Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

SIMULATION OPTIMIZATION RESEARCH AND DEVELOPMENT

Royce O. Bowden

Department of Industrial Engineering
Mississippi State University
Mississippi State, Mississippi 39762, U.S.A.

John D. Hall

TASC, Inc.
700 Boulevard South, Suite 201
Huntsville, Alabama 35802, U.S.A.

ABSTRACT

Simulation optimization is rapidly becoming a mainstream tool for simulation practitioners. Simulation optimization is the practice of linking an optimization method with a simulation model to determine appropriate settings of certain input parameters so as to maximize the performance of the simulated system. Requirements for an automated simulation optimization tool for practitioners were formulated in the early 1970s and the first widely used commercial product appeared in 1995. In this paper, the authors identify six domains that are common to any automated simulation optimization tool. The domains are Methods, Classification, Strategy and Tactics, Intelligence, Interfaces, and Problem Formulation. These domains are the cornerstones for a unified strategy for simulation optimization and should guide future research in the field and development of next generation simulation optimization tools. This paper describes the six domains, presents recent research, and discusses research issues for two-phased optimization techniques.

1 SIMULATION OPTIMIZATION

Simulation can be used to determine the state of certain controllable inputs to a system that will cause system outputs to be at their most favorable or optimal condition. Foundational work in describing the requirements for software that automatically optimizes simulated systems can be attributed to Dennis E. Smith (1973a, 1973b). Smith suggested an automated optimizer would be a computer application external to the simulation model. The optimizer would use model inputs and outputs as well as user supplied information to determine an optimal solution. The optimizer would possess the requisite intelligence to determine an appropriate optimization method for a given problem.

Dennis Pegden and Michael Gately were among the first to report the linking of an optimization algorithm with commercially available simulation packages. They linked a direct search technique, the Hooke-Jeeves pattern search method, with GASP IV and SLAM models and illustrated the utility of combining an optimization algorithm with a simulation language (Pegden and Gately, 1977 and 1980). Their optimization method was not readily supported or explained by software vendors and simulation practitioners did not commonly use it.

Recently, several simulation software vendors have introduced optimizers that are fully integrated into their simulation packages. These simulation optimizers use newer direct search techniques such as evolutionary algorithms (e.g. genetic algorithms and evolution strategies), scatter search, and simulated annealing. The optimizers include SimRunner for ProModel, OptQuest96 for Micro Saint, and Witness Optimizer for Witness. Simulation practitioners now have access to robust optimization algorithms and they are using them to solve a variety of “real world” simulation optimization problems (see Akbay, 1996). Although these simulation optimization packages are based on better optimization algorithms than those available in the late 1970s (Schwefel, 1995), further improvements can be made to this important area of simulation.

2 UNIFIED STRATEGY FOR CONTINUED RESEARCH AND DEVELOPMENT

The majority of the published research on simulation optimization focuses on a single aspect of simulation optimization without considering the subject as a whole. A framework is needed that unifies research and development across all relevant domains — the component tools, techniques, and strategies associated with simulation optimization. The synergy created by this systems view of simulation optimization can lead to better optimization tools for practitioners.

Using Smith’s requirements for an automated simulation optimizer, the authors derived six distinct domains to address when developing simulation optimization tools (Hall, 1997 and Hall and Bowden, 1998). Figure 1 presents the six domains of simulation optimization.

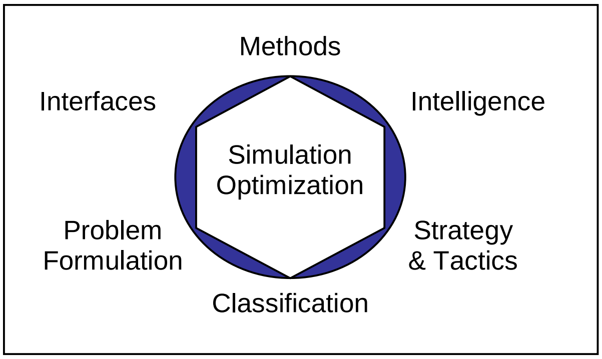

Figure 1: Domains of Simulation Optimization

The Problem Formulation Domain addresses the construction of the objective function and constraints to guide the optimizer. A poorly formulated problem can negate the effectiveness or efficiency of the best optimizer. Therefore, this domain considers tools to assist the user in designing appropriate objective functions and constraints.

The Methods Domain addresses those optimization methods used to optimize simulated systems. Most simulation optimization research falls into this domain by addressing the development and application of specific methods to optimize simulated systems. Recommendations are often made for improvements to optimization algorithms that will improve the method's performance for a specific situation. Carson and Maria (1997) present a recent summary of methods used for simulation optimization.

The Classification Domain addresses the analysis and classification of a given optimization problem. Accurate classification is important for the optimization tool to select the appropriate optimization method and strategy (Figure 2). Classification can depend on the types of decision variables (integer, real, or logical), number of decision variables, topology of the response surface, the variance of the simulation model's output, and number of available runs of the simulation model.

The Strategy and Tactics Domain addresses the employment of simulation optimization in order to make the most efficient use of computing resources and increase the accuracy of the observed optimal solution. Strategic issues may consider the optimization method or methods selected for a class of problems. Tactical issues consider the use of metamodeling techniques, variance reduction techniques, multiple comparison tests, etc. to enhance the efficiency or accuracy of the search.

The Intelligence Domain considers the intelligence embedded in the solver to select the strategic approach that will be used for an optimization study. It also considers the intelligence that guides the tactical employment of various tools based on the problem classification.

The Interfaces Domain addresses both the interface between the optimizer and the user and the interface between the optimizer and the simulation model. Since simulation optimization tools should be designed for the general simulation practitioner, effective interfaces are needed for both the user and the user's model.

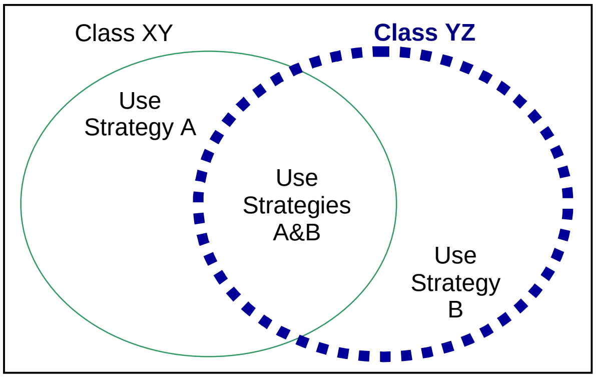

Figure 2: Problem Classification and Strategy Selection

3 AN EXAMPLE OF COMBINING DOMAINS FOR BETTER OPTIMIZATION SUPPORT

The example problem used to illustrate a more unified approach to simulation optimization is based on a scenario faced by an appliance manufacturer designing a pull production system. The objective is to determine the number of kanbans and corresponding trigger values necessary to achieve anticipated production goals while minimizing work in process. There are 33 decision variables that represent the number of kanbans and corresponding trigger values for each product being manufactured in a two-stage pull production system.

A discrete event simulation model of the proposed production system was built and interfaced with optimization routines to seek optimal values for the 33 decision variables. The optimizer evaluates solutions by averaging output results from four independent replications of the simulation model. The objective function to be maximized is:

$$f(\boldsymbol{a}) = C_1(AvgPct + MinPct) - C_2(TK_1) - C_3(TK_2)$$

where $\boldsymbol{a}$ is the solution vector that represents the number of kanban cards and corresponding trigger quantities for each product flowing through the production system, $AvgPct$ is the average throughput percentage (percentage of demand satisfied) for all product types coming off the assembly lines, $MinPct$ is the minimum throughput percentage for all product types coming off the assembly lines, TK_1 is the total number of kanban cards in the first stage of the pull production system, TK_2 is the total number of kanban cards in the second stage of the pull production system, and C_1, C_2, and C_3 are coefficients that reflect the level of importance (preference) assigned to each term by production planners. Using this function, the objective is to find a solution that will maximize throughput while minimizing the total number of kanban cards in the system. Note that there are 4.81 X 10^{41} unique solutions to this problem. See Bateman *et al*. (1997) for additional details.

The two-stage pull production system problem was originally solved using a single–phased strategy. A single-phased strategy is defined as using only one method to conduct the search for an optimal solution. To this end, the Hooke-Jeeves (HJ) pattern search method (Hooke and Jeeves, 1961) was used as a stand-alone search method. The standard Hooke-Jeeves pattern search algorithm consists of one-variable-at-a-time exploratory moves about a base point solution to determine an appropriate direction of search (pattern). Following the exploratory search, a series of pattern moves are made to accelerate the search in the direction determined in the exploratory search. Exploratory searches and pattern moves are repeated until a termination criterion is met. See Reklatis *et al*. (1983) for a detailed description of the HJ method. For this pull production kanban-sizing problem, the HJ method quickly converged on poor solutions.

Another single-phased strategy based on the evolution strategies (ES) algorithm was also used to solve the problem. Evolution strategies (ES) is a population-based direct search method requiring only a function output and does not require information on derivatives of the response surface. ES combines and mutates parent solutions in the population to produce offspring solutions during its search. An ES is able to avoid converging on local minima and is well suited for problems of high dimensionality (Biethahn and Nissen, 1994). The ES algorithm used in this study is the (μ,λ)ES as described in Hall *et al*. (1996) and employs the self-adapting feature described in Back and Schwefel (1993).

The ES method found very good solutions to the kanban-sizing problem. However, it required 4200 calls to the simulation model (approximately 24 hours of run time on a standard personal computer). Although the cost of operating a personal computer for 24 hours to find the solution was miniscule compared to the reduction in inventory-carrying costs produced by the solution, there are situations where analyst need good solutions in a more timely fashion.

3.1 Unification of Strategies, Tactics, and Methods

The above results are typical for a single-phased strategy. The more globally oriented search algorithms that are less likely to be trapped by local optima generally require longer search times. The algorithms requiring shorter search times tend to become trapped by local optima. Therefore, the authors have researched strategies and tactical tools for making simulation optimization more efficient with respect to the number of required simulation calls and solution quality.

One strategy is to use a locally oriented search method to augment a globally oriented search method. In a two-phased strategy, the globally oriented method first conducts an exploration search and then a second method conducts a more locally oriented search (exploitation). Using this scheme, the ES method could be used in the exploration phase and the HJ method could be used in the second or exploitation phase. This strategy promotes the strength of the ES to find the most promising region in the solution space while avoiding its weakness of fine-tuning the search to an optimal solution. The strategy also promotes the strength of the HJ method to locate the nearest local optimal solution. This type of strategy has been suggested in the literature; however, only one example has been found where it was actually employed (Myung *et al*., 1995). In Myung's paper, an evolutionary programming algorithm is used to find the region of the global optimum followed by a gradient based search method to find the optimum.

Another way to improve the efficiency of simulation optimization is a tactical tool based on an archival database. All solutions and their fitness scores evaluated during the search are recorded in the database. Before any solution is evaluated, the database is searched. If the solution was previously evaluated, the fitness score recorded in the database is passed to the optimizer so the model is not called. The literature reports several uses of this strategy.

3.2 Applying a Two-Phased Strategy

The two-phased strategy was applied by selecting an intermediate solution produced by the ES search as the transition solution, $\boldsymbol{a}_t$. The idea is that the transition point is near the region of the global optimum, although there is no such guarantee. The second phase consists of applying the HJ method using $\boldsymbol{a}_t$ as the starting base point.

Figure 3 presents the single-phased (4,28)ES search over its 150 generations that required 4185 simulation calls while using the archival database. The database saved 15 simulation calls from being executed during the ES search. The curve follows a characteristic pattern of rapid initial growth as the ES finds regions of good solutions followed by much slower growth as the ES tunes its strategy parameters and converges to a solution. For this problem, the best solution of 110 kanbans found by the ES meets the manufacturer's throughput requirements. This solution was found during generation 135 with 3781 simulation calls.

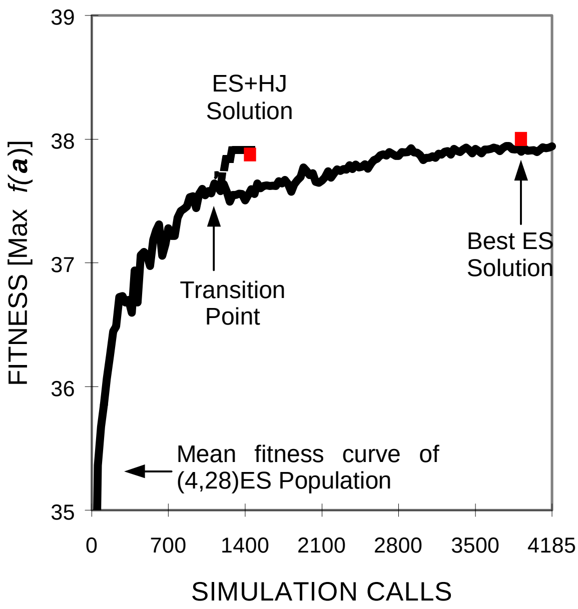

Figure 3: Characteristics of ES and ES+HJ Methods

For the two-phased search (ES+HJ), the transition solution ($\boldsymbol{a}_t$) was arbitrarily selected as the best solution found by the (4,28)ES over 43 generations where the rapid growth stops prior to a decline in the fitness (see Figure 3). The ES used 1204 simulation calls up to the transition point. The HJ method was applied using the transition solution ($\boldsymbol{a}_t$) from the abbreviated ES search as its starting point. The HJ method performed three exploratory searches and three pattern moves requiring 286 simulation calls while using the archival database. The database saved 35 simulation calls from being executed during the HJ search. Including the 1204 calls needed to reach the transition point by the ES, a total of 1490 simulation calls were used to find the solution with the two-phased ES+HJ strategy. This solution uses 114 kanbans and achieved the throughput required by the manufacturer.

For comparison purposes, four single-phased HJ searches were also done using the randomly generated solutions in the initial population of the ES as starting points. Generally, the HJ method required about 250 simulation calls but produced very poor solutions that failed to meet production throughput requirements. The best solution found using this procedure is identified in Table 1 as HJ. The HJ method does not appear to be a good globally oriented search technique for this kanban-sizing problem.

Table 1: Comparison of Single-Phased ES, Two-Phased ES+HJ, and Single-Phased HJ Methods

Method	Kanbans	Fitness [$f(\boldsymbol{a})$]	Simulation Calls
(4,28)ES	110	37.922	3781
(4,28)ES+HJ	114	37.858	1490
HJ	127	30.703	238

Final solutions generated by the single-phased ES, two-phased ES+HJ, and single-phased HJ strategies were run in the simulation model for 20 independent replications to get better estimates of the true value of $f(\boldsymbol{a})$. Table 1 presents a comparison of the three strategies. Simulation runs are the total required runs needed to find the best solution for each method. While the single-phased (4,28)ES found the better solution requiring four fewer kanbans, the two-phased ES+HJ strategy used 60% fewer simulation calls. When computer time is considered expensive, a two-phased ES+HJ strategy may provide the better solution for a user.

4 RESEARCH ISSUES INVOLVING TWO-PHASED OPTIMIZATION

The results from this applied problem suggest that a two-phased strategy can significantly reduce the number of times a simulation model is called to evaluate solutions without greatly sacrificing the quality of the solution. However, many issues need to be researched before a generalized multi-phased method can be implemented.

Perhaps the most important issue for a two-phased optimization strategy is determining when to transition from the first phase (exploration) to the second phase (exploitation). Consider a two-phased optimization strategy using an evolutionary algorithm (EA) for the first (exploration) phase to identify a region containing an optimal solution. An EA was chosen as a basis for discussion and not to promote it as the best choice for the exploration phase. This is an open research question.

Figure 4 presents a typical plot of the best fitness scores (based on an objective function) observed during the successive generations of an EA search for a minimization problem. Early in its search, the EA is more globally oriented and considers solutions from throughout the solution space. The slope of the fitness curve is generally steep while in this Exploration Region as shown in the figure. As the search converges on a solution, the slope of the fitness curve will approach zero as the EA concentrates the search in one region of good solutions. This is shown as the Exploitation Region in the figure. The Transition Region is the area where the orientation of the EA search changes from exploration to exploitation.

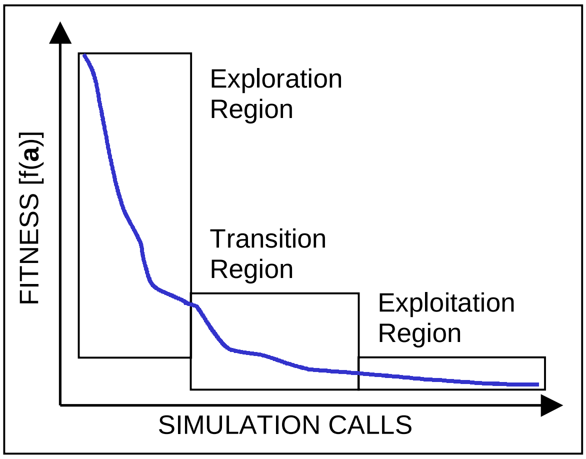

Figure 4: Typical Fitness Convergence in an EA

For a transition method to be effective, it should allow the EA to sufficiently explore the solution space during Phase 1 (exploration). The transition method should not cause a premature change to Phase 2 (exploitation). As the EA moves from exploration to exploitation, the fitness curve moves into the Transition Region. In a two-phased optimization strategy, an aggressive transition would cause the change to the Phase 2 search when the EA is in this Transition Region. A conservative transition would cause the change to the Phase 2 search when the EA is in the Exploitation Region. However, it should not allow the EA to expend a large number of simulation calls in the Exploitation Region since the Phase 2 search method should be more efficient in converging to an optimal solution. The authors are actively designing and testing various transition methods.

There are many other issues germane to multi-phased approaches. These issues span beyond the Strategy and Tactics Domain into several of the other six domains. Within the Methods Domain, an important question is what are the best optimization methods to combine in multi-phased strategies. The most effective combinations will likely change depending on the characteristics of a given problem. The characteristics of a problem must be measured and the problem classified in the Classification Domain. An important question is what are the key attributes necessary to classify simulation optimization problems.

In the Intelligence Domain, identifying the knowledge needed for the optimization tool to select appropriate strategies is an important consideration. If the response surface were unimodal, for example, then a single-phased method using a locally oriented search method would likely be most appropriate. Additionally, mechanisms are needed for selecting and deploying tactical tools and for adapting strategies and tactics as new information is gained during the course of an optimization.

5 SUMMARY AND CONCLUSIONS

Six functional domains common to automated simulation optimization tools were outlined in this paper. These six domains form the cornerstones of the authors proposed unified strategy for simulation optimization research and development. An example highlighting the sort of improvements that can be realized by employing the unified strategy was presented along with several research issues regarding the multi-phased optimization procedure. The authors are actively researching these and other issues relative to the field of simulation optimization.

It is hoped that other researchers will adopt a more encompassing view of the process of optimizing simulated systems. The results from such research will help improve next generation simulation optimization tools.

REFERENCES

Akbay, K. 1996. Using simulation optimization to find the best solution. *IIE Solutions* May:24-29.

Back, T. and H-P. Schwefel. 1993. An overview of evolutionary algorithms and parameter optimization, *Evolutionary Computation* 1:1-23.

Bateman, R., R. Bowden, T. Gogg, C. Harrell, J. Mott. 1997. *System improvement using simulation.* 5th ed. Utah: ProModel Corportation.

Biethahn, J. and V. Nissen. 1994. Combinations of simulation and evolutionary algorithms in management science and economics. *Annals of Operations Research* 52**:**183-208.

Carson, Y. and A. Maria. 1997. Simulation optimization: methods and applications. In *Proceedings of the 1997 Winter Simulation Conference* 118-126.

Hall, J. 1997. *Investigation of a two-phased strategy for simulation optimization*, Ph.D. dissertation, Department of Industrial Engineering, Mississippi State University, Mississippi.

Hall, J. and R. Bowden. 1998. A unified strategy for simulation optimization research and development. In *7th Industrial Engineering Research Conference Proceedings* 1-8.

Hall, J. R. Bowden, and J. Usher. 1996. Using evolution strategies and simulation to optimize a pull production system. *Journal of Materials Processing Technology* 61**:**47-52.

Hooke, R. and T. Jeeves. 1961. 'Direct search' solutions of numerical and statistical problems. *Journal of the Association for Computing Machinery* 8 (2):212-229.

Pegden, D., and M. Gately. 1980. A decision optimization module for SLAM. *Simulation* 34:18-25.

Pegden, D., and M. Gately. 1977. Decision optimization for GASP IV simulation models. In *Proceedings of the 1977 Winter Simulation Conference* 127-133.

Myung, H., J.-H. Kim, and D. Fogel. 1995. Preliminary investigations into a two-stage method of evolutionary optimization on constrained problems. In *Proceeding on the Fourth Annual Conference on Evolutionary Programming* 449-463. MIT Press, Cambridge, MA.

Reklaitis, G., A. Ravindran, and K. Ragsdell. 1983. *Engineering optimization: methods and applications.* New York: Wiley.

Schwefel, H.-P. 1995. *Evolution and optimum seeking*. New York: Wiley.

Smith. D. 1973a. An empirical investigation of optimum-seeking in the computer simulation situation. *Operations Research* 21 (2):475-497.

Smith D. 1973b. Requirements of an 'optimizer' for computer simulations. *Naval Research Logistics Quarterly* 20 (1):161-179.

AUTHOR BIOGRAPHIES

ROYCE O. BOWDEN is Associate Professor of Industrial Engineering at Mississippi State University and Director of the Simulation and Advanced Computation Laboratory. Prior to joining the faculty at Mississippi State, he was in manufacturing engineering positions with Martin Marietta Aerospace and Texas Instruments. His research interests include systems modeling and analysis, machine learning and optimization. Dr. Bowden's research program has been funded by several organizations including the National Science Foundation. He is an Alpha Pi Mu faculty advisor and member of IIE, INFORMS, SME, and Sigma Xi.

JOHN D. HALL is a member of the technical staff for TASC, Inc. in Huntsville, AL. He holds a B.S. in Aerospace Engineering from the U.S. Military Academy, an M.S. in Industrial Engineering from California Polytechnic State University, and a Ph.D. in Industrial Engineering from Mississippi State University. He is a member of IIE, INFORMS, APICS, and ASQ. Dr. Hall is an ASQ certified quality engineer.