https://dx.doi.org/10.22161/ijaers.4.3.17 **ISSN: 2349-6495(P) | 2456-1908(O)**

A Review on Simulation Optimization

Mobin Ahmad

Department of Mathematics, Faculty of Science, Jazan University, Saudi Arabia

Abstract*— *One of the primary and most important employments of simulations is for optimization. Simulation optimization can be characterized as the way toward finding the best info variable qualities from among all potential outcomes without unequivocally evaluating each possibility. The goal of simulation optimization is to minimize the assets spent while boosting the data acquired in a simulation experiment. The purpose of this paper is to review the zone of simulation optimization. A critical review of the methods employed and applications developed in this generally new range are introduced and striking victories are highlighted. Simulation optimization software tools are discussed. The target group is simulation practitioners and theoreticians and additionally fledglings in the field of simulation.

***Keywords*— Simulation, Optimization, important, process, resources, information, methods, develop, successes, software tools.**

I. INTRODUCTION

The mathematical model of a system is concentrated on using simulation; it is known as a simulation model. System behavior at particular estimations of info factors is assessed by running the simulation model for a settled timeframe. A simulation experiment can be characterized as a test or a progression of tests in which significant changes are made to the information factors of a simulation model so that we may observe and recognize the purposes behind changes in the output variable(s). At the point when the quantity of information factors is huge and the simulation model is perplexing, the simulation experiment may turn out to be computationally restrictive. Other than the high computational cost, a much higher expense is brought about when imperfect info variable qualities are chosen. The way toward finding the best info variable qualities from among all potential outcomes without unequivocally evaluating each plausibility is simulation optimization. The goal of simulation optimization is minimizing the assets spent while amplifying the information acquired in a simulation experiment.

A general simulation model comprises n input variables $(x_1, x_2, ..., x_n)$ and m output variables $(f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))$ or $(y_1, y_2, ..., y_m)$ (Figure 1). Simulation optimization entails finding optimal settings of the input variables, i.e. values of $x_1, x_2, ..., x_n$, which optimize the output variable(s).

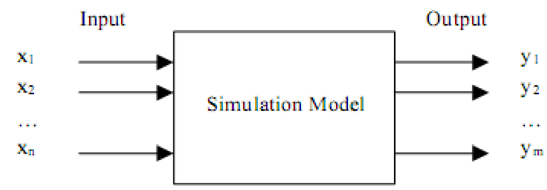

Fig.1: A Simulation Model

Such problems emerge habitually in engineering, for example, in process design, in mechanical experimentation, in design optimization, and in reliability optimization. This is the issue we will address in this paper. A simulation optimization model is shown in Figure 2. The yield of a simulation model is utilized by an optimization strategy to give criticism on advancement of the quest for the optimal solution. This thus manages further contribution to the simulation model.

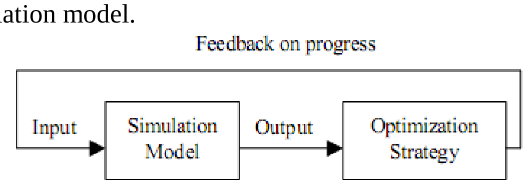

Fig.2: A Simulation Optimization Model

II. REVIEW OF LITERATURE

Simulations Optimization of true occasions can permit a complex problem to be dismembered and examined in a productive, safe, and financially savvy way. A simulation becomes a much more valuable instrument when optimizing an arrangement of parameters, especially in circumstance where experiments on this present reality framework are troublesome or impractical. Simulation optimization, as a rule, tries to minimize an objective function:

$$\min_{X \in \Theta} f(x)$$

Where $x \in \Theta$ represents an input vector of parameters, $f(x)$ is the scalar objective function and Θ is the constraint set [1, 2]. The info parameters are frequently alluded to as variables, while the yield exhibitions are known as the reactions [3]. Simulations can be subdivided into two classifications of element variable sorts: consistent or discrete. In continuous simulations, the limitations are normally the set or every single genuine number, n $\Theta \subset \Re$. In discrete event simulations, the response function is not specifically accessible, may not be nonstop, or may not be in shut structure, and subsequently standard mathematical extremism solutions cannot be utilized. In this manner, a hunt must be performed over the discrete factor set [1, 2, 3, and 4]. This comprises of picking a worth for all parameters, running the simulation until the appropriate stopping criteria are met, and after that measuring the yield of the simulation that is to be optimized. In simulations with numerous parameters, a comprehensive inquiry turns into a period expending errand. Propels in computational power, consolidated with more current strategies for decreasing the pursuit space, have permitted discrete optimization techniques to be deployed with more success.

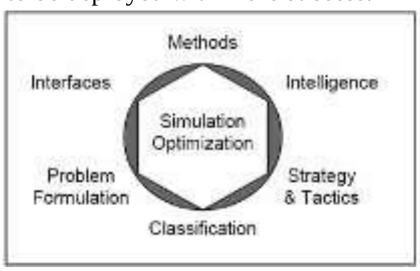

Fig.3: Six Domains of Simulation Optimization

Simulation Optimization has also become an integral part of many commercially available software simulation packages. Bowden and Hall [5] have described six domains which should be addressed when designing automatic simulation optimization tools (figure 3).

III. GRADIENT AND STOCHASTIC METHODS

Stochastic approximation methods (SAM) endeavor to discover minima by moving toward the steepest slope of the function. This is an iterative process, where every cycle comprises of evaluating the angle of the simulation model at the current decision point, and after that moving this decision point along the slope with a specific stride size. This development of the decision point can be expressed as

$$x^{n+1} = \Pi\left(x^n - \alpha_n \hat{\nabla} f\left(x^n\right)\right)$$

Where x^n the current decision point solution is $\hat{\nabla} f\left(x^n\right)$ is the estimate of the gradient, α n is the step size, and Π is a mapping onto the set Θ [2, 4, 11]. This method has received much attention, mainly because it has been proven to converge to the minima as the step size gets sufficiently smaller [2, 4]. The difficulty in using this approach is estimating the gradient, which will not be continuous for discrete simulations. The most common gradient approximation method is the method of finite differences, where a small number of output values are taken for small changes in the simulation parameters. The two-sided, central difference gradient operator, for example, is:

$$\hat{\nabla} f\left(x_i\right) = \frac{f\left(x_i + \Delta x_i p_i\right) - f\left(x_i - \Delta x_i p_i\right)}{2\Delta x_i}$$

Where Δx_i is the perturbation of input parameter i and i p is a vector with a one in the ith place and zeros elsewhere. This is essentially taking the discrete derivative, or "slope" of the function for each separate dimensional input parameter, denoted by i [2, 4, 8]. However, if the simulation output is noisy, then the gradient estimation could also be noisy, possibly making the decision point move in an inappropriate direction [4, 7].

It is clear that simulations with a higher number of variable input parameters will require more calculations to appraisal the slope. In particular, utilizing the focal distinction inclination, it will take 2q simulation measurements, where q is the dimensionality, or number of variable input parameters [2, 8]. Another method has been proposed to lessen the quantity of simulation measurement required. This technique is known as Infinitesimal Perturbation Analysis (IPA), and is summarized in [5] by the formula:

$$\hat{\nabla} f(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x_i}$$

Now, the vector Δx becomes a random perturbation vector, and the gradient estimation requires only two simulation measurements, regardless of the input vector dimensionality. Thus, the speedup using this gradient estimator is potentially q. However, this potential is realized only if the number of iterations required for convergence is not increased. It is also been shown that the Δx vector should be independently and symmetrically distributed about zero with finite inverse moments. This can be achieved by using the Bernoulli ±1 distribution.

IV. SIMULATION OPTIMIZATION APPLICATIONS

Simulation optimization methods have been connected to applications with a solitary target, applications that require the optimization of multiple criteria, and applications with non-parametric objectives.

Azadivar et al. (2010) connected a simulation optimization algorithm based on Box's perplexing hunt strategy to optimize the locations and stock levels of semi-completed items in a force sort production system.

Corridor et al. (2009) utilized ES with a simulation model for streamlining a kabana sizing problem. Dad technique to stock models where the interest has a related restoration arrival process.

Tompkins and Azadivar (2011) proposed an approach to join a GA and an article oriented simulation model generator to locate the ideal shop floor design. an approach to consolidate the process plant production operations into the design of an office by joining simulation and GA. a calculation that joined SA and simulation to locate a suitable dispatching need of operations to minimize the aggregate lateness for a business flexible manufacturing system (FMS).

V. SIMULATION OPTIMIZATION SOFTWARE

A longstanding goal among a portion of the simulation practitioners and theoreticians was having the capacity to direct a progression of simulations in the most effective way as opposed to performing "blind" analyses and accepting that no less than one of the tests will yield the best contrasting option to execute (Glover et al., 2012). Numerous simulation software developers today have turned out to be more mindful of the importance of finding optimal and close optimal solutions for applications in minutes, instead of playing out a comprehensive examination of pertinent options in days or months. Simulation software that incorporates extraordinary hunt techniques to control a progression of simulations to uncover optimal or close ideal scenarios includes: Pro-Model, Auto-Mod, Micro Saint, Lay-OPT, and Factory OPT. A brief portrayal of every software’s optimization and/or statistical module follows.

The extra optimization module for Pro-Model is called Sim Runner Optimization. This module consists of two elements for investigating and optimizing existing Pro-Model simulation models. The primary component is a factorial configuration of tests that uncovers the impact of an adjustment in info element on the objective function. The second feature is a multi-variable optimization that tries different combinations of input components to land at the combination that yields the best objective function value.

VI. GENETIC ALGORITHMS

Genetic Algorithms (GA) are a subset of Evolutionary Algorithms, which are main stream in optimization literature on account of their generality. In particular, they just require the Monte-Carlo simulation output, with no learning of capacity or information imperatives [9]. Genetic Algorithms endeavor to iteratively discover an all inclusive optimum solution by investigating the reaction surface of the simulation, and developing the best solutions in a comparable way to Darwin's hypothesis of evolution. A conceivable solution is encoded as a chromosome, with every quality in a chromosome speaking to a variety of a solitary input parameter. The wellness of a chromosome speaks to how close the chromosome's qualities will convey the simulation to its optimum value. A chromosome with low wellness will have a higher likelihood of being expelled from the population. A population is a gathering of chromosomes in one algorithm iteration. GA requires two operations, cross over and mutation, to change over one population of chromosomes to the following [3, 4]. The essential stream of hereditary calculations is demonstrated in figure 4.

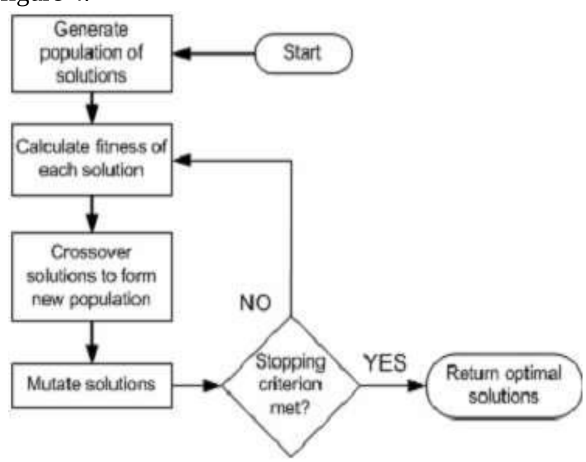

Fig.4: Genetic Algorithm flow chart

Some of the issues when implementing GA’s include gene representation, crossover operator selection, and mutation operator selection. Binary strings are a very common choice for gene representation, because they are very general, can be used for any size data type, and require minimal storage. An example of a binary string chromosome with genes representing byte sized parameters X, Y, and Z is given in figure 5.

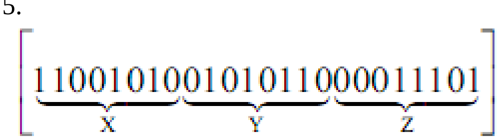

Fig.5: Chromosome of 3 bytes

The crossover operator is in charge of making new chromosomes from two existing ones. A famous way to deal with this is to choose a little number of bits from every quality and essentially swap them between the two

https://dx.doi.org/10.22161/ijaers.4.3.17 *ISSN: 2349-6495(P) | 2456-1908(O)*

chromosomes [3, 4]. The mutation operator is basic since it permits the changing of qualities to keep a solution from being caught in a local optimum. This can be expert by arbitrarily selecting a bit from every chromosome, and upsetting it with a predefined probability distribution [3, 4]. As the iterations of the calculation increase, chromosomes (solutions) with lower wellness will be evacuated in the hybrid stage, so populaces will comprise of more chromosomes with higher wellness. The calculation can end when a sought number of cycles has been come to, or the standard deviation of a population's fitness has been minimized.

VII. CONCLUSION

We have provided an introduction to simulation optimization, with emphasis on gradient-based techniques for continuous parameter simulation optimization and on random search methods for discrete parameter simulation optimization. Although simulation optimization has received a fair amount of attention from the research community in recent years, the current methods generally require a considerable amount of technical sophistication on the part of the user, and they often require a substantial amount of computer time as well. Therefore, additional research aimed at increasing the efficiency and ease of application of simulation optimization techniques would be valuable.

REFERENCES

[1] M.C. Fu, F. W. Glover, and J. April., Simulation Optimization: A Review, New Developments, and Applications, Proc. Of the 2010 Winter Simulation Conference, Orlando, FL, 83-95.

[2] S. Olafsson, and J, Kim, Simulation Optimization, Proc. Of the Winter Simulation Conference, San Diego, CA, 2011, 79-84.

[3] J. April, et al, Practical Introduction to Simulation Optimization, Proc. of the Winter Simulation Conference, New Orleans, LA, 2008, 71-78.

[4] T. M. Alkahmis, and M. A. Ahmed, Simulation- Based Optimization Using Simulated Annealing With Confidence Interval, Proc of the 2007 Winter Simulation Conference, 514-519.

[5] A. A. Prudius, and S. Andradottir, Two Simulated Annealing Algorithms for Noisy Objective Functions, Proc of the 2010 Winter Simulation Conference, Orlando, FL, 797-802.

[6] J. A. B. Montevechi, R. G. A. Filho, and A. L. Medeiros, Application of Factorial Designs for Reducing Factors in Optimization via Discrete-Event Simulation, Proc. of the Winter Simulation Conference, Monterey, CA, 2008.

[7] A. Persson, H. Grimm, A. Ng, T. Lezama, J. Ekberg, S. Falk, and P. Stablum, Simulation-Based Multi-Objective Optimization of Real-World Scheduling Problem, Proc of the Winter Simulation Conference, Monterey, CA, 2012.

[8] A. Persson, H. Grimm, and A. Ng, On-Line Instrumentation for Simulation-Based Optimization, Proc. of the Winter Simulation Conference, Monterey, CA, 2011, 304-311.

[9] Azadivar, F., J. Shu, and M. Ahmad. 2010. Simulation Optimization in Strategic Location of Semi-Finished Products in a Pull-Type Production System, Proceedings of the Winter Simulation Conference, 1123-1128.

[10] Hall, J. D., and R. O. Bowden. 2011. Simulation Optimization for a Manufacturing Problem, Proceedings of the Southeastern Simulation Conference Huntsville, AL, Society for Computer Simulation, 135-140.

[11] Tompkins, G. and F. Azadivar. 2011. Genetic Algorithms in Optimizing Simulated Systems, Proceedings of the Winter Simulation Conference, 757-762.

[12] Glover, F., J. P. Kelly, and M. Laguna. 2012. New Advances and Applications of Combining Simulation and Optimization, Proceedings of the Winter Simulation Conference, 144-152.