OVERCOMING SPREADSHEET RISK IN SUPPLY CHAIN MODELING

by

Michael R. Galbreth
University of South Carolina

and

Larry J. LeBlanc
Vanderbilt University

INTRODUCTION

In recent years, organizations have responded to opportunities for cost reductions and increased sales by globalizing their supply chains more than ever before (Guinipero et al. 2008). The result is increasingly complex supply chain networks, and many firms are using optimization modeling to support both tactical and strategic supply chain decision-making (Soudhi 2003). A wide range of firms apply optimization modeling to supply chain problems as varied as routing, vehicle and crew assignment, facility location, transportation mode selection, timing of shipments, and inventory allocations. The potential for optimization to add value to businesses has increased in recent years as advances in software and hardware have produced staggering improvements in model solution times. For example, Bixby (2007) notes that from 1988 to 2004, the time required to solve a linear program decreased by a factor of approximately 5.28 million, and further improvements are occurring every day. To put Bixby's observation in perspective, consider that a model that took two months to solve to optimality in 1988 can be solved in about *one second* today.

The ubiquity of spreadsheets, as well as their intuitive nature and familiarity among managers, make them an appealing choice for modeling complex problems. Although many managers believe that large-scale supply chain optimization cannot be implemented in spreadsheets, several very powerful solution add-ins are available for spreadsheet modelers. With these add-ins, which are both appealing to users and very convenient for model developers (Fourer 2007), spreadsheets such as Microsoft Excel can be used to develop very complex supply chain models (Smith 2003). A recent issue of the widely read management science journal, *Interfaces*, is devoted to "Spreadsheet Applications of Management Science and Operations Research," testifying to the high level of interest in this area.

Examples of spreadsheet-based models supporting the management of supply chains can be found in a wide range of industries. Glickman (1991) used spreadsheets in the choice between various tank truck configurations for the bulk transport of flammable chemicals. More recently, Lipman and Delucchi (2006) describe how a spreadsheet model was used to analyze the lifecycle costs of hybrid versus conventional vehicles. Other spreadsheet optimization models have addressed tactical problems such as the loading schedule of an automated storage and retrieval system (Jacobs et al. 2000), as well as strategic decisions regarding large-scale networks for global sourcing, production, and distribution (LeBlanc et al. 2004; LeBlanc and Galbreth 2007). Schuster et al. (2000) optimized recipes and supply chain flows in a process industry context using a spreadsheet model. Smith (2003) describes three applications of spreadsheets for supply chain analysis: direct shipment decision-making for a major food manufacturer; a warehouse consolidation decision for a pharmaceutical firm; and the assignment of a distribution center to a new market for a major grocery store chain. In each of these cases, key supply chain decisions were made based on a complex spreadsheet model. Another interesting example is found in Botter and Fortuin (2000), where a spreadsheet model is used to determine optimal stocking points of services parts throughout a supply chain. This model, which was used by a large multinational firm, addressed over 50,000 items. The authors note that the requirement that "the tool to be developed had to be simple and easy to use…resulted in a spreadsheet application" (p. 662).

Despite the advantages of spreadsheets (or in some cases, as we describe below, *because* of these advantages), there are significant risks associated with spreadsheet models. Spreadsheet risk is defined as the chance of adverse operational or financial consequences due to erroneous creation, maintenance, and/or use of spreadsheet models. Such errors arise because the intrinsic complexity of optimization models is beyond the expertise of many of the managers that use them. Spreadsheet risk is distinct from simple data entry mistakes, which are possible in virtually every aspect of business and can never be completely eliminated. The specific risks we describe can be present in a range of spreadsheet-based applications. They certainly exist in contexts other than supply chain management and logistics. However, the fact remains that spreadsheet risk mitigation is an important and relevant topic for today's supply chain manager.

A number of authors have recently discussed non-technical methods for addressing spreadsheet risk—Cummings (2008b) cites studies showing that more than 86 % of spreadsheets contain errors. In spite of this, he notes that a Deloitte poll shows only 42 % or respondents said spreadsheet risk is part of a periodic risk assessment. He recommends an inventory of spreadsheets with an estimate of each one's level of risk and comprehensive training programs. Kugel (2008) notes that past studies show that "any spreadsheet probably contains at least one error—even those that have been checked for them (which too few people do)." He even cites advice "to replace spreadsheets," noting that desktop spreadsheets pose the risk of introducing damaging errors.

In spite of the advice against using spreadsheets, we believe that they will remain pervasive in nearly all organizations. As a result, in this article we discuss how supply chain models can be developed to leverage the advantages of spreadsheets while mitigating the substantial risks inherent in a spreadsheet-based model. Our approach does not require a departure from the comfortable user interface of Microsoft Excel, nor does it require the purchase of expensive third-party software. In fact, it accomplishes all risk mitigation by building on existing Excel capabilities.

ADVANTAGES OF SPREADSHEETS FOR SUPPLY CHAIN MODELING

The advantages of spreadsheet modeling for managers can be demonstrated by contrasting it with a traditional mathematical model. Figure 1 contains two models of the exact same problem—the minimization of transportation costs between factories and warehouses. In the top portion of that figure, this is formulated in a spreadsheet, while the bottom portion contains an algebraic formulation similar to what would be entered into mathematical optimization software. In the spreadsheet model, we use the convention that cells with double-black borders contain data, and heavy-border cells represent the management decisions (decision variables or "changing cells" in Excel terminology). The data cells are the following:

- C3:I6 give the transportation costs per units shipped between each factory and warehouse
- L10:L13 give the factory capacities, and
- C16:I16 give the demands at each warehouse.

The objective is to minimize the total transportation cost while meeting all demands without exceeding the factory capacities. As many supply chain managers are aware, such transportation optimization models are becoming increasingly valuable given the recent rise in fuel costs. The cell with the overall cost to be minimized contains a formula—Excel's SUMPRODUCT function in B19. This function multiplies units shipped by unit transportation costs for each factory/warehouse pair. This model is ready to be sent to Excel's embedded optimization engine, Frontline Systems' Solver (www.solver.com)[a]. In the algebraic formulation in Figure 1, the optimization model is written by specifying the objective function, variables, and constraints using mathematical notation. A comparison of these two models clearly demonstrates the fact that spreadsheets are a more appealing and accessible tool for most managers. While many would be comfortable with the spreadsheet approach, it is likely that non-technical managers would find the algebraic formulation much less intuitive.

[a] Although the Solver delivered free with Excel is limited in terms of the sizes of problems it can solve, Frontline sells very powerful "premium" versions that can solve optimization models that are virtually unlimited in size.

FIGURE 1

A SPREADSHEET SUPPLY CHAIN MODEL AND AN EQUIVALENT ALGEBRAIC FORMULATION

Unit Shipping Costs

Factory	Atlanta	Pitts-burgh	Kansas City	St. Louis	Memphis	Denver	Phoenix
	Warehouse						
Shanghai	$ 6.44	$ 8.04	$ 9.00	$ 6.61	$ 6.16	$ 5.37	$ 6.29
Jakarta	$ 7.91	$ 5.24	$ 6.57	$ 7.60	$ 9.71	$ 5.27	$ 7.87
Tijuana	$ 6.45	$ 8.31	$ 9.93	$ 6.62	$ 7.16	$ 6.29	$ 8.04
Kuala Lumpur	$ 9.66	$ 7.89	$ 7.82	$ 9.48	$ 8.10	$ 8.49	$ 6.30

Shipping Decisions

Factory	Atlanta	Pitts burgh	Kansas City	St. Louis	Memphis	Denver	Phoenix	Total Sent From		Factory Capacities (000)
	Warehouse									
Shanghai	0	0	0	600	300	900	0	1,800	<=	1,800
Jakarta	0	800	200	0	0	0	0	1,000	<=	1,000
Tijuana	300	0	0	0	0	0	0	300	<=	1,900
Kuala Lumpur	0	0	500	0	0	0	600	1,100	<=	1,500
	300	800	700	600	300	900	600			
	=	=	=	=	=	=	=			
Demands (000)	300	800	700	600	300	900	600			

Total sent from Shanghai =SUM(C10:I10)

Total sent to Phoenix =SUM(I10:I13)

Total Cost ($000)
$ 25,778

Minimize total shipping costs: =SUMPRODUCT(C3:I6,C10:I13)

Double border cells contain data that can't change

Heavy border cells are changing cells (management decisions)

Cell comment identifies the cell to optimize (B19)

Notation	
i	index for factory
j	index for warehouse
m	number of factories—i = 1, 2, …, m
n	number of warehouses—j = 1, 2, …, n
x_{ij}	amount shipped from factory i to warehouse j
c_{ij}	transportation cost/unit between factory i and warehouse j
S_i	supply at factory i
D_j	demand at warehouse j

Optimization Model

$$Min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} \le S_i \qquad i = 1,2,\ldots,m \ \text{(supply constraints)}$$

$$\sum_{i=1}^{m} x_{ij} = D_j \qquad j = 1,2,\ldots,n \ \text{(demand constraints)}$$

SPREADSHEET RISK

Many studies have noted that end-user developed spreadsheets can contain extremely high error rates (Kreie et al. 2000; Panko and Sprague 1999). Anecdotal evidence of the potential damage of spreadsheet errors is widespread—the European Spreadsheet Risks Interest Group (EuSpRIG) maintains a Web site devoted to some of the most egregious ones (http://www.eusprig.org/stories.htm). The disturbing frequency and severity of spreadsheet errors motivated us to develop and formalize the mitigation techniques described in this article.

When spreadsheets are used for optimization, the end-users of the models themselves (i.e., managers) often engage in their actual development and/or maintenance. One explanation for the high rate of spreadsheet errors is the fact that, although most managers are generally familiar with Excel, their skills vary widely. In some cases, the familiar nature of Excel may cause managers to be *too* confident, and some convenient features such as "copy-and-paste" might actually make errors more likely, especially when the end-user developer has not been trained in basic skills such as testing and documentation of complex models. When a manager is developing an Excel-based supply chain model, the result is often a complex (possibly convoluted) spreadsheet, and the resulting model can be inefficient and/or inaccurate, creating the potential for errors with serious consequences.

If a technical expert develops the spreadsheet and then turns it over to a manager, the manager may struggle to update the model correctly as business conditions change. In fact, such ignorance of spreadsheet modeling details has been used as a defense in legal proceedings, such as the case where a defendant was charged with violating limits on the release of hazardous materials into the environment. The mistake was driven by erroneous spreadsheet formulas, and the defendant stated that it was "prepared to demonstrate at hearing that these formulas were in a typical spreadsheet format, that once these formulas are developed, they are opaque to the users of the spreadsheets" (U.S. EPA 2008).

Jafry, Marrison, and Umkehrer-Neudeck (2008) discuss spreadsheet risk in the banking industry and describe an Excel-based centralized model that incorporates the management science tool of simulation. Using a complex Excel model, they simulate cash flows from a portfolio of loans given a randomized set of potential macroeconomic outcomes. In describing the risks of such spreadsheet models, Jafry, Marrison, and Umkehrer-Neudeck (2008) note that with a decentralized model, human errors, such as running an insufficient number of simulation iterations, can create misleading results. They also note the danger of basing decisions on untested spreadsheets stored on individual users' PC's.

Although the use of decentralized, unmonitored spreadsheets is extremely common, this approach involves a dangerous risk of errors and even fraud. One of many examples is the case where a controller generated financial statements from a spreadsheet on his personal PC. The following is a direct quote from the SEC litigation documents of that case: "California Amplifier's financial statements were generated from a spreadsheet maintained by Kusatzky on his own desktop computer. This spreadsheet was wholly separate from the company's accounting system. California Amplifier had no internal control to verify the accuracy of the information in his spreadsheet, such as by tying the information to the company's general ledger. As a result, once Kusatzky improperly created lower expenses on this spreadsheet, he was able to falsify California Amplifier's financial statements because Kusatzky's fraudulent spreadsheet was incorporated directly into the company's quarterly filings with the Commission" (U.S. SEC 2008).

Supply chain optimization modeling in spreadsheets has its own significant risks, as we describe in the following subsections. The drivers for these risks go beyond simple careless or typographical errors (although these are common). The risks below arise from the inherent complexity and sophistication of optimization modeling. While spreadsheet modeling involves many risks, we focus on the ones that we perceive to be both common and significant in their potential consequences. These serve to highlight the dangers of allowing non-technical managers to use optimization models without carefully managing spreadsheet risk.

Risk: Overwriting Formulas

It is common for supply chain optimization models to be run repeatedly as the business environment changes (e.g., customers are gained or lost, demand forecasts are updated, unit costs change, capacity is expanded). A major advantage of spreadsheets for optimization modeling is that it is easy to make modifications by entering new data directly into a previously-developed model. For example, in Figure 1 a user might enter 1,600 into cell L13 to reflect an increase in Kuala Lumpur's capacity. Or she might enter 600 in I10 and 0 in I13 to assess the cost impact of meeting all Phoenix demand from the Shanghai factory.

Such parameter updates and what-if analyses are perfectly acceptable practices. Indeed, the ease with which they can be accomplished is one of the major advantages of spreadsheet models versus algebraic ones. However, this ease can lead to serious and insidious errors. For example, the user should never enter data in any cell in J10:J13 or C14:I14 of Figure 1. These cells contain formulas—row sums and column sums, respectively. The formulas are passed to Solver as constraints that shipments out of factories can't exceed capacities and shipments into warehouses must equal demands. If a user does enter data here, overwriting the formulas, then the optimization model will be corrupted—Solver's solution will ignore the intended constraint. This corruption may not be obvious to the user, whose familiarity with spreadsheets in general might create a false sense of confidence in her model updates. It is important to note that most non-technical managers would not dare to modify an *algebraic* model such as the one at the bottom of Figure 1. However, because of the intuitive nature of spreadsheets, non-technical managers often do modify spreadsheet models without a complete understanding of their actions.

Risk: Algorithmic Parameter Errors

In some cases, a user might unwittingly obtain a suboptimal solution to an optimization model by failing to specify critical, but somewhat esoteric, algorithmic parameter settings. For example, although many supply chain optimization models are linear programs, a linear model is not the default setting in some solution engines for large-scale problems (e.g., Frontline Systems' enhanced "Premium" Solver). Thus, it is possible that the solution engine could use an inappropriate non-linear optimization algorithm (which can be much less accurate) to solve linear models, and a solution that is far from optimal could result. Another example is the "tolerance" setting in Frontline's Solver for optimization models with integer restrictions. This specifies whether the exact or approximately optimal solution should be found, and the default setting is for *approximate* (within 5 % of optimality) solutions. An experienced professional would know when to adjust both of these default settings (i.e., if the model is linear and/or if an *exact* solution is required). However, for a non-technical user, errors in such technical details are quite likely, and the result might be that important managerial decisions are made based on flawed analysis.

We illustrate the risk of incorrect algorithmic settings using the supply chain setting depicted in Figure 2, which shows a network of 20 suppliers, four manufacturing sites, and 100 customer areas. This supply chain is similar to that of Mendocino Forest Products (MFP). MFP and its affiliates produce treated wood products for decks, fences, landscaping, and other specialty wood products. MFP had extended its distribution and product range through acquisition, and the firm used Excel-based LP models to optimize its expanded supply chain. Using disguised data and number of shipping sites from MFP, the cost minimizing shipments for the network in Figure 2 are determined. Since the model is linear, the appropriate option within Solver has been specified. As shown at the top of Figure 3, the minimal cost of meeting all customer demands is $128,000 per month.

Suppose that a company is considering selling manufacturing site M4. A key component of the analysis for this decision is estimating the increase in shipping costs that would result. The same spreadsheet-based supply chain model can be used, with minor modifications, for this analysis. When analyzed correctly, the optimal shipping cost without M4 in the supply chain is $149,000, implying a cost increase of $149,000 – 128,000 = $21,000 per month.

When no spreadsheet risk controls are in place, it is possible that a non-technical decision maker might solve this optimization model without specifying the correct algorithmic parameters. To demonstrate the potential impact of such an error, we solved this model with state-of-the-art Solver software *without* specifying that the model is linear. Solver returned a shipping cost with M4 excluded of $209,868, implying a monthly cost increase of over $80,000. With this erroneous result, which significantly exceeds the true cost increase, a manager might incorrectly conclude that the increased shipping costs are too large to justify selling M4. Thus, the organization would miss the opportunity for an attractive asset sale because of an avoidable error in the spreadsheet model.

Risk: Model Expansion Errors

Figure 4 illustrates another kind of spreadsheet risk in optimization modeling. The model in that figure minimizes costs to meet demands at a number of warehouses using production at four capacity-constrained factories. Suppose that the manager using the spreadsheet in Figure 4 has just updated the model to include a proposed new warehouse in Chicago (see column J). This required many careful updates to the spreadsheet. Specifically, all data (costs and demands) related to Chicago had to be specified, and new changing cells indicating shipping decisions to Chicago had to be defined. In addition, all formulas in the spreadsheet and settings in Excel's Solver dialogue box needed to be updated.

FIGURE 2

A HYPOTHETICAL SUPPLY CHAIN

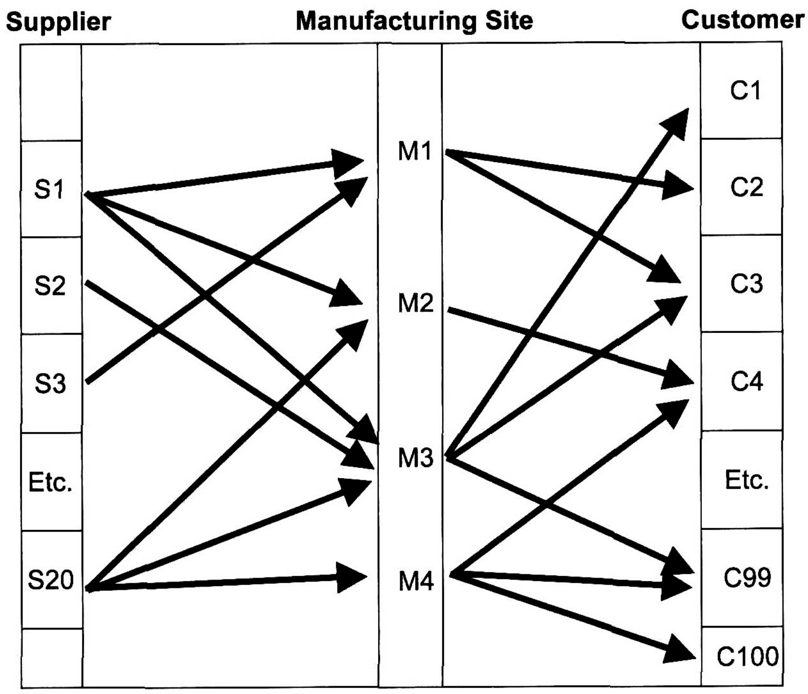


FIGURE 3

PARTIAL SCREEN SHOT OF OPTIMAL SHIPMENTS IN MFP'S SUPPLY CHAIN MODEL

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Total Shipping Cost				$	128,000		Minimize total shipping costs: =SUMPRODUCT(C6:F....)					
2													
3			Shipping Costs per Unit							Units Shipped			
4			Manufacturing Site							Manufacturing Site			
5			M1	M2	M3	M4				M1	M2	M3	M4
6	Supplier	S1	186	130	145	104		Supplier	S1	0	0	0	0
7		S2	171	145	122	130			S2	0	0	0	0
8		S3	114	138	132	117			S3	0	0	0	0
9		S4	192	146	188	106			S4	0	0	0	0
10		S5	187	169	114	117			S5	0	0	0	0
11		S6	121	191	108	128			S6	0	0	0	0
12		S7	197	189	166	126			S7	0	0	0	0
13		S8	173	167	143	125			S8	0	0	0	0
14		S9	137	117	124	122			S9	0	0	0	0
15		S10	198	111	103	90			S10	0	0	25	952
16		S11	146	106	154	118			S11	0	13	0	0

It is interesting to note that such a seemingly simple change impacted the formula for the total shipping cost, the changing cells range, and many constraints of the model. Excel's Solver dialogue box also required multiple updates (see Figure 5). However, there is a serious error in Figure 4—the user failed to update the cost formula in cell B19, which incorrectly ignores column J. In spite of this, no error message will be received, and Solver will inform the user that the optimal solution has been found. This is, of course, true—Solver has optimized the model given. However, the user error in failing to update a single cell, B19, results in a problem that does not fully capture the costs of the new supply chain problem, and any decisions based on this model will ignore the costs of shipments to the Chicago facility. Had the costs of the Chicago shipments been included in the analysis, the correct total cost in cell B19 would be $28,292(000). This is $2,514(000) more than the cost indicated by the erroneous formulation in Figure 4. Clearly, this avoidable spreadsheet error could cause the organization to make a poor investment if it based its decision on the erroneous model.

FIGURE 4

UPDATED SPREADSHEET MODEL WITH ERROR

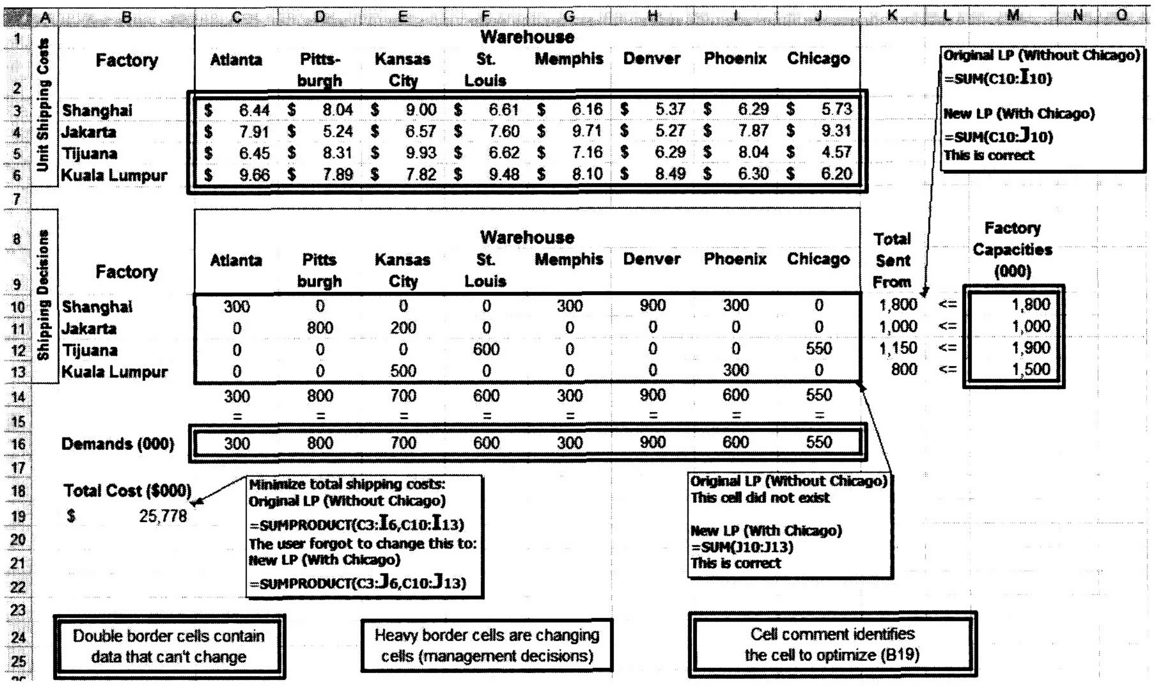
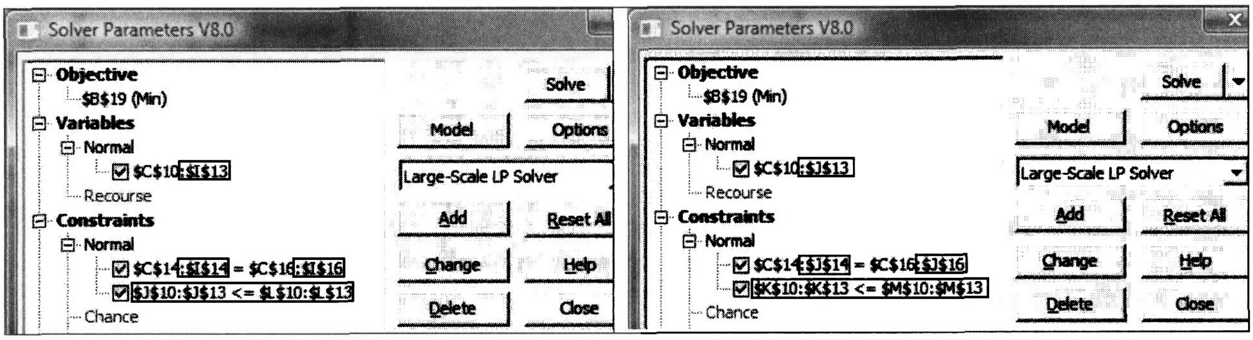


FIGURE 5

PREMIUM SOLVER DIALOGUE BOX BEFORE (LEFT) AND AFTER (RIGHT) THE ADDITION OF CHICAGO



SPREADSHEET RISK MANAGEMENT

As the above examples demonstrate, spreadsheet errors can be difficult to avoid, even in relatively simple supply chain models. It is easy to imagine how much more mistake-prone models can be as they become more complex than these examples. In this section, we describe procedures that we have developed for the control of spreadsheet risk. The first is an automated tool to facilitate the critical task of thorough spreadsheet documentation, which can reduce errors such as overwriting formulas. The second is a method for controlling supply chain model updates using an automated, centralized process. This second technique is simple to implement and can greatly mitigate risks such as algorithmic parameter and model expansion errors, since it is driven by macros that will not "forget" any of the many necessary steps in updating a supply chain optimization model.

Risk Mitigation: Spreadsheet Documentation

Mistakes such as overwriting cell formulas are driven by a lack of understanding of the structure of the spreadsheet model. This type of misunderstanding can be minimized with thorough documentation. One effective approach to spreadsheet documentation is inserting detailed comments for cells containing key formulas in the model. Such documentation is essential not only for supply chain optimization models, but also for virtually any advanced Excel analysis. However, it is tempting for a developer to omit this documentation, since she already understands how to use the model. All of the effort in documentation is borne by the developer, but all of the benefits go to other users of the spreadsheet. Although adding cell comments for documentation is tedious, the following simple macro can greatly streamline the process[b]. The user simply selects several cells and calls this macro. The macro prompts the user for a description of each cell in turn (the user can ignore this if desired) and then places that description and the cell's formula in cell comments. The macro repeats this automatically for every cell that was selected by the user.

FIGURE 6

CELL COMMENT VBA

```
For Each Cell in Selection        'Loop over all cells in the user's selection
    Explanation = InputBox("Enter description of cell" & Cell.Address)
    Cell.AddComment               'Add a new comment to this cell
    Cell.Comment.Text Text: = Explanation & vbNewLine & Cell.Formula
    Next Cell                     'Comment has user's explanation and the cell's formula
```

Figure 7 illustrates a spreadsheet before and after thorough documentation. In this case, the spreadsheet was documented by including border conventions and then running the simple macro shown above. In the undocumented spreadsheet, the source of the costs in cells B3:G8 is not obvious. In fact, these costs are the result of complex calculations, as shown in the documented spreadsheet. Without such documentation, it is more likely that a user will mistakenly enter new costs directly into B3:G8. The documented spreadsheet warns against this, and explains where updates should occur. Other formula cells are similarly described, making the model much easier to understand and modify.

Risk Mitigation: Access Control

To address risks such as the formula overwriting, parametric settings, and model expansion examples described above, we present a novel approach for implementing the requirement that all optimization models are solved using centrally-located, access-controlled spreadsheets, instead of allowing every individual to maintain his or her own model. Such a centralized approach to managing spreadsheet use in an organization is known to be effective in combating spreadsheet risk in general (Cummings 2008a). In a supply chain optimization context, these central spreadsheets would contain advanced solution engines. Since Excel's built-in Solver cannot solve the large models required for most real-world problems, a manager *must* access a central spreadsheet in order to solve the

[b] Your technical support personnel can easily install this macro so that it is available whenever you open Excel.

optimization problem. The central spreadsheets are maintained by technical experts, and individuals can only pass data describing the optimization model to a central spreadsheet and receive the optimal solution from it. This controls the risk of errors while still allowing managers to use and update the parameters in the model.

Although several packages exist to manage centralized (or "server-based") spreadsheets, these are typically very costly. Kugel (2007) notes that some are so expensive that they would be difficult to justify for firms with less than 500 employees. In the Appendix we show how to control access to the central spreadsheets and their solution engines using macros developed in Excel's built-in programming language, Visual Basic for Application (VBA)—no additional cost or software layer or technical expertise beyond Excel and VBA is required. In the Appendix we provide detailed VBA that can be used to control the risk of errors in a spreadsheet-based supply chain model, while still allowing managers to use and update the parameters in the model. It is important to note that the central optimization model is password protected so that it cannot be directly accessed by the end user. The only access is through this VBA, which is designed to properly update the central model, substantially reducing the risk of erroneous analysis. Furthermore, non-technical users cannot see this VBA code on their own PC's, since access to it is also password protected.

FIGURE 7

UNDOCUMENTED SPREADSHEET MODEL (TOP) AND THE SAME MODEL WITH BORDER CONVENTIONS AND THEN DOCUMENTATION ADDED USING THE VBA IN 6 FIGURE (BOTTOM)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Total Cost to Manufacture and Ship Converted to U.S. $												
2	FROM/TO	Mex	Can	Ven	Fra	Gar	Sun							
3	Mex	$ 67.63	$ 115.53	$ 47.41	$ 137.48	$ 113.00	$ 132.02							
4	Can	$ 77.99	$ 104.41	$ 47.83	$ 138.26	$ 114.00	$ 132.02							
5	Ven	$ 82.56	$ 114.16	$ 31.53	$ 130.84	$ 112.40	$ 133.53							
6	Fra	$ 85.08	$ 121.97	$ 49.68	$ 120.49	$ 114.00	$ 128.84							
7	Gar	$ 80.32	$ 116.12	$ 44.11	$ 135.94	$ 102.00	$ 133.79							
8	Sun	$ 81.35	$ 117.09	$ 42.01	$ 132.85	$ 115.00	$ 118.94							
9														
10				Shipments				Total Sent						
11		Mex	Can	Ven	Fra	Gar	Sun	FROM						
12	Mex	3.0	0.0	0.0	0.0	0.0	3.2	6.2	<=	22.0				
13	Can	0.0	2.6	0.0	0.0	1.1	0.0	3.7	<=	3.7				
14	Ven	0.0	0.0	4.5	0.0	0.0	0.0	4.5	<=	4.5				
15	Fra	0.0	0.0	11.5	20.0	6.8	8.7	47.0	<=	47.0				
16	Gar	0.0	0.0	0.0	0.0	18.5	0.0	18.5	<=	18.5				
17	Sun	0.0	0.0	0.0	0.0	0.0	0.0	0.0	<=	5.0				
18	Total Sent TO	3.0	2.6	16.0	20.0	26.4	11.9							
19		=	=	=	=	=	=							
20		3.0	2.6	16.0	20.0	26.4	11.9	$ 7,928						
21														
22	Manufacturing Costs (FCU)													
23		Mex	Can	Ven	Fra	Gar	Sun							
24	Site	725.0	107.0	67,712.0	78.0	102.0	11,823.0							
25														
26	Shipping Costs (FCU)													
27	FROM/TO	Mex	Can	Ven	Fra	Gar	Sun			Exchange Rates per US $				
28	Mex	-	11	34,100	11	11	1,300		Mex	Can	Ven	Fra	Gar	Sun
29	Can	111	-	35,000	12	12	1,300		10.7	1.0	2,147.3	0.6	1.0	99.4
30	Ven	160	10	-	7	10	1,450							
31	Fra	187	18	38,970	-	12	984							
32	Gar	136	12	27,000	10	-	1,476							
33	Sun	147	13	22,500	8	13	-							

Total Cost to Manufacture and Ship Converted to U.S. $

FROM/TO	Mex	Can	Ven	Fra	Gar	Sun
Mex	$ 67.63	$ 115.53	$ 47.41	$ 137.48	$ 113.00	$ 132.02
Can	$ 77.99	$ 104.41	$ 47.83	$ 138.26	$ 114.00	$ 132.02
Ven	$ 82.56	$ 114.16	$ 31.53	$ 130.84	$ 112.40	$ 133.53
Fra	$ 85.08	$ 121.97	$ 49.68	$ 120.49	$ 114.00	$ 128.84
Gar	$ 80.32	$ 116.12	$ 44.11	$ 135.94	$ 102.00	$ 133.79
Sun	$ 81.35	$ 117.09	$ 42.01	$ 132.85	$ 115.00	$ 118.94

DO NOT UPDATE THIS RANGE DIRECTLY! Update the manufacturing costs in row 24 and the shipping costs in rows 28-33: =(G$24+G28)/N$29

Shipments

	Mex	Can	Ven	Fra	Gar	Sun	Total Sent FROM		
Mex	3.0	0.0	0.0	0.0	0.0	3.2	6.2	<=	22.0
Can	0.0	2.6	0.0	0.0	1.1	0.0	3.7	<=	3.7
Ven	0.0	0.0	4.5	0.0	0.0	0.0	4.5	<=	4.5
Fra	0.0	0.0	11.5	20.0	6.8	8.7	47.0	<=	47.0
Gar	0.0	0.0	0.0	0.0	18.5	0.0	18.5	<=	18.5
Sun	0.0	0.0	0.0	0.0	0.0	0.0	0.0	<=	5.0
Total Sent TO	3.0	2.6	16.0	20.0	26.4	11.9			
	=	=	=	=	=	=			
	3.0	2.6	16.0	20.0	26.4	11.9	$ 7,928		

This column sums shipments from each site. Total sent from Mex: =SUM(B12:G12)

This row sums shipments to each site. Total sent to Sun: =SUM(G12:G17)

Solver chooses shipments to minimize this total cost: =SUMPRODUCT(B3:G8,B12:G17)

Manufacturing Costs (FCU)

	Mex	Can	Ven	Fra	Gar	Sun
Site	725	107	67,712	78	102	11,823

Shipping Costs (FCU)

FROM/TO	Mex	Can	Ven	Fra	Gar	Sun
Mex	$ -	$ 11	$ 34,100	$ 11	$ 11	$ 1,300
Can	$ 111	$ -	$ 35,000	$ 12	$ 12	$ 1,300
Ven	$ 160	$ 10	$ -	$ 7	$ 10	$ 1,450
Fra	$ 187	$ 18	$ 38,970	$ -	$ 12	$ 984
Gar	$ 136	$ 12	$ 27,000	$ 10	$ -	$ 1,476
Sun	$ 147	$ 13	$ 22,500	$ 8	$ 13	$ -

Exchange Rates per US $

Mex	Can	Ven	Fra	Gar	Sun
10.7	1.0	2,147.3	0.6	1.0	99.4

CONCLUSIONS

Optimization modeling plays an important role in the management of today's sophisticated supply chains. Spreadsheet-based supply chain optimization models are now a viable alternative to algebraic packages, given that spreadsheet modeling has reached a high level of sophistication and rigor. However, the chance of spreadsheet errors resulting in adverse operational or financial consequences is very real. These errors arise because the intrinsic complexity of optimization models is beyond the expertise of many non-technical managers. It is critical that the risks and potential pitfalls of spreadsheet modeling be acknowledged and actively managed. Supply chain managers should take an informed, conscientious approach to addressing spreadsheet risk, and the tools that we describe can help them in this effort.

Spreadsheet risk can be minimized with thorough documentation, since this reduces the chance of their misuse. However, since the developer already understands how to use the model, it is tempting to omit this documentation. Adding cell comments for documentation is tedious, but we have suggested a best-practice approach involving VBA to partially automate the process of documenting spreadsheet models. We also presented VBA to simplify the process of maintaining centralized spreadsheets without the use of third-party software. Users can pass data to these central spreadsheets and the solution is automatically placed in their worksheet. However, they cannot directly use (or mis-use) the central spreadsheets.

APPENDIX

NON-TECHNICAL READERS CAN READ ONLY THE COMMENTS IN BOLD

Sub Solve_My_LP()

'Access the master workbook. Password to the master workbook is "XYZ":
```
Workbooks.Open ThisWorkbook.Path & "\Master LP.xlsm", Password: = "XYZ"
```

'Run the VBA in master workbook passing it user's workbook name ("ThisWorkbook"):
```
Run "'Master LP.xlsm'!Solve_Users_LP", ThisWorkbook.Name
```

'Close the master workbook and end:
```
Workbooks("Master LP.xlsm").Close SaveChanges: = False

End Sub
```

'In parentheses is the user's workbook name to process:
```
Sub Solve_Users_LP(User_Workbook)
```

'Activate the <u>user's workbook</u> and prompt for data and solution ranges:
```
Workbooks(User_Workbook).Activate
Set Ship_Coef_User = Application.InputBox("Hilite the range of unit shipping costs", Type: = 8)
Set Cust_Dems_User = Application.InputBox("Hilite the range of customer demands  ", Type: = 8)
Set Fact_Caps_User = Application.InputBox("Hilite the range of factory capacities", Type: = 8)
Set Ship_Qtys_User = Application.InputBox("Hilite starting cell to place solution", Type: = 8)
```

'Activate <u>this workbook</u> and define variables corresponding to data ranges:
```
ThisWorkbook.Activate 'Start in E3 of master workbook, continuing to the right and down:
Set Ship_Coef_Mstr = Range("E3", Range("D2").Offset(          Ship_Coef_User.Rows.Count, _
                                                               Ship_Coef_User.Columns.Count))
Ship_Coef_Mstr.Value = Ship_Coef_User.Value 'Copy from user's workbook to this workbook
Set Ship_Qtys_Mstr = Ship_Coef_Mstr.Offset(Ship_Coef_Mstr.Rows.Count + 1, 0)
```

'Select 1 row beneath the shipping quantity range; insert column-sum formulas:
```
Ship_Qtys_Mstr.Rows(Ship_Qtys_Mstr.Rows.Count + 1).Select
Set Into_Custs = Selection
Into_Custs.Formula = "=SUM(" & Ship_Qtys_Mstr.Columns(1).Address(False, False) &")"
```

'Select 1 row farther down; insert customer demands:
```
Into_Custs.Offset(1, 0).Select
Set Cust_Dems_Mstr = Selection
  Cust_Dems_Mstr.Value = Cust_Dems_User.Value
```

'Select 1 column to right of shipping quantity range; insert row-sum formulas:
```
Ship_Qtys_Mstr.Columns(Ship_Qtys_Mstr.Columns.Count + 1).Select
Set From_Facts = Selection
  From_Facts.Formula = "=SUM(" & Ship_Qtys_Mstr.Rows(1).Address(False, False) &")"
```

'Select 1 column farther right; insert factory capacities:
```
From_Facts.Offset(0, 1).Select
Set Fact_Caps_Mstr = Selection
  Fact_Caps_Mstr.Value = Fact_Caps_User.Value
```

'Select single cell 2 rows lower in same column. Enter shipping cost function:
```
Cust_Dems_Mstr.Cells(3, 1).Select
```

```
Set Total_Cost = Selection
Total_Cost.Formula = "=SUMPRODUCT(" & Ship_Coef_Mstr.Address &", "& Ship_Qtys_Mstr.Address &")"

'Solve the LP. Solver in New Master LP already assumes linear & non-negative[c]:
SolverOk SetCell: = Total_Cost.Address, MaxMinVal: = 2, ByChange: = Ship_Qtys_Mstr.Address
SolverAdd CellRef: = From_Facts, Relation: = 1, FormulaText: = Fact_Caps_Mstr.Address
SolverAdd CellRef: = Into_Custs, Relation: = 2, FormulaText: = Cust_Dems_Mstr.Address
SolverSolve (True)

'Copy shipping quantities and total cost to the user's workbook:
Ship_Qtys_Mstr.Copy Ship_Qtys_User
Ship_Qtys_User.Offset(Ship_Coef_User.Rows.Count + 2, 0).Value = Total_Cost.Value

End Sub
```

REFERENCES

Bixby, Robert (2007), "The Evolution of Optimization in Business," Omega Rho Distinguished Lecture, INFORMS Annual Meeting, Seattle, 11/7/07 (quoted with permission).

Botter, Rene and Leonard Fortuin (2000), "Stocking Strategy for Service Parts—A Case Study," *International Journal of Operations and Production Management*, Vol. 20, No. 6, pp. 656-674.

Cummings, John (2008a), "Getting to Grips with Spreadsheet Risk," published on Business Finance (http://businessfinancemag.com), accessed on August 19, 2008.

Cummings, John (2008b), "Spreadsheet Risk Management: A Guide for the Perplexed," published on Business Finance [http://businessfinancemag.com], accessed on July 10, 2008.

Fourer, Robert (2007), "Software Survey: Linear Programming," *OR/MS Today*, Vol. 34, No. 3, pp. 42-51.

Glickman, Theodore (1991), "An Expeditious Risk Assessment of the Highway Transportation of Flammable Liquids in Bulk," *Transportation Science*, Vol. 25, No. 2, pp. 115-123.

Guinipero, Larry, Robert Handfield, and D. L. Johansen (2008), "Beyond Buying," *MIT Sloan Management Review: Business Insight,* posted on March 10, 2008; [http://sloanreview.mit.edu/wsj/insight/operations/2008/03/10/], accessed August 25, 2008.

Jacobs, David, John Peck, and J. Steve Davis (2000), "A Simple Heuristic for Maximizing Service of Carousel Storage," *Computers & Operations Research*, Vol. 27, pp. 1351-1356.

Jafry, Yusuf, Christopher Marrison, and Ulrike Umkehrer-Neudeck (2008), "Hypo International Strengthens their Risk Management with a Large-Scale Secure Spreadsheet Management Framework," *Interfaces*, Vol. 38, No. 4, pp. 281-288.

Kreie, Jennifer, Timothy Cronan, John Pendley, and Janet Renwick (2000), "Application Development by End-Users: Can Quality Be Improved?" *Decision Support Systems*, Vol. 29, pp. 143-152.

Kugel, Robert (2007), "New Tools Enhance Excel," published on Business Finance [http://businessfinancemag.com], accessed on April 1, 2007.

Kugel, Robert (2008), "Quick Fixes for Your Spreadsheet Snags," published on Business Finance [http://businessfinancemag.com], accessed on November 11, 2008.

[c] This VBA is for the built-in Solver. VBA for the large-scale LP Solver differs somewhat.

LeBlanc, Larry and Michael Galbreth (2007), "Designing Large-Scale Supply Chain Linear Programs in Spreadsheets," *Communications of the ACM*, Vol. 50, No. 8, pp. 59-64.

LeBlanc, Larry, James Hill, Gregory Greenwell, and Alexander Czesnat (2004), "Nu-kote's Spreadsheet Linear Programming Models for Optimizing Transportation," *Interfaces*, Vol. 34, No. 2, pp. 139-146.

Lipman, Timothy and Mark Delucchi (2006), "A Retail and Lifecycle Cost Analysis of Hybrid Electric Vehicles," *Transportation Research: Part D*, Vol. 11, pp. 115-132.

Panko, Raymond and Ralph Sprague, Jr. (1999), "Hitting the Wall: Errors in Developing and Code Inspecting a 'Simple' Spreadsheet Model," *Decision Support Systems*, Vol. 22, No. 4, pp. 337-353.

Schuster, Edmund, Stuart Allen, and Michael D'Itri (2000), "Capacitated Materials Requirements Planning and its Application in the Process Industries," *Journal of Business Logistics*, Vol. 21, No. 1, pp. 169-189.

Smith, Gary (2003), "Using Integrated Spreadsheet Modeling for Supply Chain Analysis," *Supply Chain Management: An International Journal*, Vol. 8, No. 4, pp. 285-290.

Soudhi, ManMohan (2003), "How to do Strategic Supply-Chain Planning," *MIT Sloan Management Review*, Vol. 45, No. 1, pp. 69-75.

United States Environmental Protection Agency in the Matter of Vemco, Inc. (2008), [http://www.epa.gov/oalj/orders/vemco-pad.pdf], accessed on August 25, 2008.

United States Securities and Exchange Commission in the Matter of California Amplifier, Inc. (2008), [http://www.sec.gov/litigation/admin/34-49630.htm], accessed on August 25, 2008.

ABOUT THE AUTHORS

Michael R. Galbreth (Ph.D. Vanderbilt University) is an Assistant Professor of Management Science at the Moore School of Business, University of South Carolina. He received his Ph.D. in Operations Management from Vanderbilt University. His current research interests include sustainable operations, supply chain modeling, and the intersection of operations and information systems.

Larry J. LeBlanc (Ph.D. Northwestern University) is a Professor of Operations Management at the Owen Graduate School of Management at Vanderbilt University. His Ph.D. is from Northwestern University in Industrial Engineering/Management Sciences. His research interests include teaching management science using spreadsheets, supply chain analysis, spreadsheet optimization models, implementation of algorithms for large-scale optimization models, and telecommunication network design/analysis. He has approximately 60 publications in refereed journals and 60 presentations at universities and organizations overseas.

Contact author: Larry J. LeBlanc; E-mail: Larry.LeBlanc@Owen.Vanderbilt.edu

JOURNAL OF BUSINESS LOGISTICS

Volume 31, Number 2 *Table of Contents* *2010*