THE EFFECT OF NOISE ON ARTIFICIAL INTELLIGENCE AND METAHEURISTIC TECHNIQUES

ABHIJIT GOSAVI
University of Southern Colorado Pueblo, CO 81001
Email: gosavi@uscolo.edu

Abstract

Artificial intelligence methods and search techniques such as simulated annealing and simultaneous perturbation can be used in simulation optimization of complex stochastic systems. Simultaneous perturbation has also been applied in the area of neural networks. Simulation optimization methods are often used when it is difficult to obtain the closed form of the objective function, but it is possible to estimate the objective function value at any point via simulation of the system. However, when the objective function is estimated via simulation, simulation-induced noise cannot be avoided. We theoretically analyze the effect of noise on the performance of simulated annealing. We also theoretically analyze simultaneous perturbation under some special conditions and then present some empirical results with this algorithm from a revenue management problem in the airline industry.

Keyword 1: Optimization
Keyword 2: Algorithm, Artificial Intelligence, Convergence analysis, combinatorial optimization, gradient descent algorithm, simulated annealing, stochastic control, simulation approximation, steepest descent, search heuristics, operations research, convergence time, engineering optimization, binomial distribution, local minima.

INTRODUCTION

Simulated annealing is a popular meta-heuristic that has been adapted for use in simulation-based optimization. We analyze what happens when the function value is contaminated with simulation-induced noise in Section 2. The method of gradient descent is a popular method in operations research, which has been used extensively in simulation-based optimization and also in neural network algorithms. We discuss a relatively new approach called simultaneous perturbation, which is due to Spall (1992), to gradient descent. We analyze the convergence of this approach under some special conditions in Section 3. Finally, in Section 4, we present some numerical results from the use of simultaneous perturbation in a problem of revenue management in airlines.

SIMULATED ANNEALING

The simulated annealing algorithm (Pham and Karaboga, 1998) is described below. In this algorithm, one proceeds from one solution to its "neighbor." Each solution is a vector. The current solution and the best solution so far will be denoted by $\vec{x}_c$ and $\vec{x}_b$ respectively. The algorithm is written to minimize the objective function value $f(\vec{x})$. The "temperature," T, is maintained at a fixed value during a "phase." The algorithm is performed for $p_{\max}$ number of phases. Initialize T to a large value.

Step 1. The number of phases, p is set to 0.

Step 2 Select a neighbor $(\vec{x}_n)$ of the current solution.

Step 3. Define $\Delta \equiv f(\vec{x}_n) - f(\vec{x}_c)$. If $f(\vec{x}_n) < f(\vec{x}_b)$, set: $\vec{x}_b \leftarrow \vec{x}_n$.

Case 1: If $\Delta \leq 0$, set: $\vec{x}_c \leftarrow \vec{x}_n$.

Case 2: If $\Delta > 0$, generate U, a uniformly distributed random number between 0 and 1. If $U \leq \exp(-\frac{\Delta}{T})$, then set: $\vec{x}_c \leftarrow \vec{x}_n$.

Step 4. Repeat Steps 2 and 3, which together form one interation, for the number of iterations associated with the current phase.

Step 5. Set $p \leftarrow p+1$. If $p < p_{\max}$, then reduce T and return to Step 2 for another phase. Otherwise STOP. The best solution is $\vec{x}_b$.

We next show that "noisy" simulated annealing can converge.

Theorem 1. With probability 1, the version of simulated annealing algorithm that uses simulation based estimates of the function can be made to mimic the version that uses exact function values.

Proof: The effect of noise is felt in Step 3. In Step 3, we have two cases, each of which is analyzed below.

Case 1: Denoting the simulation estimate of the function at $\vec{x}$ by $\tilde{f}(\vec{x})$ and the exact value by $f(\vec{x})$, we can write that: $\tilde{f}(\vec{x}) = f(\vec{x}) + \eta$ if η denotes the noise that can be positive or negative. Then: $\tilde{f}(\vec{x}_c) = f(\vec{x}_c) + \eta_c$ and $\tilde{f}(\vec{x}_n) = f(\vec{x}_n) + \eta_n$. In Step 3, for Case 1, the noise-free and the noisy algorithm will behave in the same way if

$$\tilde{f}(\vec{x}_n) - \tilde{f}(\vec{x}_c) \leq 0. \tag{1}$$

Now, if $\eta_1 = |\eta_n|$ and $\eta_2 = |\eta_c|$ then we have four scenarios, which can be described by: $\tilde{f}(\vec{x}_n) = f(\vec{x}_n) \pm \eta_1$ and $\tilde{f}(\vec{x}_c) = f(\vec{x}_c) \pm \eta_2$. For example Scenario 1 is: $\tilde{f}(\vec{x}_n) = f(\vec{x}_n) + \eta_1$ and $\tilde{f}(\vec{x}_c) = f(\vec{x}_c) + \eta_2$. Now let us assume that

$$\eta_1 < -\frac{\Delta}{2} \text{ and } \eta_2 < -\frac{\Delta}{2}. \tag{2}$$

Below, we will identify conditions that will make Inequations (2) true. To prove that the result is true for Case 1, it is necessary to show that

Inequation (1) is satisfied. Let us consider Scenario 1. The following can be shown for any other scenario.

$$\begin{aligned}\tilde{f}(\vec{x}_n) - \tilde{f}(\vec{x}_c) &= f(\vec{x}_n) - f(\vec{x}_c) + \eta_1 - \eta_2 \text{ (from Scenario 1)} \\ &= \Delta + \eta_1 - \eta_2 \\ &\leq \Delta - \frac{\Delta}{2} - \eta_2 \text{ (from Inequation (2))} \\ &= \frac{\Delta}{2} - \eta_2 \leq 0 \text{ (from Inequation (2))}\end{aligned}$$

The above proves that Inequation (1) is true for Scenario 1. What remains to be shown is how Inequation (2) can be satisfied. From the strong law of large numbers, η_1 and η_2 can be made arbitrarily small. In other words, with probability 1, for a given value of $\epsilon > 0$, a sufficiently large number of replications (samples) can be selected such that $\eta_1 < \epsilon$, and $\eta_2 < \epsilon$. By choosing $\epsilon = -\frac{\Delta}{2}$ the claim in Inequation (2) is true.
Case 2: In a manner similar to Case 1, it can be shown that by selecting a suitable number of replications, one can ensure that:

$$\tilde{f}(\vec{x}_n) - \tilde{f}(\vec{x}_c) > 0 \tag{3}$$

when $\Delta > 0$. What remains to be analyzed is how the probability of selecting a worse neighbor is affected by the noise and what is the limiting behavior. The probability must converge to 0 as $T \to 0$, like in the noise-free version. The probability (U), when contaminated by noise, is $\exp(\frac{\tilde{f}(\vec{x}_n)-\tilde{f}(\vec{x}_c)}{T})$. From Inequation (3), the numerator in the power of the exponential term will always be strictly positive. Hence, $\lim_{T\to 0}\exp(\frac{\tilde{f}(\vec{x}_n)-\tilde{f}(\vec{x}_c)}{T}) = 0$. The above shows that the probability U in the noisy version will also converge to 0. Q.E.D.

SIMULTANEOUS PERTURBATION

In the steps given below, k will denote the number of decision variables and $\vec{x}^{\,m}$ will denote the solution vector in the mth iteration. It is defined as:$\vec{x}^{\,m} = (x^m(1), x^m(2), \ldots, x^m(k))$. Set $m = 0$ and start with an arbitrary value for $\vec{x}^{\,m}$. Intialize A to a small value such as 0.1 and set $\mu = A$.
Step 1. Assume that $H(i)$ is a binomially distributed random variable whose two permissible, *equally likely*, values are 1 and -1. Using this distribution, assign values to $H(i)$, where $i = 1, 2, \ldots, k$.. Then compute $h(i)$ for all values of i, using the following: $h(i) \leftarrow H(i)c^m$.

Step 2. Compute: $F^+ = f(x^m(1) + h(1), x^m(2) + h(2), \ldots, x^m(k) + h(k))$ and $F^- = f(x^m(1) - h(1), x^m(2) - h(2), \ldots, x^m(k) - h(k))$.

Step 3. Set: $x^{m+1}(i) \leftarrow x^m(i) - \mu\frac{F^+ - F^-}{2h(i)} \quad \forall i$. Increment m by 1 and set: $\mu \leftarrow A/m$. If $\mu < \mu_{\min}$ then STOP; else return to Step 1.

The convergence of this algorithm has been established in Spall (1992). We present an analysis, under some special conditions, that uses a different result. We first need to define some standard step-size conditions that will be needed. If μ^m denotes the step size in the mth iteration of the algorithm, then the conditions are:

$$\sum_{m=1}^{\infty} \mu^m = \infty, \qquad \sum_{m=1}^{\infty} [\mu^m]^2 < \infty. \tag{4}$$

For the subsequent analysis, we need a result in non-linear programming, which is stated next. The result, along with its proof, can be found in Bertsekas and Tsitsiklis (1996) in a more general version as Proposition 4.1 (page 141) The result is presented next.

Theorem 2. Let us assume that a function $f : \mathcal{R}^k \to \mathcal{R}$ satisfies the following conditions: $f(\vec{x}) \geq 0$ everywhere, and $f(\vec{x})$ is Lipschitz continuous. The core of the gradient descent algorithm can be expressed as:

$$x^{m+1}(i) = x^m(i) - \mu^m \left[\frac{\partial f(\vec{x})}{\partial x(i)}|_{\vec{x}=\vec{x}^{\ m}} + w^m(i) \right] \text{ for } i = 1, 2, \ldots, k$$

in which k denotes the number of decision variables, μ^m represents the step size in the mth iteration, while $w^m(i)$ is a noise term. Let the step size satisfy the step size conditions defined in (4). The history of the algorithm up to and including the mth iteration by the set: $\mathcal{F}^m$. Then, if

$$E[w^m(i)|\mathcal{F}^m] = 0 \text{ for every } i \quad \text{(Condition 1)} \tag{5}$$

and

$$E[||\vec{w}^{\ m}||^2|\mathcal{F}^m] = A + B||\nabla f(\vec{x}^{\ m})||^2 \quad \text{(Condition 2)} \tag{6}$$

for finite A and B, with probability 1, $\lim_{m\to\infty} \nabla f(\vec{x}^{\ m}) = 0$.

The above result says that a gradient descent algorithm, with imperfect gradient, can converge, under certain conditions. The next result, which uses Theorem 2, shows that simultaneous perturbation can converge.

Theorem 3. The simultaneous perturbation algorithm described above converges to a local optimum of the objective function, $f(\vec{x})$, if (i) exact function values are used in the algorithm, (ii) the objective function satisfies the Lipschitz condition, (iii) $f(\vec{x}) \geq 0$ everywhere, and (iv) the step-size is made to satisfy the standard conditions defined in (4).

Proof: Via the Taylor series, Spall (1992) shows that in his algorithm,

$$w^m(i) = \sum_{j \neq i; j=1}^{k} \frac{h^m(j)}{h^m(i)} \frac{\partial f(\vec{x})}{\partial x(j)} \text{ for every } i \tag{7}$$

and that $E[w^m(i)|\mathcal{F}^m] = 0$. This is Condition 1 (Theorem 2). We next test Condition 2. In the algorithm, for any i, j, $\frac{h(j)}{h(i)} = \pm 1$. Therefore, for any j and a given i,

$$\left[\frac{h(j)}{h(i)}\right]^2 = 1. \tag{8}$$

Then, using the Euclidean norm, we have that:

$$\begin{aligned} ||\vec{w}^{\ m}||^2 &= [w^m(1)]^2 + [w^m(2)]^2 + \cdots + [w^m(k)]^2 \\ &\leq \sum_{j=1}^{k} \left[\frac{\partial f(\vec{x})}{\partial x(j)}\right]^2 + \sum_{j=1}^{k} \left[\frac{\partial f(\vec{x})}{\partial x(j)}\right]^2 + \cdots + \\ &\quad \cdots + \sum_{j=1}^{k} \left[\frac{\partial f(\vec{x})}{\partial x(j)}\right]^2 \qquad (9) \\ &= k \sum_{j=1}^{k} \left[\frac{\partial f(\vec{x})}{\partial x(j)}\right]^2 = k||\nabla f(\vec{x}^{\ m})||^2 \end{aligned}$$

Line (9) follows from Equations (7) and (8). Condition 2 of Theorem 2 is hereby proved. Then from Theorem 2, the result follows. Q.E.D.

REVENUE MANAGEMENT

Airline companies in order to maximize their revenues divide their customer pool into a number of fare "classes." Customers who demand special features such as flexible time, customers flying on direct flights, or customers who arrive late in the booking horizon are made to pay higher fares. In fact, typically, many companies have 5 to 15 fare classes. Fare classes have nothing to do with seating arrangement within the aircraft. It makes business sense to reserve some seats for higher fare classes

Table 1: The booking horizon is 100 days long with three equal periods in which the Poisson rates of arrival are 0.5, 1, and 2 per day. Flight capacity = 100. A = Fare Class, B = Arrival Prob., C = No-Show Prob., D = Cancel Prob., and E = Cancel Penalty.

A	1	2	3	4	5	6	7	8	9	10
B	.19	.17	.15	.13	.11	.09	.07	.05	.03	.01
C	.001	.001	.001	.001	.001	.002	.009	.015	.08	.1
D	.025	.025	.025	.05	.05	.075	.125	.2	.2	.225
E	95	90	85	80	75	70	65	60	55	50

because they yield higher revenues. At the same time, one also has to address the question of overbooking. If flights are not overbooked, one runs the risk of flying with empty seats, which is expensive, and on the other hand excessive overbooking, which can lead to bumping a large number of passengers, can also be very expensive. This is a well-studied problem in the academic and industrial literature (see McGill and van Ryzin, 1999). The problem is stochastic because the arrival rates of passengers are random variables and so are the cancellations. In formal terms, the problem can be described as: Maximize $f(x_1, x_2, x_3, \ldots, x_k)$ where $f(.)$ is the expected revenue per flight on which x_i was used as the booking limit for the ith fare class. Here if $r > l$, then fare of class r is higher than fare of class l. The booking limits are implemented in the following manner. Let y_i denote the seats sold in the ith class. Then accept an incoming customer in class i, if $\sum_{j=1}^{i} y_j < \sum_{j=1}^{i} x_j$, and reject the customer otherwise.

We next present the problem details of a numerical on which we implemented simultaneous perturbation. Its performance was compared to that of a widely used heuristic called EMSR-b. Simultaneous perturbation, which took about 15 minutes on a Pentium PC, was able to outperform EMSR-b. The usefulness of simultaneous perturbation can be gauged from the fact that one is able to solve a 10-parameter problem in a reasonable amount of computer time. Using traditional finite difference approaches for calculating the gradient, this would take about ten times as much time. Table 1 provides the data related to the numerical. The EMSR-b policy returns an expected revenue of 37263.20 dollars per flight, while simultaneous perturbation returns an expected revenue of 38158.39 dollars per flight, which is an improvement of 2.4 percent over EMSR-b. The simulations were run for 10 replications with a 1000 flights in each replication.

CONCLUSIONS

We showed that in the presence of simulation-induced noise simulated annealing can mimic its noise-free counterpart. We also established convergence of simultaneous perturbation under some conditions. Finally, we presented some empirical results obtained from the use of simultaneous perturbation on an important problem in airline revenue management.

ACKNOWLEDGEMENTS

The author would like to acknowledge support from NSF grant ITR/AP 0114007 and from Emrah Ozkaya for some computer programs.

REFERENCES

Bertsekas, D.P. and Tsitsiklis, J.N., 1996, *Neuro-Dynamic Programming,* Athena Scientific, Belmont, MA, pp. 141-142.

McGill, J.I., and G.J. van Ryzin, 1999, "Research overview and prospects," *Transportation Science*, Vol 33 (2), pp. 233-256.

Pham, D.T. and K. Karaboga, 1998, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing, and Neural Networks*, Springer-Verlag, New York, pp. 187-218.

Spall, J.C., 1992, "Multivariate Stochastic Approximation using a simultaneous perturbation stochastic approximation," *IEEE Transactions on Automatic Control*, Vol 37, pp. 332-341.