주서비스와 보조서비스를 갖는 시스템 설계*

김 성 철**

A Design Problem of a System Working at Both Primary Service and Secondary Service

Sung-chul Kim**

■ Abstract ■

In this paper, we consider a system working at both primary service and secondary service. A server can switch between the primary service and the secondary service or it can be assigned to secondary service as a dedicated server. A service policy is characterized by the number of servers dedicated to the secondary service and a rule for switching the remaining servers between two services. The primary service system is modelled as a Markovian queueing system and the throughput is a function of the number of servers, buffer capacity, and service policy. And the secondary service system has a service level requirement strategically determined to perform the service assigned. There is a revenue obtained from throughput and costs due to servers and buffers. We study the problem of simultaneously determining the optimal total number of servers, buffers, and service policy to maximize profit of the system subject to both an expected customer waiting time constraint of the primary service and a service level constraint of the secondary service and develop an algorithm which can be successfully applied with the small number of computations.

Keywords : Optimization Problem, Service Policy, Queueing System, the First Moment, the Second Moment, Marginal Analysis, Implicit Enumeration

논문접수일 : 2011년 04월 26일 논문수정일(1차 : 2011년 07월 18일, 2차 : 07월 26일)
논문게재확정일 : 2011년 07월 26일
* 본 연구는 덕성여자대학교 2011년도 교내연구비 지원에 의해 수행되었음.
** 덕성여자대학교 경영학과

1. 서 론

본 논문에서는 주서비스(primary service)와 보조서비스(secondary service)의 서로 다른 두 기능의 서비스를 제공하는 시스템을 설계하는 최적화 문제(optimization problem)를 다룬다. 시스템을 설계한다는 것은 주서비스에 도착하는 고객의 수요와 보조서비스에서 요구되는 서비스 능력을 동시에 만족시키는 최적의 서비스능력(service capacity), 대기용량(buffer size) 그리고 서비스정책(service policy)을 결정하는 것을 말한다. 주서비스와 보조서비스의 서로 다른 두 종류의 서비스를 수행할 수 있는 서버는 컴퓨터에 의하여 제어되거나 교차훈련된(cross-trained) 서버를 의미하며 서비스능력이란 제공되는 서버의 수를 말한다. 대기용량이란 고객이 서비스를 받기 위해 기다리거나 서비스를 제공받을 수 있는 대기공간의 수이다. 서비스정책(service policy)이란 두 종류의 수요에 효율적으로 대응할 수 있도록 기능의 융합(functional convergence)를 통하여 서비스혁신을 추구하는 서버의 배분정책으로 주서비스에 대기 중인 고객이 많은 경우에는 주서비스에 서버를 많이 배치하고 대기 중인 고객이 적은 경우에는 적게 배치하여 주서비스에 배치되지 않은 서버는 보조서비스를 수행하도록 하는 것을 말한다. 별도로 보조서비스만을 담당하도록 전담서버를 배치하는 것도 고려한다.

두 종류의 서비스를 갖는 시스템은 서비스 시스템, 제조시스템, 통신시스템, 그리고 설비보전시스템 등 다양하다. 그 중 한 예로서 유통을 위한 매장을 갖는 서비스 시스템을 보자. 이러한 서비스 시스템에 있어서 주서비스는 매장에 도착하는 고객을 맞이하여 상품을 판매하는 업무가 되고 보조서비스는 창고에서 상품을 입출고하고 분류 정리하는 재고관련 업무가 된다. 주서비스와 보조서비스를 모두 담당할 수 있도록 교차훈련된 종업원은 매장에서는 고객에게 상품을 판매하고 창고에서는 재고를 정리한다. 그러므로 이러한 서비스 시스템에 있어서는 매장에 도착하는 고객에게 효율적인 서비스를 제공할 수 있도록 서비스 시스템을 설계하는 것은 매우 중요하며 매장에 고객이 많은 경우에는 많은 수의 종업원이 매장에 배치되고 고객이 적은 경우에는 적은 수의 종업원이 배치되어 매장에 배치되지 않은 종업원은 창고에서 재고와 관련된 보조서비스를 수행한다. 창고만을 담당하는 전담종업원의 배치도 고려할 수 있다. 그러므로 이러한 서비스 시스템에서 요구되는 최적의 서비스능력과 대기용량의 설계와 함께 주어진 서비스능력을 효율적으로 활용할 수 있도록 서비스 정책을 통합적으로 고려하여 서비스 시스템을 설계하는 것은 매우 중요하다.

두 종류의 서비스를 갖는 시스템의 다른 예를 보자. 제조시스템에 있어서는 주서비스는 시장이나 상위단계(upstream)의 수요를 처리하는 제조이며 보조서비스는 제조에 소요되는 제반 재료를 보관하는 창고에서 수행되는 작업을 들 수 있다. 저장전송(store and forward)을 수행하는 통신시스템을 보면 주서비스는 메시지를 수신하고 저장하여 전송하는 통신업무를 담당하고 보조서비스로는 정보처리업무를 수행하며 설비보전 시스템의 경우에는 주서비스는 고장이 난 설비의 교정보전을 수행하고 보조서비스는 예방보전을 수행하는 것으로 설명될 수 있다.

다루고자 하는 문제를 두 개로 분류하면 첫 번째 문제는 주서비스에 도착하는 수요의 도착율과 제약조건을 고려하여 설계모수인 서비스능력과 대기용량을 결정하는 전략적(strategic) 문제이다. 두 번째 문제는 주서비스와 보조서비스의 융합으로 구현되는 프로세스이며 기능의 융합을 통하여 서비스정책을 결정하는 전술적(tactical) 문제이다. 그러므로 본 논문은 전략적 의사결정문제인 시스템의 서비스능력 및 대기용량의 설계와 전술적 의사결정문제인 서비스정책을 통합적으로 고려함으로써 시스템설계에 있어서 계층적 의사결정문제인 전략적 문제와 전술적 문제를 함께 다루어 통합적 최적화를 추구한다.

이를 좀 더 부연설명하면 서버와 대기용량을 확

보하는데 소요되는 비용(cost)이 소요되고 확보된 서비스능력과 서비스정책의 함수인 생산율(throughput)과 수익(revenue)이 발생하여 수익과 비용을 함께 고려한 이익(profit)이 산정된다. 반면에 주서비스에 도착하는 고객의 대기시간이 고객에 대한 서비스를 보존하기 위한 중요한 수행도 측정치가 되어 주서비스에서는 고객의 평균대기시간(mean waiting time)이 제약조건으로 주어지고 시간적으로 주서비스보다 훨씬 민감하지 않은 일상적인 업무가 수행되는 보조서비스에서는 보조서비스에서 필요로 하는 서비스능력을 제공하기 위하여 보조서비스에서 요구되는 서비스능력이 제약조건으로 주어진다.

서비스정책은 다음과 같이 설명될 수 있다. 설명을 쉽게 하기 위해서 보조서비스만을 담당하는 전담서버는 없다고 가정한다. m개의 서버가 주어졌다고 가정하자. 서버의 교체점(switching point)은 r_d, $d=0, 1, \cdots, m$로 표현되며 주서비스의 고객의 수가 $r_{d-1}+1$과 r_d사이에 존재하면 주서비스에 $d(1 \le d \le m)$개의 서버를 배치하고 나머지 $m-d$개의 서버는 보조서비스에 배치됨을 의미한다. 예를 들어 $m=5$, $d=2$, $r_1=2$, 그리고 $r_2=5$라면 주서비스에 존재하는 고객의 수가 3~5사이이면 주서비스에 2개의 서버를 배치하고 나머지 3개의 서버는 보조서비스에 배치하는 것을 말한다. 그러므로 서비스정책은 서버의 교체점들의 집합 $(r_0, r_1, \cdots, r_m)$으로 표현된다. 예를 들어 $m=5$이고 서비스정책이 $(0, 2, 4, 7, 8, 10)$은 주서비스에서의 고객의 수가 2까지는 주서비스에 하나의 서버를 배치하고 나머지 4개의 서버를 보조서비스에 배치하며, 고객의 수가 3과 4인 경우에는 주서비스에 두 개의 서버를 나머지 세 개는 보조서비스에, 고객의 수가 5, 6, 7인 경우에는 세 개의 서버를 주서비스에 나머지 두 개는 보조서비스에, 고객의 수가 8인 경우에는 네 개의 서버를 주서비스에 한 개는 보조서비스에, 그리고 고객의 수가 9와 10인 경우에는 다섯 개의 서버를 모두 주서비스에 배치하고 보조서비스에는 하나도 배치하지 않음을 의미한다. 여기에서 대기용량은 10으로 가정되었음을 인지하자. 주서비스에 너무 적은 서버를 배치하거나 너무 많게 배치하는 것은 생산율을 감소시키고 평균대기시간을 증가시키거나 보조서비스의 서비스능력을 감소시킨다.

이와 같이 서비스메커니즘(mechanism)을 제공하는 시스템은 대기시스템(queueing system)으로 모형화 될 수 있다. 그러므로 주서비스는 수요종속적인 서비스능력을 갖는 대기시스템으로 모형화되고 주어진 제약식을 만족시키면서 서버의 수, 대기용량, 그리고 서비스정책의 함수로 표현되는 수익과 서버와 대기용량에 소요되는 비용을 동시에 고려하여 이익을 최대화하는 해법이 제시된다. 일반적으로 전략적 문제와 전술적 문제는 개별적으로 다루어지고 있으며 본 논문과 같이 이를 통합적으로 최적화하는 문제는 더 복잡하고 어려운 문제이다.

Berman and Larson[2]과 Berman et al.[3] 등은 교차훈련된 서버를 갖는 서비스 시스템에서 최적 서비스정책을 결정하는 문제를 다루었다. Berman et al.[3]는 교차훈련된 주어진 마코비안(markovian) 대기시스템에 있어서 보조서비스에 요구되는 서비스능력의 제약식을 만족시키면서 주서비스에서 고객의 평균대기시간을 최소화하는 서비스정책을 결정하는 최적화문제를 다루었다. Berman and Larson[2]은 최적화 모형을 주서비스와 보조서비스 사이의 서버 교체에 따른 비용을 포함하는 경우로 연장하였다. 제조시스템을 대기시스템으로 모형화하고 서비스능력이나 대기용량을 설계하는 문제는 다양하며 특히 Buzacott and Shanthikumar[4]에 체계적으로 제시되고 있다. 본 논문과 관련하여 가장 관계가 있는 논문으로는 Shanthikumar and Yao[11]를 들 수 있으며 이는 폐쇄대기 네트워크(closed queueing network)에 있어서 서버의 수와 대기용량을 설계하는 모형을 제시하였다.

제 2장에서는 서버의 수, 대기용량, 그리고 서비스정책을 의사결정변수로 갖는 목적함수에 주서비스에서의 평균체재시간과 보조서비스에서 요구되는 서비스능력을 제약식으로 갖는 비선형 정수계획 최적화문제(nonlinear integer programming opti-

mization problem)가 모형화된다. 이를 위하여 서비스정책이 정의되고 수행도가 산정된다. 제 3장에서는 최적화문제의 최적화절차가 유도되고 제시된다. 이를 위하여 다양한 수행도의 특성들이 서비스정책과 관련되어 제시된다. 특히 설계모수에 대한 수행도의 일계특성(the first moment)과 이계특성(the second moment)이 관심의 대상이 된다. 제 4장에서는 최적화절차와 최적화결과에 대한 수치예가 제시된다. 제 5장에서는 본 논문의 주제와 결과에 대한 전반적인 의미를 기술하는 결어로서 마감한다.

2. 최적화모형

주서비스에 도착하는 고객의 수요는 기대치 λ인 포아송분포(poisson distribution)에 의하며 주서비스에서 서버의 서비스시간은 서비스율(service rate)이 μ인 지수분포(exponential distribution)를 갖는다. $\rho=\lambda/\mu$라고 정의하면 ρ는 주서비스에 부여된 부하(offered load) 즉 일의 양을 의미한다. 보조서비스에서 요구되는 최소한의 서비스율은 ν이다.

총 서버의 수를 s, 주서비스의 대기용량을 $n(s \le n)$, 그리고 서비스정책을 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$, $|r_d - r_{d-1}| \ge 1$, $r_0 \ge 0$이라고 하자. 여기에서 서비스정책, $R_{m,s,n}$은 총 s개의 서버 중 주서비스에 배치되는 서버는 최대 m이며 $s-m$개의 서버는 보조서비스의 전담서버로 배치되고 서버의 교체점이 r_d, $d=0, 1, \cdots, m$이 되는 서비스정책으로 $r_m = n$이다. 이를 부연설명하면 주서비스에 존재하는 고객의 수가 $r_{d-1}+1$과 r_d사이인 경우에는 주서비스에는 $d(1 \le d \le m)$개의 서버가 배치되고 보조서비스에는 나머지 $m-d$개의 서버와 전담서버인 $s-m$개의 서버의 합인 $s-d$개의 서버가 배치되며 $m=s$인 경우에는 보조서비스만을 위한 전담서버가 없음을 의미한다.

주서비스에 m개의 서버와 대기용량 n이 주어지고 서비스정책이 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$이 적용되는 경우 주어진 마코브 과정(markov process)은 다음의 상세균형방정식(detailed balance equation)을 만족시킨다.

$$\lambda p_{m,n,R}(x) = d\mu p_{m,n,R}(x+1) \quad d\mu p_{m,n,R}(x+1),$$
$$x = r_{d-1}, r_{d-1}+1, \cdots, r_d - 1, \ d = 1, \cdots, m. \qquad (2.1)$$

그러므로 주서비스의 상태공간(state space)은 절단되고(truncated)(Kelly[7]) 주서비스에 존재하는 작업물의 수 $X_{m,n,R}$이 x, $x=r_0, \cdots, n$,일 상태확률(state probability) $p_{m,n,R}(X_{m,n,R}=x)$은 다음과 같다.

$$p_{m,n,R}(x) = \xi_{m,n,R}(x) \quad p_{m,n,R}(r_0),$$
$$r_{d-1} < x \le r_d, \ 1 \le d \le m,$$
$$p_{m,n,R}(r_0) = (1 + \sum_{x=r_0+1}^{n} \xi_{m,n,R}(x))^{-1}. \qquad (2.2)$$

여기에서

$$\xi_{m,n,R}(x) = (\frac{\rho}{d})^{x-r_{d-1}} (\frac{\rho}{d-1})^{r_{d-1}-r_{d-2}} \cdots (\frac{\rho}{1})^{r_1 - r_0},$$
$$r_{d-1} < x \le r_d, \ 1 \le d \le m. \qquad (2.3)$$

그러므로 주어진 주서비스의 수행도를 산정할 수 있는 다음의 양들을 쉽게 산정할 수 있다. 먼저 주서비스에 존재하는 작업물 수의 기대치 $L_{m,n,R}$과 평균체재시간 $W_{m,n,R}$은 다음과 같다.

$$L_{m,n,R} = \sum_{x=r_0}^{n} x p_{m,n,R}(x), \qquad (2.4)$$

$$W_{m,n,R} = \frac{L_{m,n,R}}{\lambda(1-p_{m,n,R}(n))}. \qquad (2.5)$$

여기에서 $p_{m,n,R}(n)$은 봉쇄확률(blocking probability)로서 주서비스에 도착하는 고객은 기대치 $\lambda p_{m,n,R}(n)$로 봉쇄되므로 주서비스에 도착하는 고객의 실제 도착률은 $\lambda\{1-p_{m,n,R}(n)\}$가 된다. 또한 주서비스에 존재하는 서버의 수의 기대치를 $Q_{m,n,R}$, 보조서비스에 존재하는 서버의 수의 기대치를 $v_{m,n,R}$이라 하면 다음이 성립한다.

$$Q_{m,n,R}=\sum_{d=1}^{m}\sum_{x=r_{d-1}+1}^{r_d} d\,p_{m,n,R}(x), \tag{2.6}$$

$$v_{m,n,R}=s-Q_{m,n,R}. \tag{2.7}$$

주서비스의 생산율(throughput)을 $\theta_{m,n,R}$이라 하면 주서비스에 실제 도착한 고객은 모두 서비스를 제공받고 시스템을 떠나게 되므로 주서비스의 생산율은 주서비스에 실제 도착율과 같게 되며 봉쇄확률 $p_{m,n,R}(n)$은 주서비스의 생산율을 결정하는 중요한 모수가 된다.

$$\begin{aligned}\theta_{m,n,R}&=\sum_{d=1}^{m}\sum_{x=r_{d-1}+1}^{r_d} d\mu p_{m,n,R}(x)\\&=\lambda\{1-p_{m,n,R}(n)\}.\end{aligned} \tag{2.8}$$

이제 주서비스에서의 생산율 $\theta_{m,n,R}$의 함수로서 수익함수를 $f(\theta_{m,n,R})$서버 s를 확보하는데 소요되는 비용함수를 $h(s)$, 그리고 주서비스의 대기용량 n을 확보하는데 소요되는 비용함수를 $g(n)$이라 정의하자. 여기에서 서버와 대기용량의 확보에 소요되는 비용은 주어진 대기시스템의 생산율과 일치하도록 단위기간 당 비용으로 치환된 금액을 의미한다. 또한 수익함수 $f(\theta_{m,n,R})$은 생산율 $\theta_{m,n,R}$에 대하여 증가하는 오목함수(concave function)를, 비용함수 $h(s)$와 $g(n)$은 각각 서버 수 s와 대기능력 n에 대하여 증가하는 볼록함수(convex function)를 가정한다.

이제 의사결정변수인 설계모수 서버의 수 s 대기능력 n 그리고 서비스정책 $R_{m,s,n}$에 대한 최적화문제를 통합적으로 정리하면 다음과 같이 모형화된다.

$$\begin{aligned}&Max._{m,n,R}\ \ f\{\lambda[1-p_{m,n,R}(n)]\}-h(s)-g(n)\\&s.t.\ \ W_{m,n,R}\le W,\\&v_{m,n,R}\ge v,\\&1\le m\le s\le n\ \text{and 정수}.\end{aligned} \tag{2.9}$$

주어진 최적화모형의 목적함수는 서버의 수 및 대기용량의 함수로 산정되는 수익함수와 비용함수를 함께 고려하여 산정되는 이익함수를 정식화한 것이며 첫 번째 제약식은 주서비스에서의 작업물의 평균체재시간의 상한을, 두 번째 제약식은 보조서비스에서 최소한 요구되는 서비스능력을 제약식으로 표시한 것이다. 그러므로 주어진 최적화문제는 설계모수인 서비스의 수, 대기용량, 그리고 서비스정책의 결과로 결정되는 이익함수를 목적함수로 하고 주서비스에서의 평균체재시간과 보조서비스에서 요구되는 서비스능력을 제약식으로 하는 비선형정수계획 최적화문제이다.

3. 최적화해법

주어진 최적화문제를 풀기 위해서 가능한 접근방법은 주어진 설계모수에 대하여 수행도의 특성을 도출하는 일이라 하겠다. 여기서 수행도의 특성이란 일반적으로 수행도의 일계모멘트와 이계모멘트를 대상으로 하며 이러한 특성은 해의 공간을 현저히 감소시킴으로써 주어진 최적화과정을 매우 용이하게 하여 주는 유용한 결과를 제시한다. 그러므로 먼저 이러한 특성과 관련된 내용에 대하여 살펴보기로 한다.

특히 본 최적화문제에 있어서는 총 서버의 수 s, 대기용량 n이 주어져 있을 때 서비스정책 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$이 다양하게 적용되고 그 결과로 수행도도 다양하게 주어지므로 주어진 시스템의 수행도의 특성을 파악하기 위해서는 먼저 서비스 정책 $R_{m,s,n}$에 대하여 살펴보자.

이를 위하여 비음정수(nonnegative integer)값을 갖는 두 개의 이산적(discrete) 확률변수 Y_1과 Y_2에 대하여 누적확률분포(cumulative probability distribution)를 $F_i(x)=p(Y_i\le x)$, 생존함수(survival function)를 $\overline{F_i}(x)=p(Y_i>x)$, $i=1, 2$로 표현하면 $\le^{lr}$(likelyhood ratio ordering)과 $\le^{st}$(추계적(stochastic) ordering)은 다음과 같이 정의된다(Ross[9]).

$$Y_1\le^{lr}Y_2\ \leftrightarrow\ \frac{p(Y_1=y-1)}{p(Y_1=y)}\ge\frac{p(Y_2=y-1)}{p(Y_2=y)},$$

$$y \geq 1; \tag{3.1}$$

$$Y_1 \leq^{st} Y_2 \leftrightarrow \overline{F_1}(y) \leq \overline{F_2}(y), \ \forall y; \tag{3.2}$$

$$Y_1 \leq^{lr} Y_2 \Rightarrow Y_1 \leq^{st} Y_2 \Rightarrow E(Y_1) \leq E(Y_2); \tag{3.3}$$

이제 서비스정책 $R^1_{m,s,n}=(r_0, r_1, \cdots, r_j, \cdots, r_m, s-m)$, $R^2_{m,s,n}=(r_0, r_1, \cdots, r_j-1, \cdots, r_m, s-m)$, $j \in (0, \cdots, m-1)$이라 하자. 이는 $d \neq j$이면 $r^1_d = r^2_d$이며 $|r^1_j - r^1_{j-1}| \geq 2$, $r^2_j = r^1_j - 1$임을 의미한다.

정리 1 : $L_{m,n,R^1} \geq L_{m,n,R^2}$, $W_{m,n,R^1} \geq W_{m,n,R^2}$, $\theta_{m,n,R^1} \leq \theta_{m,n,R^2}$, $Q_{m,n,R^1} \leq Q_{m,n,R^2}$, 그리고 $\upsilon_{m,n,R^1} \geq \upsilon_{m,n,R^2}$이다.

증명 : 먼저 $r_{d-1}+1 \leq x \leq r_d$, $x \neq r_j$에 대하여 $\frac{p_{m,n,R^1}(x-1)}{p_{m,n,R^1}(x)} = \frac{p_{m,n,R^2}(x-1)}{p_{m,n,R^2}(x)} = \frac{d}{\rho}$, $x = r_j$에 대하여 $\frac{p_{m,n,R^1}(r_j-1)}{p_{m,n,R^1}(r_j)} = \frac{j}{\rho} < \frac{p_{m,n,R^2}(r_j-1)}{p_{m,n,R^2}(r_j)} = \frac{j+1}{\rho}$가 만족되므로 $X_{m,n,R^1} \geq^{lr} X_{m,n,R^2}$이 성립하고 $X_{m,n,R^1} \geq^{lr} X_{m,n,R^2} \Rightarrow X_{m,n,R^1} \geq^{st} X_{m,n,R^2} \Rightarrow L_{m,n,R^1} \geq L_{m,n,R^2}$이 성립된다. 이제 식 (3.2)에 의하여 $p_{m,n,R^1}(n) \geq p_{m,n,R^2}(n)$이 만족되므로 Little 법칙(Little[8])에 의하여 $W_{m,n,R^1} \geq W_{m,n,R^2}$이 성립하고 식 (2.8)에 의하여 $\theta_{m,n,R^1} \leq \theta_{m,n,R^2}$이 성립된다. $Q_{m,n,R^1} \geq Q_{m,n,R^2}$, 그리고 $\upsilon_{m,n,R^1} \leq \upsilon_{m,n,R^2}$의 증명은 Berman et al.[3]에 의한다.

이제 서비스정책, $R_{m,s,n}$에서 정의되는 두 서비스정책 $R^3_{m,s,n}=(n-m, n-m+1, \cdots, s-m)$과 $R^4_{m,s,n}=(0, 1, \cdots, m-1, n, s-m)$을 보자. 정리1에 의하여 모든 서비스정책에 대하여 서비스정책 $R^3_{m,s,n}$가 주서비스에 존재하는 작업물의 수의 기대치, 평균체재시간, 그리고 봉쇄확률이 가장 크고 생산율과 주서비스에 존재하는 서버의 수의 기대치가 가장 적으며, 서비스정책 $R^4_{m,s,n}$는 주서비스에 존재하는 작업물의 수의 기대치, 평균체재시간, 그리고 봉쇄확률이 가장 적고 생산율과 주서비스에 존재하는 서버의 수의 기대치가 가장 크다. 그러므로 주어진 최적화문제에 있어서 서비스정책 $R^3_{m,s,n}$는 보조서비스에서의 서비스능력의 실행가능성(feasibility)을 확인하는데, 서비스정책 $R^4_{m,s,n}$는 목적함수를 최대화하고 평균체재시간의 실행가능성을 확인하는데 적용될 수 있어 주어진 최적화 문제의 최적화 과정에 매우 유용하게 활용될 수 있다. 그러므로 주어진 최적화문제에 있어서 매우 중요한 결과를 제시할 수 있는 두 서비스정책 $R^3_{m,s,n}$과 $R^4_{m,s,n}$의 수행도의 특성에 대하여 살펴보기로 한다.

정리 2 : 다음의 결과가 성립한다.

1. 주서비스에서의 대기용량 n이 주어져 있을 때 생산율 θ_{m,n,R^3}는 서버의 수 m에 대하여 증가하는 오목함수이나 서버의 수 m이 일정하면 대기용량 n이 변하여도 생산율은 일정하다.
2. 주서비스에서의 대기능력 n이 주어져 있을 때 생산율 θ_{m,n,R^4}는 서버의 수 m, $m=1, \cdots, n$에 대하여 증가하는 오목함수이다.
3. 서버의 수 m이 일정할 때 생산율 θ_{m,n,R^4}는 주서비스에서의 대기용량 $n(n \geq m)$에 대하여 증가하는 오목함수이다.

증명 : 서비스정책 $R^3_{m,s,n}$과 $R^4_{m,s,n}$는 Gordon and Newell[6]의 폐쇄대기 네트워크에서 작업장이 하나인 특수한 경우로 적용가능하다. 그러므로 주어진 결과는 폐쇄대기 네트워크에 있어서의 Shanthikumar and Yao[11]의 결과에 의한다. 또한 주어진 결과는 직접 대수적으로 접근함으로써 쉽게 증명될 수 있다.

정리 3 : 총 서버의 수 s와 대기용량 n이 일정할 때 서비스정책 $R^4_{m,s,n}=(0, 1, \cdots, m-1, n, s-m)$, $R^5_{m+1,s,n}=(0, 1, \cdots, m-1, m, n, s-(m+1))$라 정의하면 $\theta_{m+1,n,R^5} \geq \theta_{m,n,R^4}$이다. 이제 서비스정책 $R^6_{s,s,n}=(0, 1, \cdots, s$

$-1, n, 0)$으로 정의하면 서비스정책 $R^6_{s,s,n}$의 생산율이 가장 크다.

증명 : 정리 2에 의한다. 즉 생산율 θ_{m,n,R^4}는 서버의 수 $m, m=1, \cdots, n$에 대하여 증가하는 함수이다.

그러므로 서비스정책 $R^6_{s,s,n}=(0,1,\cdots,s-1,n,0)$는 보조서비스에서 요구되는 최소한의 서비스능력의 제약을 무시한다면 주서비스에서의 생산율이 가장 크고 평균체재시간이 가장 작고 주서비스에 존재하는 서버 수의 기대치가 가장 크며 결과적으로 보조서비스에 존재하는 서버 수의 기대치가 가장 적다. 주어진 서비스정책은 최적화를 수행하는 과정에 있어서 생산율과 수익이 가장 높은 서비스정책으로 평균체재시간의 실행가능성을 확인하는데 매우 유용하게 활용될 수 있다.

이제 원 최적화문제로 돌아가자. 주어진 최적화문제의 설계모수는 서버의 수 s, 주서비스의 대기용량 n 그리고 서비스정책 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$이므로 주어진 최적화문제를 접근하는 방법으로 먼저 대기용량 $n, n=0, 1, \cdots$이 고려되고 주어진 대기용량 n에 있어서는 서버의 수 $s, s=1, \cdots, n$가 고려되고 마지막으로 주어진 배분 n과 s에 있어서는 서비스정책 $R_{m,s,n}$이 고려된다.

서버의 수 s와 대기용량 n이 주어졌을 때 생산율이 가장 크고 평균체재시간이 가장 적은 서비스정책이 $R^6_{s,s,n}=(0, 1, \cdots, s-1, n, 0)$이므로 먼저 제약식은 고려하지 않고 서비스정책 $R^6_{s,s,n}$가 적용되는 경우에 있어서 주서비스에서의 서버의 수 s와 대기용량 n을 결정하는 최적화절차를 보자.

먼저 대기용량 n이 주어져 있는 경우를 가정하고 다음의 최적화모형을 정의한다.

$$Max._{1 \le s \le n} \ \Theta(s, n, R^6_{s,s,n}) = f\{\lambda[1-p_{s,n,R^6}(n)]\} - h(s) = f(\theta_{s,n,R^6}) - h(s). \quad (3.4)$$

대기용량 n에서의 최적 서버의 수를 s_n이라고 정의하면 최적화모형 식 (3.4)는 원 최적화모형 식 (2.9)에서 대기능력 n이 일정(constant)하게 주어져 있으므로 비용함수 $g(n)$도 일정하며 결과적으로 주어진 대기용량 n에서 최적 서버의 수 s_n과 이익 $\Theta(s_n, n, R^6_{s_n s_n n})$을 구하는 문제이다. 주서비스의 대기용량 n이 주어져 있으므로 생산율이 최대인 서비스정책 $R^6_{s,s,n}$에서의 생산율 θ_{s,n,R^6}는 서버의 수 $s, s=1, \cdots, n$에 대하여 증가하는 오목함수이고 수익함수 $f(\theta_{s,n,R^6})$도 생산율 θ_{s,n,R^6}에 대하여 증가하는 오목함수이므로 서버의 수 s에 대하여 수익함수 $f(\theta_{s,n,R^6})$는 증가하는 오목함수가 되고, 비용함수 $h(s)$는 증가하는 볼록함수이므로, 결과적으로 이익함수 $\Theta(s, n, R^6_{s,s,n})$는 서버의 수 s에 대하여 오목함수가 되어 최적 서버의 수 s_n은 한계분석법(marginal analysis)(Fox[5])으로 쉽게 얻어질 수 있다. 여기에서 한계분석법이란 의사결정변수 s를 한 단위씩 증가시켜 가면서 처음으로 이익함수 $\Theta(s, n, R^6_{s,s,n})$가 감소하는 s에서 최적해를 도출하고 알고리즘을 종료하는 방법으로 처음으로 $\nabla(\Theta(s, n, R^6_{s,s,n})) = \Theta(s+1, n, R^6_{s+1,s+1,n}) - \Theta(s, n, R^6_{s,s,n}) \le 0$, $1 \le s < n$이 만족되는 s가 구하고자 하는 최적 서버의 수 s_n이 되는 최적화기법이다.

이제 최적화모형 식 (3.4)로 얻어진 결과를 원 최적화문제에 적용하면 식 (3.4)는 다음과 같이 모형화 될 수 있다.

$$Max._{1 \le n} \ \Phi(s_n, n, R^6_{s_n s_n n}) = \Theta(s_n, n, R^6_{s_n s_n n}) - g(n). \quad (3.5)$$

만약 생산율함수 $\theta(s, n, R^6)$가 서버의 수 s와 대기용량 n에 대하여 공동(jointly)의 오목함수임을 증명할 수 있다면 주어진 최적화 문제는 한계분석법으로 쉽게 최적배분을 구할 수 있다. 여기에서 한계분석법이란 최초로 $\nabla\Phi(s_n, n, R^6_{s_n s_n n}) = \{\Theta(s_{n+1}, n+1, R^6_{s_{n+1}, s_{n+1}, n+1}) - g(n+1)\} - \{\Theta(s_n, n, R^6_{s_n s_n n}) - g(n)\} \le 0$이 성립하는 대기용량 n에서 알고리즘을

종료하는 것을 말한다. 그러나 정리 2는 생산율 함수 $\theta(s, n, R^6)$가 서버의 수 s와 대기용량 n 각각에 대해서는 증가하는 오목함수임을 보이고 있으나 공동의 오목함수임은 제시하지 못하고 있다. 그러므로 전술된 한계분석법을 적용할 수 없다.

그러나 최적화모형 식 (3.5)는 얼마간의 복잡성(complexity)을 추가하여 다음과 같은 방법으로 최적 대기용량 n^*를 구할 수 있으며 그 절차는 다음과 같다.

$$Max._{1 \le n \le \underline{n}} \Phi(s_n, n, R^6_{s_{n^*}s_{n^*}n}) = \Theta(s_n, n, R^6_{s_{n^*}s_{n^*}n}) - g(n). \quad (3.6)$$

여기에서 $\underline{n}$은 고려되는 대기용량의 한계치로서 다음의 관계식을 만족시키는 가장 적은 정수이다.

$$\frac{\Theta(s_n, n, R^6_{s_{n^*}s_{n^*}n})}{n} \le g(n) - g(n-1). \quad (3.7)$$

주어진 절차는 폐쇄대기 네트워크에 적용된 Shanthikumar and Yao[11]의 절차를 본 문제에 연장하여 적용한 것으로 $n \le \underline{n}$의 경우에는 식 (3.6)과 식 (3.7)의 절차에 의하여 최적해가 보증되며 $n > \underline{n}$에 있어서는 $\Theta(s_n, n, R^6_{s_{n^*}s_{n^*}n}) - g(n) \le \Theta(s_{\underline{n}}, \underline{n}, R^6_{s_{\underline{n}^*}s_{\underline{n}^*}\underline{n}}) - g(n) \le \Theta(s_{\underline{n}}, \underline{n}, R^6_{s_{\underline{n}^*}s_{\underline{n}^*}\underline{n}}) - g(\underline{n})$가 성립되어 최적해가 보증된다. 이의 증명은 오목성과 sublinearity를 적용한 다음의 일련의 부등식으로 정리될 수 있다.

$$\begin{aligned}
&\Theta(s_n, n, R^6_{s_{n^*}s_{n^*}n}) - \Theta(s_{\underline{n}}, \underline{n}, R^6_{s_{\underline{n}^*}s_{\underline{n}^*}\underline{n}}) \\
&\le \Theta(s_n, n, R^6_{s_{n^*}s_{n^*}n}) - \Theta(s_n, \underline{n}, R^6_{s_{n^*}s_{n^*}\underline{n}}) \\
&\le (n-\underline{n}) \frac{\Theta(s_n, \underline{n}, R^6_{s_{n^*}s_{n^*}\underline{n}})}{\underline{n}} \le (n-\underline{n}) \\
&\frac{\Theta(s_{\underline{n}}, \underline{n}, R^6_{s_{\underline{n}^*}s_{\underline{n}^*}\underline{n}})}{\underline{n}} \le (n-\underline{n})\{g(\underline{n}) - g(\underline{n}-1)\} \\
&\le g(n) - g(\underline{n}). \quad (3.8)
\end{aligned}$$

지금까지는 대기용량 n과 서버의 수 s에 대하여 생산율이 가장 크고 평균체재시간이 가장 짧은 서비스정책 $R^6_{s,s,n} = (0, 1, \cdots, s-1, n, 0)$에 대하여 식 (3.4)에서 식 (3.7)로 정식화되는 최적화절차가 제시되었다. 서비스정책 $R^6_{s,s,n}$가 적용되는 경우에 작업물의 평균체재시간과 보조서비스에서의 서비스능력의 제약조건이 만족되면 이는 주어진 배분 s와 n에서 실행이 가능한 최상의 서비스정책이 되어 최적화과정에 매우 편리하게 적용될 수 있다. 그러나 주서비스에서의 평균체재시간의 제약식이 만족되지 못하면 정리 1과 정리 3에 의하여 주어진 배분 s와 n에서는 어떠한 서비스정책도 실행이 가능하지 않음을 알 수 있다. 또한 평균체재시간의 제약식은 만족되나 보조서비스에서의 서비스능력이 만족되지 못하는 경우에는 주어진 배분 하에서 다양한 서비스정책에 대하여 최적화과정이 필요하다. 즉 서비스정책 $R_{m,s,n}$에서 $m = s, s-1, \cdots, 1$이 순차적으로 고려되고 주어진 m에 있어서는 서비스정책을 고려하는 복잡한 최적화절차가 필요하다. 이 경우 주어진 주서비스에서의 서버배분 m에 있어서 보조서비스에서의 서비스능력이 가장 큰 서비스정책 $R^3_{m,s,n} = (n-m, n-m+1, \cdots, n, s-m)$이 보조서비스에서의 서비스능력 제약식을 만족시키지 못하면 주어진 서버배분 m에 있어서 모든 서비스정책이 실행이 가능하지 않음 또한 알 수 있다.

그러므로 서비스정책 $R^6_{s,s,n}$를 적용하여 제시되었던 식 (3.4)에서 식 (3.7)을 대신하여 본래의 최적화문제인 식 (2.9)에 대한 최적화절차를 수정 제시한다. 이를 위하여 주어진 배분 s와 n에서 최적 서비스정책 $R^*_{s,n}$을 정의한다. 대기능력 n이 주어졌을 때 최적 총 서버의 수 s_n을 구하는 절차는 다음과 같이 수정된다.

$$\begin{aligned}
\Theta(s_n, n, R_{s_{n^*}n}{}^*) &= Max._{1 \le s \le n} \Theta(s, n, R_{s,n}{}^*) \\
&= Max._{1 \le s \le n} f\{\lambda[1 - p_{m,n,R^*}]\} - h(s). \quad (3.9)
\end{aligned}$$

최적 대기용량 n^*를 구하는 절차도 다음과 같이 수정 제시된다.

$$Max._{1 \le n \le \underline{n}} \Phi(s_n, n, R_{s_n n}^*) = \Theta(s_n, n, R_{s_n n}^*) - g(n), \quad (3.10)$$

여기에서 $\underline{n}$은 고려되는 대기용량의 한계치로서 다음의 관계식을 만족시키는 가장 적은 정수로서 식 (3.7)과 동일하게 적용된다.

$$\frac{\Theta(s_n, n, R_{s_n n}^*)}{n} \le g(n) - g(n-1). \quad (3.11)$$

여기에서 한계대기용량 $\underline{n}$는 제조시스템에 있어서 대기용량에 대한 평균수익보다 대기용량의 한계비용이 더 커지는 대기능력을 의미한다.

원 문제인 식 (3.9)에서 식 (3.11)의 최적화절차를 수행하기 위해서는 서비스정책 $R_{s,s,n}^6 = (0, 1, \cdots, s-1, n, 0)$을 적용하는 식 (3.5)에서 식 (3.7)의 최적화과정에서 배분 s와 n에 대한 최적 서비스정책 $R_{s,n}^*$을 구하는 절차가 필요하다. 그 과정을 요약하면 다음과 같다.

1. 서비스정책 $R_{s,s,n}^6$의 결과가 주서비스에서의 평균체재시간의 제약식 $W_{s,n,R^6} \le W$과 보조서비스에서 요구되는 서비스 능력의 제약식 $v_{s,n,R^6} \ge v$을 동시에 만족시키면 주어진 결과는 주어진 배분 s와 n에서 실행가능한 해가 되고 서비스정책 $R_{s,s,n}^6$가 최적 서비스정책 $R_{s,n}^*$가 되어 다음 단계인 서버배분 $s+1$로 간다. 그러나 평균체재시간의 제약식 $W_{s,n,R^6} \le W$를 만족시키지 못하면 제약식 $v_{s,n,R^6} \ge v$의 실행가능성과 상관없이 주어진 배분, s와 n은 실행불가능하고 서버 $s+1$로 간다.
2. 평균체재시간의 제약식 $W_{s,n,R^6} \le W$은 만족시키나 제약식 $v_{s,n,R^6} \ge v$을 만족하지 못하는 경우에는 주어진 배분 s와 n에 있어서의 모든 서비스정책 $R_{m,s,n} = (r_0, r_1, \cdots, r_m, s-m)$에 대하여 제약식 $W_{m,n,R} \le W$와 제약식 $v_{m,n,R} \ge v$, $m = s, s-1, \cdots$을 동시에 만족시키면서 가장 큰 목적함수 값을 갖는 최적 서비스정책 $R_{s,n}^*$를 도출하는 절차, 즉 $Max._{m,R} \Phi(s, n, R_{m,s,n})$, $s.t.$ $W_{m,n,R} \le W$, $v_{m,n,R} \ge v$의 해를 구하는 과정이 필요하다. 그러므로 그 첫 번째 단계로 $m = s$인 경우 모든 서비스정책 $R_{s,s,n} = (r_0, r_1, \cdots, n, 0)$ 중 보조서비스에서의 서버의 수의 기대치가 가장 큰 서비스정책 $R_{s,s,n}^3 = (n-s, n-s+1, \cdots, n, 0)$에서의 제약식 $v_{s,n,R^3} \ge v$와 제약식 $W_{s,n,R^3} \le W$의 실행가능여부가 검토된다.
   - 2-1. 제약식 $v_{s,n,R^3} \ge v$가 만족되면 주어진 주서비스에 배치가능한 서버의 수 s에서의 모든 서비스정책 $R_{s,s,n} = (r_0, r_1, \cdots, n, 0)$ 중 실행가능하고 목적함수가 가장 큰 서비스정책을 도출하는 절차가 필요하다. 2-3으로 간다.
   - 2-2. 제약식 $v_{s,n,R^3} \ge v$이 만족되지 못하는 경우에는 실행가능한 서비스정책이 존재하지 않으며 2-3으로 간다.
   - 2-3. $m = s-1$에서 $m = s$에서와 같은 절차를 반복한다. 다음으로는 $m = s-i$, $i = 2, 3, \cdots$에 대하여 순차적으로 주어진 절차를 수행한다. 그러나 m이 적어질수록 보조서비스에서의 서버 수의 기대치는 증가하고 생산율과 평균체재시간은 감소하여 실행가능성이 제한되어 서버배분 s에서의 알고리즘은 종료되고 $s+1$로 간다.

이제 언급된 바와 같이 서버의 수 s, 대기용량 n, 그리고 주서비스에 배치가 가능한 서버가 m, $m = s, s-1, s-2, \cdots$,인 경우에 있어서 모든 서비스정책 $R_{m,s,n} = (r_0, r_1, \cdots, r_m, s-m)$중 실행가능하고 목적함수가 가장 큰 서비스정책을 도출하는 절차를 보자. 평균체재시간이 가장 짧고 보조서비스에서의 서버의 수의 기대치가 가장 적은 서비스정책 $R_{m,s,n}^4 = (0, 1, \cdots, m-1, n, s-m)$에서 제약식 $W_{m,n,R^4} \le W$가 만족되지 않으면 다음의 모든 서비스정책은 실행이 불가능하므로 $s+1$로 간다. 그러나 제약식 $W_{m,n,R^4} \le W$가 만족되나 $v_{m,n,R^4} \ge v$를 만족시키지 못하는 경우에는 보조서비스에서의 서버의 수

의 기대치가 가장 큰 서비스정책 $R^3_{m,s,n}=(n-m, n-m+1, \cdots, n, s-m)$에서는 제약식 $v_{m,n,R^3} \geq v$의 실행가능성이 확인되고 실행 가능한 경우에는 모든 서비스정책 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$ 중에서 실행가능하면서 목적함수가 가장 큰 서비스정책을 탐색(search)하는 과정 즉, $Max \cdot_R \Phi(s,n,R_{m,s,n})$, $s.t.\ W_{m,n,R} \leq W,\ v_{m,n,R} \geq v$의 해를 구하는 절차가 필요하다.

주어진 최적의 서비스정책을 구하는 유사한 탐색법으로는 Berman et al.[2]과 Terekhov and Beck [12] 등이 있다. 그러나 Berman et al.[2]과 Terekhov and Beck[12]의 휴리스틱은 최적해를 보장하지 못할 뿐더러 여기에서는 적절한 접근방법이 아니다. 그러므로 정리 1의 결과를 이용하여 내재열거(implicit enumeration)를 수행한다. 이를 설명하면 m이 주어지면 r_0에서 r_{m-1}까지의 m개의 순환루프(loop)를 형성하고 주어진 서비스정책이 실행가능하면 목적함수의 값을 확인하고 서비스정책 $R_{s,n}$*를 비교 갱신하고 모든 순환을 마치면 $m-1$로 간다. 가장 외부 루프가 r_0이며 가장 내부루프는r_{m-1}이 된다. 가장 외부루프, $r_0=0, \cdots, n-m$의 값을 가지며 다음의 내부루프, $r_1=r_0+1, \cdots, n-m+1$, 그리고 $i+1$번째 내부루프, $r_i=r_{i-1}+1, \cdots, n-m+i$, $i=2, \cdots, m-1$로 구성된다. 주어진 순환과정 중 서비스정책 $R_{m,s,n}=(r_0, r_1, \cdots, r_m, s-m)$에서 평균체재시간의 제약식 $W_{m,n,R} \leq W$가 실행가능하지 않으면 주어진 서비스정책보다 열등한(dominated) 서비스정책은 순환과정에서 제거된다. 여기에서 열등한 서비스정책이란 서비스정책 $R^7_{m,s,n}=(r_0', r_1', \cdots, r_m', s-m)$, $r_i \leq r_i'$, $i=1, \cdots, m-1$으로 표현될 수 있으며 서비스정책 $R_{m,s,n}$보다 평균체재시간이 큰 모든 서비스정책을 의미한다. 예를 들어 $n=6$, $s=5$, $m=3$인 경우 서비스정책 $R^4_{3,5,6}=(0, 1, 2, 6, 2)$에서 평균체재시간이 만족되었으나 순환과정 중 그 다음에 고려되는 서비스정책 $R_{3,5,6}=(0, 1, 3, 6, 2)$에서 평균체재시간을 만족시키지 못하면 서비스정책 $R^4_{3,5,6}$을 제외한 모든 서비스정책은 열등한 서비스정책이 되어 평균체재시간을 만족시키지 못하여 $m=3$에서는 더 이상 고려할 서비스정책이 없고 순환과정은 종료되고 $m=2$에서의 새로운 순환과정을 시작함을 의미한다. 제시된 내재열거법은 최적의 서비스정책을 확인하기 위해 복잡성에 대한 고려가 감소하였음에도 불구하고 실제 계산이 수행되는 서비스정책은 매우 제한되어 알고리즘의 유용성을 확인할 수 있다.

4. 수치 예

본 절에서는 수치 예를 제시한다. 주서비스에 도착하는 수요는 도착률 $\lambda=6$인 포아송분포에 의하며 서버는 서비스율 $\mu=2$인 지수분포를 갖는다. 그러므로 주서비스에 부여된 부하 $\rho=3$이 된다. 생산율 $\theta_{m,n,R}$에 대한 수익함수 $f(\theta_{m,n,R})=2\theta_{m,n,R}$, 서버의 수 s에 대한 비용함수 $h(s)=s^{7/6}$, 그리고 대기용량 n의 비용함수 $g(n)=0.32n^{5/4}$을 가정한다. 수익함수 $f(\theta_{m,n,R})$는 s에 대하여 선형(linear)으로 광의의 오목함수이며 서버와 대기용량의 비용함수 $h(s)$와 $g(n)$은 각각 증가하는 볼록함수이다. 주서비스는 평균체재시간 $W=2.0$을, 보조서비스는 서비스율 $v=2$를 제약식으로 갖는다.

먼저 식 (3.9)에 있어서 대기용량 n이 주어졌을 때 최적 서버의 수 s_n을 구하는 절차를 보자. 대기용량 $n=6$에서 서버의 수 $s=1, \cdots, 6$에 대하여 최적 서비스정책 $R_{s,n}$*을 구하고 그 결과로 최적 서버의 수 s_n을 구하는 수치 예를 제시한다. <표 1>은 서버의 수 $s=1, \cdots, 6$에 대하여 최적 서버 수 s_6와 최적 서비스정책 $R_{s_6 6}$*를 구하는 과정을 정리한 것으로 '가' 또는 '불가'는 주어진 제약식이 만족되거나 또는 만족되지 못하여 주어진 해가 실행이 가능하거나 불가능함을 의미한다.

그러므로 서버의 수 $s=1, 2, 3$에서는 실행이 가능한 서비스정책이 존재하지 않고 $s=4, 5, 6$에서는 두 개의 제약식 $W_{5,6,R^*}=0.5110054 \leq 2.0$과 $v_{5,6,R^*}=2.185827 \geq 2.0$이 모두 만족되어 실행이 가능한 서

〈표 1〉 최적 서버 수 탐색 알고리즘($n=6$)

s	m	$R_{s,6}$	$\Theta(s, 6, R)$	$W_{m,6,R}$	(가, 부)	$v_{m,6,R}$	(가, 부)
1	1	(0.6, 0)	불가	2.754	불가		
2	2	(0, 1, 6, 0)	불가	1.173	가	0.079	불가
		(4, 5, 6, 0)				0.566	불가
3	3	(0, 1, 2, 6, 0)	불가	0.705	가	0.509	불가
		(3, 4, 5, 6, 0)				1.038	불가
	2	(0, 1, 6, 1)		1.173	가	1.079	불가
		(4, 5, 6, 1)				1.588	불가
	1	(0, 6, 2)		2.754	불가		
4	4	(0, 1, 2, 3, 6, 0)		0.556	가	1.274	불가
		(2, 3, 4, 5, 6, 0)				1.618	불가
	3	(0, 1, 2, 6, 1)		0.705	가	1.509	불가
		(3, 4, 5, 6, 1)				2.038	가
		(0, 1, 3, 6, 1)		1.496	가	1.570	불가
		(0, 1, 4, 6, 1)		2.422	불가		
		(0, 2, 3, 6, 1)		4.080	불가		
	2	(0, 1, 6, 2)	2.645	1.173	가	2.079	가
5	5	(0, 1, 2, 3, 4, 6, 0)	4.718	0.511	가	2.186	가
6	6	(0, 1, 2, 3, 4, 5, 6, 0)	3.286	0.5	가	3.156	가

비스정책이 존재하며 최적 서버의 수 $s_6=5$, 그리고 최적 서비스정책 $R_{5,6}{}^*=(0, 1, 2, 3, 4, 6, 0)$이다. 그러므로 주서비스의 수익 $\Theta_{5,6,R^*}=4.71839$이다. 최적해에서 보조서비스의 서비스율 $v=2.0$을 초과하는 양 0.186은 서버 수 s가 정수임에 기인하며 전술적으로 다양한 방법으로 활용될 수 있을 것이다. 제시된 알고리즘은 아주 적은 서비스정책을 열거함으로써 매우 용이하게 최적의 서비스정책에 도달할 수 있어 주어진 알고리즘이 매우 효율적임을 알 수 있다.

<표 2>에는 몇 가지 서로 다른 제약조건에서 <표 1>에서와 같이 최적 서버의 수 s_8을 구하는 최적화 과정을 제시하고 있다. 주서비스의 대기용량 $n=8$로 총 서버의 수 $s=1,\cdots,8$에 대하여 서비스정책과 수행도를 보여주고 있다.

주어진 결과에 의하면 서버의 수 s가 늘어날수록 주서비스에서의 평균대기시간 $W_{m,8,R}$은 감소하나 보조서비스에서의 서버의 기대치 $v_{m,8,R}$은 증가함을 알 수 있다. 결과는 제약식이 없는 경우에는 $s_8=3$, $R_{3,8}{}^*=(0, 1, 2, 8, 0)$, $\Theta(3, 8, R_{3,8,}{}^*)=6.876$, $W=1.0$, $\nu=1.0$인 경우에는 $s_8=4$, $R_{4,8}{}^*=(0, 1, 2, 3, 8, 0)$, $\Theta(4, 8, R_{4,8}{}^*)=6.410$, $W=2.0$, $v=2.0$인 경우에는 $s_8=5$, $R_{5,8}{}^*=(0, 1, 2, 3, 4, 8, 0)$, $\Theta(5, 8, R_{5,8}{}^*)=$ $=5.209$, 그리고 $W=3.0$, $v=3.0$인 경우에는 $s_8=6$, $R_{6,8}{}^*=(0, 1, 2, 3, 4, 5, 8, 0)$, $\Theta(6, 8, R_{6,8}{}^*)=3.761$이 되어 제약식의 우변이 커질수록 보조서비스의 서버의 수의 기대치를 만족시키기 위해 최적 서버의 수 s가 늘어나고 그 결과로 수익함수의 값은 적어짐을 알 수 있다. 제약식이 없는 경우에는 설계모수인 서버의 수 s에 대하여 수익함수 $\Theta(s, 8, R)$가 오목함수임을 수치적 결과가 보여주고 있으며 제약식이 있는 경우에도 오목함수의 특성을 만족시켜 적용되는 한계분석법이 최적화알고리즘으로 적절함을 알 수 있다. 제약식의 우변이 커질수록 서버의 수 s가

〈표 2〉 최적 서버 수 탐색 알고리즘($(n=8)$

제약식 \ 서버의 수(s)		1	2	3	4	5	6	7	8
$W=0$ $v=0$	$R_{s,8}$	(0, 8, 0)	(0, 1, 8, 0)	(0, 1, 2, 8, 0)	(0, 1, 2, 3, 8, 0)	(0, 1, 2, 3, 4, 8, 0)	(0, 1, 2, 3, 4, 5, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 7, 8, 0)
	m	1	2	3	4	5	6	7	8
	$\Theta(s,8,R)$	3.000	5.621	6.876	6.410	5.209	3.761	2.207	0.589
	$W_{m,8,R}$	3.751	1.597	0.863	0.612	0.533	0.508	0.502	0.500
	$v_{m,8,R}$	0.000	0.034	0.380	1.138	2.063	3.038	4.028	5.024
$W=1.0$ $\nu=1.0$	$R_{s,8}$	불가	불가	불가	(0, 1, 2, 3, 8, 0)	(0, 1, 2, 3, 4, 8, 0)	(0, 1, 2, 3, 4, 5, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 7, 8, 0)
	m				4	5	6	7	8
	$\Theta(s,8,R)$				6.410	5.209	3.761	2.207	0.589
	$W_{m,8,R}$				0.612	0.533	0.508	0.502	0.500
	$v_{m,8,R}$				1.138	2.063	3.038	4.028	5.024
$W=2.0$ $v=2.0$	$R_{s,8}$	불가	불가	불가	(0, 1, 8, 2)	(0, 1, 2, 3, 4, 8, 0)	(0, 1, 2, 3, 4, 5, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 7, 8, 0)
	m				2	5	6	7	8
	$\Theta(s,8,R)$				2.826	5.209	3.761	2.207	0.589
	$W_{m,8,R}$				1.597	0.533	0.508	0.502	0.500
	$v_{m,8,R}$				2.034	2.063	3.038	4.028	5.024
$W=3.0$ $v=3.0$	$R_{s,8}$	불가	불가	불가	불가	(0, 1, 8, 3)	(0, 1, 2, 3, 4, 5, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 8, 0)	(0, 1, 2, 3, 4, 5, 6, 7, 8, 0)
	m					2	6	7	8
	$\Theta(s,8,R)$					1.327	3.761	2.207	0.589
	$W_{m,8,R}$					1.597	0.508	0.502	0.500
	$v_{m,8,R}$					3.034	3.038	4.028	5.024

적은 경우에 실행이 불가능해지거나 보조서비스 전담서버가 존재하는 결과를 갖기도 하나 보조서비스의 전담서버는 최적화의 관점에서 바람직한 결과가 아님을 알 수 있다. 특히 거의 모든 경우 서비스정책 $R^6_{s,s,8}=(0,1,\cdots,s-1,8,0)$이 적용되어 최적화과정에서 매우 중요하게 적용됨을 알 수 있다.

<표 3>에서는 식 (3.7)의 최적 대기능력 n^*를 구하는 과정을 정리한 것이다. 대기용량을 $n=8$까지만 제시한 것은 $W=2$, $v=2$인 경우에 얻어지는 한계대기용량 $\underline{n}$를 적용한 것이다.

제약식의 우변이 $W=0$, $v=0$인 경우에는 $n^*=5$, $s_4=3$, $R_{3,5}{}^*=(0,1,2,5,0)$, $\Phi(3,5,R_{3,5}{}^*)=3.55$, $W=1.0$, $v=1.0$인 경우에는 $n^*=5$, $s_5=4$, $R_{4,5}{}^*=(0,1,2,3,5,0)$, $\Phi(4,5,R_{4,5}{}^*)=2.96$, $W=2.0$, $v=2.0$인 경우에는 $n^*=5$, $s_5=5$, $R_{5,5}{}^*=(0,1,2,3,4,5,0)$, $\Phi(5,5,R_{5,5}{}^*)=1.748$, 그리고 $W=3.0$, $v=3.0$인 경우에는 $n^*=6$, $s_6=6$, $R_{6.6}{}^*=(0,1,2,3,4,5,6,0)$, $\Phi(6,6,R_{6,6}{}^*)=0.281$이 되어 제약식의 우변이 커질수록 보조서비스의 서버의 수의 기대치를 만족시키기 위해 최적 서버의 수 s가 늘어나나 목적함수, $\Phi(s_n,n,R_{s_n n}{}^*)$의 값은 서버에 대한 비용부담으로 감소함을 알 수 있다. 또한 대기능력 n이나 서버의 수 s가

〈표 3〉 대기능력과 수행도

제약식 \ 대기능력(n)		1	2	3	4	5	6	7	8
$W=0$ $\upsilon=0$	s_n	1	2	2	3	3	3	3	3
	$R_{s_n n}{}^*$	(0, 1, 0)	(0, 1, 2, 0)	(0, 1, 3, 0)	(0, 1, 2, 4, 0)	(0, 1, 2, 5, 0)	(0, 1, 2, 6, 0)	(0, 1, 2, 7, 0)	(0, 1, 2, 8, 0)
	m	1	2	2	3	3	3	3	3
	$\Theta(s_n, n, R_{s_n n}{}^*)$	2	3.402	4.444	5.311	5.943	6.359	6.655	6.878
	$g(n)$	0.32	0.761	1.263	1.810	2.393	3.005	3.644	4.305
	$\Phi(s_n, n, R_{s_n n}{}^*)$	1.68	2.641	3.181	3.501	3.55	3.354	3.011	2.573
	W_{m,n,R^*}	0.5	0.5	0.632	0.558	0.629	0.705	0.783	0.863
	υ_{m,n,R^*}	0.25	0.588	0.328	0.771	0.614	0.509	0.435	0.380
$W=1.0$ $\nu=1.0$	s_n	불가	2	3	4	4	4	4	4
	$R_{s_n n}{}^*$		(0, 2, 1)	(0, 1, 2, 3, 0)	(0, 1, 2, 3, 4, 0)	(0, 1, 2, 3, 5, 0)	(0, 1, 2, 3, 6, 0)	(0, 1, 2, 3, 7, 0)	(0, 1, 2 3, 8, 0)
	m		1	3	4	4	4	4	4
	$\Theta(s_n, n, R_{s_n n}{}^*)$		1.447	4.243	4.487	5.353	5.865	6.192	6.410
	$g(n)$		0.761	1.263	1.810	2.393	3.005	3.644	4.305
	$\Phi(s_n, n, R_{s_n n}{}^*)$		0.686	2.98	2.677	2.96	2.86	2.548	2.105
	W_{m,n,R^*}		0.875	0.5	0.5	0.526	0.556	0.585	0.612
	υ_{m,n,R^*}		1.077	1.038	1.618	1.402	1.274	1.192	1.138
$W=2.0$ $\upsilon=2.0$	s_n	불가	불가	3	4	5	5	5	5
	$R_{s_n n}{}^*$			(0, 3, 2)	(0, 1, 4, 2)	(0, 1, 2, 3, 4, 5, 0)	(0, 1, 2, 3, 4, 6, 0)	(0, 1, 2, 3 4, 7, 0)	(0, 1, 2, 3, 4, 8, 0)
	m			1	2	5	5	5	5
	$\Theta(s_n, n, R_{s_n n}{}^*)$			0.297	2.172	4.141	4.718	5.032	5.209
	$g(n)$			1.263	1.810	2.393	3.005	3.644	4.305
	$\Phi(s_n, n, R_{s_n n}{}^*)$			-0.97	0.362	1.748	1.713	1.388	0.904
	W_{m,n,R^*}			1.308	0.795	0.5	0.511	0.523	0.533
	υ_{m,n,R^*}			2.025	2.197	2.330	2.186	2.108	2.063
$W=3.0$ $\upsilon=3.0$	s_n	불가	불가	불가	4	5	6	6	6
	$R_{s_n n}{}^*$				(0, 4, 3)	(0, 1, 5, 3)	(0, 1, 2, 3, 4, 5, 6, 0)	(0, 1, 2, 3, 4, 5, 7, 0)	(0, 1, 2, 3, 4, 5, 8, 0)
	m				1	2	6	6	6
	$\Theta(s_n, n, R_{s_n n}{}^*)$				-1.07	0.969	3.286	3.607	3.761
	$g(n)$				1.810	2.393	3.005	3.644	4.305
	$\Phi(s_n, n, R_{s_n n}{}^*)$				-2.88	-1.42	0.281	0.037	-0.54
	W_{m,n,R^*}				1.775	0.977	0.5	0.504	0.508
	υ_{m,n,R^*}				3.008	3.123	3.156	3.076	3.038

정수이므로 제약식이 $v = v_{m,n,R^*}$이 만족되지 못하고 v_{m,n,R^*}의 값이 v를 더 초과할수록 목적함수, $\Phi(s_n, n, R_{s_n}{}^*)$의 값이 최적 값보다 적어짐을 알 수 있다. 그러므로 보조서비스에서의 기대 서버 수 v_{m,n,R^*}이 v와 더 큰 차이를 갖는 특수한 경우를 제외하고는 주어진 수치적 결과는 거의 모든 경우에 설계모수에 대한 목적함수의 오목성이 유지되어 한계분석법이 적절한 알고리즘임을 알 수 있다. 또한 얻어진 결과에 의하면 제약식의 우변의 값이 크고 서버의 수 s가 적은 특수한 경우에는 서비스정책 $R^4_{m,s,n} = (0, 1, \cdots, m-1, n, s-m)$가 적용되나 그렇지 않은 경우에는 대부분 서비스정책 $R^6_{s,s,n} = (0, 1, \cdots, s-1, n, 0)$가 가장 높은 수행도를 갖는 서비스정책으로 판명되어 주어진 예에서는 교차훈련된 서버가 보조서비스의 전담서버로 존재하는 것보다는 모두에서 작업을 수행하여 유연성을 확보하는 것이 더 좋은 서비스정책임을 제시하며 그 결과 서비스정책 $R^6_{s,s,n}$가 주어진 최적화과정에서 매우 유용함을 알 수 있다.

5. 결 론

서비스 메커니즘을 제공하는 시스템은 대기시스템으로 모형화되고 서비스능력이 설계되거나 수행도가 산정된다. 그러나 실제 존재하는 많은 시스템은 대기시스템이 제공하는 서비스 외에도 부수적인 서비스기능을 갖고 있어 일반적으로 다루어지는 것과 같이 서비스능력을 설계하는 전략적 접근만으로는 곤란하다. 그러므로 본 논문에서는 주어진 최적화문제를 좀 더 현실적으로 접근하여 주서비스와 보조서비스의 두 종류의 기능을 갖는 시스템을 다루었으며 높은 자본비용을 수반하는 전략적 의사결정문제인 서비스능력과 대기용량을 설계하는 문제와 전술적 의사결정문제인 서비스정책을 동시에 설계하는 최적화문제를 다룸으로써 전략적 의사결정과 전술적 의사결정의 통합적 최적화를 추구하였다.

주어진 문제는 계층적 구조를 갖는 문제로서 일반적으로는 전략적 의사결정문제인 서비스능력을 설계하는 최적화문제가 먼저 다루어지고 다음에 전술적 문제인 서비스정책을 결정하는 문제가 다루어진다. 그러나 이러한 계층적 접근은 통합적인 의미에서 최적의 결과를 제공하지 못한다. 반면에 본 논문에서는 기능의 융합을 통하여 계층적으로 주어진 두 문제를 통합적으로 최적화를 추구하였으며 이러한 접근방법은 단순히 서비스능력을 설계하거나 서비스정책을 설계하는 문제보다 더 복잡하고 어렵다.

최적화절차를 수립하기 위하여 설계모수인 서비스능력, 대기용량, 그리고 서비스정책에 대하여 주서비스의 수행도에 영향을 주는 유용한 특성들이 제시되었고 이에 기초하여 최적화알고리즘이 제시되었으며 서비스정책 $R^6_{s,s,n} = (0, 1, \cdots, s-1, n, 0)$는 주어진 최적화과정에 매우 유용하게 적용되었다. 최적화결과에 대한 수치 예로써 주어진 최적화절차에 대한 이해를 도모하였으며 제시된 최적화절차는 매우 효율적으로 적용될 수 있음을 알 수 있다. 주어진 예의 경우 보조서비스에 전담서버를 배치하는 것보다 교차훈련된 서버의 유연성을 확보하는 서비스정책이 더 효율적임 또한 알 수 있다.

주어진 최적화문제는 서비스 시스템, 제조시스템, 통신시스템, 그리고 설비보전 시스템 등에 다양하게 적용될 수 있다. 서비스 시스템의 경우를 보면 주서비스가 수행되는 매장에서는 고객에 제품을 판매하고 보조서비스로는 상품을 입출고하고 분류정리한다. 효율적인 서비스를 제공하기 위하여 매장에 고객이 많은 경우에는 종업원을 많이 배치하고 적을 때에는 적게 배치하여 매장에 배치되지 않은 서버는 창고에서 재고와 관련된 업무를 수행한다. 제조시스템에서 주서비스는 제조가 보조서비스는 제조에 소요되는 원료, 부품, 그리고 설비를 담당하는 창고업무를, 통신시스템의 저장전송의 경우에는 주서비스는 통신업무를 보조서비스는 여타의 정보처리를, 그리고 설비보전 시스템의 경우 주서비

스는 고장 난 설비의 교정보전을 보조서비스는 예방보전의 경우를 예로 들 수 있다. 그러므로 두 종류의 서비스기능을 갖는 시스템의 최적화절차는 다양한 시스템에 매우 유용하게 적용이 가능하다.

참 고 문 헌

[1] Behret H. and A. Korugan, "Performance Analysis of A Hybrid System under Quality Impact of Returns," *Computers and Industrial Engineering*, Vol.56(2009), pp.507-520.

[2] Berman, O. and R.C. Larson, "A Queueing Control Model for Retail Services Having Back Rooms Operations and Crossed Trained Workers," *Computers and Operations Research*, Vol.31(2004), pp.201-222.

[3] Berman, O., J. Wang, and K.P. Sapna, "Optimal Management of Cross-trained Workers in Service with Negligible Switching Costs," *European Journal of Operational Research*, Vol.167(2005), pp.349-369.

[4] Buzacott, J.A. and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems*, Prentice Hall, 1993.

[5] Fox, B., "Discrete Optimization via Marginal Analysis," *Management science*, Vol.13(1966), pp.210-216.

[6] Gordon, W. and G. Newell, "Closed Queueing Networks with Exponential Servers," *Operations Research*, Vol.15(1967), pp.252-267.

[7] Kelly, F.P., *Reversibility and Stochastic Networks*, John Wiley and Sons Ltd., 1979.

[8] Little, J.D.C., "A Proof for the Queueing Formula : $L=\lambda W$," *Operations Research,* Vol.9 (1961), pp.383-387.

[9] Ross, S., *Stochastic Processes*, Wiley, New York, 1983.

[10] Savaskan, R.C., "Closed-Loop Supply Chain Models with Product Remanufacturing," *Management Science*, Vol.50(2004), pp.239-252.

[11] Shanthikumar, J.G. and D.D. Yao, "Optimal Server Allocation in A System of Multi-Server Stations," *Management Science*, Vol.33 (1987), pp.1173-1180.

[12] Terekhov, D. and J.C. Beck, "A Constraint Programming Approach for Solving a Queueing Control Problem," *Journal of Artificial Intelligence Research,* Vol.32(2008), pp.123-167.