Improvement due to Flexibility for Stochastic Line Balancing

Ick-Hyun Nam
College of Business Administration,
Seoul National University

Abstract

In this paper, we tried to improve a given stochastic processing line which is perfectly balanced. By introducing flexibility to a stochastic processing line, we could achieve the benefit of resource pooling. When we apply priority sequencing to the flexible processing line, we could attain additional improvement in reducing mean throughput time. The ratio of improvement from flexibility is shown to be amazingly large by using heavy traffic approximation. We also considered the set-up time loss due to flexibility and derived the appropriate range for flexibility and set-up time. The result of this paper implies that we should consider the alternative of introducing flexibility to a processing system rather than trying to optimally allocate each segments of jobs and get a balanced processing line.

1 Introduction

In mass production we encounter the problem of grouping work tasks along production line so as to achieve a required performance. The production line consists of a series of work stations through which a product goes. This kind of production line is called an assembly line or fabrication line. Designing the production line, especially listing tasks to be performed at each station considering the times required to perform and the constraints on the order of the tasks, is called a line balancing problem. There have been developed several heuristics or optimization methods for line balancing problem.

Miltenburg and Wijngaard(1994) extended the traditional line balancing methodology by considering U-type line. The U-type production line was suggested by several authors advocating JIT system(Monden(1993), Harmon(1992)). Using the fact that there are more possibilities for grouping tasks into stations under U-type, they showed that the number of stations required on a U-line is fewer than or equal to that of traditional linear production line and the benefit of using U-line was shown by several examples. But their model did not deal with the case where the tasks require a random amount of time. Leu et al.(1994) suggested a genetic algorithms to solve the assembly line balancing problem. They showed that their algorithm is superior to the conventional heuristics based on the sample problems.

Ghosh and Gagnon(1987) is a rather comprehensive and good review paper on the assembly line balancing problem. Assembly line balancing problem can be classified by whether the task times are deterministic or stochastic. There have been extensive researches on the deterministic case. But in the deterministic line balancing problem, we have several problems. One of the most critical problems is that the work amount is usually denoted as the mean value of processing time required. Using the mean value may lead to suboptimal solution. There are many cases where we should consider the processing time at a station as a random variable and incorporate the random event of delay. Even for the case where processing is done by machines and requires a fixed amount of work, there occur machine breakdowns and therefore the completion time becomes variable.

Following the classification of Ghosh and Gagnon(1987), our model is in Single Model Stochastic category. We now give our brief review of researches in this category. Hillier and Boling(1966,1979) studied the "bowl phenomena" where middle stations are considered more critical since their variability gives more impact on the flow along the line. Thus they suggested increasing capacity of these middle stations. El-Rayah(1979), through computer simulation, confirmed that assigning lower operation times to the middle stations(bowl phenomenon) consistently gave better output rates than those of balanced lines under variable processing times. But Smunt and Perkins(1985) did a simulation studies on the more realistic environments and indicated that a balanced line is not worse than an unbalanced one when task times are modelled with typical values of task variance. Their simulation results verify the benefits of using the bowl distribution for task times allocation when line lengths are short and task time variations are large. Several researchers including Kottas and Lau(1981) studied heristics rules for solving the line balancing problem with stochastic task times.

In this paper, we consider a line balancing problem with random work amount. In addition to the randomness in work amount, we have a random arrivals of jobs. We are given a stochastic processing line and try to improve its performance. Our main focus is on introducing flexibility into the stations and constructing a manufacturing cell rather than finding an optimal allocation of tasks with variable processing time. We show that when it is applicable, the flexible manufacturing cell is strictly better than the optimally balanced line. We first analyze the processing line with N stations and derive the total benefit of incorporating flexibility. For this analysis, we use the method of heavy traffic analysis popular in queueing network. In several applications(Harrison and Wein(1989), Wein(1990), Wein(1991)), heavy traffic method seems to give an asymptotic solution to problems that are intractable in their exact forms. Simulation results are given to support our argument. And then using two station processing line, we further analyze the benefit of flexibility and devide it according to the sources of benefit. Set-up time loss, possibly a critical disadvantage from flexibility, is studied and we give the range in which the benefit from flexibility outperforms its set-up time loss. In comparing the outputs of each system, we used the mean number of jobs in the system as our performance measure. According to Little's law, this criterion is equivalent to that of mean throughput time of a job.

2 N Station Processing Line

In this section, heuristic arguments will be given to estimate the performance improvement obtainable due to flexibility under heavy traffic[1] conditions. Simulation results will be presented, which show that our heuristic estimates are reasonably accurate. In each case, readers will see that processing flexibility improves system performance by a large factor, not just a modest percentage. The following will be used in deriving our argument. By extending [15] to our multi-customer class models, we can show that :

Proposition 1 *In heavy traffic limit, the multi-server system and the single server system with correspondingly large service rate have the same unfinished workload and throughput time process in probability.*

For the simplicity of analysis, we assume exponential distributions for inter-arrival and service times of customers. It should be noted that similar results hold also for more general cases and other types of flexibility were analyzed using heavy traffic analysis in Nam(1992).

2.1 Relations Among Four Systems

We consider as our original system N stations in series with external arrival rate λ from exponential distribution, a single server at each station, and exponential service times with mean $m = 1/\mu$ at each station. The traffic intensity parameter

[1] Heavy traffic means that the traffic intensity is approximately 1, i.e., $\rho = \lambda m \approx 1$. For more detailed explanation, refer to Harrison and Wein(1989).

is then $\rho = \lambda m$ at each station. This is a product form queueing network, and we can derive that the mean throughput time of an external arrival customer is

$$W_1 = N(\frac{m}{1-\rho}).$$

We seek to introduce flexibility to this serial processing line such that the servers can process several stages of jobs. Jobs waiting for processing at station i, i.e., stage i, are called class i customers. If all N servers are interchangeable, then we have Poisson input at rate λ, $N-1$ stages of feedback for each customer, and N interchangeable servers. Then we have not only resource pooling[2] but also sequencing capability due to a single server group processing multiple customer classes. The jobs can now be processed according to *priority sequencing* rule in order to minimize system queue size.

Figure 1 shows the original processing line for a two-station case. When we improve the system flexibility such that either of the two stations can serve both stages of service, then we have the pooled station consisting of two flexible servers. This flexible system(Figure 2), called System 2, now has one feedback. When we use the flexibility of servers in System 2 and implement an appropriate sequencing priority for the two customer classes, we achieve System 3. In System 3, we can assume that the external customer carries all the service requirements (the required service as class 1 plus that of class 2) when it arrives to the system. Therefore, no customer feedback occurs in System 3. Two adjacent vertical arrows in the figure of System 3 represent the external arrival customer's simultaneous carrying of class 1 and 2 workloads. When we divide the service requirement of a customer in System 3 by 2 (number of stations in the pooled station), we get the totally pooled single server system, denoted as System 4. In System 4, we have single server which is two times as fast as regular server.

[2] Resource pooling means that multiple servers have the flexibility of processing all the classes of jobs and thus there are no idle servers as long as there are jobs waiting.

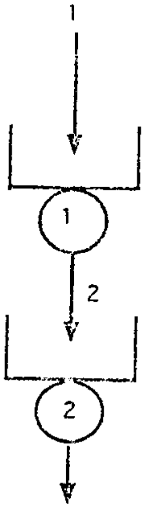

Figure 1: System 1

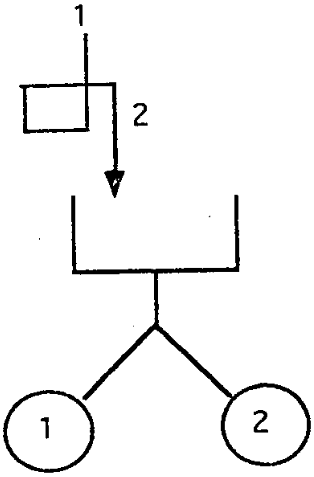

Figure 2: System 2

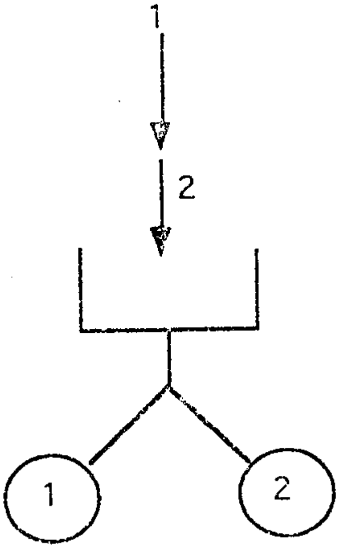

Figure 3: System 3

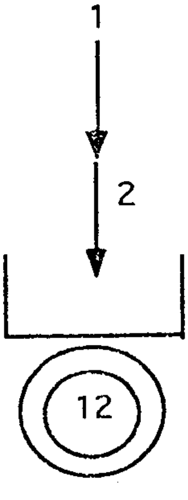

Figure 4: System 4

We now explain our sequencing control policy notations. The double slashes '//' separate the customer classes according to their sequencing priority. The customer classes on the left of '//' have higher priority than those on the right. For example, the sequencing priority (1,2//3,4) means that classes 1 and 2 have higher priority than classes 3 and 4. The classes on one side of '//' have identical priority and are served under FCFS(First Come First Serve) among themselves.

The fact that the priority scheme (2//1) gives the minimum mean system queue size in the heavy traffic limit can be shown by extending Theorem 4.2.1 of [20]. Implementing the priority scheme (2//1) to System 2, we have the same system as the $M/E_2/2(E_2$ denotes Erlang$(\mu, 2))$ of System 3 under FCFS where the k-th customer has service time $v_k^1 + v_k^2$ as far as the throughput time of a customer from outside the system is concerned(v^i denotes the random work amount for stage i). This equivalence results from the system's priority scheme. A high priority customer (class 2) is generated and continues to be served under (2//1) just when a low priority (class 1) customer completes its service. Therefore in System 3, we have the $M/E_2/2$ with FCFS where the customer k carries $v_k^1+v_k^2$ amount of work. Unlike in $M/G/1$, it is possible for both low and high priority class customers to be served simultaneously by each of the two servers. But the continuation of two consecutive services (class 1 and class 2) holds even in the $M/E_2/2$ system. As mentioned in Proposition 1, System 3 is equivalent to System 4 in throughput time in heavy traffic limit, where System 4 has one server with two times the service rate (capacity) as the individual server in System 3.

When we give high priority to class 2 customers in System 2, then they vanish in the heavy traffic limit and only low priority customers appear in the system (Theorem 4.2.1 of [20]). To compare the performance of System 3(flexible processing line with sequencing priority) with that of System 1, we should be able to analyze System 1 first. Where the product form of stationary distribution is available, we can explicitly calculate the benefits that result from flexibility. And under exponential distributional assumption, we get the product form solution.

We now consider a more general serial processing system, a tandem processing system consisting of N serial of stations. We call System 1 the original, or inflexible, system where a class k customer can be served only by station k. System 2 is called flexible when we improve flexibility to System 1 such that all the customer classes can be served by any of the available servers whose total number is N. For System 2, we had better use the priority scheme of $(N//N-1//\dots//1)$ and make System 3. This sequencing priority scheme can be shown to optimally minimize the system queue size by extending the idea for the two-station case. Using Proposition 1 the pooled system has the same throughput time as the totally pooled system in heavy traffic limit. For ρ near 1 this pooled system behaves approximately as one with a single super server working at rate N. Additionally the corresponding totally pooled system is easy to analyze in the heavy traffic limit.

2.2 Total Benefit of Flexibility

We now compare the original processing line with the totally pooled system. In the totally pooled system each *stage* of service has an exponential distribution with mean m/N, so a customer's *total* service time from the super server has an Erlang distribution with mean $N(m/N) = m$ and squared coefficient of variation $c_s^2 = N(m/N)^2/m^2 = 1/N$. The Pollaczek-Khinchin formula ([11]) gives the mean system (through all stages) throughput time of an external customer:

$$\begin{aligned} W_4 &\approx m(\frac{\rho}{1-\rho})\frac{1}{2}(1+c_s^2) \\ &= m(\frac{\rho}{1-\rho})\frac{1}{2}(1+1/N) \\ &= m(\frac{\rho}{1-\rho})(\frac{1+N}{2N}). \end{aligned}$$

And we are led to

$$\frac{W_1}{W_4} \longrightarrow \frac{2N^2}{N+1} \equiv f(N)$$

as $\rho \uparrow 1$. The ratio of improvement due to serial flexibility is approximately $\frac{2N^2}{N+1}$ in heavy traffic limit, and we can easily derive some properties of this improvement ratio function of N.

Proposition 2 *For the N station serial flexible system in heavy traffic, the improvement ratio is*

$$\frac{W_1}{W_4} \approx \frac{2N^2}{N+1}.$$

Denoting

$$f(N) \equiv \frac{2N^2}{N+1},$$

we have

1. $f(N) > N$ *for* $N > 1$.

2. *For a large value of* N, *we know that* $f(N) \simeq 2N$ *and the improvement ratio is approximately* $2N$ *for an* N *station serial flexible system.*

For the serial flexible system of N stations, the ratio of improvement is $f(N)$, which is greater than N. The benefit of flexibility results partly from resource pooling through dynamic routing of the flexible customers. By avoiding unnecessary idleness of the flexible servers, we can increase the systemwide utilization of server capacity. In addition to the resource pooling, we have the priority sequencing capability as a result of introducing serial flexibility. In a flexible processing line, since all servers can process all classes of jobs, we may choose which class of customers to process first in addition to the resource pooling effect. This option is very important in minimizing systemwide queue size since each class of customer has a different total mean service requirement until completion. By utilizing the extra option of priority sequencing in such a way that we put all the available servers first to the higher priority class customers who are nearer to completion in terms of service stage, we can reduce the systemwide queue size and throughput time together. In heavy traffic, the total benefit of flexibility is about $2N$ times when N is sufficiently large.

2.3 Simulation Results

We give some simulation results in order to check our analysis in heavy traffic setting. We used the simulation language SIMAN IV for the simulation displayed in this section. For the simulations, we used exponential distributions with corresponding parameters. We ran the simulation for $105,000$ time units with the first $50,000$ units truncated in order to delete the transient effect. The parameters for the simulation are $\lambda = 1$ and $m = 0.95$. Considering the serial flexible processing system, we have fewer customers even though we have more stations in the original system. This is interesting since we have the reverse inequality in the original system (see columns of System 3 and System 1 in Table 1). Fewer customers result in for the flexible processing line because as N increases we have a smaller variance in the unfinished workload process (i.e., the service time variance that is incorporated in the unfinished workload is $\frac{m^2}{N}$ and decreases as N increases). This decrease in variability gives a smaller mean value of workload. Restricting our attention to two station processing line, we now explicitly analyze the benefit due to flexibility.

3 Analysis of Two Station Processing Line

We assume that the job order follows a Poison process with arrival rate of λ as before. The job consists of two parts. The first of them requires processing at the first station. With the first part(denoted by v^1) being completed, the job can proceed for processing of its second part(denoted by v^2) at the next station.

Table 1: Simulation Results for N Station Serial Flexibility

N	$f(N) = \frac{2N^2}{N+1}$	*Improvement Ratios from Simulation*	*System 1 Mean Queue Size*	*System 3 Mean Queue Size*
2	2.67	2.5	29.99	12.00
3	4.5	4.14	44.63	10.78
4	6.4	6.13	61.94	10.11
5	8.33	8.47	78.00	9.20

The service or processing time at each station follows an exponential distribution with parameter of $\mu = 1/m$.

3.1 Balanced Line(System 1)

We analyze perfectly balanced line system with two serial stations. This system can be represented by a serial queueing system. Since both stations have the same service time distributions, we have perfectly balanced processing line. Using the Burke's theorem(Burke 1966), we get the following results:

$$W_1 = \frac{2}{\mu - \lambda}$$

$$L_1 = \frac{2\rho}{1-\rho}.$$

3.2 Flexible System-Manufacturing Cell(System 2)

Suppose we introduce flexibility such that each station can now process both the first and the second parts of jobs. Then we have two station queueing system with feedback. Using the famous Jackson's result(Jackson 1963), we have the same mean queue size as the $M/M/2$ system where the arrival rate is 2λ. The effective arrival rate of 2λ is derived from the following traffic equation with feedback ratio of $1/2$.

$$x = \lambda + x/2.$$

Using the result for $M/M/2$ queueing system, we get

$$p_0 = \frac{1-\rho}{1+\rho}$$

$$W_2 = \frac{1}{\mu(1-\rho^2)}$$

$$L_2 = \frac{2\rho}{1-\rho^2}.$$

Considering the ratio of L_1 and L_2, we get

$$\frac{L_1}{L_2} = 1 + \rho.$$

This means that we can achieve 100ρ % reduction in mean queue size by introducing flexibility.

3.3 Flexible System with Sequencing Priority(Systems 3 and 4)

For System 2, we introduced flexibility and thus could reduce the mean queue size or WIP. But we could have utilized the flexibility more effectively by using sequencing schedule. In System 2, we used FCFS sequencing rule. Suppose we use priority sequencing rule such that we give high priority to the jobs of which the first part has been completed, denoted by (2//1). This results in System 3. Since System 3 is hard to analyze, we deal with System 4 as a substitute for System 3. It is well known that for heavy traffic case($\rho \approx 1$) Systems 3 and 4 give approximately the same results in terms of mean queue size. For System 4, when we use the priority scheme mentioned above, the service time of a job can be represented by random variables, $\frac{v^1+v^2}{2}$, where $v^i \sim Exp(\mu)$. The reason we devide v^1+v^2 by 2 is that the service speed of the server in System 4 is two times as fast as before. The service time now has the mean of $1/\mu$ and the variance of $\frac{1}{2\mu^2}$. Using Pollaczek-Khinchin formula, we can derive

$$\begin{aligned} L_4 &= \rho + \frac{\rho^2+\lambda^2\frac{1}{2\mu^2}}{2(1-\rho)} \\ &= \frac{4\rho-\rho^2}{4(1-\rho)}. \end{aligned}$$

Comparing System 1 with System 4, we get the following ratio of reduction in mean queue size.

$$\frac{L_1}{L_4} = \frac{8}{4-\rho}.$$

We now compare System 2 with System 4. The improvement from System 2 to System 4 comes from the sequencing priority and this can be represented by the following ratio

$$\frac{L_2}{L_4} = \frac{8}{(1+\rho)(4-\rho)},$$

which can be easily shown to be larger than 1. We thus can calculate the improvement from sequencing priority as follows. The improvement fractile is the maximum k which satisfies the following inequality:

$$\frac{L_2}{L_4} = \frac{8}{(1+\rho)(4-\rho)} > k.$$

Denoting $g(k,\rho) = k\rho^2 - 3k\rho + 8 - 4k$, we need $g(k,\rho) > 0$ for $0 < \rho < 1$. This inequality is true if $g(k,1) > 0$ since $g(k,\rho)$ has the minimum at $\rho = 1$. Solving this inequality, we get

$$k < 4/3.$$

Therefore, we have at least 33.3% additional improvement from sequencing priority.

3.4 Set-Up Time Loss

In implementation, we have to consider not only the benefit from flexibility but also the disadvantage from it. One of the critical disadvantages for introducing flexibility is that we have to set up for processing different customer classes. When we actually introduce flexibility into a production line, there are many cases where we have to consider the set-up time loss. When the flexible station completes the first part of job, it may require some time for setting up for processing the second part. This time is called set-up time or swich-over time. Let us suppose that we need set-up time of s. When s is very large, the benefit from flexibility can be more than compensated by the processing time loss due to set-ups. Therefore we can derive the range of s in which the benefit from flexibility is larger than the loss due to set-ups. When we incorporate the set-up time, the mean processing time for System 4 becomes $m+s$ with the variance the same as before. Denoting $\rho_s = \lambda s$, the mean number of queue size in System 4 is

$$L_{s4} = \frac{2(\rho+\rho_s) - (\rho+\rho_s)^2 + \rho^2/2}{2(1-\rho-\rho_s)}.$$

Solving

$$\frac{L_1}{L_{s4}} > 1,$$

that is,

$$\frac{2\rho}{1-\rho} > \frac{2(\rho+\rho_s) - (\rho+\rho_s)^2 + \rho^2/2}{2(1-\rho-\rho_s)},$$

we get the range of ρ_s as follows:

$$\rho_s < \frac{1+\rho^2 - \sqrt{1-2\rho+5.5\rho^2-\rho^3+0.5\rho^4}}{1-\rho},$$

in which the benefit from flexibility outweighs the set-up time loss.

4 Concluding Remarks

In this paper, we tried to improve a stochstic processing line. We showed the amazing amount of improvement by introducing flexibility to a stochastic processing line. Introduction of flexibility gives us the benefit of resource pooling. But when we apply priority sequencing to the flexible processing line, we could achieve additional improvement. We also considered the set-up time loss due to flexibility and derived the appropriate range for flexibility and set-up time.

In addition to the set-up time loss, there is an important factor we should consider especially where each station or worker has its own objective. When we introduce the flexiblity into the system, the workers are asked to help each other. The benefit in reducing processing times comes from the resource pooling effect where one worker having slack time can help the other. But if we simply introduce the cellular system, we confront the critical free rider problem. Therefore we should introduce an appropriate compensation scheme which induces the mutual help among the workers. For the appropriate compensation scheme, readers are advised to refer to Nam(1996). We might have to combine both the individual worker's performance and the team performance altogether in order to induce the optimal assistance.

The result of this paper implies that we should consider the alternative of introducing flexibility to a processing system rather than simply trying to allocate each segment of jobs and get a balanced processing line. As shown throughout the paper, the improvement from flexibility is of a huge magnitude of scale. We can also relate our serial flexibility to other important operations areas. To reduce lead time or response time for product development, we may use concurrent engineering. Concurrent engineering relaxes the sequential constraint to shorten the timespan to complete a project. The underlying idea is similar in serial flexibility. In a serial flexible system, we relax the sequential service stage constraint by improving server flexibility, thus shortening throughput time.

References

[1] Burke, P.J., 1966. “The Output of a Queueing System,” Operations Research, 4, pp.699-704.

[2] El-Rayah, J., 1979. ”The Efficiency of Balanced and Unbalanced Production Line,” International Journal of Production Research, 17, pp. .

[3] Foshini, G.J. and Salz, J., 1978. “A Basic Dynamic Routing Problem and Diffusion,” in: Chandy, K.M. and Reiser, M. (Eds.), Computer Performance, North Holland Publishing Company, pp. 499-513.

[4] Ghosh, S. and R.J. Gagnon, 1987. “A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems,” International Journal of Production Research, 27, pp. 637-670.

[5] Harmon, R.L., 1992. Reinventing the Factory II, Free Press.

[6] Harrison, J.M. and Wein, L.M. 1989. “Scheduling Networks of Queues:Heavy Traffic Analysis of a Simple Open Network,” Queueing Systems 5, pp. 265-280.

[7] Hillier, F.S. and R.W. Boling, 1966. “The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times,” Journal of Industrial Engineering, 17, pp. 651-658.

[8] Hillier, F.S. and R.W. Boling, 1979. “On the Optimal Allocation of Work in Symmetrically Unbalanced Production Line Systems with Variable Operation Times,” Management Science, 25, pp. .

[9] Jackson, J.R., 1963. “Jobshop-Like Queueing Systems,” Management Science 10, pp. 131-142.

[10] Kelly, F.P., 1979. Reversibility and Stochastic Networks, John Wiley and Sons.

[11] Kleinrock, L., 1975. Queueing Systems, vol. I, John Wiley & Sons.

[12] Klimov, G.P., 1974. "Time Sharing Service Systems I," Theory of Probability Applications 19, pp. 532-551.

[13] Kottas, J.F. and H.S. Lau, 1981. "A Stochastic Line Balancing Procedure," International Journal of Production Research, 19, pp. 177-193.

[14] Leu, Y.Y, A.M. Lance, and P.R. Loren, 1994. "Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria," Decesion Sciences, 25, pp. 581-606.

[15] Loulou, R., 1974. "On the Extension of Congestion Theorems to Multi-Channel Systems," in: Clarke, A.B.(Ed.), Lecture Notes in Economics and Mathematical Systems, 98, pp. 185-198.

[16] Miltenburg, G.J. and J. Wijngaard, 1994. "The U-line Line Balancing Problem," Management Science, 40, pp. 1378-1388.

[17] Monden, G.J., 1993. Toyota Production System, Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, GA.

[18] Nam, I.H., 1992. Flexibility in Manufacturing: Dynamic Scheduling and Resource Pooling, Ph.D. Thesis, Graduate School of Business, Stanford Univ.

[19] Nam, I.H., 1996. "Compensation Scheme Resolving Marketing-Manufacturing Conflict," Working Paper, Seoul National Univ.

[20] Reiman, M.I., 1983. "Some Diffusion Approximations with State Space Collapse," Proceedings in International Seminar on Modeling and Performance Evaluation Methodology, Springer-Verlag, pp. 209-240.

[21] Smunt, T.L. and W.C. Perkins, 1985. "Stochastic Unpaced Line Design: Review and Further Experimental Results," Journal of Operations Management, 5, pp.351-373.

[22] Wein, L.M., 1990. "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Network With Controllable Inputs," Operations Research 38, pp. 1065-1078.

[23] Wein, L.M., 1991. "Brownian Networks With Discretionary Routing," Operations Research 39, pp. 322-340.