Vol.2012-HPC-133 No.21
2012/3/27

Efficient Monte Carlo Optimization with ATMathCoreLib

Reiji Suda[†1,†2] and Vivek S. Nittoor[†1,†2]

In this paper, we discuss a deterministic optimization method for stochastic simulation with unknown distribution. The problem to solve is the following: there is a parameter t to which a stochastic cost function $f(t, z)$ is associated, where z is the nuisance parameter. We want to estimate t with which the average of $f(t, z)$ is minimum, with as small number of evaluations of $f(t, z)$ as possible. Our method is based on Bayesian formulation, and utilizes the information given through the prior distribution.

1. Introduction

Computer simulation is the third paradigm of science — while the first two paradigms are experiment and theory. Computer simulation gives us insight into the details of a complex system, for which theoretical analysis is difficult because of the complexity and experimental study is limited by various factors, such as difficulties of full control of initial and boundary conditions or inability of experiments for such systems as cosmic systems and evolutionary systems. Accompanied by rapid progress of computer performance, the significance of computer simulation grows rapidly. Now computer simulation is an essential methodology both for science and technology.

However, methods of optimizing efficiency of computer simulation are not widely understood and utilized. Perhaps it is partly because the results of simulation must be seen, understood, and judged by human, and thus it cannot be fully automated without any human interactions. We cannot overlook the importance of human understanding of the simulation results. Still, wish for efficient computer simulation is natural and essential, since many kinds of computer simulation require much resource and long time. In this paper, we discuss an efficient method of a kind of computer simulation: optimization of a system simulated by Monte Carlo method.

There are two kinds of computer simulation. One is *deterministic* and the other is *stochastic*. The latter uses pseudo-random numbers, and thus can be called a Monte Carlo computation in a wide sense. Another classification, from the viewpoint of the aim of the computer simulation, is between *evaluation* and *optimization*. Here, "evaluation" aims to understand the behavior of the target system for a specific set of parameter values. "Optimization" aims to find the optimal parameter values for a certain criteria defined with the simulation results. In evaluation, the results should be collected over the specified domain of parameter values, but in contrast, in optimization, the number of simulation runs can be a minimum so to achieve the optimization.

In this paper, we show a method of minimization of the number of computer experiments for an optimization of parameters, where the performance is evaluated in Monte Carlo simulation. The proposed method is derived from a method of semi-optimized experimental design developed for automatic tuning. Our method is applicable to the cases where the number of candidates is very big.

The rest of this paper is organized as follows. Section 2 introduces a number of related works on efficient computer simulation. Section 3 explains the problem to be solved and our solution. Section 4 reports a small set of synthetic experiments showing the effectiveness of our method. Section 5 is a short summary with discussion on possible future research directions.

2. Related Works

Much effort has been spent to finding efficient ways of computer simulation. The methods can be classified into two classes: *deterministic methods* and *stochastic methods*. Taking the classification of computer simulation, which is discussed in the previous section, into consideration, there are eight kinds of methods, that is,

{Deterministic | Stochastic} methods of {evaluation | optimization}
with {deterministic | stochastic} simulation.

We assume that the deterministic functions and the probability distribution of the stochastic functions are unknown. If the probability distribution is known,

†1 Graduate School of Information Science, the University of Tokyo
†2 CREST, JST

the problem is classified into *stochastic programming*, which is a related but different problem. There are deterministic methods of (true) optimization for stochastic programming. However, in many problems of stochastic programming, the complexity of the algorithms is too high, and approximate methods based on Monte Carlo methods must be used. Then the some of the following methods can be used for problems of stochastic programming.

First we introduce some books that discuss effective and efficient methods of computer simulation. Note that this is not an exhaustive list.

The book by Fang et al.[1] mainly discusses deterministic methods of evaluation with deterministic simulation. The book by Santner et al.[2] also develops deterministic methods of evaluation with deterministic simulation. The latter has more theoretical flavor than the former, and the former is pragmatic and practical.

The book by Zabinsky[5] is dedicated to stochastic methods of optimization with deterministic simulation, and characterized with its developments on convergence of stochastic optimization methods to the global maximum.

The book by Gosavi[6] discusses deterministic and stochastic methods of optimization with stochastic simulation. It consists of two parts: the first part treats parametric optimization, and the second part treats optimization over Markov chain, mainly *Reinforcement Learning*. The latter is fully described in the book by Powell[7], under the name of *Approximate Dynamic Programming*. Text books on Monte Carlo methods, such as the one by Liu[3], usually contain stuff related to optimization, that is, stochastic methods of optimization with stochastic simulation. The book by Rubinstein and Kroese[4] has more extensive treatments of deterministic and stochastic methods of evaluation and optimization with stochastic simulation.

In the following, we discuss existing methods of optimization with stochastic simulation. Refer a survey paper by Tekin and Sabuncuoglu[8] and one by Fu et al.[9] for further details. We classify the existing methods into three groups.

The first group consists of methods of exhaustive evaluation of candidates with considering the stochastic variance of the simulation results. *Ranking-and-Selection*[10] applies the frequentist methods of hypothesis testing to optimization with stochastic simulation. There are two strategies to that purpose: *indifference-zone formulation* and *subset-selection formulation*. In the indifference-zone formulation, the number of trials for each candidate is optimally chosen so that a candidate within δ from the best is chosen with a probability no less than $1-\alpha$. In the subset-selection formulation, a set of candidates is determined so that it contains the best solution with a probability no less than $1-\alpha$. *Multiple Comparison* methods attain similar goals (there are several formulations) by pairwise comparisons. *Ordinal Optimization*[11] is a method similar to subset-selection: it computes a subset in which at least k of the top-n candidates are included. The key idea there is that the convergence of the estimate of the difference of two candidates $s_1 - s_2$ is $O(1/\sqrt{N})$, but the convergence of estimating the order $s_1 \leq s_2$ is exponential. *Optimal Computing Budget Allocation* (OCBA) gives a different view of the same problem, that is, optimization of the probability of correct selection of the best solution under a given number of total trials. Those methods are applicable to the problems of fairly small search space, since they require a fixed number of initial trials for all candidates. *Iterative Ranking-and-Selection*, developed by Ólafsson[12], is applicable to larger problems. It is an iterative method of optimization, for which asymptotic convergence to the optimal solution is established.

The second group consists of local (or global) search methods for stochastic functions, whose mean forms a continuous function, traditionally called *response surface*. *Response Surface Methodology* (RSM) is a class of methods where the data are locally fit into a regression model (usually of a low-degree polynomial) and the optimum solution is approximated by the optimizer of the regression model. That methodology is the one widely used in the experimental design in other application fields such as physical experiments. *Gradient-based methods* estimate the gradient of the response surface. *Finite Difference Estimate*, *Perturbation Analysis*, *Frequency-Domain Analysis*, and *Likelihood Ratio Estimators* are major methods in this class. *Stochastic Approximation* can be seen as a stochastic version of the steepest descent method. *Gradient Surface Method*[14], *Estimation of Distribution Algorithms*[15], may be categorized here. *Cross-Entropy Method*[4] and *Stochastic Model Reference Adaptive Search*[18] employ a different formulation to find a probability function over the search space that peaks at the optimal candidate. A similar approach is taken in *Probability Collectives*[19]. In addition

to them, methods that are used for local optimization of deterministic function, such as *Nelder-Mead simplex method* and *Bayesian / Sampling Method*[16], have been applied to optimization of stochastic functions.

The third group consists of stochastic global search methods for deterministic functions, which could be applied to stochastic functions. Well-known metaheuristics, such as *Genetic Algorithms*, *Simulated Annealing*, *Tabu Search*, *Scatter Search* and *Ant Colony Optimization*, are classified in this group. *Pure Random Search*, which is also called *Random Search*[17] and *Monte Carlo*, samples candidates randomly, and chooses the best observed candidate.

3. Proposed Method

Our method is classified into the first group discussed in the previous section. It assumes discrete variables, which can be finite or infinite. The biggest difference of our method from the existing approaches is that we employ Bayesian formulation. It could be compared with Bayesian / Sampling methods[16] and Stochastic Model Reference Adaptive Search[18]: they employ Bayesian formulation with different criteria of optimization.

3.1 Target Problem

The search space is discrete, and the candidates are represented as t_1, t_2, t_3, ... Let T represent the entire search space, that is, $T = \{t_i\}$. The search space can be finite or infinite, but in the case of infinite search space, we need some method for finding the next sample from the infinite search space according to the rule explained below.

For each candidate t, a cost function $f(t, z)$ is accompanied, where z represents a perturbation factor (nuisance parameter) unobservable to us. That is, the cost function is stochastic, and we want to find t which minimizes the average cost function

$$x_t = E_z(f(t, z)),$$

where $E_z()$ represents an average over z. Therefore the problem is to find t_{opt} defined as

$$t_{opt} = \mathrm{argmin}\{x_t \mid t \in T\}.$$

Our method assumes a Bayesian formulation, where we have a prior distribution $\pi_{t0}(x_t)$ for x_t. The prior $\pi_{t0}(x_t)$ represents our vague knowledge about the possible value of x_t, and we expect x_t is more probable where $\pi_{t0}(x_t)$ is larger. In a later section, we will discuss how we constructed priors in our experiments.

3.2 Bayesian Inference

Bayesian inference is derived by Bayes' Theorem

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{P(Y|X)}{P(Y)}P(X).$$

Here, $P(X)$ is the prior distribution, and $P(X|Y)$ is the posterior distribution. We assume $P(X)$ and $P(Y|X)$ are (assumed to be) known. The calculation of the marginal distribution $P(Y)$ is not compulsory, since $P(X|Y)$ can be derived from the condition that $\int P(X|Y)dX = 1$. Thus let us simplify the formula as

$$P(X|Y) \propto P(Y|X)P(X).$$

We understand the formula in this way: $P(X)$ represents the vague knowledge about the value of X before experiments, and after observing Y, the knowledge about X is updated as $P(X|Y)$.

Let x_{tj} represent the observed cost function at the jth observation of the candidate t. Applying Bayesian inference to our case, the posterior distribution after k observations of t will be

$$\pi_{tk}(x_t) \propto P(x_{t1}, x_{t2}, \ldots, x_{tk} \mid x_t)\pi_{t0}(x_t).$$

In our best knowledge, we can predict the value of the cost function of the candidate t in its $k+1$st observation, $x_{t,k+1}$, so that it follows $\pi_{tk}(x_{t,k+1})$. Further, assuming that the value x will be observed in the next observation, we can predict the posterior distribution

$$\pi_{t,k+1}(x) \propto P(x \mid x_t)\pi_{tk}(x_t)$$

with the probability $\pi_{tk}(x_{t,k+1})$. This is called the preposterior distribution.

3.3 One Step Approximation

We have proposed a sub-optimal sequential experimental design for online automatic tuning[20] and one for offline automatic tuning[21]. We call them *One Step Approximation* collectively. The method in this paper is derived from them.

Assume the situation after K observations have been done. Let k_t be the number of observations for the candidate t. Thus it holds that $K = \sum_{t \in T} k_t$. The posterior (or prior, if t is not observed yet) distribution for the candidate t at this situation is $\pi_{tk_t}(x_t)$. Without loss of generality, let us assume that t_0 is the current best:

$t_0 = \mathrm{argmax}_{t \in T}\{\xi_i\}$,
where ξ_i is defined as

$$\xi_i = \int x \pi_{t_i k_i}(x) dx,$$

where we simplify k_{t_i} into k_i. If we stop the observation now, we will choose t_0, and the expected average cost is ξ_0.

First assume that the candidate t_i $(i > 0)$ will be observed in the next observation, and will result in an observation of the cost x_{t_i,k_i+1}. In our prediction, x_{t_i,k_i+1} follows the distribution $\pi_{t_i k_i}$. After observing x_{t_i,k_i+1}, the posterior distribution for t_i will be updated into $\pi_{t_i,k_i+1}(x|x_{t_i,k_i+1})$. Let $\xi_{i,k_i+1}(x_{t_i,k_i+1})$ be its average. Here, x_{t_i,k_i+1} is inserted to emphasize that they are dependent on the unseen result x_{t_i,k_i+1}, but sometimes we will omit it. If the cost to be observed x_{t_i,k_i+1} is very low, and ξ_{i,k_i+1} is to be less than the current best ξ_0, then we will choose t_i rather than t_0 after the observation, and the expected cost will be ξ_{i,k_i+1}. Otherwise, we will choose t_0, and the expected cost will be ξ_0. Considering both cases, the mean of the expected cost can be calculated as

$$w_i = \int \min\{\xi_0, \xi_{i,k_i+1}(x)\} \pi_{t_i,k_i}(x) dx.$$

Next assume that the current best candidate t_0 will be observed in the next observation. Let t_1 be the second best candidate, without loss of generality, and ξ_1 be its expected cost. If the next observation x_{t_0,k_0+1} is to be a very large value, and the updated expected cost $\xi_{0,k_0+1}(x_{t_0,k_0+1})$ is larger than ξ_1, then we will choose t_1 rather than t_0, and the expected cost will be ξ_1. Otherwise we will choose t_0 and the expected cost will be ξ_{0,k_0+1}. Then the mean of the expected cost can be calculated as

$$w_0 = \int \min\{\xi_1, \xi_{0,k_0+1}(x)\} \pi_{t_0,k_0}(x) dx.$$

The value w_i is the mean of the expected cost after the next observation, in which the candidate t_i is chosen to be observed. So our method chooses the candidate t_i that minimizes w_i:
$i = \mathrm{argmin}_{i|t_i \in T}\{w_i\}$.

3.4 Discussion

The proposed method is a Bayesian method which requires prior distributions. This is both advantage and disadvantage of the proposed method (and the Bayesian methods in general) compared to the frequentist approaches. If the prior distribution represents the uncertainty appropriately, then the proposed method can search the optimal solution quickly with referring to the prior. However, if the prior distribution is not appropriate, Bayesian methods can be inefficient or unable to find the optimum solution.

Our method is sequential in its nature. This property is convenient in computer implementation in a classic way. However, it is difficult to extend it to parallel computing. Although the existing methods based on the frequentist formulation do not provide full freedom of controlling the number of trials, those methods request multiple trials for each step, and thus it is somewhat easier to apply them to parallel experiments.

Our method tries to find the best solution. It will not seek a good estimate of the expected cost ξ_0, and rather it computes as rough estimate as possible so that the number of experiments could be as small as possible. In our method, we can compute a quantitative estimate how much the chosen candidate t_0 is likely to be the best.

3.5 Implementation

The value of w_i is defined as an integral. If we choose Gaussian distributions for the priors π_{t0} and the conditional distributions $P(x|x_t)$, that integral can be represented with the exponential function and the Gauss error function.

We implemented our method based on ATMathCoreLib[22]. The current version of ATMathCoreLib implements the method for online automatic tuning, and it is easy to modify it for offline automatic tuning and optimization of stochastic simulation. The main difference of the method for optimization of stochastic simulation from online automatic tuning and offline automatic tuning is that the costs of experiments are not included in the objective function.

4. Experiments and Results

In this section, we report a simple set of experiments to see the effects of our method.

4.1 Experimental Setup

In the experiments, we use random numbers to generate synthetic problems.

The number of experiments is fixed to 10,000. Thus the number of the candidates is chosen to be 10,000, since no more candidates could be observed in 10,000 observations. This setting mimics the case of infinitely many candidates. However, if the candidates are actually infinitely many, our algorithm requires random sampling of the candidates, since otherwise it will be severely affected by bias induced by non-random choice of candidates.

The true means of the candidates are determined following the standard normal distribution $N(0,1)$. The true variances of the candidates are ones, uniformly.

To build the prior distribution, in the first 100 observations, 10 candidates are observed 10 times for each. From those 100 observations, we calculate the sample means and the sample variances:

$$\xi_i = \frac{1}{10}\sum_k x_{ik}, \qquad \tau_i^2 = \frac{1}{9}\sum_k (x_{ik} - \xi_i)^2.$$

With using those values, we build the initial prior distribution $\pi_i \sim N(\Xi, S^2)$ where

$$\Xi = \frac{1}{10}\sum_i \xi_i, \qquad S^2 = \frac{1}{9}\sum_i (\xi_i - \Xi)^2.$$

In our experiments, the variance of the observation is assumed to be a constant

$$\log T^2 = \frac{1}{10}\sum_i \log \tau_i^2$$

for any candidate. This formulation is known as "normal distribution with known variance" in the Bayesian statistics literature.

Those parameters of the prior distributions are recomputed in the same way after each observation; an empirical Bayes method.

4.2 Naive Method to Compare

To a comparison purpose, we also implemented a naive method, where each candidate is observed 10 times. Since there are 10,000 observations in total, only 1,000 candidates are observed in the naive method. Then the naive method choses the candidate with the minimum sample average cost.

4.3 Results

Table 1 shows the experimental results. The first column ("Exp.ID") shows the experiment ID, which is the seed of the random numbers. The second column ("Optimum") shows the true mean of the true optimum. Since the values are generated as $N(0,1)$, the minimum value is usually negative. The third column ("Naive") shows the true mean of the candidate chosen by the naive method. The fourth column ("Proposed") shows the true mean of the candidate chosen by the proposed method. The fifth column ("# Trials") shows the number of the observations of the chosen candidate in the proposed method. The corresponding value for the naive method is not shown because it is always 10. The last column ("# Observed") shows the number of candidates which are observed at least once in the proposed method. For the naive method it is always 1,000.

Table 1 Results with 10,000 observations. Lower is better.

Exp.ID	Optimum	Naive	Proposed	# Trials	# Observed
20	-3.84	-3.07	-3.61	12	9624
21	-3.75	-3.37	-3.37	16	9373
22	-3.71	-2.85	-3.17	13	9704
23	-4.38	-2.58	-4.38	6	9664
24	-3.99	-2.80	-3.58	10	9490

As is seen in the results, the proposed method gives a better result than the naive method, almost constantly. The proposed method tries more than 9,100 candidates in 10,000 observations. Note that the first 100 observations are used for initial estimate of prior distributions, and thus more than 90% of the candidates are observed only once. That implies that our method does not spend costly observations for not-promising candidates. The chosen candidates are observed 6 to 16 times in our method. Our method observes those candidates many times to see whether it is really likely to be the optimal, because the variance of the perturbation is as large as the variance of the means. Thus spending the efforts into promising candidates, our method could find much better candidates than the naive method.

5. Conclusion

In this paper, we discussed a deterministic optimization method for stochastic

simulation, typically Monte Carlo simulation. Our method is based on Bayesian formulation, and derived from our online and offline autotuning methods. We implemented our method by modifying ATMathCoreLib, and compared it with a naive experimental design where all candidates are evaluated ten times. Our method gave much better results than the naive method.

Our algorithm is sequential in nature. Our next research topic is an extension of our method into a method which utilizes high parallelism of contemporary supercomputers.

Acknowledgments This work is partially supported by Grant-in-Aid for Scientific Research (B) "Adaptive Auto-tuning Technology Aiming Complex Multicore and Multiprocessor Environments" and JST CREST "An Evolutionary Approach to Construction of a Software Development Environment for Massively-Parallel Computing Systems."

References

1) Fang, K.-T., Li, R. and Sudjianto, A.: *Design and Modeling for Computer Experiments*, Chapman and Hall / CRC, Boca Raton, FL (2006).
2) Santner, T.J., Williams, B.J. and Notz, W.I.: *The Design and Analysis of Computer Experiments*, Springer-Verlag, New York (2003).
3) Liu, J.S.: *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag, New York (2001).
4) Rubinstein, R.Y. and Kroese, D.P.: *Simulation and the Monte Carlo Method*, 2nd ed., John Wiley and Sons, Hoboken, NJ (2008).
5) Zabinsky, Z.B.: *Stochastic Adaptive Search for Global Optimization*, Kluwer Academic Publishers, Norwell, MS (2003).
6) Gosavi, A.: *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic Publishers, Norwell, MS (2003).
7) Powell, W.B.: *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, John Wiley and Sons, Hoboken, NJ (2007).
8) Tekin, E. and Sabuncuoglu, I.: Simulation optimization: A comprehensive review on theory and applications, *IIE transactions*, Vol.36, pp.1067–1081 (2004).
9) Fu, M.C., Glover, F.W. and April, J.: Simulation Optimization: a Review, New Developments, and Applications, *Proc. 2005 Winter Simulation Conference*, 13 pages (2005).
10) Kim, S.-H. and Nelson, B.L.: Selecting the Best System, Chapter 17 of *Handbooks in Operations Research and Management Science: Simulation, Volume 13*, ed. by Henderson, S.G. and Nelson, B.L., North-Holland (2006).
11) Ho, Y.-C.: An Explanation of Ordinal Optimization: Soft Computing for Hard Problems, *Information Sciences*, Vol.113, pp.169–192 (1999).
12) Ólafsson, S.: Iterative Ranking-and-Selection for Large-Scale Optimization, *Proc. 1999 Winter Simulation Conference*, pp.479–485 (1999).
13) Fu, M.C., Chen, C.-H. and Shi, L.: Some Topics for Simulation Optimization, *Proc. 2008 Winter Simulation Conference*, pp.27–38 (2008).
14) Ho, Y.C., Shi, L., Dai, L. and Gong, W.-B.: Optimizing Discrete Event Dynamic Systems via the Gradient Surface Method, *Discrete Event Dynamic Systems: Theory and Applications 2*, Kluwer Academic Publishers, Boston, pp.99-120 (1992).
15) Larrañaga, P., Etxeberria, R., Lozano, J.A. and Peña, J.M.: *Optimization by learning and simulation of Bayesian and Gaussian networks*, Technical Report EHU-KZAA-IK-4/99, University of the Basque Country, 1999.
16) Stuckman, B.E.: A Global Search Method for Optimizing Nonlinear Systems, *IEEE Trans. Sys. Man and Cyb.*, Vol.18, No.6, pp.965–977 (1988).
17) Smith, D.E.: An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation, *Operations Research*, Vol.21, No.2, pp.475–497 (1973).
18) Hu, J., Fu, M.C. and Marcus, S.I.: A Model Reference Adaptive Search Method for Stochastic Global Optimization, *Commun. Inf. Syst.*, Vol.8, No.3, pp.245–276 (2008).
19) Wolpert, D.H., Bieniawski, S.R. and Rajnarayan, D. G.: *Probability Collectives in Optimization*, Santa Fe Institute Working Paper 11-08-033,
available from ⟨http://www.santafe.edu/media/workingpapers/11-08-033.pdf⟩
(accessed 2012-02-26).
20) Suda, R.: A Bayesian Method for Online Code Selection: Toward Efficient and Robust Methods of Automatic Tuning, *Proc. Second International Workshop on Automatic Performance Tuning (iWAPT2007)*, pp.23–31 (2007).
21) Suda, R.: Mathematical Methods of Offline Automatic Tuning, *IPSJ SIG Technical Reports*, HPC-125-3 (2010).
22) Suda, R.: ATMathCoreLib,
available from ⟨http://olab.is.s.u-tokyo.ac.jp/~reiji/atmathcorelib/⟩
(accessed 2012-02-26)