

Approximate Calculation of Order Fill Rate under Purchase Dependence

Changkyu Park[†]

College of Business, University of Ulsan

구매종속성을 고려한 주문충족률의 근사적 계산

박 창 규[†]

울산대학교 경영대학

For the competitive business environment under purchase dependence, this paper proposes a new approximate calculation of order fill rate which is a probability of satisfying a customer order immediately using the existing inventory. Purchase dependence is different to demand dependence. Purchase dependence treats the purchase behavior of customers, while demand dependence considers demand correlation between items, between regions, or over time. Purchase dependence can be observed in such areas as marketing, manufacturing systems, and distribution systems. Traditional computational methods have a difficulty of the curse of dimensionality for the large cases, when deriving the stationary joint distribution which is utilized to calculate the order fill rate. In order to escape the curse of dimensionality and protect the solution from diverging for the large cases, we develop a greedy iterative search algorithm based on the Gauss-Seidel method. We show that the greedy iterative search algorithm is a dependable algorithm to derive the stationary joint distribution of on-hand inventories in the retailer system by conducting a comparison analysis of a greedy iterative search algorithm with the simulation. In addition, we present some managerial insights such as : (1) The upper bound of order fill rate can be calculated by the one-item pure system, while the lower bound can be provided by the pure system that consists of all items; (2) As the degree of purchase dependence declines while other conditions remain same, it is observed that the difference between the lower and upper bounds reduces, the order fill rate increases, and the order fill rate gets closer to the upper bound.

Keywords : Approximate Calculation, Order Fill Rate, Purchase Dependence, Quasi-Birth-and-Death Process

1. Introduction

For companies in today's competitive marketplace, order fulfillment performance has become more and more imperative in order to respond quickly to customer needs [14, 20]. This paper considers the calculation of order fill rate when

a customer order is composed of several items and the customer order is met only if all the items can be deliverable at once. The order fill rate is defined as the probability of satisfying a customer order immediately using the existing inventory.

Song [16] stated the order fill rate as the order-based performance measure. However, most standard inventory models do not consider connections between items. The models assume that each item demand is independent. This case is referred as the item-based approach. In the order-based ap-

Received 10 October 2017; Finally Revised 8 November 2017;
Accepted 9 November 2017

[†] Corresponding Author : ckparkuou@ulsan.ac.kr

proach, the following question is important, because customer order requires several items at the same time: What is the probability that a customer order can be met immediately, given any safety stock level?

Park and Seo [13] faced the same question when analyzing the inventory of a spare parts distributor for ship engines and generators. After investigating the historical purchase-order-request (POR) data, Park and Seo [13] identified that a many of orders had been cancelled because of the lack of a few spare parts. That is, if an order contained at least one item that was out of stock, the whole order was cancelled, even though the other items had large quantities of stocks. They demonstrated that over 30% was explained by the portion of orders cancelled because of stocked-out parts. In most of cases, shipping companies purchase spare parts only when all of the spare parts are available immediately, because they want to buy all of the spare parts together. In this case, if one of the spare parts is out of stock, while all of the other parts are in stock, the situation is the same as if all items were out of stock. Park and Seo [13] called this type of dependence as ‘purchase dependence.’

Purchase dependence is different to demand dependence. Purchase dependence treats the purchase behavior of customers, while demand dependence considers demand correlation between items, between regions, or over time. Purchase dependence can be observed in such areas as marketing, manufacturing systems, and distribution systems.

This paper proposes a new approximate calculation of order fill rate under purchase dependence. We calculate the order fill rate using the stationary joint distribution of on-hand inventories. The stationary joint distribution is modelled by a quasi-birth-and-death (QBD) process and it consists of the balance equations and a normalization equation. However, traditional computational methods for solving the linear equations face with a difficulty of the curse of dimensionality for the large number of equations and unknowns. Thus we develop the greedy iterative search algorithm (GISA).

This paper is organized as follows. In the following section, we review the related literature. Section 3 explains the model that calculates the order fill rate. In Section 4, we depict the procedure of GISA. Section 5 conducts the computational analysis. In Section 6, we finalize this paper by providing our conclusions.

2. Literature Review

Firstly, Song [16] examined the order-based approach. She derived an exact solution of the order fill rate for a multi-item, multi-product system with backlogging and constant lead times. A customer order was composed of several items in different amounts. Using a series of convolution of one-dimensional distributions, she calculated the order fill rate.

On the other hand, Song, et al. [17] studied a multi-component, multi-product system, where each of the components is managed by the make-to-stock and refilled by the exponential processing machine. They formulated a generalized model having both complete backlogging and lost sales as a special case. In addition, they differentiated total order service and partial order service. The total order service is that an order is fulfilled completely or rejected as a whole, while the partial order service is that partial fulfilment occurs. Iravani et al. [4] and Gao et al. [1] examined a similar situation. However, they considered a little different perspective. Iravani et al. [4] considered customer flexibility. The customer flexibility means that customers change some of their requested components not in stock. On the other hand, Gao et al. [1] considered that each item was refilled by an independent unreliable machine.

One major limitation with the above studies is their inability to analyze large systems. Their modelling approaches are difficult to calculate, because they require incorporating correlations among demands for different items. Even though the matrix-geometric techniques are used by Song et al. [17], Iravani et al. [4] and Gao et al. [1], the curse of dimensionality is still remained.

In order to solve a linear system of equations : $A\mathbf{x} = \mathbf{b}$, $A = (a_{ij}) \in \mathbb{R}^{m \times m}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^m$, we usually use a well-known Gauss-Seidel method. The Gauss-Seidel method is modified from the Jacobi method. Thus it needs less iteration to make the same degree of accuracy. In order to speed-up the convergence rate, several preconditioned iterative methods have been developed. These studies are summarized in <Table 1>. However, our linear equations system is not solved by the Gauss-Seidel method and several preconditioned iterative methods, because these methods diverge.

3. Model Description

When purchase dependence exists, we consider a retailer

<Table 1> A Summary of Preconditioned Iterative Methods

Research	Preconditioner	Remark
Milaszewicz [8]	I + C	C eliminates the elements of the first column below the diagonal of A.
Gunawardena et al. [2]	I + S	S transforms the first upper codiagonal to zero.
Usui et al. [18]	I + U	U is a strictly upper triangular part of-A.
Kotakemori et al. [7]	I + βU	β ≥ 1
Kohno et al. [5]	I + αS	α > 1
Kotakemori et al. [6]	I + S _{max}	S _{max} is constructed by only the largest element at each row of the upper triangular part of A.
Hadjidimos et al. [3]	I + αC	
Morimoto et al. [9]	I + S + S _{max}	
Noutsos and Tzoumas [12]	Generalized preconditioner	Eliminate two or more off-diagonal elements in each row.
Niki et al. [11]	I + S + R	R is referred to as the nth preconditioner.
Zheng and Miao [21]	I + S _{max} + R	
	I + S _{max} + R _{max}	R _{max} is constructed by the largest element of R.

system that has J different items. $\Omega = \{1, 2, \dots, J\}$ is the set of all item indices. For any $K \subseteq \Omega$, let $|K|$ denote the number of elements in K . The overall order process is stationary in time and follows a Poisson process. Each customer requests at most one unit of each item but may require several items simultaneously. For any $K \subseteq \Omega$, an order is of type K if it requires one and only one unit of each item in K and 0 units in $\Omega - K$. Let $T = \{1, 2, \dots, |T|\}$ be the set of order types ($1 \leq |T| \leq 2^J - 1$). We assume that the probability of type K is constant. Every order type is independent each other. This means that the demand process for each item follows also a Poisson process. This paper uses the same assumptions and notations defined in Song et al. [17].

Orders are satisfied by a First-Come-First-Serve (FCFS) basis. When an order comes and a retailer has some items in out-of-stock, the order is cancelled. Song et al. [17] called this situation as a total order service. Each item is managed independently by an base stock policy. Let s_i be the base stock level for item i . This means that if the inventory position of item i is less than s_i , then order up to s_i at each demand epoch. The inventory position is calculated by the inventory on hand plus inventory on order. It is assumed that item i has an inventory at level s_i for all i at time 0. Then, if the item i is transported to the customer, each demand for item i generates an order for that item. Hence there can exist (at most) s_i outstanding orders of item i . Orders for item i are refilled by a supplier i and the lead times are distributed exponentially with the rate μ_i . The supplier processes the orders on a FCFS basis.

3.1 The Order Fill Rate

For any $i \in \Omega$ and $K \in T$, let λ be the overall order rate, q^K be the probability that an order is of type K , λ^K be the order rate of demand type K (i.e., $\lambda^K = q^K \lambda$), q_i be the probability that an order requires item i (i.e., $q_i = \sum_{K: i \in K} q^K$), and λ_i be the aggregate demand rate of item i (i.e., $\lambda_i = q_i \lambda$).

We denote that

I_i = the on-hand inventory of item i in steady-state, $0 \leq I_i \leq s_i$,

the fill rate of type- K order is denoted by

F^K = the joint probability that all items in a type- K order are delivered immediately
 $= P(I_i > 0; i \in K)$

Then, the order fill rate of a retailer system F is obtained by

$$F = \sum_{K \in T} q^K F^K \tag{1}$$

3.2 The Stationary Joint Distribution

Due to purchase dependence, a type- K order is cancelled if at least one item i ($i \in K$) is out of stock. Since the fill rate of type- K order is the joint probability that all items in the type- K order are delivered immediately, the fill rate of type- K order is decided by the joint distribution of $(I_i :$

$i \in K$). In this section, we demonstrate how to derive the joint distribution of $(I_i : i \in K)$.

A stochastic process $\{(I_i(t) : i \in K), t \geq 0\}$, is a continuous-time Markov chain with finite state space $\{\mathbf{n} = (n_i : i \in K) | 0 \leq n_i \leq s_i, i \in K\}$, where $I_i(t)$ is the inventory on hand of item i at time t . There are $\prod_{i \in K} (s_i + 1)$ states.

A state transition can only take place if a demand or an item on order arrives. In other word, with transition rate λ^K , the state \mathbf{n} transits to the state $\mathbf{n}' = (n'_i : i \in K)$, where

$$n'_j = \begin{cases} n_j - 1 & \text{if } j \in K \text{ and } 0 < n_j \text{ for all } i \in K \\ n_j & \text{otherwise} \end{cases} \quad (2)$$

and with transition rate μ_i ($i \in K$), the state \mathbf{n} transits to the state $\mathbf{n}' = (n'_i : i \in K)$, where

$$n'_j = \begin{cases} n_j + 1 & \text{if } j = i \text{ and } n_i < s_i \\ n_j & \text{otherwise} \end{cases} \quad (3)$$

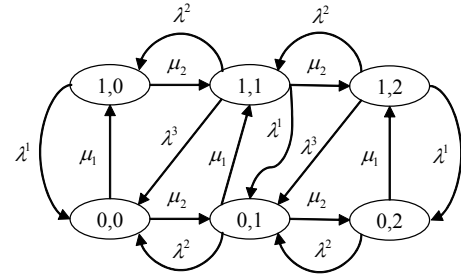
It is easy to see that this Markov chain is irreducible, so its stationary distribution exists uniquely [17]. The balance equation for each state can be formulated by utilizing equations (2) and (3). <Figure 1> shows the transition rate diagram and balance equations for the retailer system in which the type-1 order requires an item 1, the type-2 order requires an item 2, and the type-3 order requires both items 1 and 2. In the equilibrium situation, it should be that the rate of leaving any node equals to the rate of entering that node. Using the 'rate in = rate out' principle, we can make the following generalization.

Let \mathbf{I} be a vector of $(I_i : i \in K)$ and p_I be the equilibrium probabilities. Then

$$\left(\sum_{i \in K} \delta_i \mu_i + \sum_{K \in T} \xi_K \lambda^K \right) p_I = \sum_{i \in K} \mu_i p_{I - \mathbf{e}_i} + \sum_{K \in T} \lambda^K p_{I + \sum_{i \in K} \mathbf{e}_i} \quad (4)$$

where $\delta_i = 1$ for $I_i < s_i$ or 0 otherwise, $\xi_K = 1$ for $\forall I_i > 0$ ($i \in K$) or 0 otherwise, \mathbf{e}_i is a unit vector having 1 at the position of I_i , and $p_I = 0$ for any $I_i \notin [0, s_i]$.

Then, the stationary joint distribution can be obtained by solving the balance equations and a normalization equation. In order to solve the linear equations, we can use traditional computational algorithms such as Gaussian elimination. However, the direct approach can face with severe computation and space complexities for the large size of problem. Voiding the computation and space complexities, Song et al. [17],



$$\begin{aligned} (\mu_1 + \mu_2) p_{0,0} &= \lambda^1 p_{1,0} + \lambda^2 p_{0,1} + \lambda^3 p_{1,1} \\ (\mu_1 + \mu_2 + \lambda^2) p_{0,1} &= \mu_2 p_{0,0} + \lambda^1 p_{1,1} + \lambda^2 p_{0,2} + \lambda^3 p_{1,2} \\ (\mu_1 + \lambda^2) p_{0,2} &= \mu_2 p_{0,1} + \lambda^1 p_{1,2} \\ (\mu_2 + \lambda^1) p_{1,0} &= \mu_1 p_{0,0} + \lambda^2 p_{1,1} \\ (\mu_2 + \lambda^1 + \lambda^2 + \lambda^3) p_{1,1} &= \mu_1 p_{0,1} + \mu_2 p_{1,0} + \lambda^2 p_{1,2} \\ (\lambda^1 + \lambda^2 + \lambda^3) p_{1,2} &= \mu_1 p_{0,2} + \mu_2 p_{1,1} \end{aligned}$$

<Figure 1> The Transition Rate Diagram and Balance Equations

Iravani et al. [4] and Gao et al. [1] used the special structure of the QBD process. They obtained a unified, matrix-geometric solution of the stationary joint distribution. On the other hand, Ye and Li [19] developed the folding algorithm to conduct a steady state analysis of finite QBD process. The curse of dimensionality is still remained. Thus we develop the greedy iterative search algorithm to avoid the curse of dimensionality. The algorithm utilizes the special structure of the QBD process shown in equation (4) and stops the solution from diverging by adding the newly invented steps into the Gauss-Seidel method.

4. Greedy Iterative Search Algorithm

A linear equations system is established by the balance equations depicted by equation (4) and the normalization equation such that $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m \times m}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^m$. Here, we say that \mathbf{A} is the coefficient matrix, \mathbf{b} is the right-hand side vector, \mathbf{x} is the vector of unknowns (i.e., p_I , the equilibrium probabilities), and the order $m = \prod_{i \in K} (s_i + 1)$.

The linear equations system, sometimes, can be large and sparse. Then, a serious storage problem can be brought by the large number of equations and unknowns. Gaussian elimination is a very accurate, economical, and useful algorithm, if possible. But if the order m is so large and the results of Gaussian elimination cannot be stored, it would be better

Initial step
 set $k = 0$
 initialize $\mathbf{x}^{(0)}$
 $r_{\max} = \max_i [b - Ax^{(0)}]_i$
 $r_{\text{sum}} = \sum_i [b - Ax^{(0)}]_i$
 repeat until convergence
 $k = k + 1$
 for i from 1 until m do

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k-1)} \right)$$

$$r_{\max}^{\text{new}} = \max_j [b - Ax^{(k)}]_j$$

$$r_{\text{sum}}^{\text{new}} = \sum_j [b - Ax^{(k)}]_j$$

if ($x_i^{(k)} \geq 0$ and ($r_{\max}^{\text{new}} \leq r_{\max}$ or $r_{\text{sum}}^{\text{new}} \leq r_{\text{sum}}$)) then
 update $r_{\max} = r_{\max}^{\text{new}}$ and $r_{\text{sum}} = r_{\text{sum}}^{\text{new}}$
 else
 restore $x_i^{(k)} = x_i^{(k-1)}$
 end if
 end (i-loop)
 check if convergence is reached
 end (repeat)

<Figure 2> The Pseudo-Code of GISA

to solve the linear equations system by other ways that never change the matrix A and never require storing more than a few vectors of length m . For this purpose, iterative methods are recommended.

In this section, we describe the procedure of GISA. We begin with an initial solution vector $\mathbf{x}^{(0)}$, and generate a sequence of solution vectors $\mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)} \rightarrow \dots$ according to the algorithm. We hope that as $k \rightarrow \infty$, $\mathbf{x}^{(k)}$ will converge to the exact solution. <Figure 2> shows the pseudo-code of GISA. In the initialization step, an initial solution $\mathbf{x}^{(0)}$ set up and we compute the maximum absolute residual r_{\max} and the summation of absolute residual r_{sum} corresponding to $\mathbf{x}^{(0)}$. Let $[y]_i$ be the i th element of vector y .

Then we create the solution vectors $\mathbf{x}^{(k)}$ according to the following rule. Let $x_i^{(k)}$ denote the i th element of the k th iterate solution vector $\mathbf{x}^{(k)}$ (i.e., $[x^{(k)}]_i$) and b_i the i th element of the right-hand side \mathbf{b} (i.e., $[b]_i$). The algorithm adjusts the i th element of the current solution, in the order $i = 1, 2, \dots, m$, to abolish the i th element of the residual. The solution is updated immediately if the newly computed element $x_i^{(k)}$ is nonnegative, and the newly computed maximum abso-

lute residual or the newly computed summation of absolute residual does not increase. These steps are invented in this paper to prevent the solution from diverging.

The algorithm continues to create solution vectors $\mathbf{x}^{(k)}$ until a convergence is achieved. A convergence is reached when the iteration k becomes to the maximum number of iterations; the summation of absolute residual r_{sum} is less than the maximum value of tolerance; or no longer change is made in the solution vector $\mathbf{x}^{(k)}$. The solution vectors $\mathbf{x}^{(k)}$ would be affected by an initial solution $\mathbf{x}^{(0)}$ as the order m becomes so large. Section 5.2 describes the impacts of initial solution.

5. Computational Analysis

In this section, we illustrate numerical and simulation experiments that are conducted to demonstrate the GISA performance, examine the influences of an initial solution and a relaxation method, and obtain the managerial insights.

5.1 The Performance of GISA

The GISA was developed to derive the stationary joint distribution of on-hand inventories in the retailer system and calculate the order fill rate of retailer system. The order fill rate is calculated by the probability that a customer order can be met immediately. We measure the performance of GISA by the order fill rate of retailer system.

For the purpose of testing the GISA performance, the test bed was designed as follows. Combining the number of items and the number of order types, four cases are chosen. Case 1 is that $J = 3$ with $|T| = 7$. Case 2 is that $J = 3$ with $|T| = 7$ (asymmetric). Case 3 is that $J = 3$ with $|T| = 4$. Case 4 is that $J = 5$ with $|T| = 6$. All possible order types occur in cases 1 and 2, and thus $|T| = 7$. Type 1 - 3 requires each item, respectively, type 4 - 6 requires combination of two items, respectively, and type 7 requires all items. We set that $q^1 = q^2 = q^3 = 0.05$, $q^4 = q^5 = q^6 = 0.07$, and $q^7 = 0.64$ for case 1, and $q^1 = 0.04$, $q^2 = 0.05$, $q^3 = 0.06$, $q^4 = 0.08$, $q^5 = 0.06$, $q^6 = 0.07$, and $q^7 = 0.64$ for case 2. Cases 3 and 4 consider order types for individual items and one order type for all items, so $|T| = J + 1$. We set that $q^1 = q^2 = q^3 = q^4 = 0.05$ and $q^5 = 0.85$ for case 3, and $q^1 = q^2 = q^3 = q^4 = q^5 = 0.05$ and $q^6 = 0.75$ for case 4. We treat order types symmetric for the order rate, however, we dealt with asymmetric order types for case 2.

<Table 2> Comparison of GISA with the Gaussian Elimination and Computer Simulation

Case	s_i	Gaussian	GISA		Simulation	Absolute difference	
		F	F	r_{max}	r_{sum}		F
1	5	0.8476	0.8476	0.0000	0.0000	0.8448	0.0028
	10	-	0.9474	0.0000	0.0000	0.9483	0.0009
	15	-	0.9794	0.0001	0.0005	0.9796	0.0003
2	5	0.8475	0.8474	0.0000	0.0001	0.8462	0.0012
	10	-	0.9471	0.0001	0.0008	0.9468	0.0003
	15	-	0.9789	0.0001	0.0008	0.9790	0.0002
3	5	0.8098	0.8098	0.0000	0.0000	0.8142	0.0044
	10	-	0.9155	0.0001	0.0006	0.9221	0.0066
	15	-	0.9556	0.0000	0.0005	0.9616	0.0060
4	5	-	0.8407	0.0001	0.0006	0.8477	0.0070
	10	-	0.9478	0.0001	0.0005	0.9526	0.0048
	15	-	0.9852	0.0000	0.0003	0.9838	0.0014

Within all retailer systems, equal base stock levels were chosen, i.e., $s_1 = s_2 = \dots = s_J$. For these base stock levels, we assigned three values of 5, 10, and 15. We set that all replenishment rates μ_i and the overall order rate for the retailer system λ are equal to 1.0.

For the purpose of comparison, we obtained the exact order fill rate by utilizing the Gaussian elimination. However, the Gaussian elimination was not suitable for the large cases requiring the large matrix manipulation. In this case, a simulation program was developed to calculate the order fill rate. We implemented the simulation run for 5 times per each case. For each time, the simulation was implemented for 10,000 customer orders with the warm-up of 1,000 orders. Then the order fill rate was averaged the 5 order fill rates resulted from simulations.

The results of this comparison are presented in <Table 2>. For small cases, the order fill rates of retailer system calculated by the algorithm are almost same as those by the Gaussian elimination. This demonstrates the GISA performance. We can notice the fact by just reviewing that r_{max} and r_{sum} are near to zeros. For large cases, even though r_{max} and r_{sum} demonstrate the performance of GISA, <Table 2> shows the comparison between the algorithm and computer simulation. The average value of absolute difference is 0.0030. Based on the results in <Table 2>, the GISA can be considered as a reliable algorithm to obtain the stationary joint distribution of on-hand inventories in the retailer system.

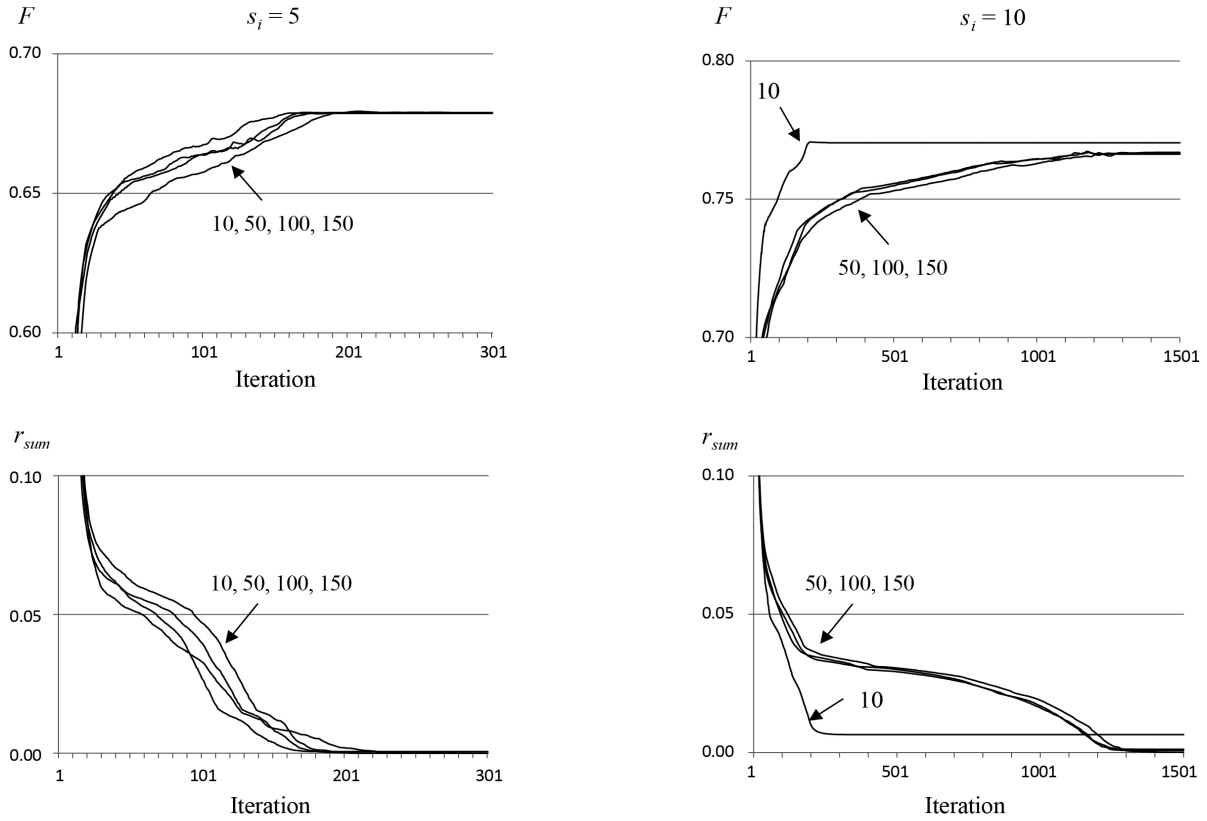
5.2 The Impacts of an Initial Solution and a Relation Method

The initial solutions are distinguished by assigning a

different number of elements having a positive value in the initial solution vector. That is, we allocated a positive value to the only controlled elements in the initial solution vector, but zeros to the rest. We calculated an equal positive value as dividing 1.0 by the number of positive elements.

In order to investigate the effect of initial solutions on the convergent speed of the GISA, we computed the order fill rates of case 4 in Section 5.1 as increasing the number of positive elements in the initial solution vector (e.g., 10, 50, 100, and 150). For the two cases of $s_i = 5$ and $s_i = 10$, <Figure 3> illustrates the convergent speeds. (This section sets that $\lambda = 1.5$ and $\mu_i = 1.0$ ($i \in K$)). From <Figure 3>, we can observe that, in general, there is no significant difference in the convergent speed depending on the initial solutions. (In the case of $s_i = 10$, even though the initial solution having 10 positive values reaches to the convergence faster than others, the accuracy is very poor.) However, it is observed that the more number of the base stock level requires the more iteration to reach to the convergence.

Meanwhile, we can recognize that the solution vectors will converge to a certain solution, because the GISA algorithm provides the steps of stopping the solution from diverging. However, it is observed that for the large cases, the GISA algorithm could converge to different solutions depending on the initial solutions, which is shown in <Table 3>. <Table 3> shows the order fill rates of the case of $s_i = 10$ in <Figure 3>. Among the 4 cases, the order fill rate of the initial solution having 50 positive values results in the highest degree of accuracy, i.e., a minimum value of r_{sum} .



<Figure 3> The Convergent Speeds Depending on Initial Solutions

<Table 3> The Fill Rates Generated by the GISA with Different Initial Solutions

Initial solution	F	r_{max}	r_{sum}
10	0.7704	0.0007	0.0065
50	0.7670	0.0000	0.0001
100	0.7665	0.0002	0.0008
150	0.7663	0.0003	0.0012

It is well-known that the convergence of Gauss-Seidel method may sometimes be improved by combining with relaxation [10, 15]. The relaxation method modifies the Gauss-Seidel procedure in <Figure 2>, as given by

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) \quad (5)$$

to

$$x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) \quad (6)$$

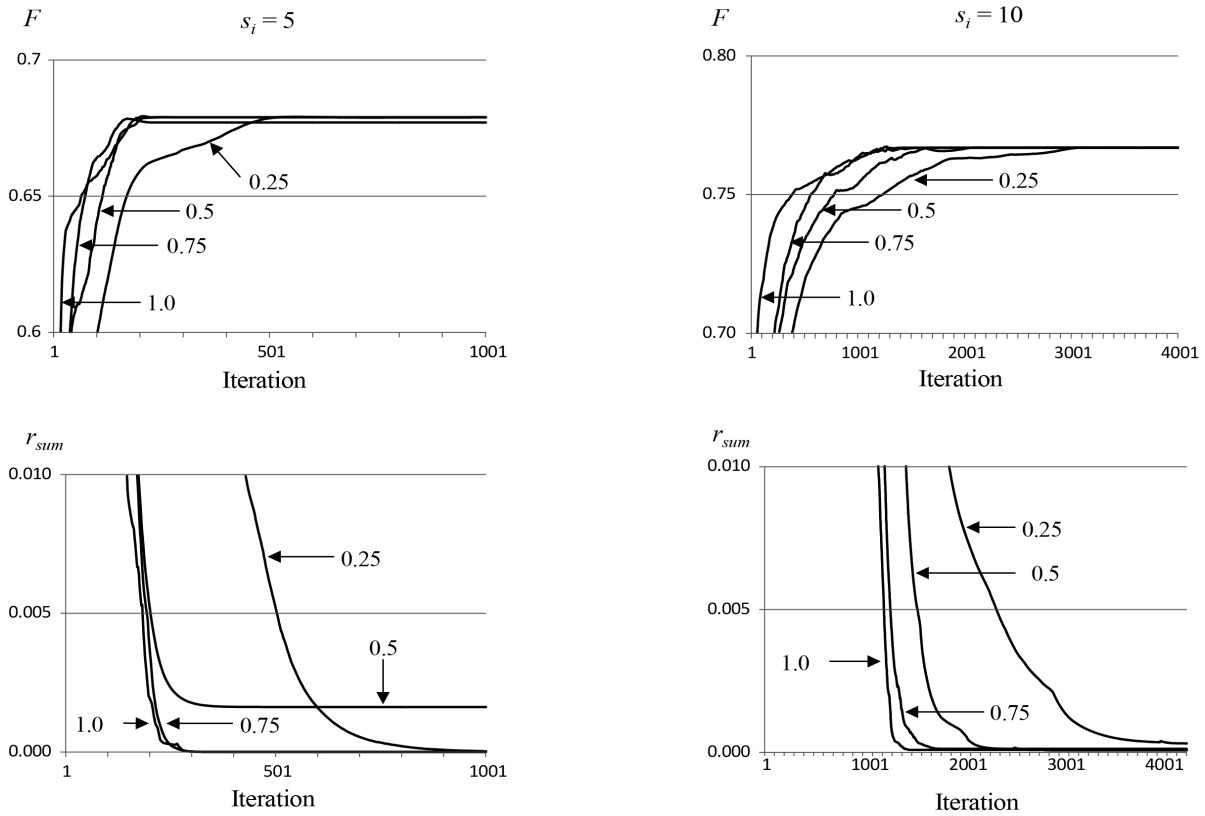
For choices of ω , the procedure is called as either an under-relaxation method with $0 < \omega < 1$ or an over-relaxation method with $1 < \omega$.

Like to the examination on the convergent speed of the GISA algorithm, we computed the order fill rates of case 4 for various values of ω . For the two cases of $s_i = 5$ and $s_i = 10$, <Figure 4> illustrates the convergent speeds. (This section sets that $\lambda = 1.5$ and $\mu_i = 1.0$ ($i \in K$.) We show only the results of under-relaxation methods because the results of over-relaxation methods are too poor. From <Figure 4>, we can observe that the relaxation methods do not improve the convergence speed of the GISA.

5.3 Analysis of Purchase Dependence

This paper calculates the order fill rate of a retailer system considering purchase dependence. Purchase dependence implies that the order is cancelled when an order comes and a retailer has some items in out-of-stock. In this section, we analyze the order fill rate as changing the degree of purchase dependence, thus deduce the managerial insights.

For analysis purpose, we utilized the case 1 in Section



<Figure 4> The Convergent Speeds of Under-Relaxation Method

<Table 4> Case 1 Values of dp and Associated q^K

K	Item			dp	q^K								
	1	2	3		0	0.25	0.325	0.475	0.55	0.745	0.85	0.925	1
1	1	0	0		0.33	0.20	0.15	0.10	0.10	0.05	0.00	0.00	0.00
2	0	1	0		0.33	0.20	0.15	0.10	0.10	0.05	0.00	0.00	0.00
3	0	0	1		0.33	0.20	0.15	0.10	0.10	0.05	0.00	0.00	0.00
4	1	1	0		0.00	0.10	0.15	0.15	0.10	0.07	0.10	0.05	0.00
5	1	0	1		0.00	0.10	0.15	0.15	0.10	0.07	0.10	0.05	0.00
6	0	1	1		0.00	0.10	0.15	0.15	0.10	0.07	0.10	0.05	0.00
7	1	1	1		0.00	0.10	0.10	0.25	0.40	0.64	0.70	0.85	1.00

5.1 as the test bed. In case 1, all base stock levels were assigned equally, i.e., $s_1 = s_2 = s_3 = 5$. We set that all replenishment rates μ_i and the overall order rate for the retailer system λ are equal to 1.0.

The degree of purchase dependence (dp) is defined as

$$dp = \sum_{K \in T} \frac{|K|-1}{J-1} q^K \quad (7)$$

The degree of purchase dependence $dp = 1$ if all orders are the one order type for the set of all items. Inversely, if all orders are different order types for individual items only, $dp = 0$. The dp assesses how much demands are con-

nected each other. <Table 4> shows the values of dp and associated q^K for case 1.

In order to investigate the attributes of order fill rate, we decomposed the order fill rate into the fill rate of a pure system. The pure system has one order type K . (We called the system as the type- K pure system.) The fill rate of type- K pure system is obtained by decomposing customer orders into item demands, in other words, by building new item demand processes. The new demand processes follow the same marginal distributions as the original order processes. For the type- K pure system, the overall order rate is approximated by averaging the demand rates of items in the type- K order. Let

$$\lambda^{pK} = \text{the overall order rate for the type-}K \text{ pure system}$$

$$= \frac{1}{|K|} \sum_{i \in K} \lambda_i.$$

Then, the fill rate of type- K pure system is calculated by F^{pK} = the joint probability of filling immediately all items in the type- K pure system

$$= P(I_i > 0 : i \in K)$$

We calculate the fill rates of type- K pure systems using the GISA after decomposing customer orders into item demands.

<Table 5> and <Figure 5> show the fill rates of both type- K pure systems and the retailer system obtained by the GISA. We can recognize some managerial insights. First, the value of order fill rate poses between values of tow pure systems. One pure system consists of all items. The other consists of only one item. That is, these two fill rates are the lower and upper bounds for the order fill rate. The lower bound can be calculated by the fill rate of pure system having all items. The upper bound can be calculated by the one-item pure system.

<Table 5> Fill Rates of Type- K Pure and Whole Systems

dp	F^{p7}	$F^{p4} = F^{p5} = F^{p6}$	$F^{p1} = F^{p2} = F^{p3}$	F
0.000	0.9930	0.9950	0.9973	0.9973
0.250	0.9644	0.9729	0.9841	0.9781
0.325	0.9498	0.9613	0.9767	0.9658
0.475	0.9130	0.9328	0.9561	0.9318
0.550	0.8915	0.9147	0.9429	0.9130
0.745	0.8297	0.8569	0.9005	0.8476
0.850	0.7950	0.8246	0.8740	0.8046
0.925	0.7703	0.8011	0.8540	0.7757
1.000	0.7460	0.7776	0.83333	0.7460

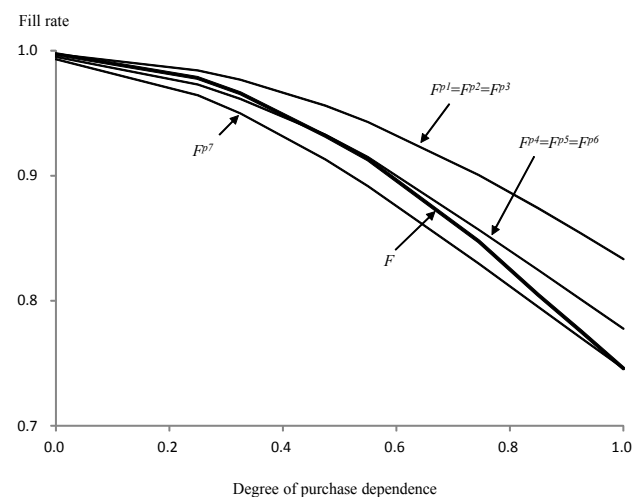
Secondly, as the degree of purchase dependence declines while other conditions remain same, it is observed that the difference between the lower and upper bounds reduces, the order fill rate increases, and the order fill rate gets closer to the upper bound. From these results, it can be inferable that if we deal with purchase dependent item demands independently, we would face either unnecessary overstocking of some items or unsatisfactory service at the order level. For better profit and lower inventory costs, treating knowledge of purchase dependence is important in designing an inventory replenishment policy.

6. Conclusion

For the retailer system having purchase dependence, we developed a new approximate calculation of the order fill rate. The order fill rate is the probability of meeting an customer order immediately from existing inventory. During the derivation of the stationary joint distribution, traditional computational methods face with the curse of dimensionality for the large problems.

This paper developed the GISA algorithm to elude the curse of dimensionality and keep the solution from diverging. The GISA algorithm is grounded on the Gauss-Seidel method. From the comparison analysis of the GISA and the simulation, this paper illustrated that the GISA is a reliable algorithm to gain the stationary joint distribution of on-hand inventories in the retailer system.

In addition, we observed some managerial insights. (1) The upper bound of order fill rate can be calculated by the one-item pure system, while the lower bound can be provided by the pure system that consists of all items. (2) As the degree of purchase dependence declines while other conditions remain same, it is observed that the difference between the lower and upper bounds reduces, the order fill rate increases, and the order fill rate gets closer to the upper bound.



<Figure 5> Fill Rates Against the dp

As the degree of purchase dependence declines, the order fill rate gets closer to the upper bound created by the pure system. It is wondered when treating purchase dependent demands as independent is tolerable. We leave this theme as the future studies.

Acknowledgement

This work was supported by the 2016 Research Fund of University of Ulsan.

References

- [1] Gao, C., Shen, H., and Cheng, T.C.E., Order-fulfillment performance analysis of an assemble-to-order system with unreliable machines, *International Journal of Production Economics*, 2010, Vol. 126, No. 2, pp. 341-349.
- [2] Gunawardena, A.D., Jain, S.K., and Snyder, L., Modified iterative methods for consistent linear systems, *Linear Algebra and its Applications*, 1991, Vol. 154-156, pp. 123-143.
- [3] Hadjidimos, A., Noutsos, D., and Tzoumas, M., More on modifications and improvements of classical iterative schemes for M-matrices, *Linear Algebra and its Applications*, 2003, Vol. 364, pp. 253-279.
- [4] Iravani, S., Luangkesorn, L., and Simchi-Levi, D., On assemble-to-order systems with flexible customers, *IIE Transactions*, 2003, Vol. 35, No. 5, pp. 389-403.
- [5] Kohno, T., Kotakemori, H., and Niki, H., Improving the modified Gauss-Seidel method for Z-matrices, *Linear Algebra and its Application*, 1997, Vol. 267, pp. 113-123.
- [6] Kotakemori, H., Harada, K., Morimoto, M., and Niki, H., A comparison theorem for the iterative method with the preconditioner (I+Smax), *Journal of Computational and Applied Mathematics*, 2002, Vol. 145, No. 2, pp. 373-378.
- [7] Kotakemori, H., Niki, H., and Okamoto, N., Accelerated iterative method for Z-matrices, *Journal of Computational and Applied Mathematics*, 1996, Vol. 75, No. 1, pp. 87-97.
- [8] Milaszewicz, J.P., Improving Jacobi and Gauss-Seidel iterations, *Linear Algebra and its Applications*, 1987, Vol. 93, pp. 161-170.
- [9] Morimoto, M., Harada, K., Sakakihara, M., and Sawami, H., The Gauss-Seidel iterative method with the preconditioning matrix (I+S+Sm), *Japan Journal of Industrial and Applied Mathematics*, 2004, Vol. 21, No. 1, pp. 25-34.
- [10] Niethammer, W., The successive order relaxation method (SOR) and Markov chains, *Annals of Operations Research*, 2001, Vol. 103, No. 1, pp. 351-358.
- [11] Niki, H., Kohno, T., and Morimoto, M., The preconditioned Gauss-Seidel method faster than the SOR method, *Journal of Computational and Applied Mathematics*, 2008, Vol. 219, No. 1, pp. 59-71.
- [12] Noutsos, D. and Tzoumas, M., On optimal improvements of classical iterative schemes for Z-matrices, *Journal of Computational and Applied Mathematics*, 2006, Vol. 188, No. 1, pp. 89-106.
- [13] Park, C. and Seo, J., Consideration of purchase dependence in inventory management, *Computers & Industrial Engineering*, 2013, Vol. 66, No. 2, pp. 274-285.
- [14] Park, K.J., Effect of customer demand on total inventory cost and order fill rate in a supply chain, *Journal of Society of Korea Industrial and Systems Engineering*, 2009, Vol. 32, No. 3, pp. 93-98.
- [15] Philippe, B., Saad, Y., and Stewart, W.J., Numerical methods in Markov chain modeling, *Operations Research*, 1992, Vol. 40, No. 6, pp. 1156-1179.
- [16] Song, J.S., On the order fill rate in a multi-item, base-stock inventory system, *Operations Research*, 1998, Vol. 46, No. 6, pp. 831-845.
- [17] Song, J.S., Xu, S., and Liu, B., Order-fulfillment performance measures in an assemble-to-order system with stochastic lead times, *Operations Research*, 1999, Vol. 47, No. 1, pp. 131-149.
- [18] Usui, M., Niki, H., and Kohno, T., Adaptive Gauss-Seidel method for linear systems, *International Journal of Computer Mathematics*, 1994, Vol. 51, No. 1-2, pp. 119-125.
- [19] Ye, J. and Li, S., Folding algorithm : a computational method for finite QBD processes with level-dependent transitions, *IEEE Transactions on Communications*, 1994, Vol. 42, No. 234, pp. 625-639.
- [20] Yoon, S.C., A study on inventory control method for an item with stock keeping units, *Journal of Society of Korea Industrial and Systems Engineering*, 2015, Vol. 38, No. 1, pp. 124-130.
- [21] Zheng, B. and Miao, S.X., Two new modified Gauss-Seidel methods for linear system with M-matrices, *Journal of Computational and Applied Mathematics*, 2009, Vol. 233, No. 4, pp. 922-930.

ORCID

Changkyu Park | <http://orcid.org/0000-0002-8250-9470>