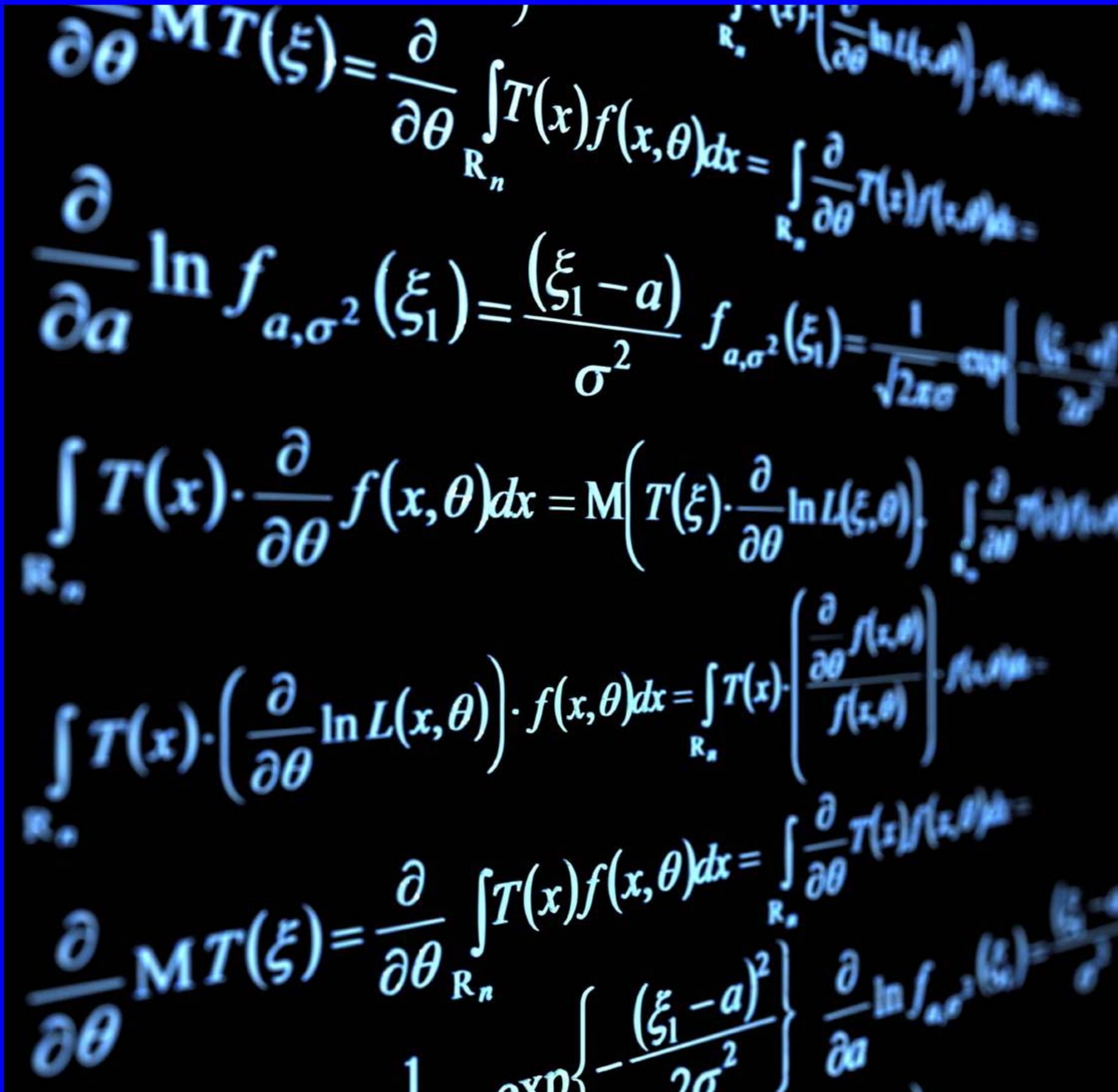

The Purchasing & Supply Management Podcast Series℠
What Is Supply Chain Optimization?

Next Level Purchasing

Helping Purchasers Become Indispensable

July 24, 2007

This free transcript of the Next Level Purchasing podcast Optimization: An Introduction, Part I and Part II, is brought to you courtesy of ToP KaTS Consulting and edited by Michael Lamoureux of the Sourcing Innovation blog.

Editor's Preliminary Notes:

- ***Italicized bold is used to convey those parts of the conversation spoken by the president of Next Level Purchasing, Charles Dominick.***
- Regular text is used to convey those parts of the conversation spoken by Michael Lamoureux of ToP KaTS Consulting and the Sourcing Innovation blog.
- *Italicized text is used to convey additional notes added by the editor to further explain or clarify parts of the conversation and related topics.*
- Regular text within *italicized text* is used to convey italics within the italicized text, which generally means a direct quote is being made.

Hello everyone, and thank you for tuning into another edition of the Purchasing & Supply Management Podcast Series. This is your host Charles Dominick, the founder and president of Next Level Purchasing. In this podcast, we'll be talking about what is arguably the most complex issue in supply chain management: optimization.

Editor's Note: Other editions in the Next Level Purchasing podcast series can be found here.

Joining me today is Michael Lamoureux. Michael is the editor of the very popular Sourcing Innovation Blog and is president of ToP KaTS Consulting- a consulting firm with expertise in modeling, optimization, and supply chain processes. Good morning, Michael.

Good morning Charles.

Thanks for joining us today.

You're welcome.

Michael, you have some very impressive educational credentials. So before we even get into optimization, can you tell our listeners a little bit about your background? I just think that this will give them a better appreciation of the intellectual horsepower that is necessary to master what is "under the hood" of some of today's advanced technologies.

Good Question and Tough Question.

Academically, I have a Doctorate of Philosophy in Computer Science, specializing in something called multi-dimensional and spatial data structures and computational geometry, which followed my Masters in Computer Science in multidimensional data structures and algorithms and my Bachelors in Mathematics. During my undergraduate studies and the first part of my Masters, I spent a lot of time studying dynamical systems, complexity, and fractal geometry. During my Master's specialization, this shifted a lot into the fields of automata, finite systems, and advanced combinatorics. The only constants throughout were set theory and abstract algebra. In plain English, I have a lot of education in the fundamentals of mathematical algorithms and computation, or what it takes to design, create, and implement very sophisticated algorithms and models.

Editor's Note: Dynamical Systems, or at least in the context of my studies, is essentially the study of non-linear mathematical systems and their unpredictable nature. [Wikipedia] defines the concept as a mathematical formalization for any fixed "rule" which describes the time dependence of a point's position in its ambient space *and it* has a state determined by a collection of real numbers, or more generally by a set of points in an appropriate state space; small changes in the state of the system correspond to small changes in the numbers. *It's the foundation for the study of chaotic dynamical systems, and most people have probably heard of the most popular pop-culture definition, the butterfly effect, which encapsulates the more technical notion of sensitive dependence on initial conditions and essentially asks if the flap of a butterfly's wings in brazil can set off a tornado in Texas.*

Complexity refers to complexity theory, which is sometimes known as computational complexity theory, and as per the [Wikipedia] definition, describes the scalability of algorithms, and the inherent difficulty in providing scalable algorithms for specific computational problems. *Algorithms and problems are categorized into complexity classes, and many people familiar with the discipline of algorithmics are probably at least acquainted with the concepts of P, polynomial-time, and NP, non-deterministic polynomial time, which are used to described whether the algorithm is guaranteed to complete in polynomial time (i.e. the run time of the algorithm can be described using a fixed-order polynomial equation) or not (i.e the algorithm may require exponential run-time with respect to the size of the inputs). The basic premise here is that polynomial algorithms, on a large enough machine, can be guaranteed to be solvable in reasonable amounts of time, whereas exponential algorithms cannot.*

Fractal Geometry is the study of fractals, which are defined as rough or fragmented geometric shapes that can be subdivided in parts, each of which is (at least approximately) a reduced-size copy of the whole ([Benoit Mandelbrot], the father of fractals). Fractals are special types of geometric objects that have a very fine structure at arbitrarily small scales that are too irregular to be described in traditional Euclidean geometry, that are self-similar, and that have simple, recursive definitions.

[Automata theory] is the study of abstract machines and the problems they are able to solve. It's a type of mathematical model for a finite state machine (FSM), and is often classified by the formal language (from a computational viewpoint) it is able to represent. There are many different types of automata, and examples of finite automata include pushdown automata (PDA), linear bounded automata (LBA), and the [Turing Machine], which is mathematically equivalent to an algorithm and the basis for modern computers. Turing Machines were first described by [Alan Turing] who also devised the famous Turing Test, which was a proposal for a test of a machine's capability to demonstrate thought. In the Turing Test, a human judge engages in a natural language conversation with two other parties, both hidden, one a human and the other a machine; if the judge cannot reliably tell which is which, then the machine is said to pass the test.

Set theory, which is simply the study of sets (a mathematical concept that is far more involved than what your high school or even undergraduate studies may lead you to believe) was a constant because fractal geometry, which is closely related to dynamical systems theory through the concept of a strange attractor, has a Hausdorff dimension which is defined by a Hausdorff measure, which is computed on point sets; automata are defined on a fixed set of states; advanced combinatorics are defined on discrete sets of possibilities; and you get the picture.

Abstract algebraic measures are used in computational complexity theory, finite systems, advanced combinatorics, and in algorithms, computational geometry, and data structures, the latter being the areas of specialization for my graduate theses.

Professionally speaking, in addition to teaching University throughout half of the 90's, starting part way through my Master's degree as a lecturer, I started consulting with Industry as well, first on professional development and training, then on system design and implementation, and finally on processes and process development. Also, in the 90's, it started out in proper internet and web development methodologies and design, such as the classic two-and-three tier web development network topologies in particular, and then moved into e-commerce, primarily B2B and B2C. By the time the naughts hit, I had completed my graduate degrees, and had moved into the sourcing, procurement, and supply chain industry full time, starting in 2000 with a then unknown start-up called MindFlow Technologies.

Editor's Note: The following is additional information on Michael G. Lamoureux's background that was left out of the podcast to allow more time on the topic of optimization.

MindFlow was one of the first companies to focus primarily on the application of decision optimization to strategic sourcing, and despite competition's claims to the contrary, was one of the true market leaders from late 2000 to late 2003. Emptoris may have been easier to use at the time, as their model was simpler and their model optimized for the solver they used, and CombineNet definitely had the more powerful solver, but Emptoris initially lacked a lot of the constraints and CombineNet's model was based on a market-clearing auction model that no one really used. However, fast forward to today and CombineNet, who shifted away from the market-clearing auction model and into a "CombineNet-inside" best-of-breed optimization platform strategy and Emptoris, who ultimately acquired MindFlow and upgraded their model considerably, are likely two of your top three choices as a procurement professional.

After MindFlow, I did some work on a novel B2C and C2C platform that could have been a next-generation eBay for power resellers and small businesses, but the company was two to three years ahead of its time and the investors didn't want to wait two years to break even and three years to make a profit (despite the fact that Amazon proved that you can take years to make a profit and still make a killing down the road) and then for a defense sub-contractor on a modeling platform for military optimization models, including the military equivalent of supply chains.

Then, for the last few years, I moved back into the supply chain and sourcing space full time as an independent consultant, and an analyst from a blogging perspective, focusing on primarily advanced sourcing and procurement technologies, and, as everyone knows, I started the Sourcing Innovation blog last June.

Editor's Note: The URL for the Sourcing Innovation Blog is: blog.sourcinginnovation.com.

My CV or the "short-form" CV I just gave you doesn't really define the types of "intellectual horsepower" that is required to master what's "under-the-hood" of today's optimization technologies.

They're very, very complex and advanced. That's why only a few companies are working on them. Mastering optimization generally requires a mastery of the underlying mathematics, the model construction techniques, and the strengths and weaknesses of the particular model solution algorithms that are available "under-the-hood".

Editor's Note: The following is additional information on what types of underlying mathematics must be mastered and what is involved in mastering model construction techniques. Some of this information was left out of the podcast for reasons of brevity, and some of this information is included to help you dig deeper into the response.

Most off-the-shelf optimizers, like ILog's CPlex or Dash Optimization's XPress or Sunset Software's XA, use what's called "Mixed Linear Integer Programming" (MILP). (A short introduction to MILP, sometimes referred to as MIP, includes this page by Sandia National Laboratories.) Thus, this would imply that you need to master systems of linear equations and branch-and-bound combinatorial algorithms, appropriate model construction techniques (including piece-wise linear constraints, the theory of logical variables, and other tricks to represent "real-world" constraints in the tight confines of the mathematical constructs available), and the solution algorithm pitfalls - such as which constraint types are "hard", which formulations are likely to be too restrictive and prevent the optimizer from finding a solution even if one exists, and which constraint formulations are likely to cause the optimizer to "cycle" or get trapped in a local minima - to insure that the models you construct are as "optimized" as possible.

But even this definition leaves out the fact that there is really no general purpose methodology for constructing a model. If you pick up a modeling text, it "teaches" modeling by walking you through different models for different problems. If you want a model for a problem not in the book, you have to pretty much develop it through trial and error, starting with the closest model, or model fragments, you can find and altering it until you can "prove" that it works. This requires an intuition that only comes with training and experience that is both broad and deep. That's why we have the current situation, where not only are there not a lot of "true" optimization experts out there, but most of the "optimization" solutions are not always appropriate for your problems.

I think that gives us a very good background of your capabilities and your experience and so forth. So let's move on to the topic of the day, optimization. In the most simple terms possible, can you tell us what optimization is?

I can try, but, as I partially alluded in my background description and on my sourcing innovation blog when you're talking about optimization in the sourcing / procurement / supply chain space, there's optimization, there's decision optimization, and then there's strategic sourcing decision optimization.

Can you tell us about the differences between those three?

No problem.

Plain old optimization, which you can even go look up on Wikipedia, is simply the mathematical process of trying to find a minimum or a maximum of a real function on one or more variables subject to fixed ranges on those variable values and one or more constraints defined on the variables. Basically, this goes all the way back to your basic college mathematics, or university mathematics, where you had a function $f(x_1, x_2..., x_n)$, and you want to find the values of x_1, x_2, ..., x_n that minimize or maximize the value of f. Pretty straight-forward, pretty non-descriptive to our situation.

Decision Optimization, which is an extension of optimization and the application thereof, which is based on rigorous analytical and mathematical techniques to well-defined real-world scenario representations in an effort to arrive at the absolute best decision out of all the possible alternatives, typically in a fashion that is rigorous, repeatable, and provable. And that was a mouthful.

Basically, when you are talking about decision optimization for the real world, there are four key components that have to be there in the optimization or optimization solution you're using.

First of all, the analytical and mathematical techniques that you use must be sound and complete - this means that they must be capable of analyzing every possible solution ("*complete*") and do so in a correct fashion ("*sound*"). If you get an optimization solution, and I'm going to quote-unquote the optimization that uses heuristics, it may not be capable of analyzing every possible solution so even though it might give you a better answer than what you'd get by hand or on your own, it still may not be true optimization if it's not looking at potential solutions because the heuristic thinks that they won't give you a better answer than the solutions it is looking at.

Secondly, the solution must be well-defined and capable of accurately representing the problem at hand. If the solution forces you to over simplify your problem to use the tool, then you're not optimizing the right model which means you're not optimizing at all. Closely related models don't necessarily have closely related solutions, mathematically speaking.

Thirdly, if you give the solution enough time, where that time varies depending upon model size, constraints, complexity, hardware, processing power, and so on, the optimization must be capable of eventually producing the absolute best decision. People who've used optimization tools in the past will recognize this. Sometimes you build a model and you can get the optimal solution in a few seconds. Sometimes you build a model that you intuitively think is not that much more complicated but it takes a few minutes. Almost all solutions allow you to stop the process after an initial solution has been reached; but there must be the capability that if there is enough processing time the optimizer can give you the best solution.

And, finally, the process must be repeatable. It can't be totally random or totally heuristic. If you put your model in, press solve, and it gives you a solution, you should be able to put that model back in tomorrow, press solve, and get either the same solution or another equivalent solution. This sort of differentiates optimization from statistical modeling, using techniques such as Monte Carlo where it randomly tries a whole lot of scenarios to kind of show you what the different possibilities are and allow you to pick the one that's the lowest. That is not optimization that is simulation. Simulation is good for understanding what types of solutions you might be able to come up with, but it's not good for giving you the optimal solution unless you have an awful lot of time.

Editor's Note: Simulation is simply the modeling of a system in an attempt to get (further) insight into its function. Monte Carlo is a non-deterministic stochastic method for simulating many possible ways an algorithm or event can play out using (pseudo) random numbers and iterative calculations. They're useful in areas of mathematics where computations are essentially intractable and where a "lucky guess" can find the right result. I don't know about you, but I don't want to leave my award allocations to chance!

And, of course, this leads us to Strategic Sourcing Decision Optimization, which I'm sure is what your listeners are most interested in as procurement professionals. This is the application of those rigorous mathematical techniques to a well-defined sourcing scenario to produce an optimal award allocation from what I call, and what a few others are starting to call, a total value management perspective.

What's a Total Value Management Perspective?

Total Value Management, or TVM, refers to the use of a comparative cost metric that quantifies a sourcing plan or sourcing award according to the overall cost of each acquired unit of product relative to the overall value of the buy over the lifetime of the project, which may be subject to the organization's sourcing strategy and supply chain goals. It's basically taking the Total Cost of Ownership viewpoint to the next step. In traditional TCO modeling, you generally stop once all of your direct costs are taken into account, such as the unit cost, the shipping cost, the import and export duties, and the temporary storage costs. TVM looks not only at the acquisition costs of those products but the processing and eventual life-cycle costs and includes all of the indirect costs and impact costs such as the support costs that you can predict will occur if you source a lower quality product. For instance, if one supplier has a defect rate (*that is*) five times (*that*) that of another supplier, you know you're not going to catch all those defects and you can probably expect your customer support calls on that particular product to triple, quadruple, or even quintuple down the road. So there needs to be a cost associated with that when sourcing.

It (*TVM*) also takes into account the value associated with using a certain supplier over another. For instance, if one supplier is well known, you may be able to use "intel-inside" branding or sustainable business practices branding and that has a value. That's less marketing dollars that your marketing organization has to spend to build up the value of your brand because you can build on someone else's. And, I don't want to get too far into this because this is really an interview and podcast on optimization, but if your readers want more information on TVM, over on the eSourcingWiki, www.eSourcingWiki.com, I co-authored a wiki-paper called "Strategic e-Sourcing Best Practices" that talks about Total Value Management and how it compares to Total Cost of Ownership. It's just the logical evolution - making sure you consider all of the life-cycle costs associated with a sourcing decision.

So, to summarize, Strategic Sourcing Decision Optimization is based on true cost modeling - you need to be able to accurately represent your costs, (*perform*) sophisticated constraint analysis - you have real world constraints both from the supplier perspective and your perspective (plus you need to mitigate risks), and what-if? capability. And the model must be capable of capturing all of those costs, all of those constraints, and do so in flexible ways because suppliers bid in different ways - some use tiered bidding, some use discounts, some will break out the fixed and variable components of a production run to let you to more accurately cost what it will cost you to use them. It should let you make copies of that scenario and run it multiple times under different constraints to understand what happens under different business rules and different possible eventualities.

Editor's Note: In addition to multiple types of bids, the model must be able to capture the different types of constraints, which should include at a minimum capacity, basic allocation, risk mitigation, and qualitative constraints that define the scenario. For more information, see the Sourcing Decision Optimization wiki-paper that is coming soon!

Now, how are you seeing supply chain professionals using optimization?

I'm sad to say that for the most part, I'm still not seeing that many supply chain professionals use optimization. And I personally find this very disappointing having been one of the innovators in this space, especially considering that Aberdeen Group has found, both in their June 2005 Study that they called "Success Strategies in Advanced Sourcing and Negotiations: Optimizing Total Costs and Total Value for the Next Wave in e-Sourcing Savings" and in their January 2007 study that they called "The Advanced Sourcing and Negotiation Benchmark Report: The Art and Science of the Deal" that Best-in-Class companies that apply advanced sourcing strategies, and advanced cost modeling and optimization in particular, typically save 12% above and beyond what their peers save - even those peers that use e-Sourcing tools and e-Auctions.

And those that are using optimization are typically using the boxed solutions that support category-based strategic sourcing - in a few cases, the old MindFlow solution, but more commonly now the Emptoris solution, the templated solutions for freight optimization from CombineNet, and, just starting, the new Iasta solution. But only a select few organizations, that are more advanced and best-in-class, contract the specialty providers like CombineNet or APL Logistics to redesign their distribution networks or sourcing processes and take them to the next level. And this is unfortunate because, with optimization, on large multi-million dollar categories I've seen many instances where a single sourcing event has literally saved the company millions of dollars.

So, there are some supply chain organizations using optimization. What did they do before they used optimization and what benefits are they realizing from using optimization? You mentioned the additional savings, but maybe you can elaborate on that a bit.

Well, let's start with what did they do before. They did what many of their counterparts still do. They used a spreadsheet - a tool that was never designed to support large, complex models. Spreadsheets were developed for simple accounting back in the 70's - to create an electronic version of the general ledger that would automatically do the math and reduce adding and subtracting errors. They were never created for the models that are being thrown at them today. And to quote a recent study / press release from Oregon State University, that's why up to 90 percent of the 100 million plus spreadsheets created in the US every year have non-trivial errors in them. And it's also why, as some readers of my blog know, I am fundamentally against the use of spreadsheets as a modeling or optimization tool, and to some extent, even a reporting tool.

Going back to those using optimization, the benefits are the same benefits that their peers using other e-Sourcing tools are seeing, only greater. It reduces cycle time and can drastically reduce the analysis required to do a sophisticated award allocation and, in most cases, it increases cost savings. And what I've seen is the average savings for companies that are doing e-Sourcing well is usually between 5 and 15 percent and even the most complex awards can usually be decided in a day or two. Simple awards sometimes in a matter of hours, or almost automatically after the sealed bid.

Let's dig a little deeper. Can you describe a simple scenario with just a few bidders and a couple of constraints and walk through how a sourcing professional would go about solving it manually and then describe how optimization technology would make it easier to solve?

Sure, but I hope you don't mind I'm going to use one of my boxed examples and go into a lot of details here because only by going into a fair number of details can you really get at the complexity of even a simple award and illustrate how optimization can be of great value, even if you think you know what the solution already is.

So let's assume we have three bidders, Alice, Bob, and Carol, and they're bidding on your organization's RFP for nuts and bolts. Let's further assume, to make this real, that you need 20,000 nuts and bolts at your Atlanta warehouse, 15,000 units at your Boston warehouse, and 25,000 units at your Chicago warehouse and, furthermore, Alice cannot supply Atlanta, Bob cannot supply Boston, and Carol cannot supply Chicago.

For those working at home, the A's, B's, and C's go together.

Yes, just to make things simple.

Okay, great!

Alice is going to bid $1.10 for a nut and bolt pair - I'm not saying these prices are real, I'm just saying that these are easy numbers to work with - but give you a 10 cent discount if you buy 20,000 units; Bob is going to bid $1.05 for a nut and bolt pair, but give you a 3 cent discount if you buy 10,000 units or a 6 cent discount if you buy 20,000 units; and Carol is going to bid $1.11 for a pairing, but a 9 cent discount if you but at least 25,000 units. And to top it all off, you're implementing a dual sourcing strategy to mitigate supply risk down the road, and you're going to insure that any selected supplier is going to get at least 25% of the business.

Okay, fair enough, continue.

Now you have a goal to figure out which two suppliers get the business, and how much business each supplier gets. Now, in order to simplify the number of scenarios that we have to work through to get at an optimal answer, or at least one that we expect to be optimal, I'm going to assume your sourcing professional is very well trained, very skilled, and knowledgeable about something called the theory of constraints and the optimization of convex spaces. In other words, I'm going to assume that your sourcing professional knows that the dual sourcing constraint is the constraint that most tightly constrains the possible solutions, and thus the possible optimal solution(s), and that this constraint should be used to guide the process overall. And I'm also going to assume that the sourcing professional suspects that the solution space, mathematically speaking, is something we call convex, and this is important because this implies that the optimal solution likely falls on a boundary point, or an extreme point, of the space, where a boundary point or an extreme point is the point on the border if you mathematically map the n-dimensional solution space. And, in more simple terms, it's usually a point that's either at the end of your ranges or somewhere where ranges overlap.

So, this very smart sourcing professional would start by trying to figure out the optimal solution for each of the possible award scenarios from the dual-source perspective. In other words, she'd realize that it's either going to be Alice and Bob, Alice and Carol, or Bob and Carol and that you can find the overall optimal solution by solving these three sub-optimization problems and taking the lowest cost solution. Are you with me so far?

I'm following you. We may need to throw up a spreadsheet for people following along at home.

I think we will have to create one and give it to them as a download. *See the spreadsheet on the last page.*

So I'm going to start with Alice and Bob. The optimal solution to this sub problem will have Alice supplying between 15,000 and 40,000 nut and bolt pairings and Bob supplying between 15,000 and 45,000.
The 15,000 to 40,000 for Alice comes from the fact that she must get at least 25% of the total award, which is 15,000, and the fact that she cannot supply Atlanta, which means that she can't get more than 40,000. We do similar calculations for Bob, based on the minimum supply that he must get and the fact that he cannot supply Boston to get that 15,000 to 45,000. If we then go and look at the discounts, or the award allocations at which new costs come in, which give us quote-unquote our "extreme points", and we assume convexity, we can assume that our lowest cost is going to be at one of these points, or more specifically, where Alice gets 15,000, 20,000 or 40,000 units or where Bob gets 15,000, 20,000, or 45,000 units. If we then look at these two possible award sets in parallel, and work out their overlap, we see that we're going to have to

hand-calculate at least three award scenarios, but because of the assumptions we're making, we're only going to work out these three. Specifically, where Alice gets 15,000 units and Bob gets 45,000 units, where Alice gets 20,000 units and Bob gets 40,000 units, and where Alice gets 40,000 units and Bob gets 20,000 units. If we work out the math, we find out that the respective costs of these three scenarios are $61,050, $59,600, and $59,800. We're already seeing the importance of having some sort of tool that can automate these calculations because three similar scenarios have three entirely different costs. You could say not by much, but enough that you might want to worry about it.

*Editor's Note: More details on the calculations are as follows. Since the total demand is 60,000 units, that means each supplier must supply at least 15,000 units; however, since Alice cannot supply Atlanta, with a demand of 20,000 units, and Bob cannot supply Boston, with a demand of 15,000 units, this places an upper limit on Alice of 40,000 units and an upper limit on Bob of 45,000 units, one of which is less than the upper bound of 45,000 on both of them since neither can get more than 75% of the award. Factoring in the discounts, and assuming convexity, we're looking at the award scenarios where we give Alice 15,000, 20,000, or 40,000 units and where we give Bob 15,000, 20,000, or 45,000 units since Alice gives us a discount at 20,000 units and Bob also gives us a discount at 20,000 units. Note that Bob's discount at 10,000 units does not need to be evaluated separately because, if Bob is selected, he gets at least 15,000, so the first discount price replaces the base price for Bob. Merging these awards gives us three award scenarios: where Alice gets 15,000 units and Bob gets 45,000 units, Alice gets 20,000 units and Bob gets 40,000 units, and where Alice gets 40,000 units and Bob gets 20,000 units. This is because an assignment to Bob of 45,000, 20,000, or 15,000 is equivalent to an assignment to Alice of 15,000, 40,000, and 45,000, and since Alice cannot get more than 40,000 units, we see we only have to evaluate the scenarios where Alice gets 15,000, 20,000, or 40,000 units. Their respective costs are found to be $61,050 (15,000*1.10 + 45,000*0.99), $59,600 (20,000*1.00 + 40,000*0.99), and $59,800 (40,000*1.00 + 20,000*0.99).*

So now I'm going to move on to Alice and Carol.
Doing the same sort of calculations tells us that Alice will supply somewhere between 15,000 and 40,000 units and Carol will supply somewhere between 15,000 and 35,000 units. Again we do all the math and we find that we are going to come up with three potential award scenarios again. Specifically where Alice gets 25,000 and Carol gets 35,000, Alice gets 35,000 and Carol gets 25,000, and Alice gets her maximum potential award of 40,000 and Carol gets 20,000. We'll then work out the costs of these scenarios and find three different costs: $60,700, $60,500, and $62,200. Again, a bit of a spread.

Editor's Note:
As noted, the optimal solution to this sub problem will have Alice supplying between 15,000 and 40,000 units and Carol supplying between 15,000 and 35,000 units, since Carl cannot supply Chicago. Factoring in the discounts, and assuming convexity, we're again looking at award scenarios where Alice gets 15,000, 20,000 or 40,000 units and we're also looking at award scenarios where Carol gets 15,000, 25,000, or 35,000 units - Carol's minimum, discount breakpoint, and maximum. Merging these awards gives us three distinct award scenarios: where Alice gets 25,000 and Carol gets 35000 units, Alice gets 35,000 units and Carol gets 25,000 units, and Alice gets 40,000 units and Carol gets 20000 units. Their respective costs are

*$60,700 (25,000*1.00 + 35,000*1.02), $60,500 (35,000*1.00 + 25,000*1.02), and $62,200 (40,000*1.00 + 20,000*1.11).*

And then we'll go through the process again and tackle Bob and Carol. Bob has to get somewhere between 15,000 and 45,000 units, as we worked out in the first case and Carol needs to get somewhere between 15,000 and 35,000 units, as we worked out in the second case. We take their two sets of quote-unquote extreme points, which is 15,000, 20,000, or 45,000 for Bob and 15,000, 25,000, or 35,000 for Carol. And again, when we merge those, we find, luckily, three award scenarios and, when we work those out, we get costs of $60,450, $60,150, and $61,200.

Editor's Note:
*As noted, the optimal solution to this sub problem has Bob supplying between 15,000 and 45,000 units and Carol supplying between 15,000 and 35,000 units. Again factoring in the discounts and assuming convexity, we're looking at the award scenarios where we give Bob 15,000, 20,000, or 45,000 units, as per the work we did when we evaluated Alice and Bob, and where we give Carol 15,000, 25,000, or 35,000 units, as per the work we did when we evaluated Alice and Carol. Merging these awards again gives us three award scenarios: where Bob gets 25,000 and Carol gets 35,000 units, Bob gets 35,000 units and Carol gets 25,000 units, and Bob gets 45000 and Carol gets 15,000 units. Their respective costs are $60,450 (25,000*0.99 + 35,000*1.02), $60,150 (35,000*0.99 + 25,000*1.02), and $61,200 (45,000*0.99 + 15,000*1.11).*

And after all of this work, we can conclude that the optimal solution, because we can only take the process so far by hand, is $59,600. But to illustrate a point, if the sourcing professional had instead used one of the built in optimization tools that come with some of today's leading e-Sourcing suites, she could have had this solution on a high-end workstation likely in less than a second with a single button push after defining the exclusion constraints, i.e. Alice can't supply Atlanta, Bob can't supply Boston, and Carol can't supply Chicago, and the award constraints that dictate each supplier must get at least 25% and the award must go to exactly two suppliers.

But more importantly, what we need to note here is that even in this simple problem our optimal solution of $59,600 was roughly 4.4% off of the least optimal solution we worked out of $62,200. Four-point-four percent might not sound like a lot when we're only dealing with thousands but let's imagine instead that we weren't dealing in units but we were dealing in thousands of units. In this case, instead of our buyer losing up to $2,600 on a bad award, which could look good after working out a few scenarios because even if we didn't work them all out I can guarantee you that there are scenarios worse than the ones we worked out, this sourcing professional would be losing $2,600,000. This sort of illustrates why optimization not only makes it easier but is important to apply even if you don't go with the award it spits out but a very similar award. It will save you money if there is money to be saved and I've never encountered a real world scenario where there wasn't money to be saved. And I kind of think that's the view you teach to your students.

Editor's Note:
One example of a worse solution is giving Carol 24,000 units and Alice 36,000. In this case you would pay $62,640 (24,000 x 1.11 + 36,000 x 1.00).

Absolutely.

Sometimes it's auctions, sometimes it's optimization, sometimes it's supplier development - but the one thing optimization does do is it lets you know how much money there is to be saved just on negotiations alone.

Editor's Note: End of Part I

And the interesting thing is that if you are doing all of these calculations by hand, even on a simple problem like you just described, human error has the capability of sneaking in there and really pushing you to a sub-optimal solution.

Definitely.

That's definitely worth mentioning. And, actually, while you were going through the scenario, I was just going through this by hand. What's interesting is that I was saying to myself, "What would an average buyer do in this case? How would an average buyer handle it?" Basically, I was saying if you look at this and you look at all of those three supplier's maximum discounts, you know you would find that Bob would have the best price at $0.99 after taking all the discounts so you give Bob all of the business you can, which is Atlanta and Chicago but not Boston, and then you give the next lowest bidder Boston. And that sounds logical, and it sounds like it would make sense, and produced a good result, but just using that kind of off-the-cuff methodology, that solution was more expensive than the optimized solution. Like you said, when you're dealing in the tens of thousands, that may not sound like much but when you're dealing in awards where optimization really can play a key role in making a profit impact, that's a lot of profit left on the table.

Editor's Note: Giving Bob Atlanta and Chicago gives Bob 45,000 units at $0.99 and the remaining 15,000 goes to Alice at $1.10 per unit. This scenario then has a cost of $61,000 (44,500 + 16,500). Considering the optimal solution was $59,600, this off-the-cuff solution was off by 2.3% and cost you $1,400. If this had been thousands of units, you would have lost $1,400,000 on the award.

That's a good point. Often, if you don't have a lot of constraints, if you start with the discounts, you will usually get to a near optimal solution but very often it won't be the optimal solution. And let's say in this example I broke out freight costs and instead of having to deal with three bids at discounts you also had to deal with tiered freight cost bids. I guarantee your method would likely be off by a lot more. You could get the (near) optimal unit cost, but you can very easily get a very bad freight cost if you happen to go with the most obvious lanes because, in the shipping industry, sometimes you get the best deals on the lanes that aren't as commonly used because these companies need to fill capacity on those lanes so they will give you a better price.

So, how much more complex would we need to get before it became too unwieldy for a standard spreadsheet and you needed optimization technology to arrive at an optimal answer?

Unfortunately, not very much. Let's just assume that you also had Dallas and El Dorado locations, which, let's just say required 40,000 and 25,000 nut and bolt pairings just to have more numbers. Let's further assume that you invited a fourth bidder, call him Daniel, and that you can accept either a dual-source or a tri-source arrangement. Let's add on to that a six-sigma defect rating for each supplier, add the constraint that you need average defects to be less than one tenth of a percent, add capacities on each supplier between 75,000 and 150,000, to sort of insure that not all of them will be able to satisfy 75% of the business, and add an existing contract in place that gives Bob at least 20% of the business because Bob's the only minority supplier. I'm sure you could work this out in a spreadsheet, given a few hours or so, but I wouldn't want to try and I haven't because, being someone who's worked on the underlying models to accurately model this to give you an optimal solution, there is so much hidden complexity there (that) I can guarantee you that just making the assumptions I made above, I'd probably get near optimal going with my methodology for sort-of working out these extreme points and breakpoints and so on, but I could pretty much guarantee I could not get optimal. I'd be lucky to get within a few thousand dollars. And that's if I got really lucky. That's sort of the point where you get thinking, "Well maybe the spreadsheet isn't the best tool".

So, we have four suppliers now, let's double that to eight; we have five DCs (but) let's say we're global and move that to ten and instead of essentially one item, let's say I have ten items in this category. In addition to nuts and bolts, I have screws, nails, hinges, and so on - all basic run of the mill, standard, traditional manufactured parts. And my freight costs are being bid separately by each bidder as well as by five independent freight carriers, and in addition to these tiered bids, each bidder is going to give me a discount if I buy multiple items in bidder defined bundles.

Each bidder will have different bundles then.

Yes, So bidder A says I'll give you a discount if you buy nails and screws, bidder B says I'll give you a discount if you buy nuts and bolts and hinges and so on. This is not something you could work out with a supercharged spreadsheet even in a week, but a very common quote-unquote simple sourcing scenario in many companies. So, this illustrates how a good sourcing tool with pre-packaged optimization is almost necessary but what's the real kicker here is that for this average size problem on today's high end workstations your best in class tools will typically solve this in a minute or two.

So you get your optimal solution and you get it much faster than you could get doing it by hand. If you could even do it by hand.

And, for the problem I just described, I don't think you could.

O.K. That's all great stuff. I'm sure the non-math fans out there in the audience are having their heads spin right now so let's move on. What other type of companies, really what types of companies or supply chain operations can benefit the most from implementing optimization solutions?

Tough question because optimization is not restricted to a type of company or supply chain department or even a supply chain problem - all companies, all departments, and all problems can, at least theoretically, benefit. And what determines the degree of the benefit is, among other things, the size of the company, the amount of the spend associated with the top categories, and the complexity of the problem. The reality is that optimization is going to add another step to your process, which might lengthen it a bit, or a little, in some cases because you have to collect all the data and build a model. It's going to necessitate a tool or service that costs you money up front, and its only going to save what the other e-Sourcing tools in your arsenal miss. So, when you take all of this high level stuff into account, optimization is best suited to categories of moderate or more spend with some complexity either in your cost modeling, constraint requirements, or where there is flexibility in the final award.

For instance, you probably wouldn't use optimization to source office supplies, even if you were a multi-billion dollar corporation sourcing tens of millions of dollars of office supplies annually, since these are commodities that are usually interchangeable from one vendor to another, there are few or no constraints, and the bids are usually standard and simple. One number, because you know your volume in advance and you can tell your suppliers that. Similarly, you likely wouldn't use it to source a manufactured part of high complexity produced by a contract manufacturer with which you have a strategic partnership and shared, and potentially patented, IP. (Intellectual Property) In the first case you'd just use an e-Auction, follow some best practices, invite enough interested suppliers, and basically award to the lowest, or the two lowest, bidders. In the second case, if you did want to move away from that shared partner, who was charging you a lot because they had a stranglehold on shared IP, you wouldn't use optimization at all. You'd instead use an invention-on-demand strategy to design a suitable replacement part and source that to someone else. Optimization might come into the process then if you can share that design with potential suppliers, but up front it's not an optimization problem.

As to where you would use it, you would use it to source high-value core commodities, parts, and materials for which there are capacity constraints, associated risks, and multiple potential suppliers. If you were in food-service, you could use it to source tomatoes. Even though tomatoes are more-or-less a commodity, you can go pick them up in any grocery store, if you're a large international food-service chain, you generally need to lock in your buy in mid-term contracts to ensure supply and to use a dual-sourcing strategy among geographically dispersed suppliers to mitigate risks in the face of unpredictable natural disasters, such as hurricanes or insect infestations. Having two separate distributors that both have agreements in place with the same Florida producers doesn't cut it and that could happen if you just used an e-Auction. Plus, you might want to simultaneously optimize your distribution network since tomatoes are perishable and not all freight options get them from point A to point B in the same amount of time. Speed is of the essence so you don't lose what you've bought.

Another example is if you were a custom computer manufacturer, you could use it to source bare-bone kits. In this case you're modeling a make-vs-buy option. Do you have your contract manufacturers quote for the entire kit, or do you have them just quote the board, processor, fan, chassis, and power supply separately and you build the kit yourself and send it to the customer, or do you look at manufacturers who do partial assembly, i.e. just stick the motherboard and the processor in the chassis but you get your fan and power supply from somewhere else.

A third example would be if you're an international apparel retailer, you could use it to optimize your distribution network and associated logistics costs. Because typically you'd have your manufacturing locations in different low-cost countries, some of which would be under long-term contract, some with a short-term contract, and some that you were considering replacing or adding. You'd have your distribution centers in each country you did business in, some would be fixed physical assets and some rented. And then, of course, you'd have your various transportation options which include air, land, and sea which can often be handled by various providers. And the question is what are the proper manufacturers combined with the proper carriers on the proper lanes from (and to) the right distribution centers to optimize your logistics costs for the next year, two years, five years since you can easily save 20% over a two or three year period with a well designed network, whereas a poorly designed one, too many DCs, too few (DCs), the wrong carriers, the wrong shipping methods will just eat up logistics costs and cancel out all the savings you hope to get by sourcing in low cost countries.

And of course there are a slew of more examples, but this should give you and your listeners a jist of the flexibility of optimization and the types of problems to which it is well suited for. And we could even talk about large combinatorial problems, non-traditional sourcing problems, and regret minimization problems. Some of these get a little advanced. I'm not going to go into these in this podcast since this is supposed to be more of an introductory podcast on optimization, but I do want to refer your listeners to a post I wrote on November 10 of last year entitled "CombineNet IV: BoB's Unique Talents" which sort of describes what some of these advanced optimization problems are.

Editor's Note: Here is a direct link to CombineNet IV: BoB's Unique Talents.

Okay. Now, are there specific types of organizations that are not particularly well-suited to optimization?

Well, going back to the fact that I just said optimization is not restricted to a type of company, department, or problem in that all departments, companies and problems can theoretically benefit, we have to note that what really determines the degree of benefit is the size of the spend, the size of the company, and the complexity. That basically tells me if I'm a small company who does not have at least a handful of moderately complex categories with spend in the seven figure range annually, I'm probably not going to be able to save a lot with optimization over a well run e-Sourcing process because there's just not going to be that much left on the table. Similarly, if I'm a mid-size or large organization that outsources much of my procurement operations, I wouldn't want to use optimization in-house, although I would insist that my providers have that technology for the appropriate categories.

You definitely have to get return on investment if you're going to acquire the rights to use that technology. Do you have any feel for the premium that a purchasing department would pay for optimization technology compared to just a standard e-Sourcing platform?

Well, it really depends upon if you're getting it included in the suite or you're buying a best-of-breed tool or where you're buying an unlimited license vs a category-by-category event (license). Premiums range for the simple built-in tools, some of which are not yet complete, anywhere from $100,000 a year to a few million. It really depends on how much you need. And it usually ends up being a lot less than what you save over the course of the period that you buy a license to use the tool for.

So it's not necessarily the process of optimization that excludes smaller companies but more the cost of the technology then.

Yes, it requires, as you've figured out, a lot of intelligence and a lot of manpower to build these things. It's not a cookie cutter development process. It's not something you can outsource to any low cost country development organization and say "build me an optimizer that allows me to define different models and solve them". There's only a few companies that do it, fewer still that do it well, and fewer still that have specialized sourcing models and most of these have spent millions, if not tens of millions, in R&D to build these tools.

It definitely takes some pretty advanced intelligence to arrive at the conclusion that these platforms do. Is there any way that a supply chain professional can try to verify that the technology came up with the best solution? In other words, how can a supply chain professional defend his or her decisions other than saying "It has to be right. The computer said so!"

Editor's Note: For the sake of brevity, this little rant was left out.

Optimization uses some pretty advanced mathematics. Optimization is not "intelligent", just as spend analysis is not "intelligent". (I wrote a post on Sourcing Innovation on this subject as well on July 30, 2006 entitled There's No Such Thing As Spend Intelligence). Algorithms are not "intelligent". They do what they were programmed to do. Their power lies in the fact that when programmed by experts, they can recreate the expert's most-likely decision under similar circumstances and, on today's top-of-the-line hardware, do more calculations in a few minutes than we humans can do in a lifetime. Furthermore, in our case, if the algorithm can be demonstrated 'sound' and 'complete', we know that if we let it calculate long enough, it will eventually find the optimal answer.

The simple answer is he or she can't, he or she just can not do it because, in reality, for a model of any complexity at all, there is no practical way that a supply chain professional can verify that the technology came up with the absolute best solution. The only way to prove a solution to be the absolute best optimal solution is to prove that there is no other solution that is better than the optimal solution. And this requires essentially evaluating at least fragments of every other possible quote-unquote "extreme" or "breakpoint" solution, where these terms have some mathematical meanings, and proving that not one of these other solutions can be better. For models of even moderate complexity, this could translate into thousands or tens of thousands of potential solutions to be evaluated.

The best a supply chain professional can do is illustrate that it is likely optimal, or at least near-optimal since good solvers will short-circuit when a solution is proved optimal within a small tolerance, usually between 1/10,000th and 1/100th of a percentage point. And these optimizers do this, specifically these optimizers that use mixed-integer linear programming techniques, by comparing the solution to the best the solution could be based upon previous calculations.

Now the way a sourcing professional might try to demonstrate near optimality is to compare the solution to the solution they would have chosen by hand, the solution a spreadsheet model comes up with, and the solution defined by a simple greedy algorithm that simply uses best-bids or best-discounts from the auction or RFP for each award. And when they see that the optimal solution is better, sometimes significantly better, they can demonstrate that it may not be optimal, but it's likely near optimal because the optimizer reported that it believed the solution to be within 99.99% of optimal, couldn't find a better one, and it's better than all of the solutions they are able to find by other methods.

So what I sort of recommend is that they take care to choose an optimization solution that has the ability to indicate whether or not the solution algorithm ran to completion, or was it short-circuited either due to a weak tolerance setting or user defined time-out, and, can tell you in the worst case, using some advanced mathematical techniques, how far off the solution could be. Typically this number will be larger than the solution really is off because the only way you could prove an exact lower bound is to have the exact optimal answer. But there are techniques to prove that if I'm at $10,000,000 it can't be less than $9,990,000, and here are the equations, which are usually too numerous to understand, why. And to have it report, every time, in the worst case, how far off it is, especially when you have it report back after a few seconds and then after a few minutes on a model that might take an hour or two. So that's pretty much the best you can do, verify by basically using solutions that produce inferior answers and then selecting solutions that tell you how bad it is in the worst case, which is usually 1/10th, 1/100th, or 1/1000th of a percentage point.

Editor's Note: Generally speaking, the way optimization works, is at any given time, it can tell you how far off the current solution is in the worst case, but it can't necessarily tell you whether or not it's within a given range without finding the optimal solution. Thus, be sure not to set your tolerances too low, or you might find that even though you find a near optimal solution quickly, and the optimal solution in a timeframe that's the same order of magnitude, it might have to evaluate all possible solutions to prove that the solution is within 0.0001%. It might prove that it's within 0.0002% in a minute, but take two hours to evaluate the branch of the branch-and-bound tree (that divides the solution space) that allows it to assertively state it's within 0.0001%.

OK. Many of us have heard terms like "expressive bidding" and "combinatorial auctions." What are the relationships between the processes that these terms represent and optimization?

I'm going to start with "combinatorial auction". This refers to the fact that you are auctioning a number of items or item bundles and a number of bidders are placing bids on those items or item bundles. It's combinatorial since you have m, when m is the number of items, and n, where n is the number of bidders, and, thus, m x n bids, and the award for each item, absent any constraints, is to the lowest bidder. In other words, only m of the m x n bids receive award.

If there are no constraints, there is no relationship to optimization since you can come up with a deterministic algorithm, take the lowest bid for each item, and there's your award. However, if there are constraints, you can use a quote-unquote "combinatorial auction" to collect bids and then you use optimization to determine the final award. In this situation, the relationship is that the combinatorial auction serves as the front-end data collection mechanism, which captures both the bids and the constraints, and the back-end optimization solution determines the award subject to the bids and the constraints, such as dual-source 80-20 to mitigate risk, respect supply capacity, and insure the average quality of the sourced items, say measured as average defects per part-per-million (*PPM*), is 99.99%.

"Expressive Bidding" is a term invented by CombineNet, and then 'adopted' by Emptoris, to describe a methodology of capturing bids that will eventually be used in their optimization models. It's really a fancy way of saying "this a bid collection format that allows you to capture fixed costs, variable bids, and discounts, in a tiered format if necessary, or any other way you need to as to accurately represent the quotes your supplier gave you and build a correct sourcing scenario". Basically, what happened is CombineNet is one of many companies that not only started with VC support, but still has VCs involved. And these Venture Capitalists love to protect their investments, or at least take steps to make it look like they are, and this usually translates into a push to have all of their companies patent everything under-the-sun that has not been patented already. And since you can't patent algorithms since algorithms are mathematics and mathematics is a law of nature, the only way that software can apply for a patent is to basically come up with a business process methodology. So, coming up with "Expressive Bidding" gives them a business process methodology that they can patent and it also gives them something they can market that's a unique term compared to their competition.

So, like combinatorial auctions, it's a front end, but much more flexible than a typical combinatorial auction in that you can pretty much enter bids anyway you can think to define them and then it does behind-the-scenes mathematical translations to put all the bids in an apples-to-apples bidding fashion versus the applies-to-oranges format the suppliers will give them to you in, in an effect to hide the true cost of doing business with that supplier, usually. So, does that answer your question?

Yes. That's definitely pretty interesting. Now what else should supply chain professionals know about optimization?

Well, I'm going to say it's not what they should know about optimization, it's best to leave what's under-the-hood to the experts because it is so complicated. What they need to know is the utilization of decision optimization for strategic sourcing. Like any good software tool, the point is not to replace the sourcing professional in award allocation decisions, but to augment his or her capabilities. It's usually not about the solution the optimizer produces, but the solution it leads you to. And not about the lowest possible cost you can squeeze out of your supply base, but the value you can create and the knowledge of how much value there is there to capture.

For a given scenario, there might be more than one optimal solution, and, more importantly, there might be dozens of solutions that are near optimal within a percent or two. If the optimal solution requires switching three suppliers, whereas a near optimal solution, only 1% off, only required switching one supplier, you might be better off going with that specific near optimal solution, which is going to be more palatable to you and your boss. No one likes to switch half or more of their supply base at one time.

Or, for example, the optimal solution might exclude a supplier that you feel could be strategic, in this case you could use the tool, create another what-if scenario, force that supplier to get 20% of the business, and maybe find that you're only going to pay two extra percentage points, but it's going to come with the expectation of a lower year-over-year cost in the future because they've committed to implementing new technology.

And as long as this near-optimal solution still saves you a significant amount of money, it might be better for you as a company than the optimal solution because models are mathematics and math can only basically take into account anything that you can objectively quantify in a hard way. There's always going to be soft strategies that, in the long term, will give you more value than just what the hard answer gives you in the short term and only a good sourcing professional will understand that you use the tool to understand what the absolute best cost is, what the cost of your preferred scenario is, and how you work either inside your company or with your suppliers to adopt a solution that's not very far off from what you want but very close to the optimal solution.

And, to kind of jump back a bit, this really demonstrates the importance of that what-if? requirement that I gave for decision optimization and strategic sourcing decision optimization in particular. The tool needs to give you the ability to create multiple scenarios, potentially creating one off of a copy of another but adding a few constraints or removing a few constraints, including one user-defined award scenario, to sort of help you understand what the lowest cost solution is, what the lowest cost solution is within your existing supply base, and what the lowest cost solution is if you give a percentage of the award to potentially strategic suppliers or minority suppliers so that you can do something with your CSR initiatives. It's about using the tool to understand the solution-space you are working in and getting the best overall solution and not just the best cost solution. That's something you can't do with a spreadsheet. You just can't create multiple scenarios, have them solve almost instantaneously, and compare side-by-side the cost and benefits of each of those scenarios, usually in the same report. It's a tool that helps you get to the best solution by telling you what the absolute best you can do is and no other tool can tell you what the absolute best you can do is and that's, that's, the true power. Not what's under-the-hood, but what it does for you.

Editor's Notes: CSR refers "Corporate Social Responsibility" in this instance. For more information, start with the Wikipedia overview. Tim Minahan of Supply Excellence has been blogging a lot on the topic of social responsibility as well.

Well, Michael, we have just a few minutes left for you to tell us a little bit about your company and how someone can get in touch with you.

Well, as you noted in the intro, my company is ToP KaTS Consulting, at www.topkats.com, and basically it specializes in the sourcing and procurement aspects of supply chain from a technology and process perspective. Just go to the website and click on corporate for the Contact information. You can also reach me through the about or welcome section of my blog, Sourcing Innovation blog.sourcinginnovation.com, and the associated free resource site I maintain, www.sourcinginnovation.com, that maintains an ever-growing list of various resources useful to sourcing and procurement professionals, such as links to relevant blogs, publications, journals, vendor resource centers, societies, analyst firms, and solution providers.

If you're looking to build or adopt a technology solution or to streamline or revise your affiliated sourcing processes to allow you to get your benefits from today's technology, that's pretty much what I do and what my company does. Optimization is an important component of most solutions, but it's not the only component. And if you don't have the other appropriate components that both lead into it and flow out of it, you won't get the most bang for your buck. Basically, Supply Chain Technology, falls into the sourcing, procurement, or logistics & distribution buckets, is very broad and intensive because in each these buckets I can name at least half-a-dozen distinct solution types. And my company tries to help you figure out where optimization falls in each of these buckets and where it can benefit you. I tend to work a lot with companies building solutions, as well as companies implementing them.

So, you work with the vendors as well.

Yes. I actually like to work with the vendors who are trying to build these solutions to help the vendor build a solution that will benefit multiple companies versus just working with a single company to give them a custom solution because I think the answer is on-demand Software-as-a-Service technology that the vendor continually updates to help you. But I won't work with a company if it says hands-off; you can't talk to our customers to find out what they really need. So, it's trying to find that balance. Sometimes (*I work*) with a really big company to understand what the problems are, to not only help them pick the right solution, but to help them identify where they need to work with a provider to build a better solution or build one in house, and then sometimes with a vendor that's trying to build a better solution for a certain vertical.

Well, there you have it folks. Hopefully, you know a little more about optimization because if you have to embrace it at some point in your supply chain career, it certainly helps to get a jump-start because there is certainly a lot to learn about it.

Yes. Definitely. And I'd also recommend that your interested listeners check out some of the optimization posts on my blog, they just have to scroll down to the category archives on the right-hand sidebar and select "Decision Optimization", and they should watch for the forthcoming wiki-paper, over on the e-Sourcing Wiki, "Sourcing Decision Optimization", which will probably be up in a week or two.

Editor's Note. It will likely be here.

Okay, great! So that wraps it up for today. Michael, thank you for joining me today.

You're very welcome. I hope this helps your students and your listeners.

I'm sure it will.

And on behalf of Next Level Purchasing, I'd like to say a special thank you to all of our listeners out there. I hope the information in today's program helps you all in having a more successful and rewarding purchasing and supply management career. Good day!

Manual Optimization Calculation

Bidders	Bid	Discount Amount	Discount Volume
Alice	1.10	0.10	20000
Bob	1.05	0.03	10000
		0.06	20000
Carol	1.11	0.09	25000

Limits	Least amt. supplied due to dual source (min. 25%)	Largest amt. supplied
Alice	15,000	40,000
Bob	15,000	45,000
Carol	15,000	35,000

Location	Nuts & Bolts	Possible Suppliers
Atlanta	20000	Bob, Carol
Boston	15000	Alice, Carol
Chicago	25000	Alice, Bob
Total	60000	

Alice and Bob

Possible Solutions	Supplier		Supply	Price/ unit	Cost/ Supplier
Solution 1	Alice		15,000	1.10	16,500
	Bob		45,000	0.99	44,550
		total	60,000		**$61,050**
Solution 2	Alice		20,000	1.00	20,000
	Bob		40,000	0.99	39,600
		total	60,000		**$59,600**
Solution 3	Alice		40,000	1.00	40,000
	Bob		20,000	0.99	19,800
		total	60,000		**$59,800**

Alice and Carol

Possible Solutions	Supplier		Supply	Price/ unit	Cost/ Supplier
Solution 7	Alice		25,000	1.00	25,000
	Carol		35,000	1.02	35,700
		total	60,000		**$60,700**
Solution 8	Alice		35,000	1.00	35,000
	Carol		25,000	1.02	25,500
		total	60,000		**$60,500**
Solution 9	Alice		40,000	1.00	40,000
	Carol		20,000	1.11	22,200
		total	60,000		**$62,200**

Bob and Carol

Possible Solutions	Supplier		Supply	Price/ unit	Cost/ Supplier
Solution 4	Bob		25,000	0.99	24,750
	Carol		35,000	1.02	35,700
		total	60,000		**$60,450**
Solution 5	Bob		35,000	0.99	34,650
	Carol		25,000	1.02	25,500
		total	60,000		**$60,150**
Solution 6	Bob		45,000	0.99	44,550
	Carol		15,000	1.11	16,650
		total	60,000		**$61,200**

Optimal Solution		Solution 2	
Alice	20,000	1.00	20,000
Bob	40,000	0.99	39,600
total	60,000		**$59,600**
Least Optimal Solution		Solution 9	
Alice	40,000	1.00	40,000
Carol	20,000	1.11	22,200
total	60,000		**$62,200**
		Cost Difference	**$2,600**