Module 11: Supply Chain Models

Reading Material: 11.1 – Introduction to Supply Chain and Assignment Models

In this module, we continue to expand our LP modeling capabilities. Here, we look at foundational supply chain models. Ultimately, these models set the groundwork for creating even more general (and useful) network models. This module may go by quickly as it builds on previously mastered knowledge of using matrix-style decision variables.

Some of the most useful applications of LP are in getting objects from points A to B (to C to D to . . .) in the most efficient (optimal) way possible. This module starts us on our way.

Assignment models are also introduced in this module. Assignment problems look at optimizing the assignment for example, of teachers to classes, workers to jobs, and employees to projects. These types of models merge supply chain model attributes with our next topic, the use of 0/1 (binary) variables in decision situations.

As before, let's begin the fun with the following three in-class exercises, followed by the usual set of practice problems.

Reading Material: 11.2 – T.B.P. Rent-A-Wreck Company

11.2.1 Problem Description

The T. B. P. Rent-a-Wreck car rental company, headquartered near the large structure located at Duck and Hall of Fame Avenue in Stillwater, Oklahoma, is determining how to move its vehicles in the least costly manner to match demand for the next period.

Presently, there are two vehicles in Stillwater, two vehicles in Perkins, and four vehicles in Pawnee (all in Oklahoma).

Rentals for tomorrow are: four vehicles needed in Ponca City, one vehicle needed in Guthrie, and three vehicles needed in Cushing (also all locales in Oklahoma).

Distances between the cities are given in the table below. Get cars to Ponca City, Guthrie, and Cushing such the total overall amount of miles traveled by the eight cars is minimized.

	Ponca	Guthrie	Cushing
STW	42	35	25
Perkins	54	26	16
Pawnee	45	62	26

11.2.2 Problem Formulation

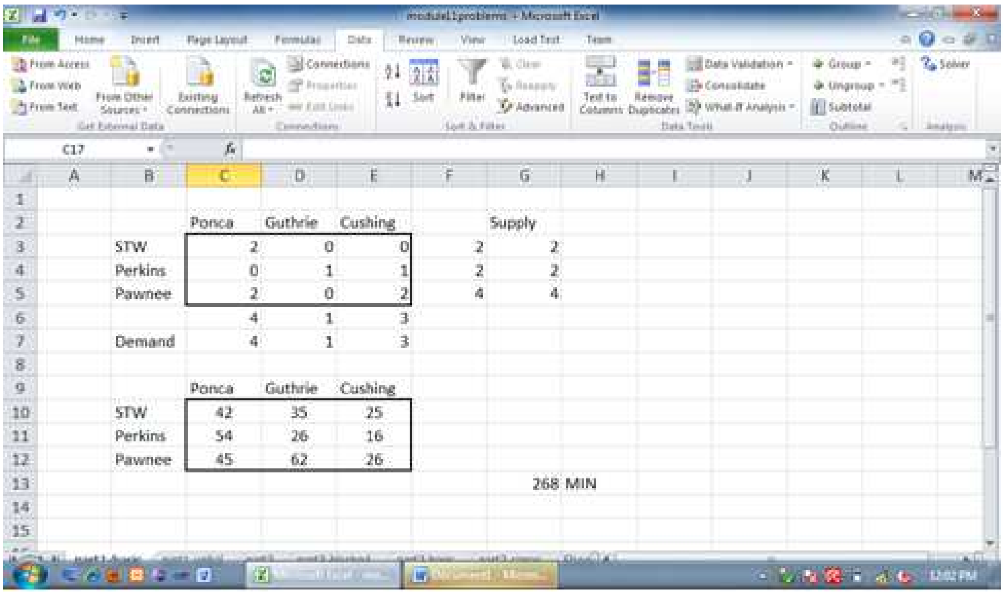
This scenario is a classic transportation problem. Many years ago, before we had such powerful, easy-to-use implementations of tools like the EXCEL Solver, we would teach a specialized solution algorithm for solving problems of this type. It was kind of fun and also insightful into how management science researchers would sometimes create specialized solution approaches based on problem characteristics. Today, this is not critical for foundational problems like transportation models. (But the concept of specialized algorithms is still extremely important – but beyond the scope of our goals in this class.)

Anyway, this is a transportation problem since we are trying to optimally get objects (cars) from where they are presently located (supply) to where they are needed (demand). Logistics and supply chain issues confront all of us. The basics learned here can be used for supply chain models of all complexity and size.

We are trying to find the best way to get cars from three locations (Stillwater, Perkins, and Pawnee) to three other locations (Ponca City, Guthrie, and Cushing). These are all real towns or cities in North Central Oklahoma. Have you figured out what the initials TBP might represent (especially given what is approximately – actually, just a little west – at the corner of Duck and Hall of Fame in Stillwater)?

Hmmm – three supply locations to three demand locations. Sounds like a 3×3 kind of problem! Yes, there are nine (3 × 3 = 9) specific decisions for the model: the nine combinations of the number of cars to move from city (Stillwater, Perkins, and Pawnee) to city (Ponca City, Guthrie, and Cushing). Thus, a 3×3 matrix is reasonable decision variable representation of an LP model.

Therefore, in building the EXCEL representation of the model, we borrow previously used matrix representations. Figure 11.1 is the EXCEL representation of this scenario. Decision variable placeholders are in Cells C3:E5. Also, remember the automatic row and column sums one should do when using a matrix-style representation? Relevant here, so Column F are the row sums (e.g., the formula in F3 is =SUM(C3:C5)) and Row 6 are the column sums (e.g., the formula in C6 is =SUM(C3:C5)). In words, Column F represents the sum of the cars leaving the supply locations (the present location of the cars), and Row 6 represents the sum of the cars coming to the demand locations (where the cars need to be in the morning). These row and column sums will be useful in specifying constraints.



	Ponca	Guthrie	Cushing		Supply
STW	2	0	0	2	2
Perkins	0	1	1	2	2
Pawnee	2	0	2	4	4
	4	1	3		
Demand	4	1	3		

	Ponca	Guthrie	Cushing
STW	42	35	25
Perkins	54	26	16
Pawnee	45	62	26

268 MIN

Figure 11.1

Our objective is to minimize the total overall mileage traveled by all the cars. A mileage chart in matrix form is provided (how convenient!). Every car that travels between Stillwater and Ponca City (cell C3) adds 42 miles (Cell C10) to our final mileage tally. Thus, the total mileage traveled by all cars is a simple sumproduct: = SUMPRODUCT(C3:E5, C10:E12). This sumproduct aligns all nine distances with the corresponding nine decision variables, multiplies them, and sums up all nine terms. This formula is in Cell G13 and will be the target cell.

Finally, transportation models have two core constraints – supply constraints and demand constraints. As luck would have it, the number of cars available is exactly the number of cars needed tomorrow. Let us call this a *balanced* problem – when demand exactly meets supply. This is simplest to model (versus when supply does not equal demand, an unbalanced problem).

Column G lists the number of cars presently at our supply locations. Row 7 lists the number of cars needed at the demand locations tomorrow. Given that the problem is balanced, the supply and demand constraints are equalities. The supply constraints are F3:F5 = G3:G5, and the demand constraints are C6:E6 = C7:E7. Figure 11.2 shows the Solver inputs for the problem.

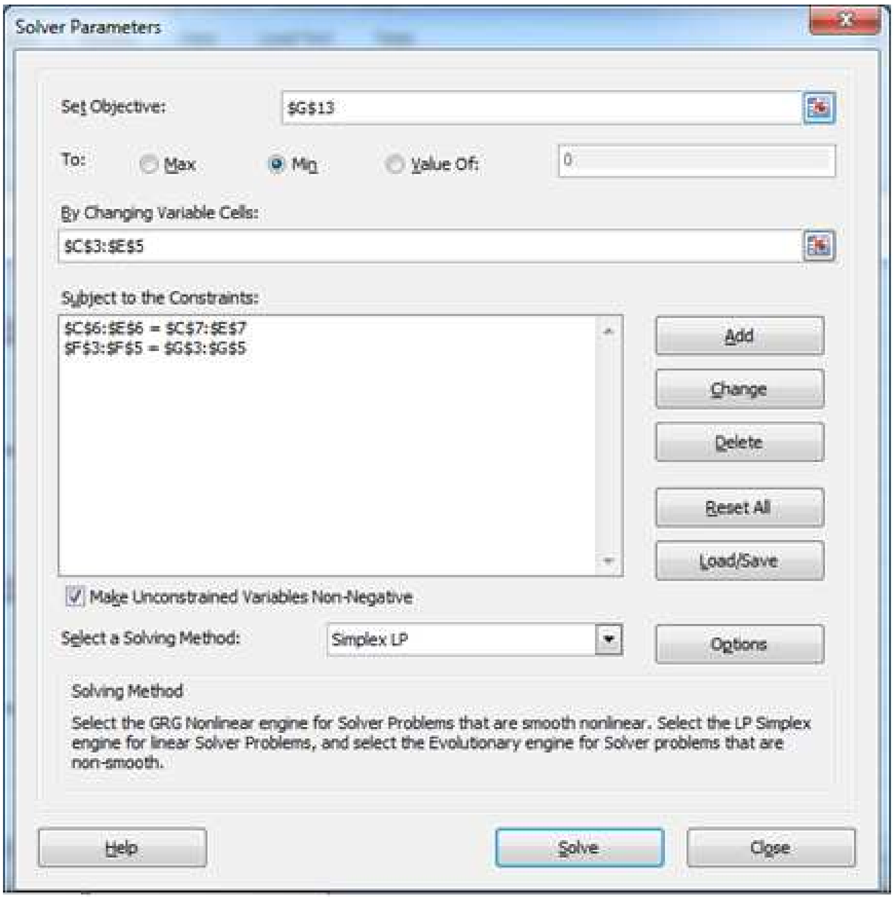


Figure 11.2

11.2.3 Model Solution

The optimal solution requires 268 miles of travel by the eight cars. The two cars in Stillwater will travel to Ponca City (C3 = 2), one car in Perkins will need to go to Guthrie (D4 = 1), and one will need to go to Cushing (D5 = 1). Finally, two cars will need to be sent from Pawnee to Ponca City (C5 = 2) and two cars from Pawnee to Cushing (E5 = 2).

Note that we did not need to explicitly worry about whole numbers. If you examine the algebraic representation of the model constraints, one will find only ‘1’ coefficients (or ‘0’ coefficients) on the left-hand side (LHS) of these constraints. Because of this algebraic structure, one is guaranteed all integer solutions. If additional “side” constraints were added to the model that caused solutions to be fractional, one could easily add integer constraints.

11.2.4 Modifying the problem – Unbalanced

For completeness, we modify the original problem to illustrate how to handle supply not equaling demand. Suppose the demand for cars at Cushing is two cars instead of three, while all else remains the same from the original problem.

Figure 11.3 shows the new EXCEL spreadsheet (only change is found in cell E7). Figure 11.4 shows the only change necessitated in the Solver. As we have more cars available (supply) than needed (demand), one deals with this by specifying supply constraints as less than or equal to, and demand constraints still as equal to (because all supply will not be used!). Specifically, Column F is set to less than or equal to Column G in the unbalanced model.

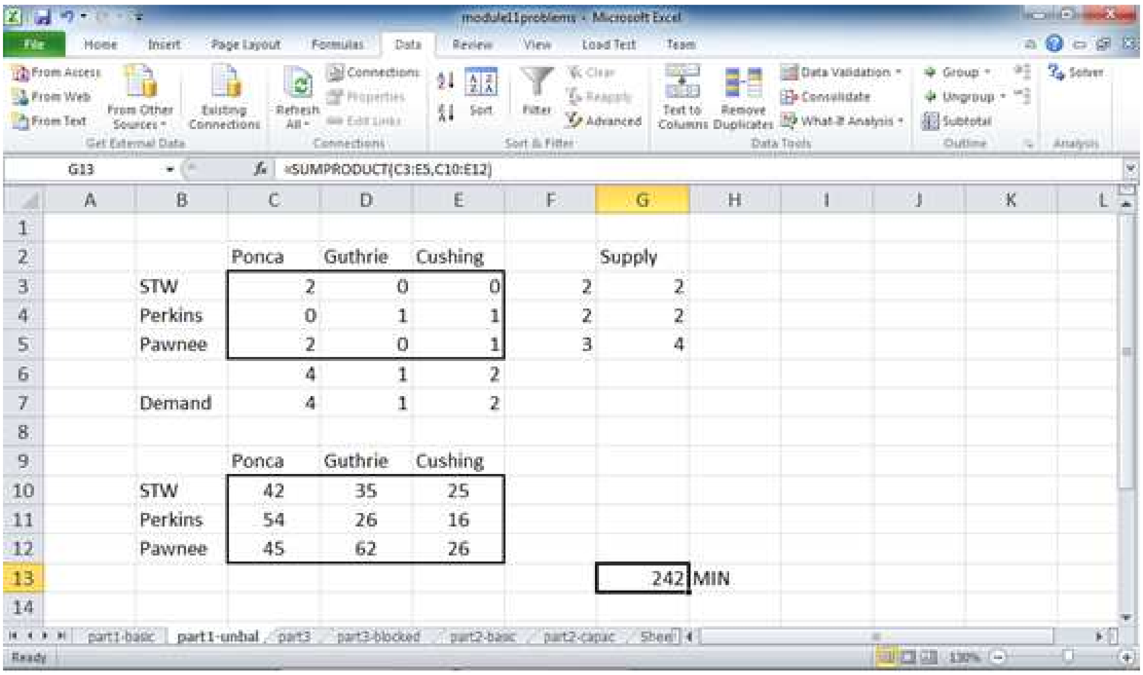


Figure 11.3

Figure 11.4

Note that in the new solution, Pawnee is left with the extra car.

One might think – why don't we always model both the supply and demand constraints with less than or equal to constraints in every case, and then we don't have to consider if supply is less than, equal to, or greater than demand! Consider the implications of this in a model in which an objective is being minimized (distance, costs, etc.). The least cost solution with all less than or equal to constraints – is to ship nothing! (a cost of 0!). Thus, our model must force the units of interest to their desired destination through wise use of the constraints.

The next example expands our supply chain scenario by considering moving cars from where they are located to where they are needed – but through an intermediate inspection station. We are adding a stage to our supply chain – this type of problem is termed a 'transshipment' model.

Reading Material: 11.3 – Example 2: T.B.P. with Inspection Station

11.3.1 Problem Description

The T. B. P. Rent-a-Wreck car rental company, headquartered near the large structure located at Duck and Hall of Fame Avenue in Stillwater, OK, is determining how to move its vehicles in the least costly manner to match demand for the next period.

Presently, there are two vehicles in Stillwater, two vehicles in Perkins, and four vehicles in Pawnee (all in Oklahoma).

The vehicles need to all pass through an inspection station – there are two of them available to clean and inspect the vehicles. Upon inspection they will then be sent to where rentals are needed for the next day.

	INSP #1	**INSP #2**
STW	27	20
Perkins	17	41
Pawnee	35	12

Rentals for tomorrow are: four vehicles needed in Ponca City, one vehicle needed in Guthrie, and three vehicles needed in Cushing (also all locales in Oklahoma).

Distances between the cities and inspection stations are given in the table below. Get cars to Ponca City, Guthrie, and Cushing while minimizing the total overall amount of miles traveled by the eight cars. This includes distance from their present location to the inspections stations and from the inspection stations to Ponca City, Guthrie, and Cushing.

	INSP 1	**INSP 2**
PC	14	52
Guthrie	31	12
Cushing	34	17

Let's introduce a new mantra for our supply chain problems - "IN = OUT" (pronounced IN equals OUT). The core model constraints that we have in these distribution problems seek to make sure we account for the goods we are transporting, and don't lose any. Thus, what comes IN to an entity must leave, or go OUT of an entity.

In the previously scenario of 11.2, the simple supply and demand constraints we in fact IN=OUT constraints. We must make sure that we do not send more units (cars) from our supply locations that what we have, and that we send no more than the amount needed at the three demand locations. Thus, IN=OUT (not worrying semantically about balanced and unbalanced for now).

In the present scenario, an intermediate step has been added to the supply chain for TBP Rent-a-Wreck. The cars presently in three locations (STW, Perkins, Pawnee) must first travel to an inspection station (INSP 1 and INSP 2) before being driven to the demand locations (Ponca, Guthrie, Cushing). So, not only must we make sure we appropriately deal with the ends of the supply chain (supply and demand), we must also make sure we properly account for the cars as they travel through inspection. Thus, the cars that come into INSP 1 must equal the cars that leave INSP 1 and are moved to where they are needed the next morning. IN=OUT.

As mentioned before, this type of generic scenario is called a transshipment model. It is actually a specialized case of generalized network models, as was the earlier transportation model. Our initial LP model for these types of problems will build on the matrix approach of earlier models. In preparation for future additional network scenarios, and as a good building block for future scenarios you may face in practice, the last section of this module will show an alternative way of modeling this situation, applicable to all network-related problems (what I call the "From-To" notation, nothing more than a row/column representation on its side "with a twist.") Okay, back to the problem.

11.3.2 Model formulation – Decision Variables and Objective Function

Figure 11.5 pictures (not geographically correct!) the relationships among the different locations in this problem.

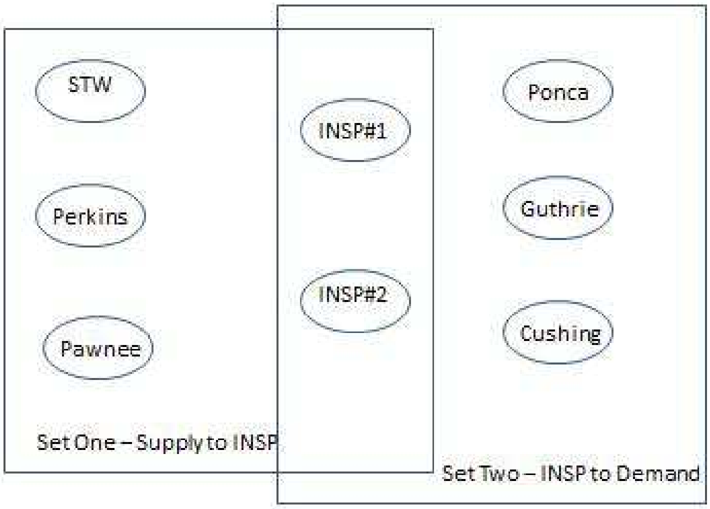
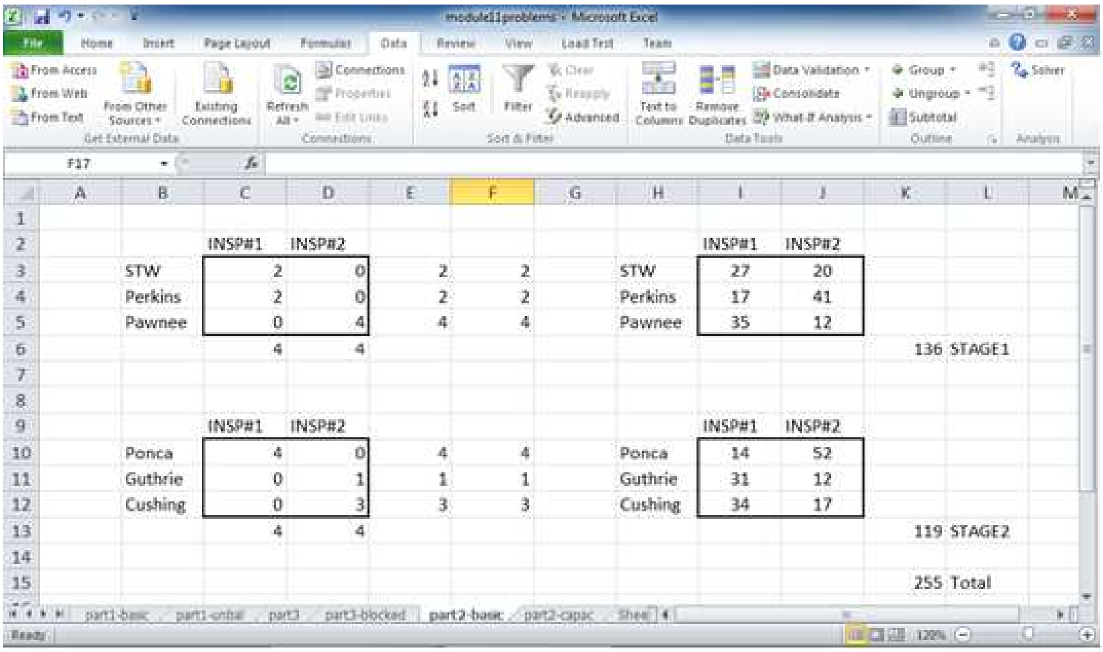


Figure 11.5

Cars must travel from the three supply locations to the two inspection stations – a total of six different paths. That corresponds to the first set of 6 decision variables. Each path represents the number of cars that travel from a specific supply location to a specific inspection station.

Cars must also travel from the two inspection stations to the three final demand locations. Again, this is a total of six different paths – that is, decisions (the number of cars traveling from the inspection station to the demand locations).

Thus, there are a total of 12 decision variables, 6 of each class. They naturally fall into two different matrices. Figure 11.6 depicts the spreadsheet template.



	A	B	C	D	E	F	G	H	I	J	K	L
1												
2			INSP#1	INSP#2					INSP#1	INSP#2		
3		STW	2	0	2	2		STW	27	20		
4		Perkins	2	0	2	2		Perkins	17	41		
5		Pawnee	0	4	4	4		Pawnee	35	12		
6			4	4							136	STAGE1
7												
8												
9			INSP#1	INSP#2					INSP#1	INSP#2		
10		Ponca	4	0	4	4		Ponca	14	52		
11		Guthrie	0	1	1	1		Guthrie	31	12		
12		Cushing	0	3	3	3		Cushing	34	17		
13			4	4							119	STAGE2
14												
15											255	Total

Figure 11.6

The first stage decision variables (supply to inspection) are represented in the matrix C3:D5. The second stage decision variables (inspection to demand) are represented in cells C10:D12. For my own convenience (and it is just a style preference), I have kept the inspection stations as columns. You will see my rationale for that when we implement IN=OUT.

Column E in both matrices is the infamous “row sums.” For instance, Cell E3 has the formula =SUM(C3:D3). All relevant rows have similar formulas.

Row 6 and Row 13 are the column sums. These rows sum up the total number of cars coming into the inspection station (Row 6) and leaving the inspection station (Row 13). Why do both? IN=OUT; coming up in just a bit.

The formulas in Rows 6 and 13 are probably self-evident, but, for example, Cell C6 has the formula =SUM(C3:C5). All other column sum cells have similar formulae.

The corresponding cost matrices related to decision variable tables are found in Columns I and J.

The target cell (objective function value) for this model has been placed in K15. Cell K15 adds together the distance traveled in Stage 1 travel (Cell K6) and the distance traveled by the cars in Stage 2 (Cell K13). Thus, the formula in K15 is =K6+K13.

Cells K6 and K13 calculate the individual stage distances. They are just basic SUMPRODUCT functions – K6 is =SUMPRODUCT(C3:D5, I3:J5) and K13 is =SUMPRODUCT(C10:D12, I10:J12).

11.3.3 Model Formulation – Constraints and Solution

In the model, there are three families of constraints: supply constraints, demand constraints, and IN=OUT constraints. Actually, all are IN=OUT constraints, but the intermediate inspection station constraints are the new additional aspects of modeling in this example.

As with the original first example, supply = demand. So the supply and demand constraints will be equalities. If this is not the case in practice, the same rules apply as discussed earlier with transportation models.

As Column E (Rows 3 through 5) represents the sum of all cars leaving the initial locations, setting E3:E5 = F3:F5 implements the supply constraints. Likewise, Column E (Rows 10 through 12) represents the sum of all cars coming to the desired destinations; setting E10:E12 = F10:F12 implements the demand constraints.

The final step – making sure the number of cars going into the inspection station equals the number that drive out. As mentioned earlier, Row 6 sums the cars coming into the stations; Row 13 sums the cars leaving. Thus, we very easily add C6:D6 = C13:D13 to the model to "conserve flow." IN=OUT.

Figure 11.7 restates the Solver input. Note that when we have multiple areas on our spreadsheet used to represent decision variables, we delineate the different ranges in the Solver dialog box by use of a comma between each range (see the Changing Variable Cells input line).

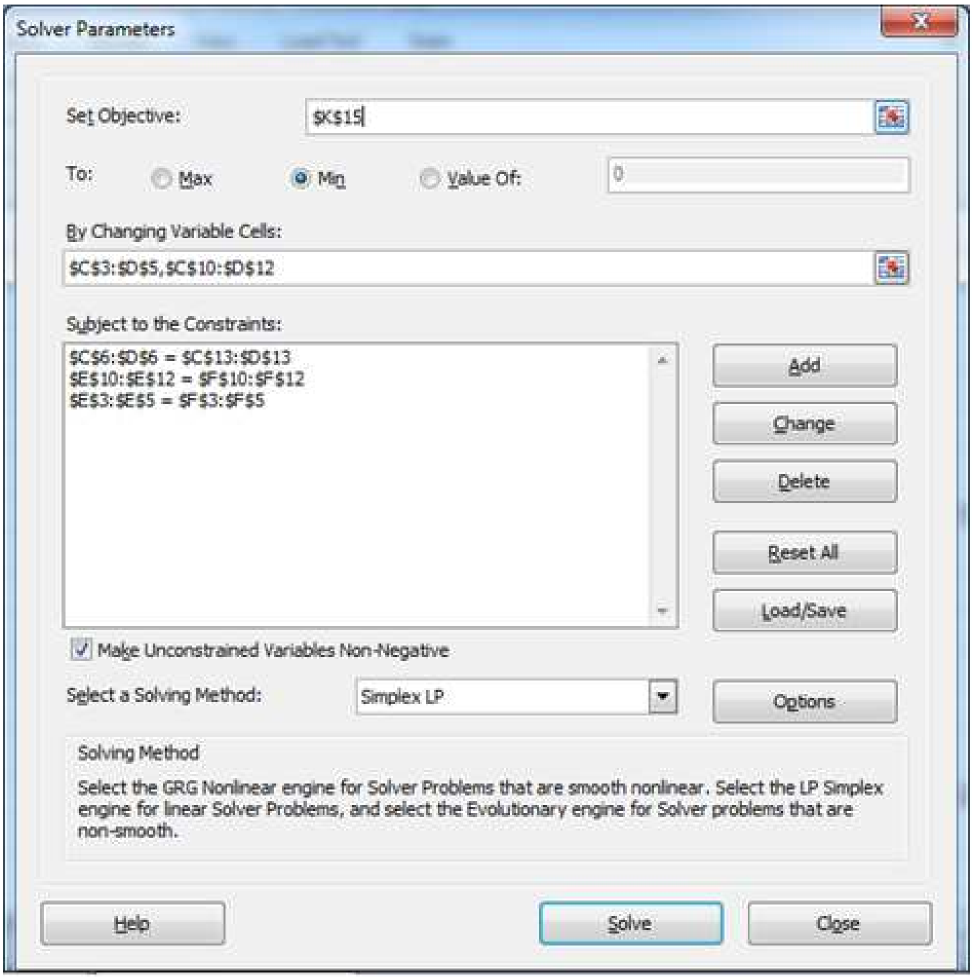


Figure 11.7

Total distance traveled by the cars in our optimal solution is 255 miles. Note how Stillwater and Perkins cars (two each) go to Inspection Station 1, and the four cars from Pawnee go to Inspection Station 2. Inspection Station 1's four cars all go to Ponca City, and the four cars from Inspection Station 2 are split among Guthrie and Cushing appropriately. If you know the geography of North Central Oklahoma, don't try to figure out the latitude and longitude of the inspection stations from the distance matrices. They may not be real places!

11.3.4 Intermediate Stage Constraints – Part 2 of the example.

Let us augment the original problem with capacity restrictions on the inspection stations: How does the optimal shipping of the rental cars change if INSP 1 can handle at most three vehicles and INSP 2 six vehicles?

It is reasonable that any intermediate stage in a supply chain could have its own set of capacity requirements. Because the original determined optimal solution had four cars going to each inspection station, adding this capacity restriction will alter the optimal solution and increase the mileage traveled.

As with any "side" constraints added to a core model, they can be placed anywhere convenient (heck, they can be placed in inconvenient spots if we wanted to, but . . . you know what I mean!). Figure 11.8 shows that we have simply added Row 7 to represent the right-hand side (RHS) (capacity) of the individual inspection stations. The additional constraint added with be Row 6 <= Row 7 (specifically, C6:D6 <= C7:D7). This will constrain the number of cars that pass through each station. Figure 11.9 shows the Solver input needed for this constraint.

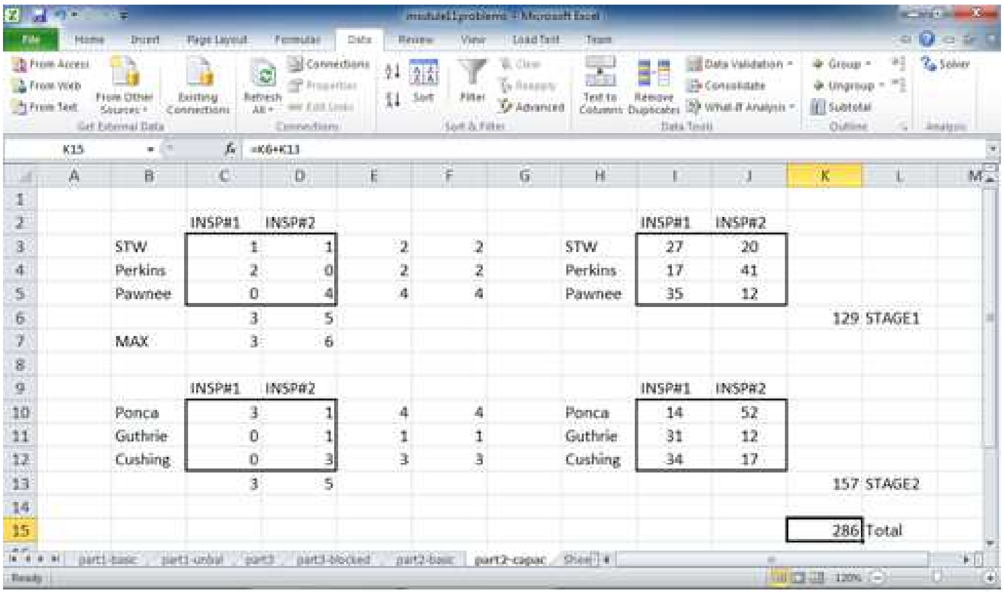


Figure 11.8

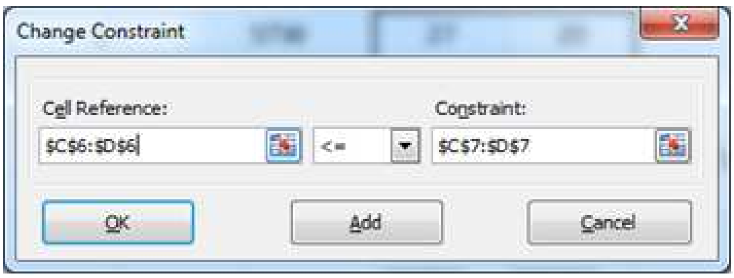


Figure 11.9

We could have also implemented this constraint by adding: C13:D13 <= C7:D7. The LHS value of the two constraints are equivalent because of the IN=OUT constraint for the inspection stations.

Note that the solution changed considerably, adding 31 more miles to the optimal distances and altering the manner in which the cars travel through the supply chain.

11.3.5 Final Comments about Transshipment and Supply Chain Models

The first two examples of this module just scratch the surface of supply chain models. Yes, these are all very simplistic representations of supply chains, but one can begin to add additional parts within the overall framework to create comprehensive models. Some of the practice problems add production issues with supply chain components, again in a simplified manner, to give you a taste of how we are gaining strength in modeling building blocks that we can configure to create bigger and bigger LP applications and save the world.

As the stages of the supply chain increase, one can just keep adding matrices of decision variables and keep remembering the mantra IN=OUT.

If the supply chain is not quite as nice and neat as what we are modeling here – say a situation in which cars could bypass the inspection station and not HAVE to go from Stage 1 to Stage 2 to Stage 3, etc. – one can still model this problem, even using matrices. However, a hybrid approach, one that is more generalizable to all network-type problems, might be helpful as well. In Section 11.5 we will revisit this problem with an alternative formulation. It is not necessary at present to master both core model formulations, but the alternative formulation will be helpful when we look at routing and flow models in subsequent modules.

Now, using the same basic problem setting, let's look at a specialized type of transportation model that also serves as a great lead in to expanding our integer programming experiences.

Reading Material: 11.4 – Example 3 – T.B.P. and the Volatile Earth

11.4.1 Scenario Description

An earthquake hits North Central Oklahoma – 6.3 on the Rick-ter scale (pun intended). And, earthquakes DO occur all the time in Oklahoma.

Anyway, you own three franchises of the T.B.P. Rent-a-Wreck car rental company, located in Ponca City, Guthrie, and Cushing.

As part of your disaster preparedness plan, you always have supervisors on call. When the earthquake hits, the first three supervisors on your call list live in Stillwater, Perkins, and Pawnee.

You need to have one supervisor check on each franchise location. Determine the best assignment of supervisors to franchise location, using the criteria of minimizing total overall miles traveled by the supervisors.

The distance between the supervisor locations and the franchise locations is repeated below.

	Ponca	**Guthrie**	**Cushing**
STW	42	35	25
Perkins	54	26	16
Pawnee	45	62	26

11.4.2 Model Formulation and Solution

This type of problem is termed an assignment problem because we are (not surprisingly) trying to find the optimal assignment of people to locations. It is not really a supply chain scenario in the strictest sense, so why here?

Consider this – we could treat this as a transportation model – we have a supply of one for each supply location (homes of the supervisors) and a demand of one at each franchise location in Ponca City, Guthrie, and Cushing. Thus, it does fit here!

Core assignment problems are specialized transportation models in which supply and demand are equal to 1. They can be balanced and unbalanced, and in some ways it really doesn't matter what we call them. But there is one specific difference – we ARE making assignments, so the decision variables have a different connotation. More on that in a bit.

Figure 11.10 is an EXCEL implementation of this model. It is exactly like Figure 11.1. The only change – the RHS values for supply and demand are all 1. All the model constraints are the same. Please refer back to that earlier module section if any of this is unclear.

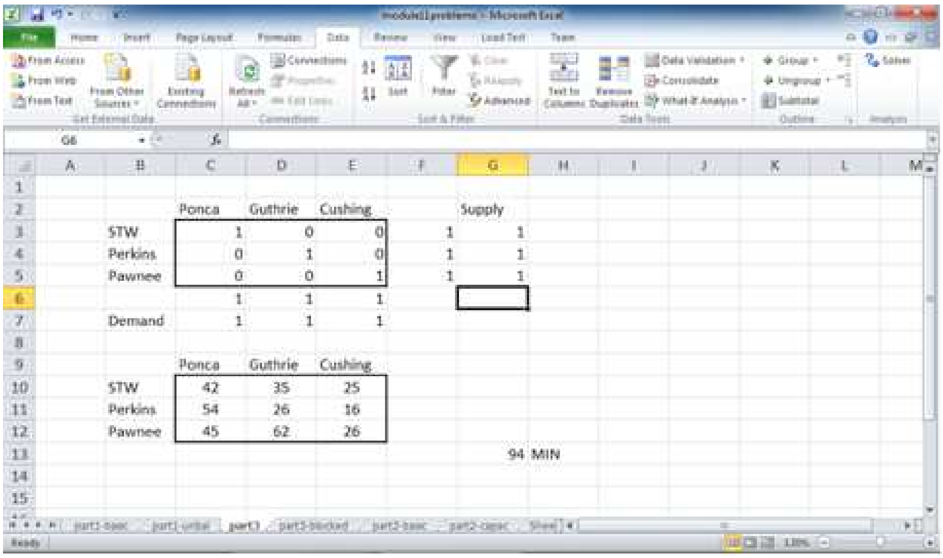


Figure 11.10

The interpretation of the decision variables is a little different though. Cells C3:E5 represent implicit yes/no variables. We see in the optimal solution that the supervisor in Stillwater is assigned to check on the Ponca City location (C3=1), the person in Perkins is assigned to Guthrie (D4=1), and our friend in Pawnee is heading to Cushing (E5=1), with total distance traveled 94 miles. The solution values are all zeroes and ones. A "1" means that the assignment represented by that decision variable is selected as optimal – a "0" means that it is not optimal. For assignment models, we want our solutions to be whole numbers, or in fact, a 0 or a 1. These are binary variables, a subset of integer variables.

For simple models, because of the core constraints, we are assured of either a 0 or 1 value. As we add additional constraints and requirements, and allow multiple assignments, etc. – we may start to see fractional values.

We really want the assignment decision variables to always be 0 or 1. But as we mentioned previously, when we start placing integer (binary) requirements on the Solver, our computational requirements go up dramatically. Just keep in mind that should you be involved in creating a model to do assignments, and you are getting fraction values – feel free to add the constraint shown in Figure 11.11. We will talk more about binary variables in the next module.

Figure 11.11

11.4.3 One More Trick – Unwanted Assignments

Revisit our earthquake situation. Suppose the earthquake makes state highway 33 impassible between Perkins and Guthrie and that the person in Perkins cannot be assigned to Guthrie (and no other routes are impaired). How could we implement this constraint?

In practice, often when assigning people to tasks, for example, there are incompatibilities, certifications, or experience levels that have to be considered and might make some assignments totally undesirable (or unworkable). So, this is a very common occurrence in practice.

There are a number of ways to implement assignment restrictions. One way would be to simply exclude the corresponding assignment decision variable from the model. This is "clunky." Most assignment models use matrix-style approaches and the decision variable area of EXCEL is typically created with one quick swoop of the mouse. Not suggested.

A second way would be to add an explicit constraint to the Solver that says, in our case, D4 = 0.

This forces the model to not assign the person in Perkins to Guthrie. This is workable and as long as there are not a lot of these to do, reasonable.

A third way is shown in Figure 11.12. It is a modeling trick, but very useful. The objective function is used to make the potential assignment very unattractive. An artificially large number is put in the distance table (see 99999 in D11), large because this is a minimization problem. By doing this, the Solver, if the model is feasible, will not use this assignment because of the large cost associated with it.

If our objective was to maximize some criteria and we wanted to exclude a particular assignment, a large negative number would have the desired effect.

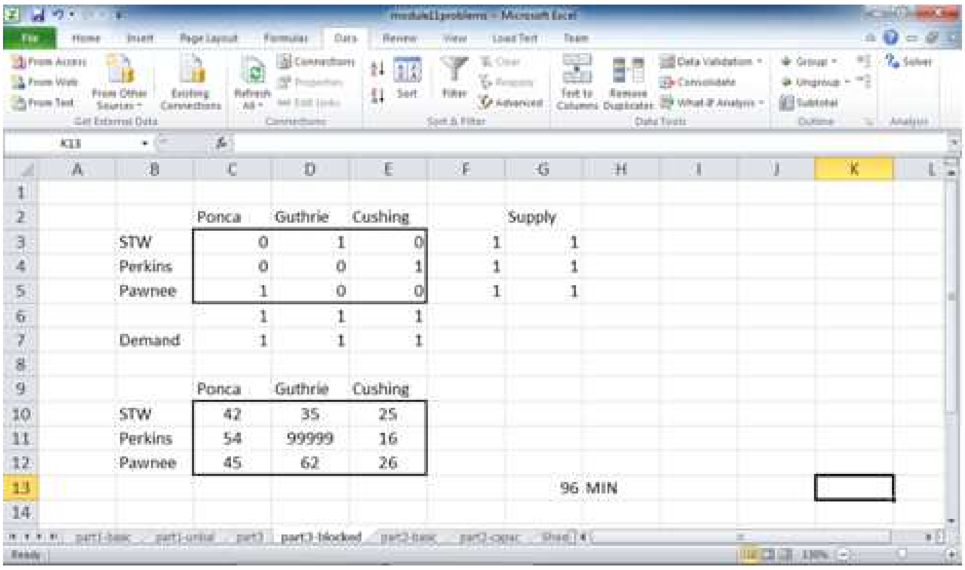


Figure 11.12

The solution is modified (distance increases) as shown in Figure 11.12 when we make the change excluding the Perkins-Guthrie assignment.

Assignment models are some of the most useful core models in practice and to your past colleagues. By the time we've gotten to this point in the semester, many past students have used this type of model to schedule their staff over the holidays, or concessions at ball games, or assigned overtime responsibilities for the weekends. Frankly, the models are very easy to implement, and one can use subjective information as objective function value – part of the human/model insight interface.

Now, the last section revisits general supply chain models and looks at "from-to" modeling styles.

Reading Material: 11.5 – Alternative LP Formulation for Generalized Network Models – Including Transportation and Transshipment

11.5.1 Introduction to the "From-To" Modeling Format

This section can be skipped without loss of ability to attack the practice problems . . . for now. In Module 13, when we look at shortest route and maximum flow models, we will refer back to the model template style used here. However, it can be useful for the situations in this Module, so it is covered in detail now.

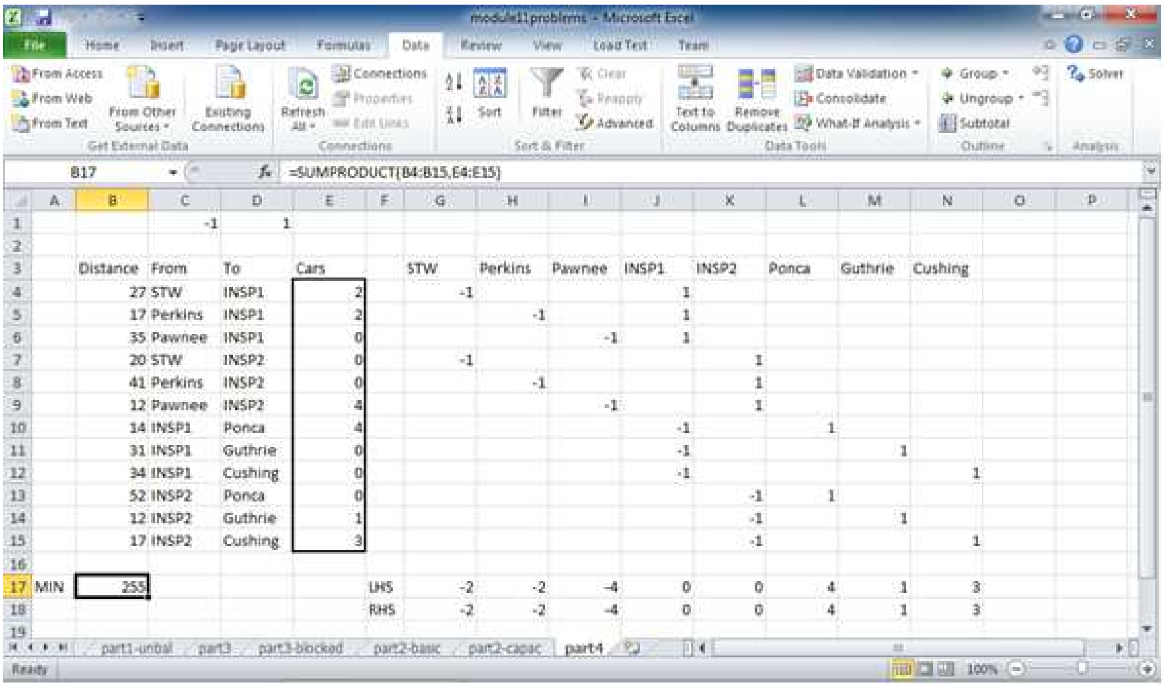
Both transportation and transshipment models can use this alternative implementation style, balanced or unbalanced. We will try to maintain restraint on how much algebra used to explain concepts in this section, realizing that the equivalency between the two different forms of representations may not be apparent without a more detailed examination than what we provide.

This alternative style we will call “from-to” representation. Much like we had “row/column” form and matrix form, now we have “from/to” form. Interestingly, it is similar to row/column on its side (or, you could call it column/row!).

Any situation in which units flow from one node (location) to another can take advantage of this type of EXCEL implementation. Decision variables here represent the number of units being transported between two nodes – so, using network vocabulary, the link between nodes. The T.B.P. Example using inspection stations will be re-solved to illustrate the concepts of this different EXCEL template.

11.5.2 Decision Variables and Objective Function

Each decision in the T.B.P. Inspection Station problem represents the number of cars that travel between a source location (the “from” location) to a destination location (the “to” location). Each row in the spreadsheet will then represent a decision variable – a “from-to” pair. This is shown in the left side of the spreadsheet in Figure 11.13.



B17 =SUMPRODUCT(B4:B15,E4:E15)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			-1	1										
2														
3		Distance	From	To	Cars		STW	Perkins	Pawnee	INSP1	INSP2	Ponca	Guthrie	Cushing
4		27	STW	INSP1	2		-1			1				
5		17	Perkins	INSP1	2			-1		1				
6		35	Pawnee	INSP1	0				-1	1				
7		20	STW	INSP2	0		-1				1			
8		41	Perkins	INSP2	0			-1			1			
9		12	Pawnee	INSP2	4				-1		1			
10		14	INSP1	Ponca	4					-1		1		
11		31	INSP1	Guthrie	0					-1			1	
12		34	INSP1	Cushing	0					-1				1
13		52	INSP2	Ponca	0						-1	1		
14		12	INSP2	Guthrie	1						-1		1	
15		17	INSP2	Cushing	3						-1			1
16														
17	MIN	255				LHS	-2	-2	-4	0	0	4	1	3
18						RHS	-2	-2	-4	0	0	4	1	3

Figure 11.13

Column E is the placeholder for the decision variables/changing cells. Column C notes the "from" location and Column D the "to" location. Column B is the mileage for the link. There are 12 rows, corresponding to the 12 decision variables – the different combinations possible of moving vehicles.

The objective function is calculated in cell B17 – as usual, a sumproduct of the link distance multiplied by the number of cars shipped over the link. The formula: SUMPRODUCT(B4:B15, E4:E15). This will be the target cell to be minimized.

Going through the formalism of listing the FROM and TO nodes (typed as labels) will assist us in constructing the constraints.

11.5.3 A Constraint for Every Node, a Chicken in Every Pot, and a Car for Every Renter

In the core problem, there is one constraint for each node. For supply nodes (locations), it limits us to logically supplying no more than what is available. For demand nodes (locations), the constraints limit us to meet the specified demand. And for the intermediate nodes, in this case the Inspection Stations, the constraints are our friends IN=OUT.

These constraints can be created using some simple rules, given the manner in which the decision variables are defined. These rules are algebraically based and will be further clarified at the end of the section.

The columns in the spreadsheet correspond to model nodes. Note Row 3 from Columns G through N – a column for every location. We are going to use these columns to create our eight constraints (three supply locations, two intermediate inspection stations, and three demand locations).

For each row, constraint coefficients will be created in each column using the following rule. If the heading of a column (i.e., the constraint of interest) matches the "from" node of the decision variable, enter "–1." If the heading of a column (i.e., the constraint of interest) matches the "to" node of the decision variable, enter "1." If the heading of a column (i.e., the constraint of interest) does NOT match either the "from" node or the "to" node of the decision variable, enter "0" or leave it blank (same effect).

Look at Row 4, corresponding to the decision variable that represents the number of cars that go FROM Stillwater (STW) TO Inspection Station 1 (INSP1). In the STW column (Column G), there is a "from" match – thus, –1 is entered. In the INSP1 column (Column J), there is a "to" match, so 1 is entered. All other entries remain blank (thus, 0).

We repeat this for all rows (you can validate that we have done it correctly). Yes, this has an algebraic basis.

The key to victory, and where this all turns into constraints, occurs in the bottom of the spreadsheet, Row 17 (LHS) and Row 18 (RHS). Row 17 is a SUMPRODUCT function much like the one in Column B that calculates the distance traveled. For instance, the formula in G17 is =SUMPRODUCT($E4:$E15, G4:G15). As before, the judicious use of the absolute address allows us to type the formula only one time and copy it through column N.

Row 18 is simply a value – for supply nodes, it is the number of cars present (with a negative sign), for demand nodes, the number of cars needed, and for intermediate nodes, 0. Then, the only constraints in the core model: G17:N17 = G18:N18. Figure 11.14 shows the Solver input.

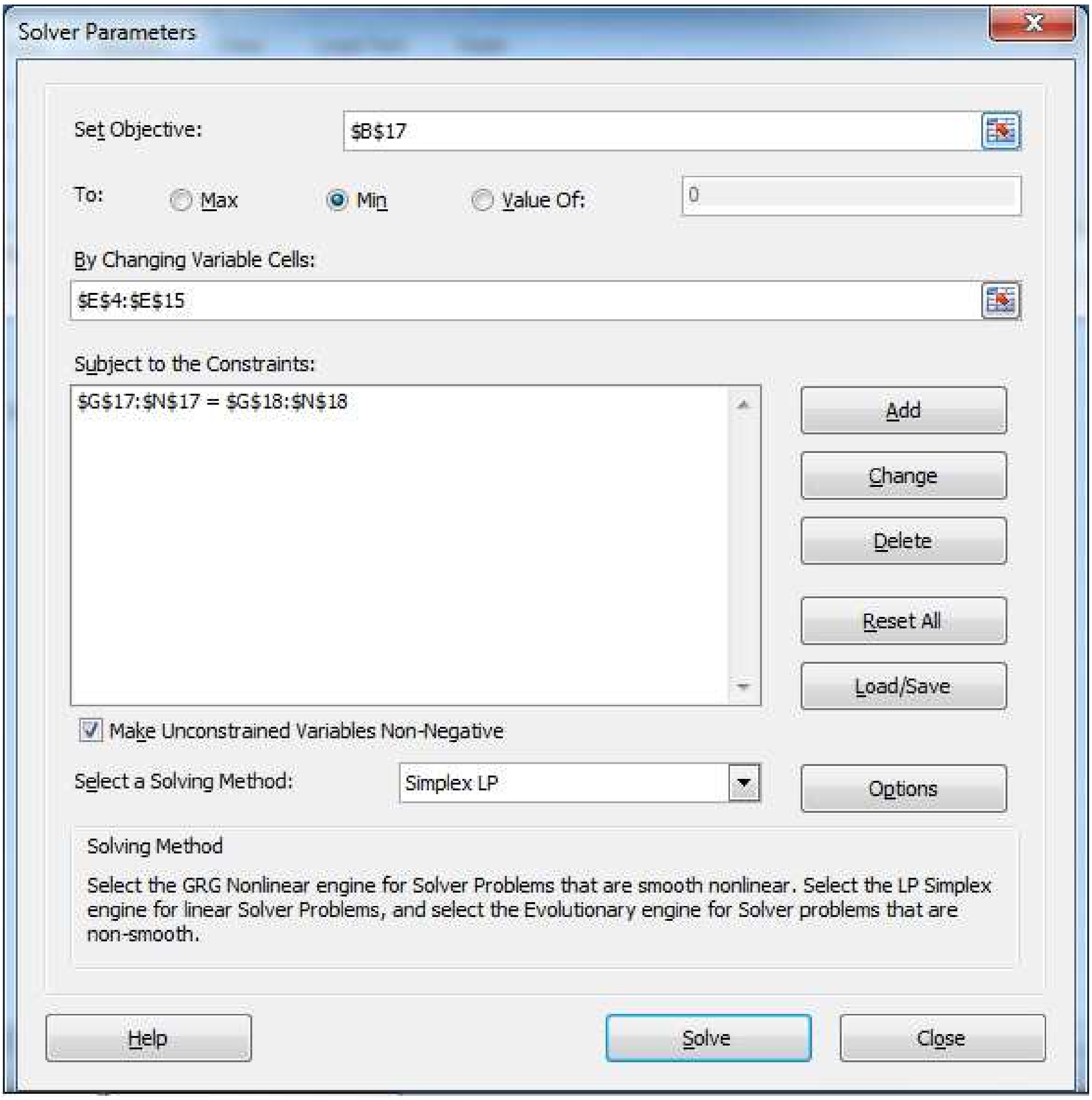


Figure 11.14

Figure 11.13 was solved, and, as one would expect, the same optimal solution was found as in Figure 11.4. It is easier in this format to see that the model consisted of 12 decision variables and 8 constraints.

This template can handle unbalanced supply and demand as well. Great care needs to be specified though due to the unique way which supply is coded (as negative numbers). For instance, if we had extra supply, the constraints for Columns G, H and I would be GREATER-THAN constraints. This would allow smaller negative numbers in the LHS than is found in the RHS – and smaller negative numbers means "greater than."

11.5.4 Why Does This Work? I'm Not Sure I Like My Chicken in Every Node!

Basically, the reader can trust me that this works and go on. But what if it seems like some random algorithm? Read further.

This method of setting up the spreadsheet model is based on the premise of "I'd like to have rules that I never break, that I can apply every time, and that under duress, are repeatable." Sounds like a good golf swing. The negative number for supply nodes might be our biggest 'Huh?" at present. I opted to do supply nodes that way, rather than change how we code the from-to coefficients. Repeatability.

First, let's look at what is going on in the constraint for INSP1 (Column J). The sumproduct of the decision variables times the coefficients in Column J (the formulas in Row 17) is equivalent to the following algebraic statement:

STW/IN1 + PER/IN1 + PAW/IN1 – IN1/PON – IN1/GUT – IN1/CUS

(I think my decision variable names are understandable). The first three terms are Rows 4 through 6, the last three Rows 10 through 12. When we create the constraint by taking Row 17 = Row 18, we get:

STW/IN1 + PER/IN1 + PAW/IN1 – IN1/PON – IN1/GUT – IN1/CUS = 0.

If we rearrange the terms and move the negative coefficient terms to the RHS, our equation becomes:

STW/IN1 + PER/IN1 + PAW/IN1 = IN1/PON + IN1/GUT + IN1/CUS

Which, in words, says: "All cars coming into Inspection Station 1 must equal all cars leaving Inspection Station 1, or, even more simply, "IN=OUT."

See, there is a method to the madness.

Let us look at a demand node (Cushing, Column N). Doing the same analysis (mimicking the sumproduct in Row 17, and setting it equal to Row 18), we get:

IN1/CUS + IN2/CUS = 3.

Which is exactly the demand constraint needed for the model.

Finally, a supply node (say Stillwater, Column G). Mimicking the sumproduct in Row 17 and setting it equal to Row 18 we get:

–STW/IN1 – STW/IN2 = –2

This is equivalent to:

STW/IN1 + STW/IN2 = 2

Which is exactly the supply constraint needed for the Stillwater node.

The odd-looking negatives in supply nodes occur as a result of how we systematically code from and to nodes. This small little oddity is well worth the consistency, repeatability, and ease of coding our constraints the same way every time for every node – in my opinion of course.

Thus ends the algebraic discussion. And the description of the alternative format – the "from-to" format.

Reading Material: 11.6 – Conclusion

This module covered supply chain models and assignment models. Assignment models will help lead us into the next module, in which we will further expand our modeling capability by examining different uses of binary variables, integer variables that can take only the values of 1 or 0.