퍼터베이션 분석을 이용한 대기행렬 네트워크의 최적화*

Optimization of Queueing Network by Perturbation Analysis

권치명**
Chimyung Kwon

Abstract

In this paper, we consider an optimal allocation of constant service efforts in queueing network to maximize the system throughput. For this purpose, using the perturbation analysis, we apply a stochastic optimization algorithm to two types of queueing systems. Our simulation results indicate that the estimates obtained from a stochastic optimization algorithm for a two-tandem queuing network are very accurate, and those for closed loop manufacturing system are a little apart from the known optimal allocation. We find that as simulation time increases for obtaining a new gradient (performance measure with respect to decision variables) by perturbation algorithm, the estimates tend to be more stable. Thus, we consider that it would be more desirable to have more accurate sensitivity of performance measure by enlarging simulation time rather than more searching steps with less accurate sensitivity. We realize that more experiments on various types of systems are needed to identify such a relationship with regards to stopping rule, the size of moving step, and updating period for sensitivity.

Key Words; Stochastic Optimization Algorithm, Perturbation Analysis, System Throughput.

* 이 논문은 (1997)년 한국학술진흥재단의 학술연구비에 의하여 지원되었음.
** Division of Management Information Systems, Dong-a University

1. Introduction

In this paper, we consider how simulation can be used to design queueing systems to yield optimal expected performance. We assume that the system performance of interest depends on the values of the input parameters and that we want to determine the optimal values of these parameters subject to certain constraints. More specifically, we are interested in maximizing the throughputs of queueing systems by distributing the total service rate across service stations in an optimal way.

Usually the performance of throughput, TP, is estimated by total service-completed entities per unit time. That is, $TP=P/S$, where P is total amount of service -completed entities for a simulation period S. Under a condition of the constant effort of service stations, we try to find an optimal service rate of each station such that the system throughput is maximized. We let t_i be the mean service time of station i. Then we can formulate this problem as an optimization problem with constraint given as follows:

$$\text{Maximize } TP=P/S \qquad (1)$$

$$\text{subject to } \sum_{i=1}^{m} t_i = \text{constant, and } t_i \geq 0 \quad \text{for } i=1, 2, \cdots, m,$$

where m is the number of service stations in the system.

Although the objective is simple, its fulfillment is in general very complicated since the objective function is basically stochastic and its value at given point (service times of m stations) can be realized with noise through simulation. Also the set of variables affecting the system throughput is not small. For instance, in production system, such variables include buffer sizes between adjacent stations, production rates for machines, failure rates for machines, and etc [15].

Due to the complexity and stochastic nature of system, it is difficult to obtain the optimal service rate of each service station by mathematical models. For most cases, discrete event simulation is the only way to model these types of systems [3, 13]. To find the optimal point with respect to considered decision variables, brute-force parametric analysis using direct simulation and response surface methodology is not appropriate since optimization through direct simulation invariably require a large number of simulation runs to be made.

To remedy this kind of problem, recently, a stochastic approximation algorithm is introduced in the simulation optimization problem [12]. This method is based on informations obtained from sample observations of performance measures and parameters of interest during a single run of simulation. Thus this method requires less computational efforts than other commonly used simulation methods since, unlike other simulation analysis methods, this algorithm gives an estimate of the optimum in a single simulation run.

In this research, we apply the stochastic optimization algorithm to open queueing system of simple tandem queues and closed queueing system of production system, which is called as a closed loop manufacturing system, and explore the efficiency of this algorithm under certain conditions. We also discuss the issues in implementation of the stochastic optimization algorithm.

2. Perturbation Analysis

To apply the stochastic optimization algorithm, we need the derivatives of performance measure with respect to decision variables. Perturbation analysis (PA) provides a mean to estimate the gradient of throughput with respect to the mean service time, t_i of service station i, that is, dTP/dt_i ($i=1, 2, \cdots, m$). Even though we can't identify the function of TP, we can obtain dTP/dt_i if we measure the effect of a small change of t_i to TP. For a fixed amount P, TP is inversely proportional to simulation period S, which depends on mean service time of station i, t_i ($i=1, 2, \cdots, m$). Thus, the sensitivity of TP with respect to t_i is given by

$$\begin{aligned} dTP/dt_i &= dTP/dS \times dS/dt_i \\ &= -P/S^2 \times dS/dt_i. \end{aligned} \quad (2)$$

PA has been used for estimating derivative of considered performance measure in various queueing networks [1, 4, 5, 7, 11, 14]. We now briefly describe perturbation algorithm for obtaining dS/dt_i. PA is a technique that has the capability of producing gradient information or sensitivity vector in real time, without the necessary duplication of the discrete event dynamic system (DEDS) n times, provided we can obtain some relatively simple processing of the system behavior as it evolves. Ho (1983) first applied this technique to analyze the outputs of sample path of a discrete event simulation. Consider a DEDS as illustrated in Figure 1. Suppose that the DEDS has certain parameters of interest t, and an output which is a performance measure such as the throughput of system. We observe the behavior and the performance measure of the DEDS over a period of time [0, S].

Now, imagine that we have an identical twin to the DEDS in Figure 1 in the sense that this twin duplicates exactly the behavior of the original DEDS. In the case of simulation, we can create this twin system by using the same random seeds. However, for this identical twin DEDS, we are endowed with the additional capability of being able to modify the input parameter t by a small amount Δt. Clearly, the output of the twin DEDS will in general become different from the original by some amount. We denote the perturbed performance measure as $TP(t+\Delta t)$ and compute the sensitivity of TP as follows:

$$\text{Sensitivity of } TP = [TP(t+\Delta t) - TP(t)]/\Delta t \quad (3)$$

By perturbation, we mean the difference between the time instants of two corresponding events in the nominal and perturbed paths. If a parameter changes by a small amount dt, (we denote dt as Δt goes to 0), this change usually induces one or more perturbations in time of occurrence of system events along the nominal path. This is called as perturbation generation.

The main feature of PA is its ability to predict the effect of small changes in service times on the throughput by observing a single sample trajectory of simulation run. That is, a change in parameter is hypothesized and the effect of such a change

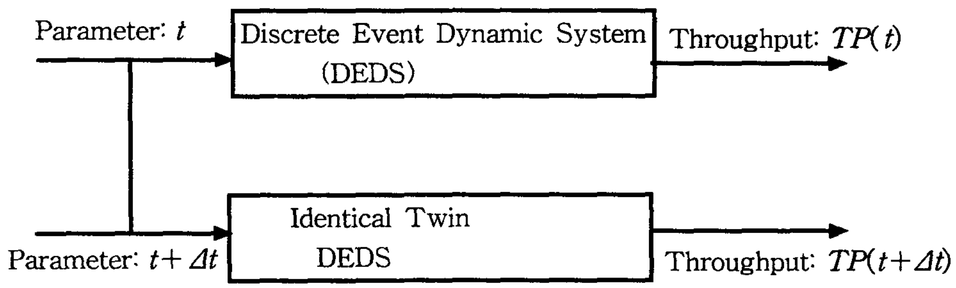

Figure 1. Sensitivity of Throughput in DEDS

is estimated without an actual change in simulation run. During observing the sample path, we continuously check the server's state of each station. If the ith station's mean service time t_i is decreased by a small amount Δt_i, we investigate the possible occurrences of its effect to services of other stations. This procedure of investigation is referred to as a perturbation propagation.

Perturbation Generation

We suppose that the random service time S_i of the ith station is exponentially distributed with mean t_i. Then the cumulative distribution function of S_i with mean t_i is given by

$$F_{t_i}(s_i) = 1 - \exp(-s_i/t_i) \quad (4)$$

which implies that

$$s_i = F^{-1}_{t_i}(u_i) = -t_i \log(1 - u_i), \quad (5)$$

where u_i is the outcome from uniform distribution in U(0, 1). Suppose that t_i is perturbed by Δt_i from nominal path. Then the perturbed outcome of S_i would be

$$s_i + \Delta s_i = F^{-1}_{t_i+\Delta t_i}(u_i), \quad (6)$$

Thus, in terms of actual observations, we get the relationship

$$s_i + \Delta s_i = F^{-1}_{t_i+\Delta t_i}(u_i) = F^{-1}_{t_i+\Delta t_i}(F_{t_i}(s_i)) \quad (7)$$

From this equation and equation (5), we have the perturbation in outcome to be propagated as follows:

$$\Delta s_i = F^{-1}_{t_i+\Delta t_i}(F_{t_i}(s_i)) - s_i$$
$$= -(t_i + \Delta t_i)\log[\exp(-s_i/t_i)] - s_i \quad (8)$$

Thus we have

$$\Delta s_i / \Delta t_i = s_i / t_i \quad (9)$$

Therefore an infinitely small change of t_i for s_i gives that $ds_i/dt_i = s_i/t_i$. This relationship implies that the perturbation in the service time of s_i of station i by ds_i can be translated into a ds_i perturbation in the service time duration of the station along the nominal sample path, and that its gradient is equal to ds_i/dt_i (For the more detailed description, see the reference [17]). Also when t_i is a location parameter for the distribution of S_i, Suri [1987] showed that $ds_i/dt_i = 1$.

Perturbation Propagation

If a system is perturbed in time by Δs_i, how will this perturbation evolve and propagate to the rest of the system? The evolution of this perturbation is governed by perturbation rules, which is due to the result of service to service interaction and capacity of queue. Consider the ith server. It may stop at times because of interaction with other servers in the network. For instance, (i) if the queue Q_i waiting for the ith service is empty, then the ith server can not service and (ii) if the Q_i which is a space for the completed entities from the $(i-1)$th server is full, then the $(i-1)$th server can not service.

If any of these two conditions occur, the system reduces its service rates. If such a state is caused by the first condition, it is called No Input (NI). On the other hand, if it is caused by the second, it is called Full Output (FO). If an entity, upon the completion of its service at the ith server, is destined to the $(i+1)$th service, and at that time the buffer of Q_{i+1} is full, then this entity has to stay in the ith server until the $(i+1)$th server finishes its service to its current entity. At the same time, the ith server is forced down from serving other entities. In this case, we say that the ith server meets a FO and the entity is blocked at the ith server. Thus the perturbation rules can be summarized as follows:

(i) if the ith server has a perturbation when it completes its service to its entity, this perturbation is propagated to the completion time of the next server, unless this server meets a NI or FO.

(ii) If a NI of the ith server is terminated by an item from server j, then after this NI, the server i will acquire the same perturbation as server j.

(iii) if a server i meets a FO caused by server j, then, after this FO, server i will acquire the same perturbation as server j.

From a nominal path with parameter t_i, $(i=1, 2, \cdots, m)$, using these perturbation generation and propagation rules, we can obtain an entire perturbed path with parameter $t_i + \Delta t_i$, from which we can calculate performance measure TP for perturbed system. The value of dTP/dt_i gives an estimate of the derivatives of TP with respect to the mean service time of t_i. We finally illustrate PA algorithm for obtaining the dTP/dt_i as follows:

PA algorithm for estimating derivatives of TP with respect to t_i

(0) Initialization: We set $A(i, j) = 0$ for $i, j = 1, 2, \cdots, m$
When $i=j$, $A(i, j)$ denotes the accumulator for perturbation generated at station i.

When $i \neq j$, $A(i, j)$ denotes the accumulator for propagation of perturbation generated at station i to station j.

(1) Perturbation Generation: Each time station i ends its service, we add the gradient of random service time s_i with respect to t_i, ds_i/dt_i, to accumulator $A(i, i) = A(i, i) + ds_i/dt_i$, where $A(i, i)$ denotes accumulated perturbations generated at station i.

(2) Perturbation Propagation:

a) If an entity leaving station j going to station k terminates an idle period of station k, we move $A(i, j)$ into $A(i, k)$ for $i = 1, 2, \cdots, m$.

b) If an entity leaving station j going to station k terminates a blocked period of station j, we move $A(i, k)$ into $A(i, j)$ for $i = 1, 2, \cdots, m$.

(3) Estimation of Derivatives: At the end of simulation, we estimate derivative as follows

$$dTP/dt_i = dTP/dS \times dS/dt_i$$
$$= -P/S^2 \times A(i, m), \text{ for } i = 1, 2, \cdots, m$$

3. Stochastic Optimization Algorithm

A stochastic approximation algorithm for level crossing problem was first introduced by Robbins and Monro(1951). This method tries to find the real values x^* so that $M(x^*) = p$, where $M(x^*)$ is the expectation of observable function of $Y(x)$ and p is a real constant. With appropriate assumptions on $M(x)$ and $Y(x)$, they showed that the following iterative scheme converges to x^*:

$$x^{(n+1)} = x^{(n)} - a/n[\,Y(x^{(n)}) - p]$$

as n goes to infinity, (10)

where an initial value of $x^{(1)}$ is arbitrary chosen, and a is a constant.

The single run optimization method for the level set problem can be used for simulation optimization problem by incorporating PA to estimate the derivative of performance measure (PM) with respect to system parameter [2, 4, 6]. The basic idea is that rather than running whole a new experiment each time to compute the derivatives, we obtain them within a single simulation run. Through a single run, we compute a gradient, which has a noise, of considered PM (obtained by simulation) with respect to system parameter, and we use this noisy gradient effectively for hill climbing of unknown function of PM.

Suppose that we simulate the system with given service times of stations, s_i ($i = 1, 2, \cdots, m$) until the first L parts are completed. Based on this simulation run, we compute the gradient of TP with respect to t_i by using PA algorithm and then we simulate the system with the updated service times of stations repeatedly in a single run. In other word, for searching the maximum TP under the condition of constant total service time of system, we would continuously move other points of t_i's (service times of stations) with maintaining the feasibility such that we have the improved TP in a single run. In this way, an estimate of the optimum service time of each station can be obtained at the end of a simulation run. This searching scheme is similar to that of the non-linear optimization with constraints.

When updating the new t_i's, in order to maintain the feasibility of constant system service time, we take the average the sample gradients of dTP/dt_i ($i=1, 2, \cdots, m$.) and assign appropriate weights to them. Thus, a hill climbing optimization scheme uses an iterative algorithm, which is of form given as follows:

$$t_i^{(n+1)} = t_i^{(n)} + a_n[dTP/dt_i^{(n)} - 1/m \sum_{i=1}^{m} dTP/dt_i^{(n)}] \qquad (11)$$

where n stands for the nth iteration; $t_i^{(n)}$ and a_n are the updated service time of station i and the moving weight of t_i at the nth iteration respectively [2, 9]. Since the second term of this equation is the projected gradient on the constraint hyperplane $\sum_{i=1}^{m} t_i = constant$, we always have the constant system service time. We now present stochastic optimization algorithm and its discussions follow the algorithm.

Stochastic Optimization Algorithm

(0) Initialization: Set the iteration number $n=1$ and choose initial values of t_i $(i=1,2,\cdots,m)$

(1) Simulate the system at $t_i^{(n)}$ until L parts are completed and estimate the derivatives of dTP/dt_i using PA algorithm in Section 2.

(2) Update the service time of each station as follows:

$$t_i^{(n+1)} = t_i^{(n)} + a_n[dTP/dt_i^{(n)} - 1/m \sum_{i=1}^{m} dTP/dt_i^{(n)}],$$

where $a_n = A/n$ with $A=5$.

(3) Check the non-negativity of mean service time $t_i^{(n+1)}$ and adjust $t_i^{(n+1)}$ to be positive:

If $t_i^{(n+1)}>0$ ($i=1,2,\cdots,m$), then go to step 4.

Otherwise reduce the moving size in (2) by changing the value of $a_n = A/2n$.

(4) Stopping criterion: If $a_n[dTP/dt_i^{(n)} - 1/m \sum_{i=1}^{m} dTP/dt_i^{(n)}] < \varepsilon$, then stop and find estimates for the optimum t_i^*. Otherwise go to step (1) with increasement of n by 1.

At step (1), the gradient dTP/dt_i is stochastic and its value depends upon the simulation time or the number of total-service completed entities, L. So its estimate may have biases due to the initial transient period of simulation run and effects of updating the $t_i^{(n)}$ with noises. Similarly to the stochastic optimization procedure of Robins and Monro(1951) , step (2) adopts a_n to satisfy the conditions;

$$\lim_{n\to\infty} a_n = 0, \quad \sum_{n=1}^{\infty} a_n = \infty \text{ and } \sum_{n=1}^{\infty} a_n^2 < \infty;$$

and simply chooses as a constant multiple of reciprocal of the iteration number n. Such a choice results in larger moving length of t_i in the first few steps, on the other hand, the moving length will be shorter as n increases.

Step (3) is for prohibiting the operation time of each station to be negative. When estimates of derivatives include large errors and moving sizes of first-few-stages are not small, the updated $t_i^{(n)}$ may be negative. In this case, we shorten the length of moving step by half to ensure the positive $t_i^{(n)}$. The interpretation of stopping criterion is that if the maximum moving length for $t_i^{(n)}$ $(i=1,2,\cdots,m)$ is less than the prescribed value ε, we stop the algorithm. A choice of this value gives a trade-off relationship between run length and accuracy of the estimated optimum.

4. Numerical Example

We apply the stochastic optimization algorithm to two types of queueing system: a simple tandem queueing system and a closed loop manufacturing system (CLMS). We conduct a set of simulation experiments on these models to evaluate the performance of a stochastic optimization algorithm and offer a summary and results.

4.1 Simple tandem queueing system

For a simplicity, we consider a two-node tandem network, where the first node is an M/M/1 queue (with mean inter-arrival time of 2 and mean service time of t_1) whose output feeds a second queueing node with a single exponential server of mean service time, t_2 (see Figure 2). For this system, we try to obtain the optimal service time of each node which yields the maximum throughput by applying a simulation optimization technique when the total service time, (t_1+t_2) is 3.2.

We can easily find that the maximum throughput occurs when the service time of each station is 1.6. Assuming that the optimal service times are unknown, we apply the stochastic optimization algorithm to estimate them. A simulation model is coded in SLAM II network model. We simulate this system with the initial values of $t_i^{(1)}$=(1.8, 1.4) and statistics are cleared at a time 0. We get the sensitivity of TP with respect to t_i each time when L products are completed, and then we update $t_i^{(n)}$ with increasement of n by 1.

We use a stopping rule of algorithm that the maximum deviation percentage of an estimate for the optimal service time, t_i^* is in the range of less than 3% from t_i^*, and we explore the efficiency of algorithm with L = 30, 40, 50, 60 and 70.

A summary of simulation results is

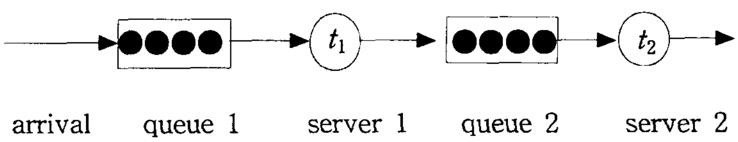

Figure 2. Two-node tandem queue system

Table 1. Estimates for Optimal Service Time

L	Run Length	Estimate for t_1^*	Estimate for t_2^*
30	67,490	1.65	1.55
40	1,247	1.64	1.56
50	704	1.62	1.58
60	503	1.61	1.59
70	461	1.62	1.58

presented in Table 1, where run length and an estimate for t_i^* are shown in terms of number of completed parts L. As L increases, the run length decreases and deviation from the optimum tends to be smaller. In this simple example, the estimate for the optimal service time is quite accurate.

4.2 Closed loop manufacturing system

A diagram of CLMS is depicted in Figure 3. Six stations are connected together in a closed loop. The workpiece starts at load station, pass through each station in sequence, emerge from the last station (station 6), and go to unload station as the final product. Six pallets carry material in a given sequence for appropriate service of each station. We consider CLMS operating with no rework loop.

One of buffer space is afforded between adjacent stations. A service time of each station is a deterministic for simplicity. We also suppose that a failure for operating each station happens randomly with probability 0.05. If a station failure occurs, it needs repair, but operations of other stations are independent to failed station. The repair time of failed station is uniformly distributed [6, 66]. Finally, we assume that a moving time of pallet from a station to adjacent one is zero.

For a system described above, we try to find the optimal service time of each station which yields the maximum throughput by applying a simulation optimization technique in a case that total service time of 6 stations is 36. This system is also coded in SLAM II network model.

In this example, we can easily realize that by symmetry, the maximum throughput happens when the service time of each station is 6. Assuming that the optimal service times are unknown, we apply the stochastic optimization algorithm to estimate them. We start simulation with the initial values $t_i^{(1)}$ = (9, 6, 3, 7, 8, 3). We obtain the sensitivity of TP with respect to t_i each time when L products are completed, and then we update $t_i^{(n)}$ with increasement of n by 1. If a simulation satisfies the stopping condition, then we find the estimate for the

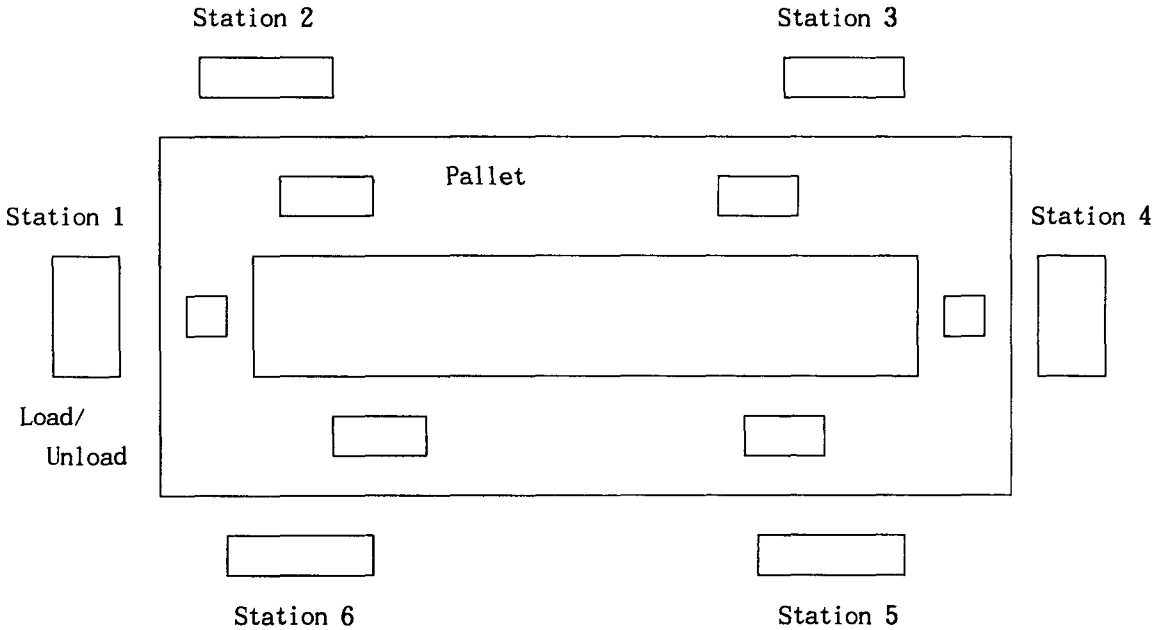

Figure 3. Closed Loop Manufacturing System

optimal service time of each station, t_i^* ($i = 1, 2, \cdots, 6$). We investigate the efficiency of the algorithm with L = 20, 30, 40, 50, 60 and 70.

Table 2 summarizes the simulation results obtained by 4 independent replications. Run length, an estimate for t_i^* and its maximum deviation from the optimum are shown in terms of number of completed parts L. Statistics are cleared at a time 0. The maximum deviations are in the range from 0.50 to 1.90. The largest deviation of the estimated optimal from the true optimum occurs when L=30 at replication 4, which is 1.90. As L increases, the maximum deviation tends to be smaller and stable. Figure 3 presents these results graphically.

The maximum deviation for a case that L=30 at replication 1 is the least one, which is a quite different from our expectation. We consider that such a result may be due to the stochastic property of multi-dimensional sensitivities. We need a much larger set of experiments for identifying the relationship between the accuracy of algorithm and run length.

Table 2. Estimates for Optimal Service Time and its Deviation from Optimum

L \ Rep.		Rep 1	Rep 2	Rep 3	Rep 4
	Estimate for t_i^*	5.77, 6.57, 6.21 6.84, 5.78, 4.84	6.89, 4.90, 6.04 6.36, 5.47, 6.34	5.49, 6.88, 6.23 6.15, 6.38, 4.87	6.00, 6.63, 5.46 6.15, 6.18, 5.59
L=20	run length	151,600	99,210	190,300	174,500
	Max deviation from optimum	1.16	1.10	1.13	0.63
	Estimate for t_i^*	5.66, 6.50, 6.21 5.84, 6.16, 5.64	5.37, 6.72, 6.29, 5.70, 5.98, 5.93	4.95, 6.88, 6.52 5.98, 6.40, 5.27	6.63, 4.10, 5.68 6.55, 6.66, 6.28
L=30	run length	246,000	294,900	275,700	156,900
	Max deviation from optimum	0.50	0.63	1.05	1.90
	Estimate for t_i^*	4.95, 6.07, 6.32 6.35, 6.22, 5.49	5.86, 4.64, 6.58 5.56, 6.28, 7.09	5.56, 6.52, 6.12 5.61, 5.94, 6.24	5.10, 6.69, 6.45 6.23, 5.54, 6.00
L=40	run length	370,900	146,700	391,600	428,700
	Max deviation from optimum	1.05	1.36	0.52	0.90
	Estimate for t_i^*	5.44 ,6.52, 6.37 6.02, 5.69, 5.97	5.89, 6.50, 6.29 5.06, 5.92, 6.35	5.27, 6.70, 6.05 5.96, 5.97, 6.05	5.35, 6.62, 5.99 6.20, 6.71, 5.14
L=50	run length	500,400	515,100	501,800	420,400
	Max deviation from optimum	0.52	0.94	0.73	0.86
	Estimate for t_i^*	5.64, 6.63, 6.38 5.67, 5.63, 6.06	5.37, 6.58, 6.30 6.00, 5.85, 5.91	5.30, 6.63, 6.48 5.75, 5.46, 6.37	5.22, 6.63, 5.90 6.32, 5.83, 6.11
L=60	run length	602,000	560,700	622,900	594,400
	Max deviation from optimum	0.63	0.63	0.70	0.78
	Estimate for t_i^*	5.63, 6.73, 6.43 5.03, 6.09, 6.09	5.47, 6.66, 6.23 6.17, 5.68, 5.80	5.35, 6.59, 6.42 5.44, 5.91, 6.29	5.35, 6.59, 6.12 5.65, 5.95, 6.34
L=70	run length	749,500	697,100	733,500	705,200
	Max deviation from optimum	0.97	0.66	0.65	0.65

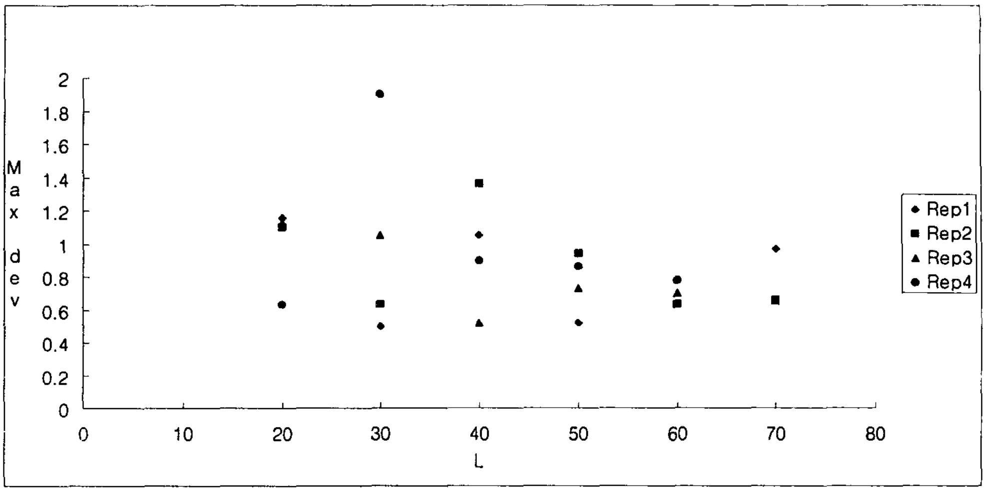

Figure 4. Maximum Deviation from Optimum with Respect to L

5. Conclusion

We apply a stochastic optimization algorithm to a simple tandem two-node queue network and a specific type of production system. We use the perturbation analysis for estimating the gradient of station's service time with respect to the system throughput, and implement it in SLAM II. Using the estimated gradients, we also implement a stochastic optimization algorithm in SLAM II. Thus far, much of research on the PA focused experimental results demonstrating its accuracy for various systems [6, 9, 10]. As a step in this direction, we here study PA applied two queueing systems to estimate the throughput.

Our simulation results show that the estimated optimum for the two-node tandem queue network is quite accurate, but that for CLMS is a little apart from the known optimum. We consider that this is due to the stochastic property of the estimated gradient and some bias occurred in updating the sensitivity for transient period. We also find that as simulation time for obtaining the new gradients increases, the maximum deviation of the estimated optimum from the true optimum tends to be small and stable.

Thus in the procedure of updating a new service time of each station (finding the moving direction for a new point), it would be desirable to have more accurate sensitivity by enlarging simulation time rather than to have more searching steps with less accurate sensitivity even though we should pay the cost of more running.

Of course, we need more experimentations to identify such a relationship between the accuracy of algorithm and run length of updating the sensitivity, and to evaluate the efficiency of considered algorithm with regards to stopping criterion, moving step, and updating period for sensitivity. Future research includes these problems in addition to convergency of the algorithm and applications of algorithm to more optimization problems.

References

[1] Ackere, A. V. and P. Ninios, "Simulation and Queueing Theory Applied to a Single -Server Queue with Advertising and Balking," *J. of the Operational Research Society*, Vol. 44 (1993), pp. 407-414.

[2] Andradottir, S., "A New Algorithm for Stochastic Optimization," *Proceedings of the WSC* (1990), pp. 364-366.

[3] Andradottir, S., "Simulation Optimization," Handbook of Simulation Principle, Methodo -logy, Advance, Applications, and Practice, ed. by J. Banks, John Wiley & Sons, Inc. (1998), pp. 308-333.

[4] Atkinson, J. B., "A note on a Queueing Optimization Problem," *J. of the Operational Research Society*, Vol. 47 (1996), pp. 463-467.

[5] Bashynam, S and M. C. Fu, "Application of Perturbation Analysis to (s, S) Inventory Systems," *Proceedings of the WSC* (1991), pp. 994-1003.

[6] Carson, Y. and A. Maria, "Simulation Optimization Methods and Applications," *Proceedings of the WSC* (1997), pp. 118-126.

[7] Chong, E. K .P. and P. Ramadge, "Optimization of Queues Using Infinitesmal Perturbation Analysis-Based Stochastic Algorithm with General Update Times," *SIAM J.*, Vol. 31, No. 3. (1993), pp. 698 -732.

[8] Dupis, P. and R. Simha, "On Sampling Controlled Stochastic Approximation," *IEEE Transactions on Automatic Control*, Vol. 36, No. 8 (1991), pp. 915-924

[9] Ecuyer, P. L., N. Giroux and P. W. Glynn, "Stochastic Optimization by Simulation: Numerical Experiments with the M/M/1 Queue in Steady-state," *Management Science*, Vol. 40, No. 10 (1994), pp. 1245-1261.

[10] Ecuyer, P. L. and P. W. Glynn, "Stochastic Optimization by Simulation: Convergence Proofs for the GI/G/1 Queue in Steady -state," *Management Science*, Vol. 40, No. 11 (1994), pp. 1245-1261.

[11] Glasserman P. and S. Tayur, "Sensitivity Analysis for Base-stock Levels in Multiechelon Production-inventory Systems," *Management Science*, Vol. 41, No.2 (1995), pp. 263-281.

[12] Ho, Y. C. and Cao, X, "Perturbation Analysis and Optimization of Queueing Networks," *Journal of Optimization Theory and Applications*, Vol. 40, No. 4 (1983), pp. 559-582.

[13] Ho, Y. C. and X. Cao, Perturbation Analysis of Discrete Event Dynamic Systems, Kluwer Academic Publishers, Boston, 1991.

[14] Lin, C., C. Madu, T.W. Chien and C. Kuei, "Queueing Models for Optimizing System Availability of a Flexible Manu -facturing System," *J. of the Operational Research Society*, Vol. 45 (1994), pp. 1141-1155.

[15] Owen, A. E., Flexible Assembly Systems, Plenum Press, New York, 1984.

[16] Robbins, H and S. Monro, "A Stochastic Approximation Method," *Annals of Math -ematical Statistics* Vol. 22 (1951), pp. 400-407.

[17] Suri, R., "Implementation of Sensitivity Calculations on a Monte Carlo Experiment," *Journal of Optimization Theory and Applications*, Vol. 40, No. 4 (1983), pp. 625-630.

● 저자소개 ●

권치명
1978년 서울대학교 산업공학과 졸업
1983년 서울대학교 대학원 산업공학과 졸업
1991년 VPI & SU 산업공학과 박사
현 재 동아대학교 경영정보과학부 교수
관심분야: Simulation Modeling & Output Analysis, Simulation Optimization, FMS