47 Journal of the Korea Management Engineers Society
Vol. 21 No. 4, 2016. 12. (pp. 47-59)

M/G/c/c 대기행렬 모형을 이용한 대학 도서관 보유 장서 최적화

Application of M/G/c/c queueing models to optimize book circulation process in university library

김 우 성* Kim Woo-sung
최 혜 봉** Choi Hye-bong
홍 신*** Hong, Shin

Abstract

Queueing models have been used to model library management system for about 40 years. While the analytic solutions of the models provide intuition to understand various performance measures of the library system, it is difficult to employ the queueing models to optimization problems due to its complexities. In this paper, we studies the optimization problems in library management by employing M/G/c/c queueing models. To estimate the probability that arrivals find all books unavailable(loss probability), Erlang loss formula in M/G/c/c queue is used. The mathematical model developed here is to decide number of books to buy to ensure a given service quality. Since Erlang loss formula is a nonlinear function of system loading and number of server, the problem belongs to the domain of nonlinear optimization problem. To find an optimal solution, a dynamic programming of the 0-1 Knapsack problem is employed.

Keywords : Library optimization, Nonlinear optimization, Queueing application, Dynamic programming

* 한동대학교 경영경제학부 교수, E-mail : wskim@handong.edu
** 한동대학교 ICT창업학부 교수, E-mail : hbchoi@handong.edu
*** 교신저자, 한동대학교 전산전자공학부 교수, E-mail : hongshin@handong.edu

투고일 2016.07.29 수정일 2016.11.26 게재일 2016.12.31

1. 서 론

Morse, P. M.(1968)이 저서에서 대기행렬 모형을 통하여 도서관의 도서 대출 모형을 제시하여 분석한 이래로, 다양한 대기행렬 모형들이 약 40여 년 동안 도서관 시스템을 분석하기 위하여 사용되어 왔다. 대기행렬 모형에 관한 분석들을 통하여 얻어진 성능척도에 관한 해석적인 해들(Analytical solution)과 근사해법(Approximations)은 도서관 시스템의 성능척도에 관한 유용한 정보를 제공하지만, 이러한 정보들을 통하여 도서관의 실질적인 운영 전략을 도출해 내기는 쉽지 않다. Warwick, J. (2009)는 대기행렬 모형의 분석 결과들이 갖게 수리적인 복잡성 때문에 대기행렬 이론들의 결과들을 실제 도서관 운영에 적용하는 것이 쉽지 않다고 언급하며, 실제 도서관 운영에 도움이 되는 수리 모형의 개발의 중요성을 언급한다. 이러한 이유로 본 논문에서는 도서관 시스템을 모델링할 때 자주 사용되는 M/G/c/c 대기행렬 모형을 도서관 운용에서 발생하는 최적화 문제에 적용하고 이를 위한 수리 모형을 개발, 분석한다.

대기행렬 모형은 Morse, P. M.(1968)이 처음 사용자의 도서 대출/반납 프로세스(book circulation process)를 분석하기 위하여 사용된 이래로 오랜 시간동안 도서관의 성능척도를 분석하는 데에 사용되어 왔다. Morse, P. M.(1968)에서는 고객의 대출 요청을 고객의 도착으로, 대출 기간을 서버의 서비스 시간으로 모델링 하여 대기행렬 모형을 분석하였다. 대출하려는 책이 없을 때 고객의 대응에 따라 예약이 가능하여 전통적인 무한용량의 대기행렬 모형으로 분석 가능한 경우, 예약이 불가능하여 고객이 시스템을 이탈하는 경우, 고객 중의 일부만 예약을 신청하는 세 가지로 나누어 모형을 분석하였다. 이 때 예약이 불가능한 경우는 M/G/c/c 모형으로 모델링이 가능하며, 책 종류에 따른 보유 장서 수의 증가에 따라 고객의 만족도를 제시하였다. Morse, P. M.(1968)이 제시한 모형들은 이후에 Chen, C. C. (1976)와 Beheshti, J., and Tague, J. M. (1984)에 의하여 실제 도서관 데이터를 통하여 검증되었다. 이 후, 도서관 시스템을 분석하기 위하여 다양한 대기행렬 모형이 개발되었다. Rouse, W. B. (1976)는 도서관 시스템을 분석하기 위한 대기행렬 네트워크 모형을 개발 분석하였다. 이후, 도서관 전산 시스템이 도입되면서 Rouse, W. B., and Rouse, S. H. (1977)는 전산 시스템을 고려한 도서관 네트워크의 성능척도를 분석하여 결과를 제시하였다. Warwick, D. J. (1994)는 한 고객이 대여할 수 있는 책의 수가 제약되어 있을 경우를 가정한 닫힌 대기행렬 네트워크 모형을 개발, 분석하였다. 도서관을 분석하기 위하여 개발된 다양한 대기행렬 모형들에 관한 결과들은 Warwick, J. (2009)을 참고하길 바란다. Morse, P. M.(1968)을 시작으로 많은 모형들이 개발되었지만, Warwick, J. (2009)가 언급했듯, 대기행렬 모형을 실제 사례에 응용하여 도서관의 의사결정에 도움을 줄 수 있는 최적화 문제로 접근한 결과는 많지 않다. 실제로 논문에서는 대기행렬 모형을 도서관 시스템에 적용하는 연구가 1990년대 이전에 활발했던 것에 비해 이후에 줄어들게 된 원인들을 언급하고 있는데, 그 중 주요한 원인들을 대기행렬 모형의 수리적 복잡성과 함께 시대의 흐름에 따른 도서관 시스템의 역할변화로 언급하고 있다. 일반적으로 실제에 가까운 사람의 행태를 대기행렬 모형에 반영하기 힘들며, 많은 요소들을 반영할수록 분석을 통하여 해석적인 해를 얻기는 힘들어 지는 것이 사실이다. 많은 1990년대 이전의 연구들의 대기행렬 네트워크 모형들을 살펴보면, 도서관 시스템의 특정한 프로세스를 대기행렬 모형을 이용하여 분석하

는 것에 치중하고 있다. 이렇게 개별적인 프로세스들을 구체적으로 모델링하게 될 경우 모형을 통하여 성능척도에 관한 해를 얻을 수 있지만, 그 집합인 통합적인 시스템의 경우에는 대기행렬 모형의 상태 공간이 지수적으로(exponentially) 증가하고 모형의 성능척도가 비선형 함수의 형태이기 때문에 당시의 컴퓨터를 통하여 분석하기가 쉽지 않았다. Rouse, W. B., and Rouse, S. H. (1977)에서는 도서관 간의 통합된 시스템을 대기행렬 네트워크를 통하여 분석하였지만, 이 경우의 대기행렬 모형의 파라미터(parameter)들은 개별 프로세스를 고려한 것이 아닌 한 도서관 자체에 관한 파라미터 들이었다. 이 경우 파라미터들을 정확히 추정해 내는 것이 문제가 된다. 이렇듯, 1990년대 이전의 논문들은 대기행렬 모형을 사용하여 개별적인 프로세스를 분석한 경우, 통합된 시스템의 최적화 문제로 이어지지 않았고, 통합된 시스템을 대기행렬의 서버로 가정할 경우에는 파라미터들을 도출해 내기가 쉽지 않았다. 예를 들어, 많은 기존의 연구에서 고객의 대출 요청을 대기행렬에서의 고객의 도착으로 보고 대출 기간을 서비스 기간으로 가정하여 도서 대출 프로세스를 분석한다(보유 장서 수를 서버 수로 볼 수 있다.). 이 경우 개별적인 장서의 사용 효율(utilization)이나 평균 대기 시간(mean waiting time)등을 대기행렬 모형을 통하여 도출할 수 있지만, 기존의 연구들에서는 이러한 결과들을 통해 도서관 전체의 성능척도를 도출하는 데에는 한계가 있었다. 이러한 대기행렬 모형의 성능척도를 고려한 최적화 문제는 일반적으로 비선형계획법 문제에 해당하며, IT 기술이 발전하면서 현재에 이르러 이러한 문제의 해를 찾을 수 있게 된 것이다. 그렇지만 Ackere, A. et al(2006)에서 대기행렬 모형을 시뮬레이션을 통하여 분석한 것을 제외하면 도서관 최적화 문제에 대기행렬 모형을 적용한 사례를 찾기 쉽지 않은데, Warwick, J. (2009)와 Wells, A. (2007)에서는 IT기술이 발전하였지만, 이러한 기술의 발전으로 인해 도서관의 역할과 수요에 변화가 생겼기 때문이라고 언급하고 있다. 인터넷이나 컴퓨터의 발전으로 인해 정량화된 분석이 용이해진 것은 사실이지만, 그와 동시에 전자책 서비스와 같은 다양한 시스템이 개발되고 도서관이 다양한 사람들의 학문적 수요를 충족시킬 수 있게 되면서 도서관에 관한 현재 연구들이 정량적인 최적화 기법이나 수리모형의 분석을 통한 도서관 시스템의 개선보다는 고객의 다양한 수요를 만족시키기 위한 시스템 구축에 초점을 맞추고 진행되고 있는 것이다(김정수, 이철규, and 강민형 (2013)). 그럼에도 대기행렬 모형에 기반을 둔 비선형 최적화 문제는 제조, 운송 시스템 등 다른 분야에서는 활발하게 나타나고 있다(Osorio, C., and Chong, L. (2015), Smith, J. M. (2015)). 따라서 현대의 도서관 시스템 또한 이러한 기법을 통하여 분석, 개선될 수 있으며 본 연구의 동기는 이러한 연구의 부재에 기인한다.

본 연구에서는 도서관에서 고객의 대출 요청이 있을 때, 모든 도서가 대출 중일 확률(손실 확률)을 고려하여 도서관의 적절한 보유 장서 수를 결정하는 문제를 비선형 최적화 기법을 통하여 분석한다. 제시되는 수리 모형은 고정된 예산이 주어졌을 때, 장서 마다 일정한 손실 확률을 보장하도록 적절한 도서 보유 계획을 결정하는 문제인데 제약식과 목적함수 식을 적절히 바꿈으로써 일정한 손실확률을 보장하며 비용을 최소화하는 문제로도 분석이 가능하다. 위에 언급한 손실확률은 M/G/c/c 대기행렬 모형에서 제시된 얼랑 손실 함수에 의해 계산될 수 있는데 얼랑 손실 함수는 비선형 함수이기 때문에 본 연구에서 다루는 최적화 문제는 비선형 최적화 문제에 해당한다. 일반적인 선형계획법과 같은 기법을 통하여 분석할

수 없기 때문에 본 논문에서는 개발된 수리모형의 최적해를 찾기 위해 수리 모형을 제약조건을 가지는 0-1 Knapsack 문제로 치환하여, 동적 계획법(dynamic programming)에 기반한 알고리즘을 통하여 해를 찾는다. 본 논문에서는 포항에 소재한 한동대학교 도서관의 6년간의 대출 데이터를 사용하여 모형을 도출, 적용하고 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 M/G/c/c 대기행렬 모형의 결과들을 제시하고 관련된 변수를 제시한다. M/G/c/c 대기행렬의 안정상태 확률에 관한 결과를 통하여 현재 대학 도서관의 장서 별 손실 확률을 구할 수 있다. 3장에서는 본 연구에서 개발된 수리 모형을 제시하고 분석하기 위하여 개발한 알고리즘을 소개한다. 본 논문에서 제시하는 수리 모형의 최적해는 동적계획법을 통하여 찾을 수 있으며, 이를 통하여 손실확률을 최소화하기 위한 도서관의 적절한 장서 보유수를 결정할 수 있다. 4장에서는 개발된 모형을 한동대학교 도서관 데이터에 적용하여, 장서의 최적 수와 그 때의 도서관의 손실확률에 관한 결과를 제시한다. 또한 예산 제약 하에서, 도서관의 도서 구매 계획을 수립하기 위하여 유용한 정보를 제공한다. 5장에서는 본 논문의 의의와 결과를 정리한다.

2. M/G/c/c 대기행렬 모형의 응용

본 장에서는 M/G/c/c 대기행렬 모형을 통하여 도서관의 대출 프로세스를 모델링한 결과를 소개하고 현재 대학 도서관 시스템의 대출 데이터를 사용하여 손실 확률을 제시한다. 이미 기존에 언급한 많은 문헌들에서 M/G/c/c 대기행렬 모형을 사용되었기 때문에 본 장에서 제시하는 결과들은 새로운 결과가 아니며 기존의 결과를 실제 데이터에 적용한 것이다.

N가지의 서로 다른 책을 보유하고 있는 도서관을 가정하자. 각각의 도서를 나타내는 인덱스를 i로 정의하고(i=1, ... , N) 도서 i를 n_i권 씩 보유하고 있다고 가정한다. 이러한 도서관에 고객의 도착에 따라 도서 i에 대한 대출 요청이 발생하며 발생 과정은 도착율이 λ_i인 포아송 과정을 따른다고 가정한다. n_i권 중 대출 가능한 책이 있으면 대출은 즉시 진행되며, 모든 도서 i가 대출 상태이면 고객은 도서관을 이탈한다고 가정한다. 고객이 도서 i를 대출 했을 경우, 대출 기간(대출에서 반납까지 걸리는 기간)은 평균이 $1/\mu_i$인 일반분포를 따른다. 이 경우 도서 i의 대출 과정은 도착율이 λ_i, 평균 서비스 시간이 $1/\mu_i$, 서버가 n_i대인 M/G/c/c 대기행렬로 모델링 가능하다. M/G/c/c 대기행렬 모형의 경우, 서버의 수가 대기공간의 용량과 동일하기 때문에 모든 서버가 서비스 중일 경우 고객은 시스템에 들어오지 못하고 이탈하게 되며, 이는 도서관 시스템에서 고객이 대출 요청을 했을 때 모든 책이 이미 대여 중일 경우 고객이 도서관을 이탈하는 사건을 나타낸다. 이러한 사건이 발생할 확률은 얼랑 손실 함수(Erlang loss function)로 표현되는데 도서 i을 n_i권 보유하고 있을 때 손실 함수($P_{loss}(i,n_i)$)는 아래와 같다.

$$P_{loss}(i,n_i) = \frac{(\lambda_i/\mu_i)^{n_i}/n_i!}{\sum_{k=0}^{n_i}(\lambda_i/\mu_i)^k/k!}. \qquad (1)$$

위 식에서 알 수 있듯, 얼랑 손실 함수는 대출 기간의 분포에는 영향을 받지 않으며, 평균 대여 기간에만 영향을 받는다.

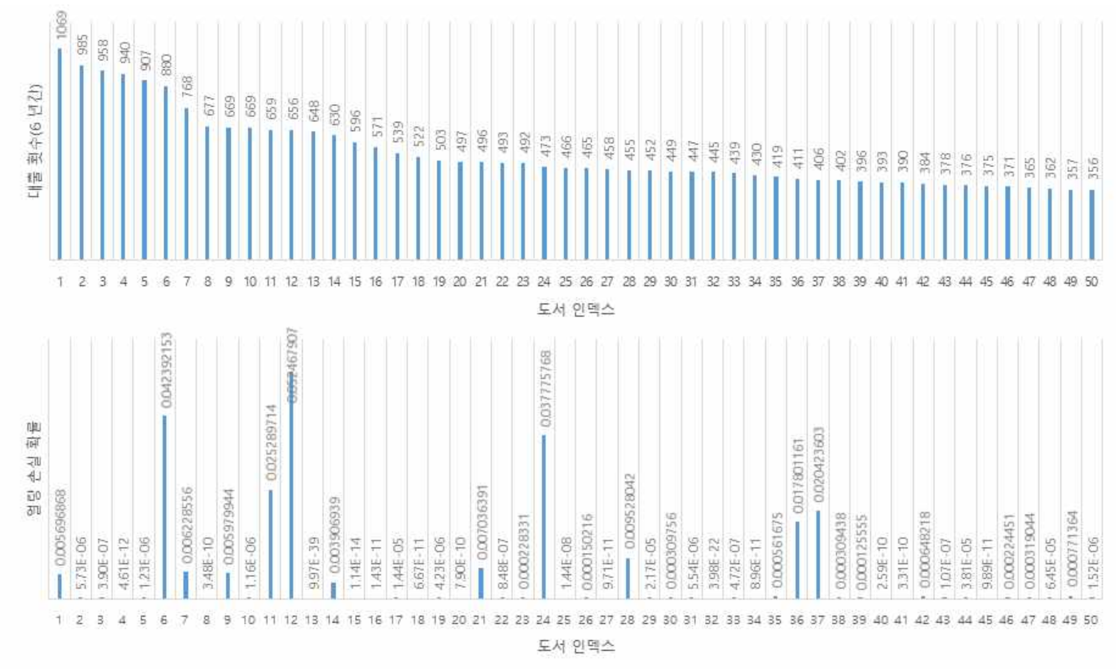
얼랑 손실 함수를 통하여 현재 도서관에서 손실이 발생하는 사건의 발생 빈도를 추정할 수 있다. 표 1은 6년간 한동대학교 도서관의 대출 기록을 이용하여 가장 대출 횟수가 많았던 책 10권에 대한 데이터를 나타낸 것이다. 대출 횟수가 많았던

< 표 1 : 도서 대여 기록(2010~2015) >

도서 인덱스	대출 횟수 (6년간)	장서 수	평균 대출 기간 (일)
1	1069	13	12.4471
2	985	18	11.4268
3	958	19	10.6545
4	940	26	10.7574
5	907	18	11.0375
6	880	8	10.8202
7	768	10	11.7161
8	677	21	11.8493
9	669	9	11.2616
10	669	15	10.4634

순서대로 도서 인덱스를 부여하였다. 위 데이터로부터 손실함수를 구하기 위한 파라미터들을 얻을 수 있다. 단위시간을 1일로 가정하면 λ_i은 대출횟수를 (365*6)으로 나눈 값이다. 예를 들어, λ_1은 0.4881회/일로 나타난다. 평균 대출 기간이 $1/\mu_i$가 되며, 서버의 수를 나타내는 n_i는 장서 수에 해당된다. 위 데이터를 통하여 가장 대출 횟수가 많았던 도서 50권에 대한 대출 횟수와 손실함수 값이 그림 1에 나타나 있다. 그림에서 보듯 손실 함수의 값은 대출 횟수나 대여기간과 함께 장서 수의 영향을 받는데, 대여 횟수나 대여기간은 고객의 수요에 해당하기 때문에 도서관의 입장에서 일정한 손실 함수 이하를 유지하기 위해 결정할 수 있는 것은 장서 수이다. 이는 표 1과 그림 1에도 나타나게 되는데, 표에 있는 10권의 도서 중 가장 큰 손실함수의 값을 갖는 도서는 6번 도서에 해당하며 이는 수요에 비하여 적은 장서 수를 보유하고 있기 때문이다.

이 때, 도서관의 손실 확률은 다음과 같이 나타나게 된다.

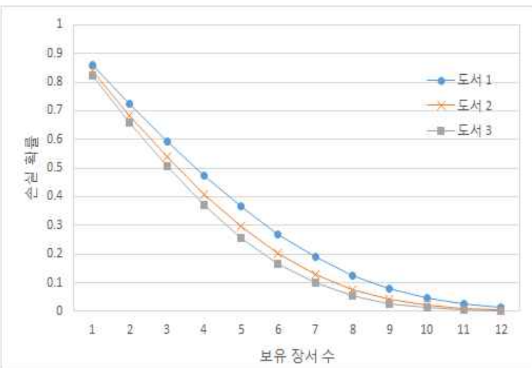


<그림 1 : 대출 횟수가 가장 많은 도서 50권의 대출 횟수 및 손실 함수 >

$$\sum_{i=1}^{N} P[t=i]P[loss|t=i] \qquad (2)$$
$$= \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_s} \frac{(\lambda_i/\mu_i)^{n_i}/n_i!}{\sum_{k=0}^{n_i} (\lambda_i/\mu_i)^k/k!}.$$

λ_s는 $\sum_{k=1}^{N} \lambda_k$로 도착율의 총합이며, $P[t=i]$는 임의의 고객 대출 요청이 발생했을 때 요청이 도서 i 에 관한 대출 요청일 확률을 나타낸다. loss는 대여 요청한 도서가 모두 이미 대여중인 사건을 나타낸다. $P[t=i]$는 포아송 과정의 분해 특성(decomposition property)에 의해 쉽게 λ_i/λ_s로 계산될 수 있다. 위 식으로 얻어진 확률 값은 도서관에 대출 요청이 도착했을 때 해당 도서가 모두 대출 중이라 대출 요청이 실패할 확률을 나타낸다. 도서관의 관점에서 적절한 서비스를 제공하기 위해서는 일정한 손실 확률을 만족시키기 위하여 적절한 장서 수를 보유해야 한다. 본 연구의 목적은 각각 도서를 구매하거나 손실 함수를 고려한 적정 보유 도서 수를 결정하는 수리 모형을 제시하는 것이다.

도서 1, 2, 3에 대하여 장서 수의 증가에 따른 손실 확률을 그래프로 나타내 보면 그림 2와 같다. 식 (1)에서 볼 수 있듯, 얼랑 손실 함수는 선형 함수가 아님을 참고하자. 따라서 본 연구에서



< 그림 2 : 보유 장서 수에 따른 손실확률 >

다루는 문제는 비선형 최적화 문제에 속하게 된다. 도서관의 적정 장서 보유 수를 결정하기 위한 수리 모형과 분석 알고리즘은 다음 장에서 제시한다.

3. 보유 도서 장서 수 결정 문제

본 장에서는 도서관이 적절한 장서 수를 보유하기 위한 수리 모형을 소개한다. 먼저 변수들을 정의하고 수리 모형을 제시한 후, 수리 모형을 분석하기 위한 동적 계획법을 통한 알고리즘에 대하여 소개한다.

3.1 변수 정의 및 수리 모형

먼저, 수리 모형에서 사용되는 변수는 각각 다음과 같이 정의된다. 이전 장에서 정의한 변수들은 생략한다.

x_i : 도서 i 의 적정 보유 수, $i = 1, 2, \cdots, N$.

c_i : 현재 보유 하고 있는 도서 i의 장서 수, $i = 1, 2, \cdots, N$.

b_i : 도서 i의 가격, $i = 1, 2, \ldots , N$.

B : 총 가용한 예산.

S : 총 가용한 공간(보유할 수 있는 최대 책의 수).

r_i : 장서 i 의 최소 보장 확률(장서 i의 손실 확률을 보장하기 위한 기준 값).

손실 확률을 최소화하기 위한 수리 모형은 다음과 같이 제시된다.

$$Minimize \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_s} \frac{(\lambda_i/\mu_i)^{x_i}/x_i!}{\sum_{k=0}^{x_i} (\lambda_i/\mu_i)^k/k!}, \quad (3)$$

$$s.t. \ \sum_{i=1}^{N} x_i \le S, \quad (4)$$

$$\sum_{i=1}^{N} b_i(x_i - c_i) \le B, \quad (5)$$

$$x_i \ge c_i, \ \forall i \quad (6)$$

$$\frac{\frac{(\lambda_i/\mu_i)^{x_i}}{x_i!}}{\sum_{k=0}^{x_i} \frac{(\lambda_i/\mu_i)^k}{k!}} \le r_i, \ \forall i, \quad (7)$$

s.t는 ‘subject to’ 의 약자이며 비음 제약식 (non-negativity condition)은 식 (6)으로 인하여 생략 할 수 있다. 이 때, 목표는 식 (4)-(8)을 만족시키며 목적함수 식 (3)을 최소화 시키는 x_i값들을 도출하는 것이다. 식 (3)은 이전 장에서 언급했다시피 임의의 대출 요청이 실패할 확률을 나타낸다. 식 (4)는 공간 제약 때문에 일정 수 이상의 도서를 보유할 수 없음을 나타내는 제약식이며, 식 (5)는 예산 제약식을 나타낸다. 식 (6)은 이미 보유하고 있는 장서 수를 고려하기 위한 것이며 식 (7)은 각각의 장서에 대해서 최소 손실 확률을 보장하기 위한 것이다. 위 모형의 다양한 변형이 있을 수 있다. 만일 새로운 도서관을 디자인 하게 될 경우에는 모든 c_i값은 0으로 설정한다. 만일 수리 모형의 목표가 손실 확률의 최소화가 아닌 예산을 최소화하는 것이라면, 목적함수 (3) 대신 함수 (5)의 좌변의 식을 목적함수로 쓸 수 있다.

3.2 최적해 계산

본 장에서는 앞서 제시된 수리 모형을 분석하는 알고리즘을 제시한다. 3.1절에서 제시한 목적 함수 (3)의 최적해를 구하는 문제는 제약 조건을 가지는 0-1 Knapsack 문제로 치환될 수 있으며, 동적 계획법(dynamic programming)에 기반한 알고리즘을 통해 해를 찾을 수 있다(이운식 and 김병수(2011), 서원철 and 이운식(2014)). 이러한 최적해를 구하는 알고리즘의 계산 복잡도(computational complexity)는 서로 다른 서적의 수(N)에 대해 선형(linear)이다. 본 절에서는 주어진 제약식들 (4), (5), (6), (7)을 만족하며 목적 함수 (3)의 값을 최소화하는 문제가 0-1 Knapsack 문제로 어떻게 치환되는지를 세 단계의 과정에 걸쳐 설명한 후 (3.2.1절), 치환된 문제를 해결하는 알고리즘을 제시하고 분석한다(3.2.2절).

3.2.1 최적화 문제를 0-1 Knapsack 문제로 치환

0-1 Knapsack 문제: 0-1 Knapsack 문제는 특정 비용 값과 특정 효용 값이 정의된 N 개의 아이템들과 가용한 자원의 양이 주어졌을 때, 비용의 합이 가용한 자원의 양을 넘지 않으면서 효용의 합이 최대가 되게 하는 아이템들의 조합을 찾는 문제다(Cormen, T. H.,, Leiserson, C. E, Rivest R. L. and Stein, C. (2003)). 즉, 각 아이템의 비용과 효용을 p_i와 e_i이고 가용한 자원의 양을 B 라고 할 때, 0-1 Knapsack은 0 혹은 1 값으로 p_i의 선택 유무를 표현하는 벡터 x 중 식 (9)를 만족하며 식 (8)의 값을 최대화하는 해를 찾는 문제이다.

$$Maximize \ \sum_{i=1}^{N} x_i e_i \quad (8)$$

$$s.t. \ \sum_{i=1}^{N} x_i p_i \le \ B \quad (9)$$

보유 도서 장서 수 결정 문제의 치환: 앞서 정의한, 세 가지 종류의 제약 사항을 모두 만족하며 손실 확률을 최소화하는 것을 목적으로 하는 보유 도서 장수 결정 문제는 다음의 세 과정을 통해서 0-1 Knapsack 문제로 치환할 수 있다.

- 과정1: 도서 i가 최소 보장 확률(r_i)을 따르도록 장서를 최소한 추가로 구입하는 과정.
- 과정2: 최적화의 목적 함수(식 (3))를 장서 추가 구입에 따른 기대 효용의 합을 최대화하는 형태로 치환하는 과정.
- 과정3: 도서의 장서를 한 권씩 추가하는 선택을 아이템으로 정하고, 각 아이템의 기대 효용의 계산하여, 최적화 문제를 0-1 Knapsack 문제로 치환하는 과정.

과정1－3을 통하여 보유 도서 장수 결정 문제는 각 도서가 특정 장서 수를 확보 할지 말지를 결정하는 문제로 치환된다. 과정1－3의 구체적인 내용은 다음과 같다.

과정1. 최소 보장 확률에 따른 장서 구입: 식 (7)의 제약 조건을 만족하는 해를 구하기 위해, 먼저 각 도서 i에 대하여 아래의 식 (10)을 만족하는 최소의 x_i 인 m_i를 구한다.

$$\frac{\dfrac{(\lambda_i/\mu_i)^{x_i}}{x_i!}}{\sum_{k=0}^{x_i}\dfrac{(\lambda_i/\mu_i)^k}{k!}} \le r_i \qquad (10)$$

이 때 도서 i의 현재 장서 수 c_i가 m_i보다 작을 경우에, 최소한 $m_i - c_i$ 만큼의 장서를 추가로 구매해야 식 (7)의 제약 조건을 만족할 수 있으므로, 과정1을 이후에는 도서 i를 $m_i - c_i$ 만큼 추가로 구입한 상황을 가정하기로 하자(즉, $x_i \ge m_i$).

이때 최대 손실 확률 보장을 위한 장서 추가 이후 공간 소요 S_r은 식 (11)과 같고, 총 가용 공간 S가 S_r보다 작을 경우, 문제의 해를 정의할 수 없다. 이와 마찬가지로, 최대 손실 확률 보장을 위해 추가로 장서를 구입하는데 소요되는 비용 B_r은 아래의 식 (12)와 같고, 만약 식 B_r이 예산 B 보다 큰 경우에는 보유 도서 장서 수 결정 문제의 값을 정의할 수 없다.

$$\sum_{m_i > c_i | i} m_i - c_i \qquad (11)$$

$$\sum_{m_i > c_i | i} b_i\,(m_i - c_i) \qquad (12)$$

장서의 추가 구입으로 인해 과정 1 이후 가용 예산은 $B' = B - B_r$으로, 가용 공간은 $S' = S - S_r$이 된다.

과정2. 목적함수 형태 변환: 과정1에서 설명한 바와 같이 최대 손실 확률 보장을 위한 추가 장서 구매를 수행한 이후, 각 도서 i의 장서 수를 c'_i라고 하자.

과정1 이후 도서 i를 a_i권 추가로 구입한다면(즉, $x_i = c'_i + a_i$), 도서 i의 손실 확률은 아래의 식 (13)만큼 감소하게 되는데, 그 값을 도서 i를 a_i만큼 추가 구입하는데 따른 ‘손실 확률 감소 효용’으로 볼 수 있다. 이 효용에 따라 공간은 a_i만큼이 소요되며, 비용은 $b_i\,a_i$가 소요된다.

$$\frac{\lambda_i}{\lambda_s}\frac{(\lambda_i/\mu_i)^{c'_i}/c'_i!}{\sum_{k=0}^{c'_i}(\lambda_i/\mu_i)^k/k!} - \frac{\lambda_i}{\lambda_s}\frac{(\lambda_i/\mu_i)^{a_i}/a_i!}{\sum_{k=0}^{a_i}(\lambda_i/\mu_i)^k/k!} \qquad (13)$$

논의의 편의를 위하여, 도서 i를 a_i만큼 추가 구입하는데 따른 식 (13)의 값을 $e_i(a_i)$로 표기하자.

도서관 보유 장서 수 문제, 즉 식 (3)의 최적화 문제는 앞서 정의한 손실 확률 감소 효용의 관점에서는 아래의 식 (14)－(16)과 같이 손실 확률 감소 효용의 최대화 문제로 표현할 수 있다.

$$Maximize \sum_{i=1}^{N} e_i(a_i), \qquad (14)$$

$$s.t. \sum_{i=1}^{N} a_i \le S', \qquad (15)$$

$$\sum_{i=1}^{N} b_i\,a_i \le B' \qquad (16)$$

즉, 도서관 보유 장서 수 결정 문제는 식 (15)와

식(16)의 제약 조건을 모두 만족하며 식 (14)의 값을 최대화하는 도서별 추가 장서 구입량 a를 찾는 문제로 치환된다.

과정3. 최적 조합을 찾는 문제로 변환: 과정 2에서 선언한 최적해 a에서 도서 i에 대한 a_i은 [0, $\min(\lfloor \frac{B'}{b_i} \rfloor, S-S_r)$] 중 한 값을 가진다. 편의를 위해, 이 구간의 최대값을 l_i라고 부르자.

도서 i에 대한 $d_{i,j}$ ($1 \le j \le l_i$)는 '도서 i를 $j-1$권 구입하기로 결정한 상황에서 1권을 추가로 더 구입할지에 대한 결정'으로 정하자 ($d_{i,j} \in \{0,1\}$). 이 때 $j<k$이며 $d_{i,j}=0$ 이고 $d_{i,k}=1$인 경우는 허용하지 않는다(식(20) 참고).

이러한 설정에서 도서 i에 대한 a_i의 값을 정하는 문제는 곧 $d_{i,1}, d_{i,2}, \ldots, d_{i,l_i}$의 값을 정하는 문제로 볼 수 있다. 예를 들어, l_i가 5인 도서 i의 a_i를 3으로 정한 결정은 $d_j=(1,1,1,0,0)$ 혹은 $d_{i,1}=1$, $d_{i,2}=1$, $d_{i,3}=1$, $d_{i,4}=0$, $d_{i,5}=0$으로 표현할 수 있다.

도서 i를 $j-1$권 보유한 상황에서 1권을 추가하여 j권 보유하기로 정하는 결정, 즉 $d_{i,j}$의 값을 1로 정하는 결정을 효용은 $e_i(j)-e_i(j-1)$이며 이 값을 $e'_{i,j}$로 부르기로 하자. 이 결정에 따른 공간 소요는 1이며 비용 소요는 b_i로 j의 값과 상관없이 같은 도서 i에 대해 일정하다.

앞서 기술한 일련의 과정을 거쳐서, 식 (3)－(7)에 기술된 보유 도서 결정 문제는 아래의 식 (17)－(20)과 같은 이진 판단의 문제, 즉 식(18)－(20)의 제약 조건이 있는 상황에서 효용을 최대화(식(17))하도록 1의 값을 가질 $d_{i,j}$의 조합을 정하는 문제로 치환된다.

$$Maximize \sum_{i=1}^{N}\sum_{j=1}^{l_i} d_{i,j}\, e'_{i,j}, \tag{17}$$

$$s.t. \sum_{i=1}^{N}\sum_{j=1}^{l_i} d_{i,j} \le S', \tag{18}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{l_i} b_i\, d_{i,j} \le B', \tag{19}$$

$$\forall i.\ \forall j.\ \forall k.\ (j<k<l_i) \rightarrow d_{i,j} \ge d_{i,k} \tag{20}$$

3.2.2 최적해 계산 알고리즘

식 (17)－(20)에서 정의한 보유 장서 수 결정 문제는 동적계획법 알고리즘을 통하여 도서 수, 가용 공간, 가용 예산에 대해 다항식 수준(polynomial time)의 복잡도를 가지는 계산을 통해 최적해를 구할 수 있다.

동적계획법을 이용한 최적화 문제 해결: 일반적으로 동적계획법은 어떤 제약조건 $r<R$ 하의 최적해를 제약조건 $r<R'$ $(R'<R)$ 하의 최적해들을 이용해 재귀함수 형태로 정의한 후, 재귀식을 점진적으로 계산하는 알고리즘을 통해 해를 구한다. 예를 들어, 전형적인 0-1 Knapsack 문제, 즉 가용 자원 P일 때 N개의 아이템 중 일부를 택하여 최대 효용을 얻는 문제에 대한 동적계획법은 가용 자원이 P보다 작은 상황에서 구한 최적해들과 $N-1$개의 아이템을 이용하는 상황에서 구한 최적해들을 이용해 가용 자원 P와 N개 아이템에 대한 최적해를 재귀적으로 정의한다.

알고리즘 설계: 보유 장서 수 결정 문제를 동적계획법으로 해결하기 최적해를 다음 재귀함수로 표현하도록 한다:

$$w[i,j,s,p] = \begin{cases} \max(w[i,j-1,s,p], \\ \quad w[i,j-1,s-1,p-b_i]+e'_{i,j}) & \text{if } 1<j \\ \max(w[i-1,l_{i-1},s,p], \\ \quad w[i-1,l_{i-1},s-1,p-b_i]+e'_{i,j} & \text{if } 1<i \wedge j=1 \\ e'_{i,j} & \text{if } i=1 \wedge j=1 \end{cases} \tag{21}$$

식 (21)에서 $w[i,j,s,p]$는 도서 i의 j번째 추가 장서 구입을 고려한 상황에서 가용 공간을 최대 s만큼, 가용 자원을 최대 p만큼 사용하여 얻을 수 있는 최대 효용(최대 손실 확률 감소)을 뜻한다. 이 때 i, j, s, p 중 하나가 0 혹은 음수 일

경우, $w[i, j, s, p]$의 값은 0이다. 모두 양수인 i, s, p에 대하여, $w[i, j, s, p]$는 다음 둘 중 큰 값으로 결정된다:

- 도서 i의 j번째 장서 추가 구입을 고려하기 전 상황의 에서 가용 공간 s와 가용 예산 p에서 최대 효용.
- 도서 i의 j번째 장서 추가 구입(공간 비용 1과 예산 비용 b_i 소요, 효용 e_i 추가)을 선택했을 때의 최대 효용

이 때 도서 i의 j번째 장서 추가 구입을 고려하기 전 상황은 i와 j의 값에 따라서 도서 i의 j-1번째 장서 추가 구입까지만 고려한 상황($1 < j$), 혹은 도서 i-1의 l_i번째 장서 추가 구입을 고려한 상황 ($1 < i \wedge j = 1$) , 혹은 0이 된다($i = 1 \wedge j = 1$). 최종적으로 알고리즘의 실행이 끝 난 후, 가용 공간 B'과 가용 예산 S'를 이용하여 얻을 수 있는 최대 효용은 $w[N, l_N, S', B']$의 값이 된다.

알고리즘 1은 식 (21)의 재귀함수의 값을 점진적으로 구하는 알고리즘이다. 알고리즘1은 앞서 정의한 도서의 수 N, 각 도서별 소요 예산 b_i, 각 도서별 최대 구입가능 장서 수 l_i, 장서별 추가 구입에 따른 효용 $e'_{i,j}$, 가용 공간 S', 가용 예산 B'을 입력으로 받은 후, 달성 가능한 최대 효용과 최대 효용을 만드는 선택을 반환한다. 전체적으로 봤을 때, 알고리즘 1은 3－19행, 4－18행의 두 반복문에서 각 도서의 추가 구매 선택을 차례로 탐색하며, 각 선택에서는 7－15행, 8－14행의 두 반복문을 통해 각각의 가용 자원 경우(공간 s, 비용 p)에서 최대 효용($w[s, p]$)을 식 (21)에 따라 계산한다(9－13행). 이외에 알고리즘 1의 구문에 대한 설명은 다음과 같다:

- 1행은 S'개의 행과 B'개의 열을 가지는 2차원 정수 배열 w를 선언하고, 각 원소를 0으로 초기화한다. 궁극적으로, $w[s,p]$의 값은 가용 공간 s, 가용 예산 p인 상황에서 최대 효용을 나타낸다.
- 2행은 S'개의 행과 B'개의 열을 가지는 2차원 배열 x를 선언하고, 각 원소를 공집합으로 초기화 한다. 집합 $x[s, p]$는 $w[s,p]$ 값을 가지게 하는 선택의 집합이다. 따라서 집합 $x[s, p]$의 각 원소는 정수 쌍 (i, j)가 되며, $(i, j) \in x[s, p]$인 경우, 최적해에 도서 i의 j번째 추가 구매를 수행한 결정이 반영됐음을 뜻한다.
- 5행에서는 도서 i의 j번째 추가 구매 선택을 정의하기 위해 임시로 w값을 저장할 배열 w'을 선언하고 초기화한다. 이 시점에서 배열 w에서는 직전 도서 추가 구입을 반영한 상황에서 최적해가 저장돼 있다. w'에 대한 모든 계산이 끝난 후(16행), w의 값을 w'으로 갱신한다.
- 6행에서는 $w'[s,p]$ 값을 결정하는 도서 구매 선택의 집합을 임시로 저장할 배열 x'을 선언하고 초기화한다. w'의 모든 값이 결정된 후, x'는 x 값으로 갱신된다(17행).
- 19행에서 알고리즘1은 최종적 결과로 $w[S', B']$값과 이때의 도서 추가 구매 결정의 집합인 $x[S', P']$을 반환한다.

```
Input
N : 도서의 수
b[1..N] : 각 도서별 소요 예산(가격)
l [1..N] : 각 도서별 최대 구입가능 수
e′ [1..N, max(l)] : 장서 추가 구입 효용 e′_{i,j}
S′ : 가용 공간
B′ : 가용 예산

 1 w[1..S′, 1..B′] = 0
 2 x[1..S′, 1..B′] = ∅
 3 for i from 1 to N do
 4 |   for j from 1 to l[i] do
 5 |   |   w′[1..S′, 1..B′] = 0
 6 |   |   x′[1..S′, 1..B′] = ∅
 7 |   |   for s from 1 to S′ do
 8 |   |   |   for p from 1 to B′ do
 9 |   |   |   |   w′[s, p] ← w[s, p]
10 |   |   |   |   if p > b[i] ∧
   |   |   |   |     w′[s, p] < w[s −1, p −b[i]] + e′[i, j] then
11 |   |   |   |     | w′[s, p] ← w[s −1, p −b[i]] + e′[i, j]
12 |   |   |   |     | x′[s, p] ← x[s −1, p −b[i]] ∪ {(i, j)}
13 |   |   |   |   end if
14 |   |   |   end for
15 |   |   end for
16 |   |   w = w′
17 |   |   x = x′
18 |   end for
19 end for
20 return w[S′, B′], x[S′, B′]
```

< 알고리즘 1 : 최대 손실 확률 감소 계산 >

알고리즘 분석1. 제약 조건의 만족: 알고리즘 1

이 구하는 해는 식 (18)−(20)의 제약 조건을 만족하는 동시에 가능한 모든 해 중 최대 효용을 가지는 해(식 (17))이다. 식 (17)−(19)에서 기술한 제약조건은 일반적인 0-1 Knapack 문제의 동적 프로그래밍 알고리즘이 만족하는 성질로, 알고리즘1 역시 만족함을 자명하게 알 수 있다. 반면, 식 (20)의 경우, 0-1 Knapsack 문제를 해결하는 일반적인 동적 프로그래밍 알고리즘은 고려하지 않는 특이한 제약조건이다. 본 절에서는 알고리즘 1이 식 (20)을 만족하는 해를 구함을 귀류법(Proof by Contradiction)을 이용해 간단히 논하도록 하겠다.

식 (20)은 도서 i의 j번째 장서 추가 구매가 해에 포함되는 경우, 반드시 도서 i의 첫 번째 장서 구매부터 $j-1$번째 장서 구매 역시 해에 포함 되어야 한다는 조건이다. 알고리즘 1이 식 (20)을 만족하지 않는 어떤 해 x, 즉 어떤 i, j, k에 대하여 $(i, j) \in x$ 이며 $(i, k) \notin x$ $(1 \le k < j)$인 x를 가진다고 가정하자. 이 가정은 곧 $w'[s, p] < w[s-1, p-b[i]] + e'[i, j]$을 만족하면서 동시에 $(i, k) \notin x[s\text{-}1, p\text{-}b[i]]$을 만족하는 어떤 s와 p가 존재함을 의미한다. 하지만, 앞선 두 조건들을 동시에 만족하는 해는 존재할 수가 없는데, 이는 식 (13)의 특성으로 인해 $e'_{i,k} > e'_{i,j}$이 언제나 성립하고, 따라서, $x[s-1, p-b[i]] + e'[i, k] > w[s-1, p-b[i]] + e'[i, j]$가 만족되기 때문이다. 이러한 모순으로 인하여, 앞서 설정한 가정이 잘못되었음을 확인할 수 있으며, 고로 알고리즘 1이 구한 최적 해는 식 (20)을 항상 만족함을 알 수 있다.

알고리즘 분석2. 계산 복잡도: 보유 장서 수 문제의 해는, 가용 예산 B'이 각 도서 i의 가격 b_i보다 충분히 크며, 가용 공간이 각 도서의 최대 보유 가능 수 l_i보다 충분히 큰 상황에서 최대 $\prod_{i=1}^{N} l_i = \prod_{i=1}^{N} \lfloor \frac{B'}{b_i} \rfloor$ 개수의 서로 다른 해를 가질 수 있다. 따라서 모든 가능한 해를 순전히 검토하는 알고리즘은 $O(B^N)$의 계산 복잡도를 갖는다.

동일한 상황에서 알고리즘 1의 경우, 3−19행 반복문의 반복 횟수는 N번, 4−18행 반복문의 반복 횟수는 최대 가용 예산 B에 비례, 7−15행의 반복문의 반복 횟수는 가용 공간 S에 비례, 마지막으로 8−14행 반복문의 반복 횟수는 가용 예산 B에 비례하므로, 알고리즘의 계산 복잡도는 $\Theta(NSB^2)$이다. 알고리즘 1은 도서 종류 수 N과 가용 공간 S 에 선형이며 가용 예산 B의 제곱에 비례한 계산복잡도로 최적 해를 구한다. 문제의 총 해의 수가 도서 수 N과 가용 예산 B 에 대하여 지수적임을 고려할 때, 제안한 알고리즘 1은 효율적으로 최적 해를 찾는 방법으로 볼 수 있다.

4. 사례 연구

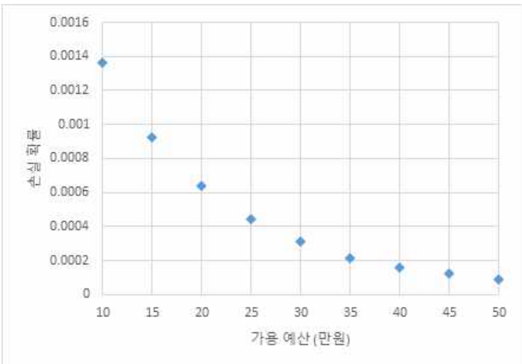
본 연구에서는 앞서 개발된 수리 모형을 현재 한동대학교 도서관 데이터에 적용하여 분석을 실시하였다. 6년간의 대출, 반납 데이터를 통하여 수요와 평균 대출 기간을 계산하였고, 도서 별 보유 장서 수는 현재 대학 도서관이 보유하고 있는 장서 수를 그대로 사용하였다. 책들의 가격은 현재 인터넷 서점에서 판매하고 있는 가격으로 설정하였다.

현재 도서관이 보유하고 있는 장서 중에서 손실 확률을 최소화하기 위해 가장 대출 횟수가 많은 도서 10권 중 어떤 책을 추가로 구매해야 하는지 살펴보자. 현재 보유 중인 각 도서별 장서 수는 표 1에 제시되어 있다. 가용한 예산에 따라 고객 손실 확률을 최소화하기 위하여 추가로 구매 하는 도서들을 계산한 결과는 표 2에 나타나 있다. 가용 공간을 나타내는 변수 S는 200으로 설정하였으며, 모든 r_i값은 0.01로 설정하였다. 먼저 표의 2행을 보면 현재 도서관의 10권에 대한 손실확률이 계산되어 있다. 식 (3)을 통하여 계산되었지만, 대출 요청은 10권 중 한권이라는 조건을 취해 얻어진 결과임을 참고하자. 나머지 행들에 예산에 따른 추가 구입하는 도서의 장서들과 손실 확률이 표 2에 나타나 있다. 예를 들어, 추가로 도서를 구매할 때 사용할 수 있는 돈이 10만

< 표 2 : 최다 대출 횟수를 지닌 도서 10권에 대한 알고리즘의 적용 >

가용 예산 (원)	도서 i의 추가 구매 권수 (권)										손실확률
	1	2	3	4	5	6	7	8	9	10	
0 (현재)	0	0	0	0	0	0	0	0	0	0	0.0061238
50,000	1	0	0	0	0	4	1	0	0	0	0.0011162
100,000	2	0	0	0	0	6	2	0	1	0	0.0003721
150,000	4	0	0	0	0	6	3	0	2	0	0.0001041
200,000	5	0	0	0	0	7	5	0	3	0	0.0000267
250,000	6	0	0	0	0	8	5	0	5	0	0.0000066
300,000	7	0	0	0	0	10	6	0	6	0	0.0000022
350,000	8	0	0	0	1	11	7	0	7	0	0.0000012
400,000	9	1	0	0	1	10	8	0	7	0	0.0000005
450,000	9	2	0	0	1	11	8	0	7	1	0.0000003
500,000	10	2	0	0	2	12	9	0	8	1	0.0000001

원인 경우, 1번 도서를 2권 도서를 6권, 7번 도서를 2권 9번 도서를 1권 구매하는 것이 손실확률을 최소화하며, 이 때 손실확률은 약 0.037%로 계산된다. 이와 같이 손실 확률을 최소화하기 위하여 추가로 어떤 도서를 구매해야 하는지 결정할 수 있다. 가용 예산에 따른 대출 횟수가 많은 도서 50권에 대한 결과가 그림 3에 나타나 있다.



< 그림 3 : 가용 예산에 따른 손실확률 >

5. 결 론

본 논문에서는 M/G/c/c 대기행렬 모형의 얼랑 손실 함수를 이용하여, 도서관의 서비스 품질을 보장하기 위한 최소 장서수를 계산하고 분석하였다. 본 연구에서 개발된 수리 모형은 도서관의 예산계획을 수립하거나, 도서관의 적정 장서 보유량을 결정하는데 유용한 정보를 제공할 수 있다. 얼랑 손실 함수는 비선형 함수이기 때문에 수리 모형을 분석하기 위해, 수리 모형을 제약조건을 가지는 0-1 Knapsack 문제로 치환하여 동적 계획법(dynamic programming)에 기반한 알고리즘을 통하여 최적 해를 도출하였다. 개발된 알고리즘을 한동대학교 도서관의 데이터에 적용하여, 가용 예산에 따른 추가 구매 도서들을 결정할 수 있다.

본 연구는 향후 다양한 방향으로 확장될 수 있다. 현재의 수리 모형은 M/G/c/c 대기행렬 모형을 이용하기 때문에 고객이 찾는 도서가 대출 불가능할 경우 고객은 시스템을 이탈하는 것으로 가

정하고 있다. 구체적인 대출 기간의 분포를 데이터를 통해 찾아내어 G/G/m 근사해법의 식을 이용하거나 관련된 대기행렬 모형의 식을 이용한다면 대출 불가능한 도서에 대하여 예약이 가능한 도서관 시스템에서의 대기시간을 계산할 수 있다. 이를 통하여 도서관 시스템을 최적화 하는 것이 하나의 방법이 될 수 있을 것이다. 또한 현재는 도서관에서 도서를 구매하는 것으로 가정하고 있지만, 최근에는 한 도서관이 다른 도서관들에서 대출하여 도서관 시스템을 운용하기도 한다. 이러한 상황을 고려한다면 본 수리모형을 확장하여 유용한 정보를 도출할 수 있을 것이다.

References

[1] 김정수, 이철규 and 강민형 (2013). 모바일 전자책 서비스품질, 고객만족 및 충성도 간의 구조적 관계에 대한 연구, 한국경영공학회지, 18(1), 99-116.

[2] 서원철 and 이운식 (2014). 유한 재생산률을 고려한 폐기물 동적 회수계획과 다중제품 재생산계획 문제, 한국경영공학회지, 19(2), 1-13.

[3] 이운식 and 김병수 (2011), 유한 재생산율을 갖는 동적 구매-재생산계획문제, 한국경영공학회지, 16(3), 261-274.

[4] Ackere, A. V., Haxholdt, C., and Larsen, E. R. (2006). Long term and short term customer re action: a two stage queueing approach. System Dynamics Review, 22(4), 349-369.

[5] Beheshti, J., and Tague, J. M. (1984). Morse's Markov model of book use revisited. Journal of the American Society for Information Science, 35(5), 259-267.

[6] Chen, C. C. (1976). Applications of operational research models to libraries, M.I.T. Press.

[7] Morse, P. M.(1968), Library Effectiveness: A Systems Approach, M.I.T. Press.

[8] Osorio, C., and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. Transportation Science, 49(3), 623-636.

[9] Rouse, W. B. (1976). A library network model. Journal of the American Society for Information Science, 27(2), 88-99.

[10] Rouse, W. B., and Rouse, S. H. (1977). Assessing the impact of computer technology on the performance of interlibrary loan networks. Journal of the American Society for Information Science, 28(2), 79-88.

[11] Smith, J. M. (2015). Optimal workload allocation in closed queueing networks with state dependent queues. Annals of Operations Research, 231(1), 157-183.

[12] Warwick, D. J. (1994). A queuing network model for book circulation. Collection management, 19(1-2), 69-80.

[13] Warwick, J. (1998). A queuing theory model for book reservations and circulation. Collection management, 23(1-2), 125-137.

[14] Warwick, J. (2009). On 40 years of queuing in libraries. Library Review, 58(1), 44-55.

[15] Wells, A. (2007). A prototype twenty-first cen tury university library: A case study of change at the University of New South Wales Library. Library Management, 28(8/9), 450-459.

[16] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2003). Introduction to Algorithm, 2nd edition, M.I.T Press.