Does Supply Chain Optimization Matter?

By Mark Woeppel

Recently, I read a white paper on supply chain optimization by AMR (Advanced Manufacturing Research), Inc. It stated that supply chain optimization presents an enormous opportunity for improving supply chain performance. The paper advanced the idea that the computer's ability to do rapid calculations allows managers to improve their decision making by focusing on objective data and relationships and relying less on their intuition and judgment.

When I sent the paper out to a number of my colleagues, some of them challenged me on the objectivity and validity of the underlying hypothesis. Following that, I read Eliyahu M. Goldratt's new book, *Necessary But Not Sufficient*, where he takes the position that optimization/technology is important, but insufficient to achieve the results managers want.

Thus the question: Does optimization matter? Many have read *The Goal*, made changes to their business processes and subsequently realized great improvement to their bottom lines. I've spent the better part of my professional life preaching that it's process that matters while software is immaterial (or at least secondary). Which is it? Process or software? Software suppliers make very little mention of business process. Management consultants downplay the role of technology in business improvement. Dr. Goldratt says both are needed.

Optimization in 500 Words or Less
Optimization software is primarily a planning tool, different from ERP, which is an execution tool. Optimization algorithms/software, like advanced planning and scheduling, is a wonderful tool that allows you to plug in the variables and constants of a problem, press a button, then gives you get the "best" result. Everyone wants what is best, don't they? Of course. The problem is, what is defined as "best?"

There are two fundamental issues with optimization. One of them is deciding on the objective of the problem—or in other words, what aspects of the problem are you trying to balance? Achieve the maximum throughput with least impact on delivery performance? Achieve the maximum throughput with the least cost? Maximize customer service with the least cost? Have the lowest cost with the least damage to customer service? Getting the objectives optimized will determine the course of the end solution. Therefore, having an optimization strategy starts with defining your goal.

The second issue is the methodology. What type algorithm do you use? Generic algorithms? Graph theory? Drum-Buffer-Rope? OPT? Finite forward loading? Backward finite loading? Linear programming? Mixed integer linear programming? Simulated annealing? Rules-based heuristics? How is someone to choose? AMR states it doesn't matter which one you choose. If that's the case, why do software companies differentiate their product on the algorithm? I believe methodology matters because different methods bring different results. For example, forward discrete event simulation cannot predict

"Optimization software claims to tell you how to fish."

in advance where the constraint(s) will be, then synchronize all other elements to those constraints. It can't see them until it encounters them. Will using this method improve the results over what you can do manually? Probably. Will one optimization method yield better results than another? Definitely.

Optimized or Optimal?

The goal of most manufacturers in implementing optimization software is to improve over the current state. Managers are not necessarily interested in the best solution, but a better solution. What's a better solution? Something practical that will improve the bottom line results. The main concern with using such software is: Can you implement it? An optimized solution will advance the best possible solution, but that solution may be some-

thing that cannot be implemented. Therefore, managers look for optimum solutions, those that may not provide "peak" system performance, but provide "better" system performance. They are realistic in their expectations of their own ability to execute the solution (there is some debate as to whether any system can deliver an "optimized" solution).

To my (biased) mind, here's what I would look for in choosing optimization software:

- Can someone explain the algorithm to me in 15 minutes or less? If they can't, I will not be able to interpret or validate the results. This is important for the implementation of the solution and for longevity. It is very difficult to get buy-in to a solution no one understands. Furthermore, one must make provision for handoff of the tool from the current staff using

the software to the next group. Frequently, people will stop using a tool they don't understand or they will use it in the wrong way.
- Can I examine the system's constraints before I choose to "optimize" them? Many times, a constraint (such as raw material delivery date) is not really "hard," meaning that if I want to, I can change it (through expediting or some other method). Certainly, one wouldn't want to subordinate the entire organization to a variable that is easily changed.
- How does the subordination mechanism work? Is there one? Assuming your goal is to reduce inventory and operating expense (that is your goal, isn't it?) while increasing throughput, all non-constraints would be subordinated to and synchronized with the defined constraint(s).

- How does the system compensate for process variation? How are buffers introduced into the system (is there a provision for buffering at all)? Is there a mechanism to manage protective capacity? Buffers immunize the schedule against disruption and give me assurance the schedule can actually be run.

From the above, you would think I'm looking for an optimization algorithm that looks like Drum-Buffer-Rope. I'm not. However, I am looking for algorithms that will allow me to identify, exploit, and subordinate (and elevate) the system's constraint(s). You can use the ERP algorithm to do that (roughly, with great difficulty). My point is: understand the algorithm so you can determine if it does what you want it to do.

Let's suppose you're sold on the idea of optimization to help your business. Does that mean that you're halfway there to improving your business? No, but you are about a quarter of the way there.

The Problem With Optimization

Optimization software (APS, supply chain, etc) is a different animal than ERP/MRP software. Enterprise systems focus on managing transactions and providing summary data to evaluate the condition of the business. Optimization software is a decision support tool that has decision rules and math to help you choose what do with the business. It's like the difference between a fish finder and a human guide. The fish finder will tell you

there are fish present, the approximate depth, and if there are many or few, but will not tell you how to get them into the boat. The guide will likely know what kind of fish they are, what they like to eat, and how to present the bait to get them to jump on your hook. In the same way, ERP says you have customers, products, and costs, but doesn't tell you how to put them together in the way that will generate the most profit (or shortest delivery, etc.).

Optimization software claims to tell you how to fish. This approach—using an automated guide to plan the business—is an alien concept to managers. The question of how one uses such a tool is left up to the users. Most managers don't know how to use it. Using something like this to help manage the business is completely outside of their experience. It's still a tool for the "early adopters" and "visionaries."

In planning the business, managers have come to rely on aggregations of each individual department's (link's) performance. Strategic planning in manufacturing and supply chains is seldom done using quantitative methods with a feedback loop to the sales and marketing organization.

Instead, each part of the business plans their area of responsibility (markets, products, sales, operations, etc), then gets together to reconcile the separate plans into a single plan everyone can follow. There are no tools to integrate then evaluate one scenario over the other, only intuition and rolled-up financials. The tradeoffs that occur between resource consumption and market/product strategy are unknown or guessed. Managers find the approximate area to fish and then drop lines into the water based on their experience catching the kind of fish they are looking for (say, catfish). They don't even look for other opportunities (bass), because they don't have a way of finding them.

Optimization software allows you to evaluate the entire organization across functional boundaries. The focus of these tools is strategic integration. Who will take responsibility for this tool? There is no well-defined answer. The barriers to answering the question are significant. As businesses are structured now, each function has a different component of the financial equation. Each function, until

now, has been responsible for "optimizing" their particular variable. We have traditionally operated under the assumption that if each department or function within the department does the "best" it can do, the entire organization will do well. This assumption (the sum of the local optima is the optimum of the whole) has been discredited, but managers have not had the ability and/or the opportunity to run their companies any other way. Now we have tools that will optimize the entire system. However, in order to optimize the system, some elements within that system will be sub-optimized! What is that manager to do? How will his performance be measured? Making effective use of such a tool requires managers to redefine "good" performance in light of their ability to subordinate themselves to the constraint(s) of the business.

Implications

Using optimization software requires a commitment to a competency. Just as implementing MRP/ERP systems required a commitment to information technology infrastructure and people, ongoing use of optimization software requires a commitment to planning by the numbers. Many large companies have strategic planning departments, but now the technology is in the reach of small and mid-size companies that want to hone their competitive edge. Organizations should have a chief integration officer (you come up with your own title) to integrate the strategies and realities of the various parts of the business. This person would be responsible for reconciling the plans and making recommendations to management about potential opportunities within the business. The job is very similar to the master production scheduler, except that instead of dealing with units of capacity, the person would be dealing with dollars. Instead of trying to fit the marketing forecast into the available capacity, he or she would be making recommendations on product mix and operations investments.

Making effective use of the optimization software means changing the way you're doing things. New technology enables new methods, and training is required to understand it. In the past, the training that has been done has been focused on the staff. This tool will require staff training *and* management training. We have the same kind of paradigm shift we had when MRP became available. Those of you in your 40's and older will remember the obstacles that had to be overcome in order to have that technology widely accepted into management practice.

We have to make allowances in order to change fundamental business processes and thinking. This is real paradigm shift material. The problem is, change can be hard. The solution? Manage your implementation as a change issue and not a technology issue.

Does it Matter?

Does optimization matter? Yes. It presents one of the biggest opportunities for profit improvement since MRP was introduced. With optimization software you can quantitatively know which products are the best ones to sell based on your infrastructure and market. You can squeeze more products out of the plant through better scheduling and sequencing. You can make better sourcing decisions. You can do "what-if" scenarios.

Does the method matter? It matters a great deal. But, as long as you're focused on the constraint, the method doesn't matter as much as the software vendors would like you to think.

Mark Woeppel, CPIM, is a nationally known speaker, lecturer and author of The Manufacturer's Guide to Implementing the Theory of Constraints (APICS/ St. Lucie Press). He is President of Pinnacle Manufacturing Consulting and may be contacted by e-mail at: mwoeppel@mfgexcellence.com