Simulation Based Optimization of Supply Chains with a Surrogate Model

Xiaotao Wan, Joseph F. Pekny[*], Gintaras V. Reklaitis
School of Chemical Engineering, Purdue University
West Lafayette, IN, 47906, U.S.A

Abstract

Simulation is widely used in the decision making processes associated with supply chain management. In this paper, we present an extension of the simulation based optimization framework which has been previously proposed for analyzing supply chains. The extension consists of the iterative construction of a surrogate model based on systematically accumulated simulation results to capture the causal relation between the key decision variables and supply chain performance. The decision variables can then be optimized using the surrogate model in place of individual simulation runs to economize on the overall computational effort. The extended framework is illustrated using a small example and then applied to optimize the inventory levels in a three stage supply chain.

Keywords: simulation based optimization, supply chains, surrogate model

1. Introduction

Monte Carlo simulation is one of the most important tools for analyzing supply chains in the presence of uncertainties. Compared to analytical techniques, simulation provides the flexibility to accommodate arbitrary stochastic elements, and generally allows modelling of all of the complexities and dynamics of real world supply chains without undue simplifying assumptions. However, as a descriptive method, simulation can only be used to perform optimization through "what if" case studies involving the comparison of several cases or scenarios.

The "what if" case approach is particularly ineffective in optimizing continuous decision variables (for example, inventory levels of each entity in a chain). Thus having a method to perform optimization efficiently in continuous decision spaces is of importance. Even for scenario analysis, such a method is a necessary because inventory optimization generally appears as a sub-problem within each scenario. For instance, consider the problem of determining the best network design from several network candidates. One performance criterion for assessing the quality of a design is to evaluate the material hold-up under this particular design. To accomplish this, an inventory level optimization problem must be solved.

Optimization in continuous decision space can be carried out through simulation based optimization (Fu, 2002). Under this framework, an optimization module is coupled with

[*] Author to whom correspondence should be addressed : pekny@purdue.edu

a simulation model. The optimizer searches the decision space systematically using the simulation model as function evaluation. While this basic framework is well understood, simulation based optimization is not yet widely used in supply chain analysis practice mainly because of its extremely high computational requirements. Typically the optimization search process is slow and inefficient due to the presence of statistically and numerically caused noise in the simulation output as well as the fact that each simulation run itself takes significant execution time.

This paper presents an extension to the concept of simulation based optimization by introducing a surrogate model for use in optimizing continuous supply chain decision variables, with the aim of mitigating the computational burden of existing methods. Section 2 describes the proposed framework; section 3 validates the framework and applies it to a case study and conclusions are drawn in section 4.

2. Simulation Based Optimization Framework

An overview of the proposed extended simulation based optimization framework is shown in Figure 1. Unlike classical response surface methodologies which build local surfaces, the surrogate model fits a single surface for the whole domain. Essentially, this framework iteratively builds a series of temporary surrogate models (or meta-models) by graduately increasing the number of sampling points in the decision space, and uses these temporary models to guide the incremental sampling. The iterative refinement process is terminated when a suitable stopping criterion is met. The surrogate models are constructed with LSSVM (least square support vector machine). The incremental sampling is carried out using DACE (design and analysis of computer experiment) so as to balance exploration of the decision space against the exploitation of the already extracted model information. A domain reduction technique is also incorporated in each step to reduce the complexity of the surrogate models.

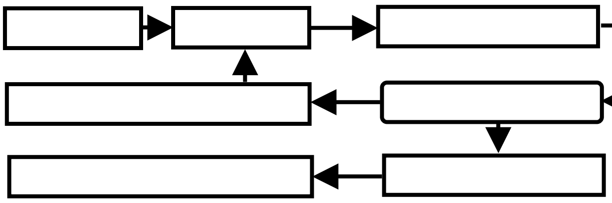

Figure 1. Simulation based optimization framework

2.1 Sampling

The initial sampling is performed with the Latin hypercube sampling (LHS) technique. LHS disperses the sampling points as evenly as possible in the whole domain, thus providing a good basis for building the first temporary surrogate model when there is no prior knowledge of where the good solutions are located.

2.2 Domain reduction

The number of points that must be sampled and the complexity of the surrogate model are roughly proportional to the size of the domain. Domain reduction can significantly decrease the overall computational burden by cutting off the regions where good solutions are unlikely to exist. This is achieved by comparing the average performance of different regions through fitting a regression tree to the simulation results.

2.3 Model Construction

One of the crucial decisions in the proposed framework is to use LSSVM (Van Gestel et al., 2002) to construct the surrogate models. LSSVM has two important features: it is capable to efficiently extract functional relations from noisy data based on structure risk minimization (Vapnik, 1999) as well as to capture complex relations embedded in data because LSSVM corresponds to fitting a linear regression with an infinite number of basis functions. The actual fitting procedure consists of the solution of a hierarchical optimization problem involving three levels of parameters in the LSSVM model. The optimization is performed under the Bayesian evidence framework, which establishes analytical relations between data and parameters of each level.

2.4 DACE model and incremental sampling

With a temporary surrogate model, the noise in simulation data is essentially removed because the LSSVM model gives a deterministic performance prediction at each point in the decision space, where the prediction approximates the expected performance value of the corresponding point. With this observation, the DACE model (Sacks et al., 1989), originally developed for deterministic computer experiments, is introduced to perform an experimental design. Two design criteria are considered: maximizing Bayesian information and maximizing expected improvement (Sasena et al., 2002). By combining the two criteria together, the incremental sampling is controlled such that both global and local sampling are considered with emphasis being gradually shifted form global exploration to local exploitation.

2.5 Stopping criteria

Several criteria are implemented to control the termination of the loop in Figure 1. The two most important criteria are absolute expected improvement and relative improvement. These criteria will stop surrogate model building once they detect that further improvement drops below a prescribed value. In addition, total simulation time and total number of sampled points are also used as stopping criteria.

3. Case Studies

3.1 Framework validation

We first validate the surrogate model framework before applying it to supply chain analysis. The purpose of the validation is to show that the surrogate model is able to capture the underlying input and output relations over the decision space of interest by only observing the performance values at sampled points. For this purpose, we choose the ‘bra’ function (Dixon and Szegö, 1978) to return the performance values in place of the simulation; thus the input and output relation of the data is known. If the framework is effective, then the surrogate model it generates should adequately represent the structure of the ‘bra’ test function. That is, the surrogate model should indicate the neighbourhoods in which the optimal solutions are located.

Figure 2 shows the contour profile of the ‘bra’ function, which has three local minimums in the domain of interest. For each sampling point x, a performance value bra(x)+N(0,900) is returned, i.e., random noise following a normal distribution N(0,900) is superimposed on the ‘bra’ function. The scaling is such as to cause the average noise to signal ratio to be as high as 0.3. Figure 3 shows the contour profile of the resulting surrogate model. Note that the initial LHS uses 20 points and the final model uses 29 sampling points where each point incurs one simulation run. It is noteworthy that, even with relatively high level of noise presented in the data, the underlying structure of ‘bra’ function is still captured with the three local minimums clearly identified. Although the location of the minima points of the surrogate do not agree exactly with those of the test function, the results are absolutely acceptable since these minimum points do correspond to good solutions (the optimal objective function value is no larger than 10 of the true optimum 0.4) of the original function (Figure 2). Computational efficiency, the major strength of our proposed framework, is also manifested by this simple example. For purpose of comparison, we have applied the well-known algorithm SPSA (Spall, 1998), which computes gradient directly from simulation results and search along the steepest decent direction, to the test function. The results obtained show that even with smaller noise level, N(0,100), and a local optimum as the initial point, 500 steps of search (where each step incurs two simulation runs) with the search yields an objective function value 30 larger than the true optimum. Further increasing the search to 5000 steps does not improve the solution quality.

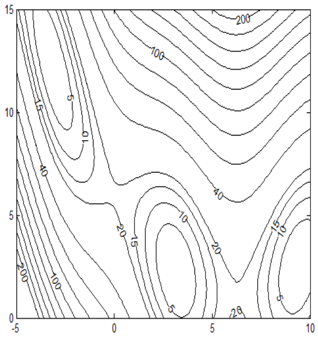

Figure 2. Contour of ‘bra’ function for ‘bra’ function

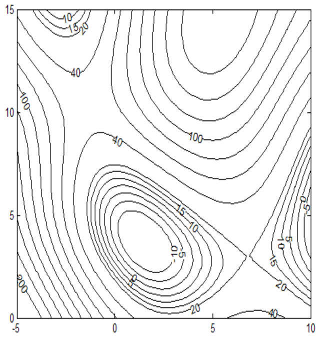

Figure 3. Contour of the surrogate model

3.2 Optimising inventory levels for a supply chain

The proposed simulation based optimization framework is next applied to a three stage supply chain. The network structure is represented in Figure 4. Product 1 and product 2 are produced in the node production 1, and then shipped to the node production 2 where further processing is carried out. The finished goods from the node production 2 are transported to warehouses to meet customer demands. Warehouse 1 only stores product 1 and warehouse 2 only stores product 2.

The transportation times between nodes are random. The operation of each production node is modelled as a simple queuing process consisting of a single machine whose service times follow exponential distributions. Note that there is no additional implication for the optimization if other more complex methods like MILP are used in the simulation to model the production. The customer demands are random both in terms of the interarrival intervals and quantities. The unmet demands are fully backlogged by warehouses. The supply chain operates under a base stock policy. Further, we assume that the chain operates under a pure "pull" mechanism, thus neither demand forecasting nor production planning are employed in this case study. The objective of the optimization is to minimize the backlogging costs at warehouses and the holding costs at all nodes by setting the ten base stock levels (each production node has inventories for raw materials and products).

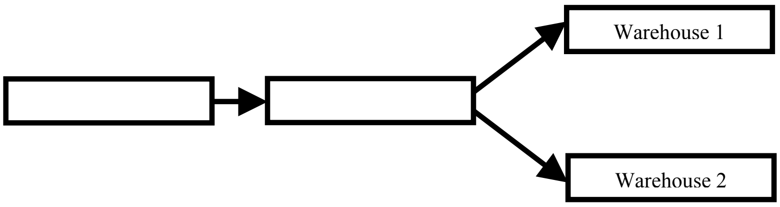

Figure 4. The supply chain network structure

The two products have quite different characteristics. Demand quantities for both products have the same triangular distribution Tri(4,8,6), but the demand arrival intervals for product 1 follow a Gamma distribution Gamma(12,3), while that for product 2 follows a Gamma(4,3). Table 1 shows the holding costs and backlogging costs at the different locations of the supply chain. Note that h_{i1} i=1, 2 are the holding costs of raw materials at node production i, h_{i2} $i=1,2$ are the holding costs of products at node production i, h_{wj} j=1,2 are the holding costs at warehouse j, and b_{wj} $j=1,2$ are the backlogging costs at warehouse j. Both products are produced in batch mode with a fixed batch size of 4. Table 2 lists the average processing time considered, where t_{ij} i=1, 2, j=1, 2 represent average processing time of product i at production node j. Further, two production queuing modes are considered—mode 1 is simply first in first out (FIFO), while mode 2 gives priority to product 1, i.e. always produce product 1 if both products have demands in the queue.

Table 3 shows some of the computational results. The subscripts in the headings follow the conventions introduced in table 1; CIMJ, I=1,2,, J=1,2 is the combination of average processing time in table 2 with the two production modes; e.g. C1M1 indicates the production time of case 1 under the FIFO mode. Comparison of production mode 1with production mode 2 shows that production mode 1 incurs a higher cost; the intuitive reason is that FIFO leads to the expensive and fast moving product 1 having to undergo longer production lead time by waiting in the queue for machine time. The results also reveal that the average processing time has significant effect on the total cost. This is manifested by the much higher cost of case 1 which has longer processing time for product 2. Again, comparison with SPSA indicates that the surrogate framework is much more efficient: 1000 simulation runs of SPSA fail to find solutions with costs below 1000 for C1M1 and C1M2, while the surrogate framework on average incurs 600 simulation runs to get the results shown in table 3; although SPSA locates solutions for C2M1 and C2M2 with qualities equivalent to the surrogate framework, SPSA needs 700 simulation runs on average for both cases, while the surrogate approach only uses 350 runs for C2M1 and 500 runs for C2M2 respectively.

Table 1. Holding costs and backlogging costs

	h_{11}	h_{12}	h_{21}	h_{22}	h_{w1}	h_{w2}	b_{w1}	b_{w2}
Product 1	3	3	6	6	9	--	60	--
Product 2	1	1	2	2	--	3	--	20

Table 2. Average processing time

	t_{11}	t_{12}	t_{21}	t_{22}
Case 1	1	1	2.2	2.2
Case 2	1	1	2	2

Table 3. Base stock policy

		I_{11}	I_{12}	I_{21}	I_{22}	I_{w1}	I_{w2}	cost
C1M1	Prod 1	11.8	21.5	5.5	46.5	19.5	--	722
	Prod 2	26.4	77.6	25.4	75.3	--	73.8	
C1M2	Prod 1	23.6	22.1	2.5	11.7	9.0	--	603
	Prod 2	1.3	40.7	60.9	75.6	--	98.7	
C2M1	Prod 1	0	22.2	5.3	14.0	7.5	--	378
	Prod 2	0	18.1	24.5	51.3	--	44.8	
C2M2	Prod 1	0	8.3	0	0	16.2	--	282
	Prod 2	27.6	55.1	5.2	10.5	--	66.3	

4. Conclusions

An extension to the simulation based optimization framework is proposed for optimizing supply chains under uncertainty. This framework employs a surrogate model to extract structure information from noisy simulation results; the supply chain decisions can then be efficiently optimized using this surrogate model. This framework is shown to correctly capture the gross structural features of a test function superposed with large random noise and to use simulation runs parsimoniously. Application to optimize the base stocks for a three stage supply chain further demonstrates the power of the method .

References

Dixon, L.C.W. and G.P. Szegö, 1978, Towards Global Optimization 2. Eds, L.C.W. Dixon and G.P. Szegö, North-Holland, Amsterdam, 1.

Fu, M.C., 2002, INFORMS Journal on Computing, 14, 192.

Sacks, J., W.J. Welch, T.J. Michell and H.P. Wynn, 1989, Statistical Science, 4, 409.

Sasena, M.J., P. Papalambros and P. Goovaerts, 2002, Engineer Optimization, 34, 263.

Spall, J.C. 1998, IEEE Transactions on Aerospaces and Electronic Systems, 34, 817.

Van Gestel, T., J.A.K. Suykens, D. Baestaens, A. Lambrechts, G. Lanckriet and B. Vandaele, 2002, IEEE Transaction on Neural Networks, 12, 808.

Vapnik, V.N., 1999, IEEE Transaction on Neural Networks, 10, 988.