
Calculated Values 0 0 0 0
Objective 100 -2 200 -1 0 MAX
Constraint 2 3 0 <= 1000
Solver Parameters
Set Target Cell: $F$7
Equal To: Max Min Value of: 0
By Changing Cells:
$B$3:$C$3
Subject to the Constraints:
$B$3:$C$3 = integer
$B$3:$C$3 >= 0
$F$8 <= $H$8
Solve
Close
Guess
Options
Add
Change
Delete
Reset All
Help

Solver Options
[do not check “assume linear”]



Solver Results



F7 =B6*B7+C6*C7+D6*D7+E6*E7

	A	B	C	D	E	F	G	H
1		Variables						
2		X1	X2					
3		25	100					
4								
5	Terms	X1	$X1^2$	X2	$X2^2$			
6	Calculated Values	25	625	100	10000			
7	Objective	100	-2	200	-1	11250	MAX	
8	Constraint	2		3		350	<=	1000

	A	B	C	D	E	F	G	H
1		Variables						
2		X1	X2					
3		25	100					
4								
5	Terms	X1	$X1^2$	X2	$X2^2$			
6	Calculated Values	25	625	100	10000			
7	Objective	100	-2	200	-1	11250	MAX	
8	Constraint	2		3		350	<=	1000
9								

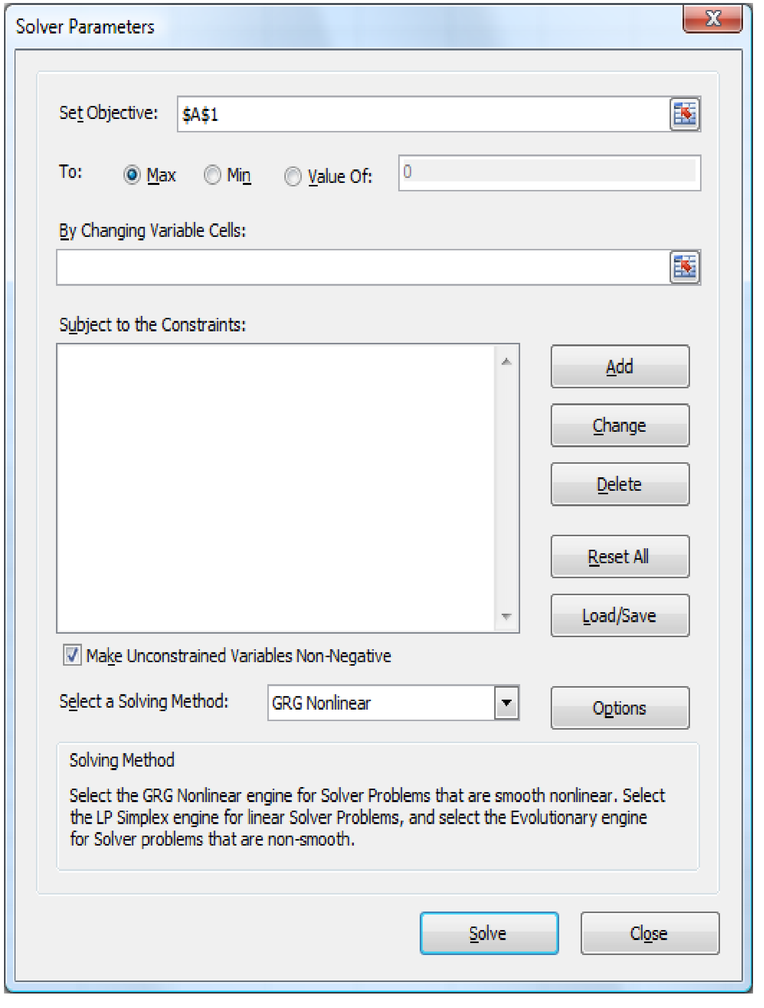
Constraint not binding

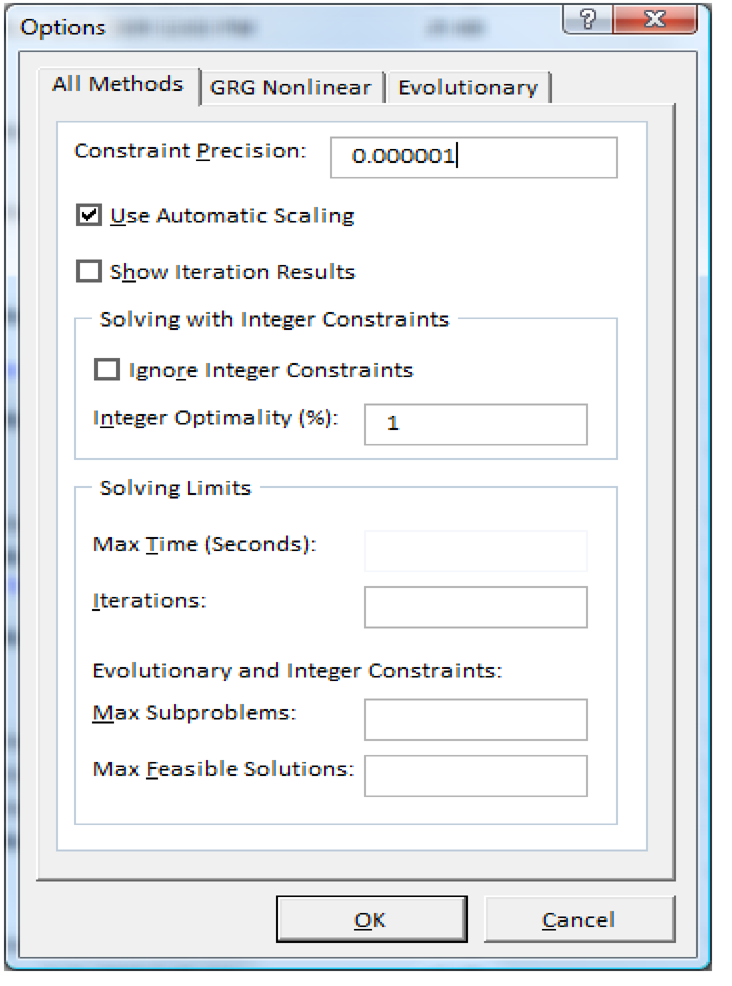
	A	B	C	D	E	F	G
1	Constraint		Isoprofit X2's				
2	X1	X2	11246	11247	11248	11249	11250
3	24	317.3333	101.4142	101	100	#NUM!	#NUM!
4	24.2	317.2	101.6492	101.3115	100.8485	#NUM!	#NUM!
5	24.4	317.0667	101.8111	101.51	101.1314	100.5292	#NUM!
6	24.6	316.9333	101.9183	101.6371	101.2961	100.8246	#NUM!
7	24.8	316.8	101.9799	101.7088	101.3856	100.9592	#NUM!
8	25	316.6667	102	101.7321	101.4142	101	100
9	25.2	316.5333	101.9799	101.7088	101.3856	100.9592	#NUM!

Excel Solver Algorithms

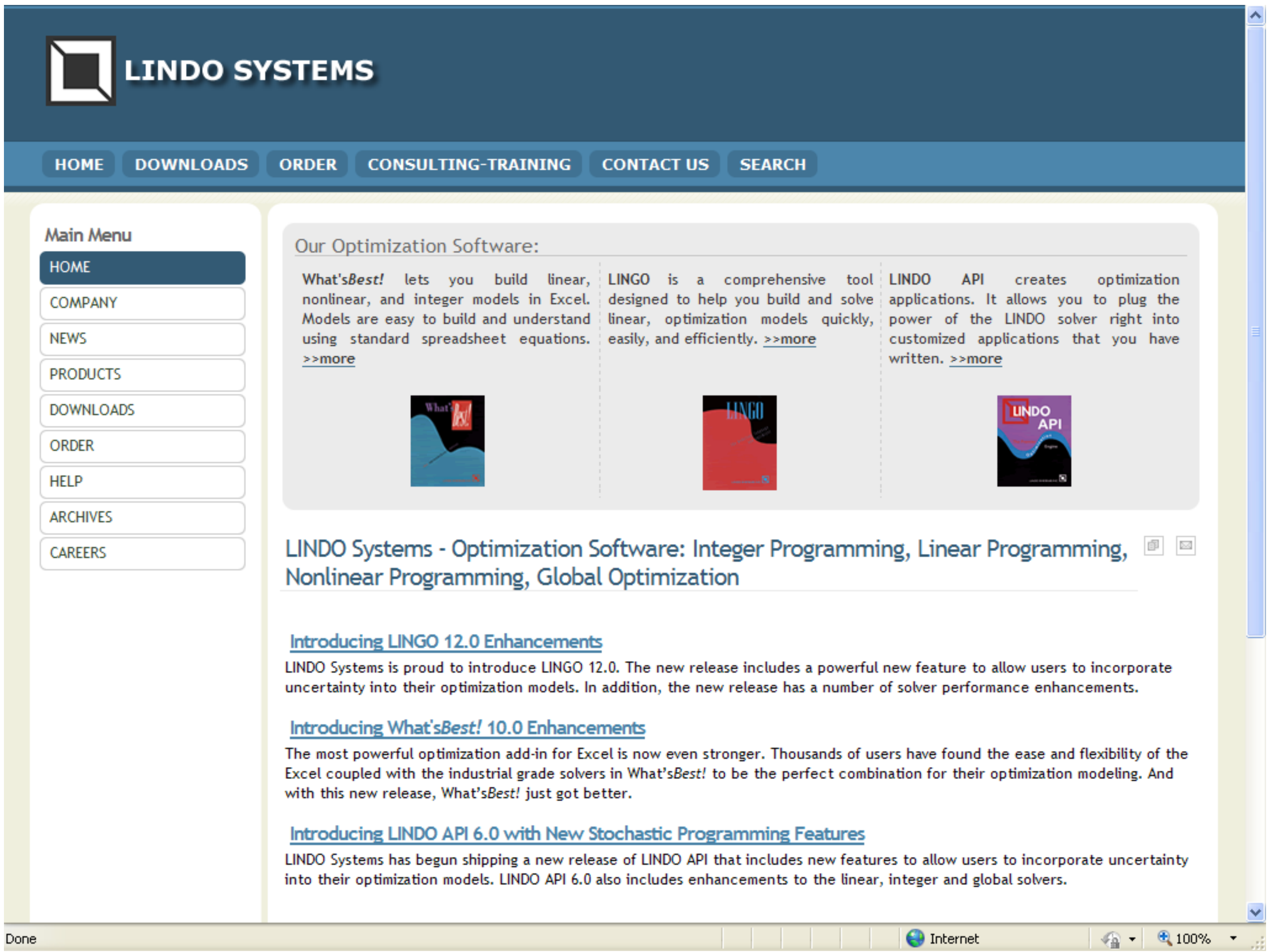
- Simplex Method
  - Revised simplex method for regular LP problems
    - Auto switch to dual simplex
  - For integer or binary uses branch and bound
- GRG Nonlinear – Generalized Reduced Gradient
- Evolutionary Method
  - Uses a variety of genetic and local search methods

Excel Solver Algorithms (con't)

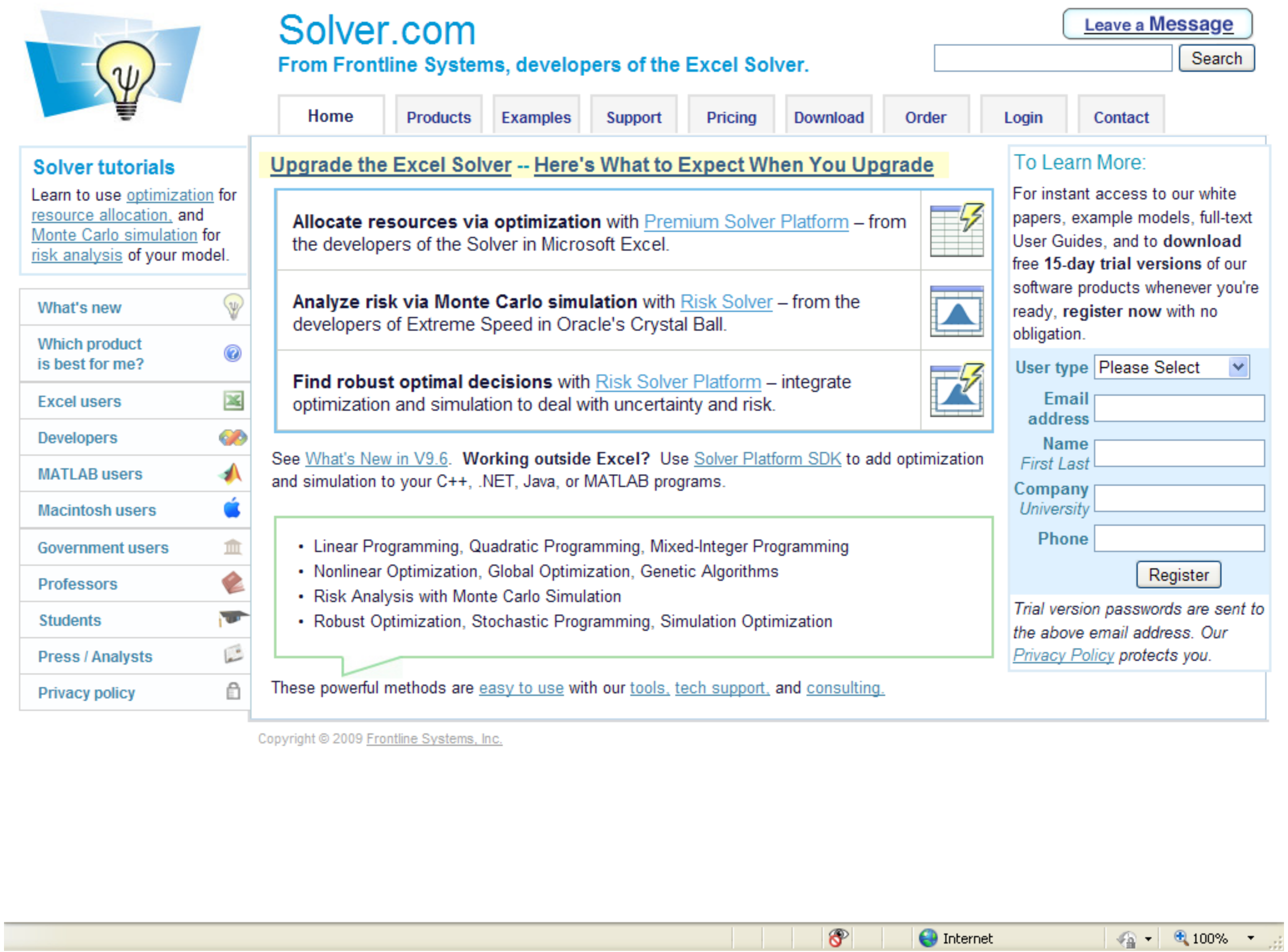




Other Excel Solvers



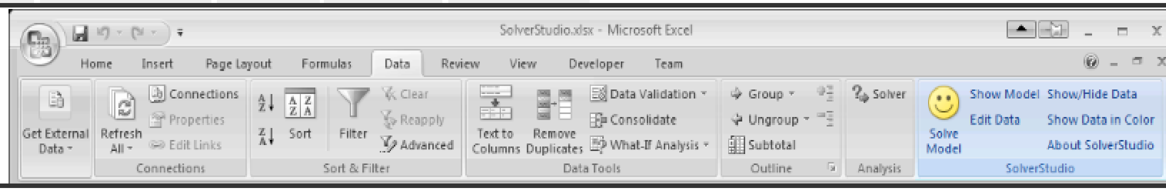
Other Excel Solvers (con't)



Other Excel Solvers (con't)

SolverStudio for Excel
An integrated environment for optimisation using modelling languages within Excel

SOLVERSTUDIO | SCREENSHOTS | DOWNLOAD & INSTALL | USING SOLVERSTUDIO | LANGUAGES | DEVELOPMENT | HELP & FEEDBACK



SolverStudio

SolverStudio 0.5.4 (2013.06.25) has been released, with numerous bug fixes and improvements. This also includes an experimental all-user installation setup. Many thanks to Ted Ralphs for using SolverStudio in his classes and giving me extensive feedback that has improved this release.

Major New SolverStudio Release SolverStudio 0.5 has been released for beta testing. This is a major new release which adds support for the **COOPR/Pyomo** modelling environment, the Python simulation tool **SimPy**, solving of both **AMPL** and now **GAMS** models (thanks, GAMS, for providing the GDX DLL"s) using the cloud optimisation service **NEOS**, and better support for **Gurobi** (including the ability to use Gurobi models without installing Python and a new license manaager — thanks, Gurobi, for help with this). There are also numerous small improvements, as documented in **this post**. All feedback is welcomed.

Welcome to SolverStudio

SolverStudio is an add-in for Excel 2007 and 2010 on Windows that allows you to build and solve optimisation models in Excel using either of the following optimisation modelling languages:

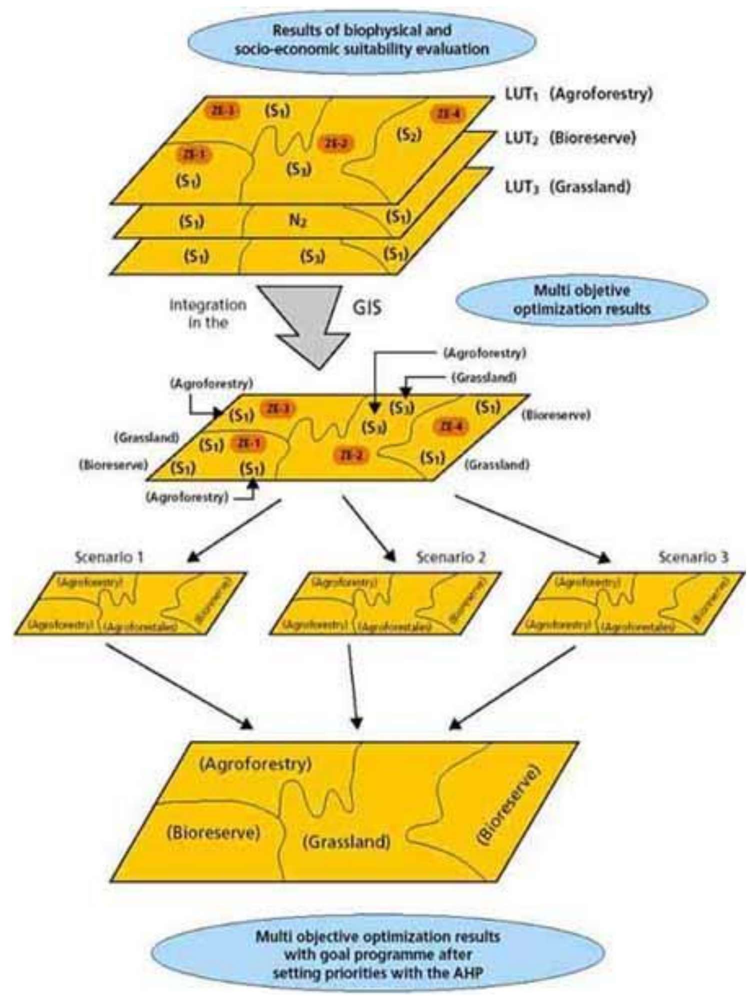
1. **PuLP**, an open-source **Python**-based **COIN-OR** modelling language developed by Stu Mitchell. PuLP is included with SolverStudio.
2. **COOPR/Pyomo**, an open source **COIN-OR** modelling language for Python which extends Pulp with abstract models, support for stochastic programming, and a larger range of solvers.
3. **AMPL**, a commercial modelling language. OpenSolver either requires AMPL to be installed by the user, or can install a **free trial version** of AMPL that allows up to 300 variables and 300 constraints. We have a **tutorial on using AMPL with SolverStudio**.
   We also support running AMPL models in the cloud using the excellent **NEOS server**; see **AMPL on NEOS**.
4. **GMPL** (GNU MathProg Language), an open source AMPL look-alike developed as part of GLPK (GNU Linear Programming Kit). GMPL is included with SolverStudio.
5. **GAMS**, a commercial modelling language. SolverStudio requires GAMS to be installed by the user. GAMS provide a **free trial version**. Support for solving GAMS models using the **NEOS server** is working in our new beta version.
6. **Gurobi**, a commercial solver which can be accessed from SolverStudio using the **Gurobi Python modelling interface**. This requires the Gurobi solver be installed. SolverStudio provides a license manager to help manage Gurobi licenses.
7. **SimPy**, an open source Python simulation language, which is now included with the SolverStudio download.
8. Any other Python software that runs under either **IronPython** or **standard Python**, both of which are included in SolverStudio.
9. We hope to add **Zimpl** in the near future, and allow GMPL to be used with CBC.

AMPL neos COIN|OR GUROBI OPTIMIZATION

SolverStudio allows you to create and edit your optimisation model without leaving Excel, and to save your model inside your workbook. You can also easily link data on your spreadsheet to sets, parameters, constants and variables used in the model. SolverStudio can run the model to solve the problem and then put the answer back onto the spreadsheet.

Goal Programming

- The line between objectives and constraints may not be clear
- When there are <u>many objectives</u>, some of them may be treated as constraints



Goal Programming Example

- A company needs to develop an advertising campaign
- They would like to reach three groups of customers:
  - **High income women (HIW)**
  - **Teenage boys from affluent families (TB)**
  - **Middle age or retired men (RM)**
- They can advertise on TV or in magazines and their budget is $600,000
- What is the product the company makes ?

Ad Reach (millions) and Cost

	HIW	TB	RM	Cost
TV	7	10	5	$100,000 per spot
Magazine	3	5	4	$60,000 per ad

Goal Programming (con't)

- If there was only one group of customers, the problem would be easier to solve
- For example, if they did not need to be concerned about TB and RM, then the problem could be solved by direct linear programming:
  - Maximize z = 7x + 3y
  - Subject to:
    - 100x + 60y <= 600
  - Where x = # of TV spots, and y = # of ads
  - With solution: x = 6, y=0, z = 42

	HIW	Cost
TV	7	$100,000 per spot
Magazine	3	$60,000 per ad

Goal Programming (con't)

- We could take each of the individual objectives as "goals" and introduce a variable (deviational variable) which indicates how much we fall short (s's) of our goals:
  - $7x + 3y + s1 = 42$ (42 is the optimal just for HIW, from last slide)
  - $10x + 5y + s2 = 60$ (60 is the optimal just for TB)
  - $5x + 4y + s3 = 40$ (40 is the optimal just for RM)

Goal Programming (con't)

- Then we want to minimize the overall shortfall, where we score the shortfalls relative to each other, for example:
  - Minimize: z = 200s1 +100s2 + 50s3
  - In goal programming, the objective function is written in terms of the goal variables
  - Where we value HIW twice as much as TB, and TB twice as much as RM – we are prioritizing our goals
- Still with the budget constraint:
  - 100x +60y <= 600
- And the equality constraints on the previous slides involving x1, x2, s1, s2, s3

Goal Programming (con't)

- The solution here is:
  - X = 6 (we still spend all our money on TV)
  - Y = 0
  - S1 = 0
  - S2 = 0
  - S3 = 10
- Thus we meet the first two goals and fall short on the third
- We could change the weights and may get another solution
- Thus goal programming involves the solution of several nested linear programming models

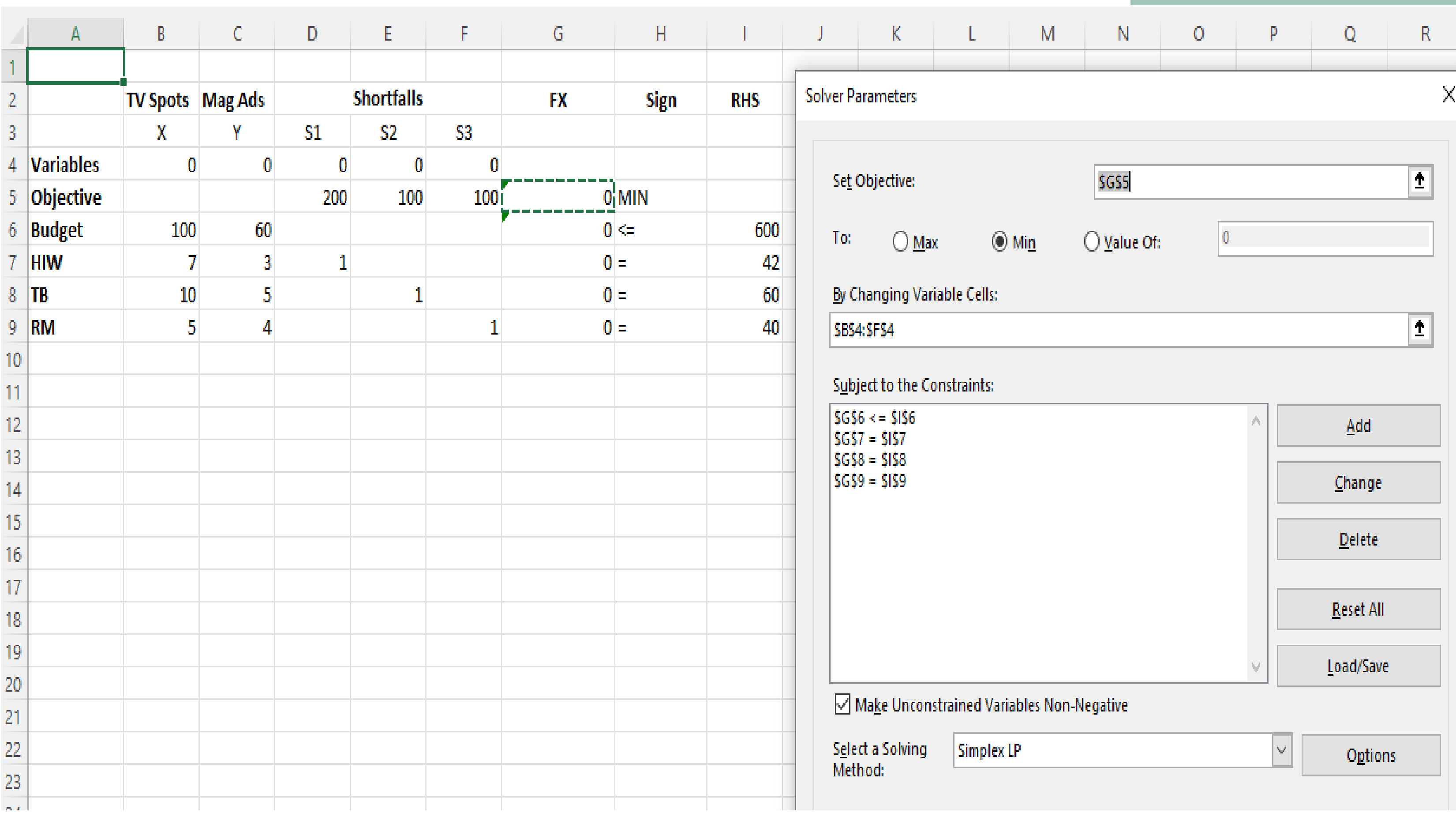
Excel LP Formulation

	A	B	C	D	E	F	G	H	I
1									
2		TV Spots	Mag Ads		Shortfalls		FX	Sign	RHS
3		X	Y	S1	S2	S3			
4	Variables	0	0	0	0	0			
5	Objective			200	100	100	0	MIN	
6	Budget	100	60				0	<=	600
7	HIW	7	3	1			0	=	42
8	TB	10	5		1		0	=	60
9	RM	5	4			1	0	=	40
10									

Excel LP Formulation (con’t)

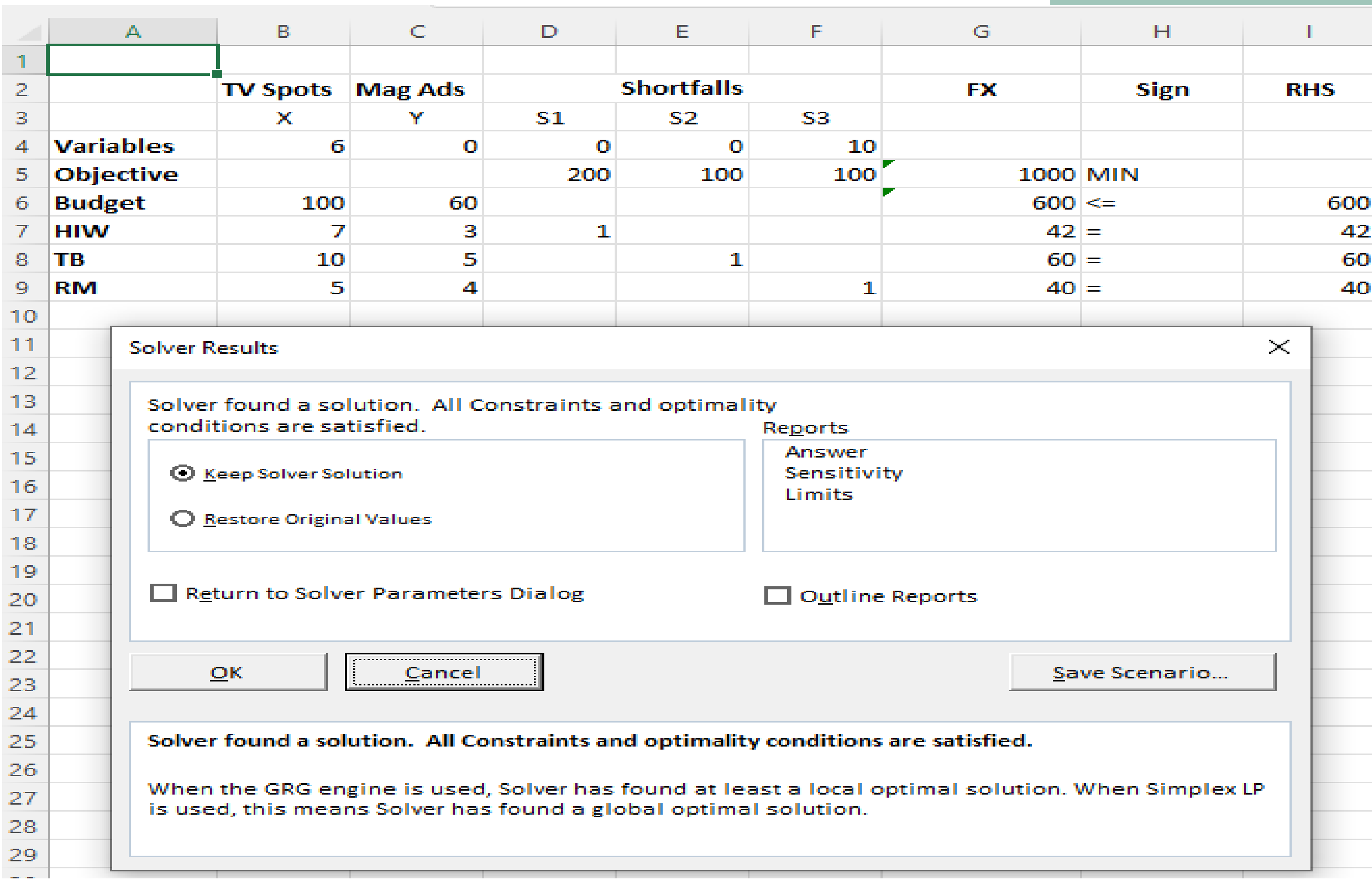
	A	B	C	D	E	F	G	H	I
1									
2		TV Spots	Mag Ads		Shortfalls		FX	Sign	RHS
3		X	Y	S1	S2	S3			
4	Variables	0	0	0	0	0			
5	Objective			200	100	100	=SUMPRODUCT(D4:F4,D5:F5)	MIN	
6	Budget	100	60				=SUMPRODUCT(B4:C4,B6:C6)	<=	600
7	HIW	7	3	1			=SUMPRODUCT(B4:F4,B7:F7)	=	42
8	TB	10	5		1		=SUMPRODUCT(B4:F4,B8:F8)	=	60
9	RM	5	4			1	=SUMPRODUCT(B4:F4,B9:F9)	=	40
10									
11									

Excel LP Formulation (con't)



	A	B	C	D	E	F	G	H	I
1									
2		TV Spots	Mag Ads		Shortfalls		FX	Sign	RHS
3		X	Y	S1	S2	S3			
4	Variables	0	0	0	0	0			
5	Objective			200	100	100	0	MIN	
6	Budget	100	60				0	<=	600
7	HIW	7	3	1			0	=	42
8	TB	10	5		1		0	=	60
9	RM	5	4			1	0	=	40

Excel LP Formulation (con’t)



	A	B	C	D	E	F	G	H	I
1									
2		**TV Spots**	**Mag Ads**		**Shortfalls**		**FX**	**Sign**	**RHS**
3		X	Y	S1	S2	S3			
4	**Variables**	6	0	0	0	10			
5	**Objective**			200	100	100	1000	MIN	
6	**Budget**	100	60				600	<=	600
7	**HIW**	7	3	1			42	=	42
8	**TB**	10	5		1		60	=	60
9	**RM**	5	4			1	40	=	40

Overages

- Sometimes one has situations were there may be “overages” such as over production or over time hours
- In this situation overage variables can be used, and there will normally be penalty rates for overages
- In some problems there may be penalties for both overages and shortages
- Overage and shortage variables are often called “deviational” variables

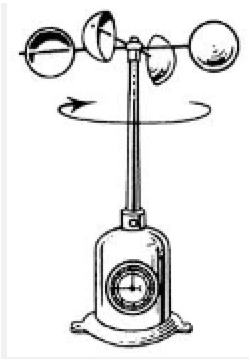
Over/short Example

- A company produces two types of wind gauges
- The labor required to produce gauge 1 is 4 hours and the profit contribution is $5
- The labor required to produce gauge 2 is 3 hours and the profit contribution is $2
- 32,000 hours of labor are available
  - A $2 penalty is incurred for each hour of overtime labor (over 32,000)
  - A $1 penalty is incurred for each unused labor hour (under 32,000)

Over/short Example (con't)

- At least 7000 units of gauge 1 must be produced and 10,000 units of gauge 2 produced to met minimum demand
- For each unit of either product by which production falls short of minimum demand there is a stockout loss of $5
- The organization's goals are:

1. Have $48,000 in profit
2. Not exceed 32000 hours labor
3. Meet demand for gauge 1
4. Meet demand for gauge 2



- What are the objective function variables ? Decision variables ?
- What are the constraints ?

- Do not look ahead !

Over/short Example (con't)

- The decision variables are X1 and X2 (# of each gauge type)
- The goal (deviation) variables are the overages and shortages for each goal (G1 thru G4):
  - G1o – goal one overage
  - G1s – goal one shortage
  - G2o – goal two overage
  - G2s – goal two shortage
  - G3o – goal three overage
  - G3s – goal three shortage
  - G4o – goal four overage
  - G4s – goal four shortage

Over/short Example (con't)

- The objective function (to minimize) in terms of the goal variables is the sum of:
  - G1s – under $48,000 in profit
  - G2o – overtime (at $2/hr)
  - G2s – unused labor (at $1/hr)
  - G3s – not meeting demand for gauge 1 (at $5 per unit)
  - G4s – not meeting demand for gauge 2 (at $5 per unit)
- All goals are equal priority and measured in $
- Min Z = G1s+2*G2o+G2s+5*G3s+5*G4s

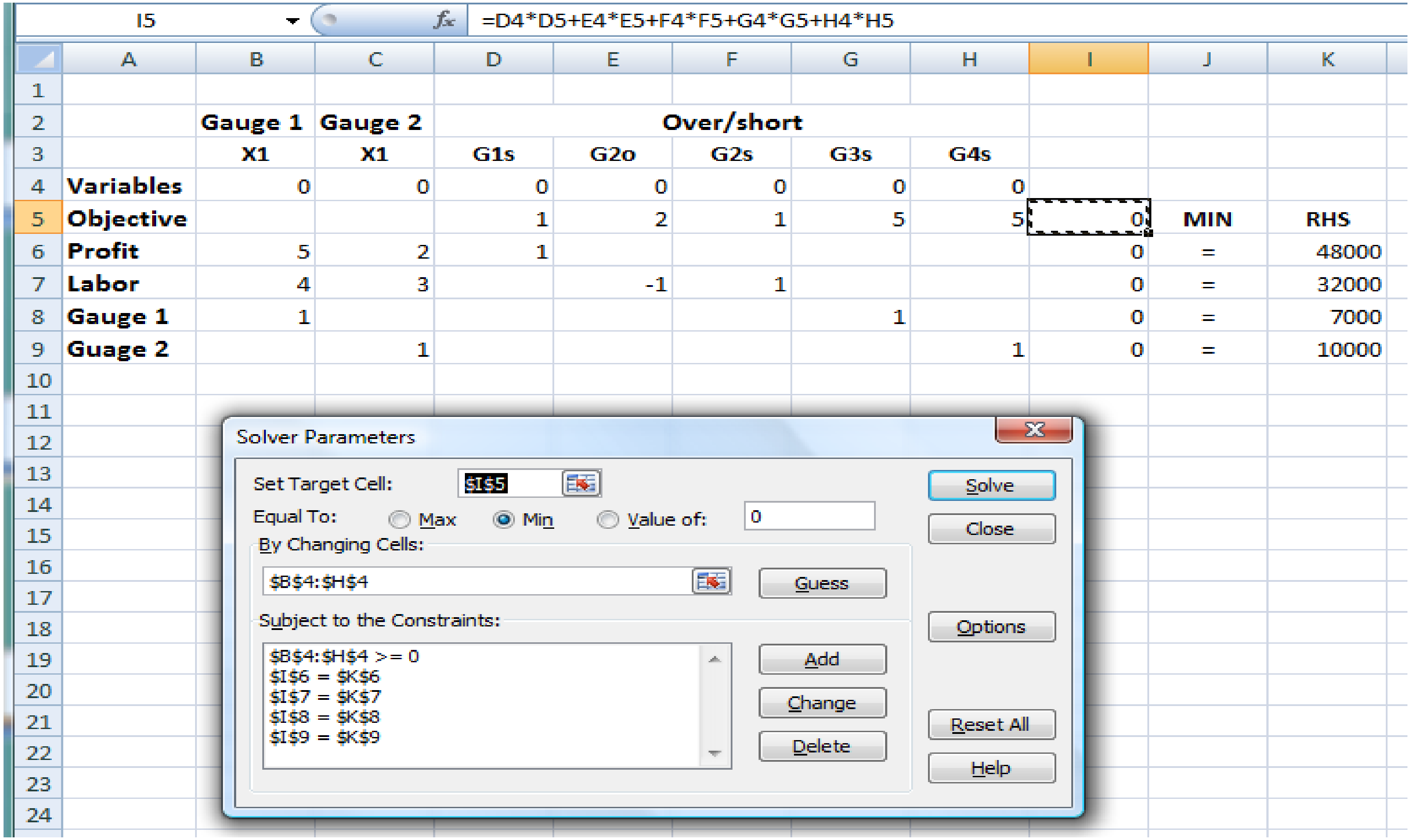
Over/short Example (con't)

- Constraints (variables in red not needed):
  - Definition of over/short:
    - Profit: 5*x1 + 3*x2 + G1s – G1o = 48000
    - Labor: 4*x1 + 2*x2 + G2s – G2o = 32000
  - Demand:
    - Gauge 1: x1 + G3s – G3o = 7000
    - Gauge 2: x2 + G4s – G4o = 10000
  - Non-negative:
    - X's and G's are >= zero

Spreadsheet Formulation

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Gauge 1	Gauge 2	Over/short							
3		X1	X1	G1s	G2o	G2s	G3s	G4s			
4	Variables	0	0	0	0	0	0	0			
5	Objective			1	2	1	5	5	0	MIN	RHS
6	Profit	5	2	1					0	=	48000
7	Labor	4	3		-1	1			0	=	32000
8	Gauge 1	1					1		0	=	7000
9	Guage 2		1					1	0	=	10000

Using Solver

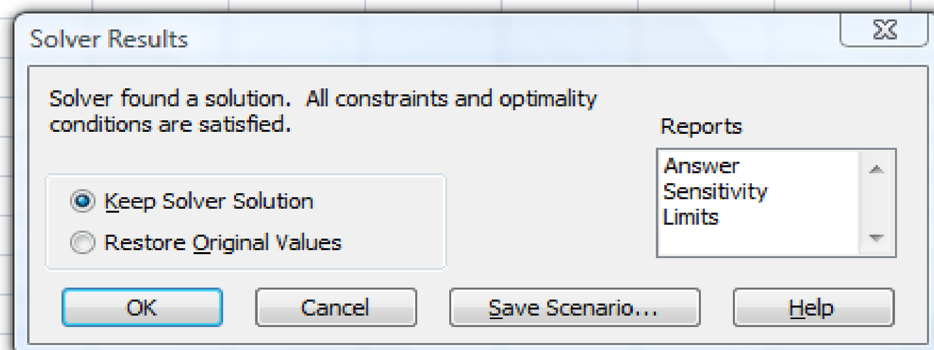


I5 | =D4*D5+E4*E5+F4*F5+G4*G5+H4*H5

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Gauge 1	Gauge 2			Over/short					
3		X1	X1	G1s	G2o	G2s	G3s	G4s			
4	Variables	0	0	0	0	0	0	0			
5	Objective			1	2	1	5	5	0	MIN	RHS
6	Profit	5	2	1					0	=	48000
7	Labor	4	3		-1	1			0	=	32000
8	Gauge 1	1					1		0	=	7000
9	Guage 2		1					1	0	=	10000

Optimal Solution

	A	B	C	D	E	F	G	H	I	J	K
1											
2		**Gauge 1**	**Gauge 2**			**Over/short**					
3		**X1**	**X1**	**G1s**	**G2o**	**G2s**	**G3s**	**G4s**			
4	**Variables**	5600	10000	0	20400	0	1400	0			
5	**Objective**			1	2	1	5	5	47800	**MIN**	**RHS**
6	**Profit**	5	2	1					48000	=	48000
7	**Labor**	4	3		-1	1			32000	=	32000
8	**Gauge 1**	1					1		7000	=	7000
9	**Guage 2**		1					1	10000	=	10000



Homework

- Textbook Chapter 10
- Quiz on these slides and Chapter 9 & 10
- Discussion Questions to be answered: 3, 4, 5, 6, 8 from Chapter 10
- Project Eight →

Project 8

- The school system has been given a grant of 6 million dollars to expand its service facilities
- Four different types of facilities have been requested by parents and students:
  - Gymnasiums
  - Athletic fields
  - Wi-Fi hot spots
  - Student lounges

- The following table shows the demand for each, the capital cost, the monthly maintenance costs, and the expected usage (students/week)

Project 8 (con't)

Facility	Capital Cost ($10,000)	Maintenance ($1000/mo)	Usage (students/week)	Number Requested
Gymnasium	80	4	1500	7
Athletic Field	24	8	3000	10
Wi-Fi hot spot	15	3	500	8
Student lounge	40	5	1000	12

Project 8 (con't)

Project 8 (con't)

- The school board has located 50 thousand dollars in their monthly budget for maintenance although more might be found
- The school board has set the following list of prioritized goals and weights:
  - 1. The total grant must be spent (weight=100)
  - 2. The facilities should be used by 20,000 or more students per week (weight=75)
  - 3. If more maintenance dollars are needed, the additional amount should be limited to 10 thousand per month (weight=50)
  - 4. The board would like to meet the wishes of the students and parents in terms of the number of facilities, however this priority should be weighted according to the people expected to use each facility (weight=25)
- How many of each type facility should be built ?