

# Routing 문제의 유한구간제어

■ 한수희

(서울대학교 공과대학 기계·항공공학부)

## 1. 서론

지난 호에서는 마코프 의사 결정 모델에서 최적화를 하는 방법에 대해 일반적으로 소개했다. 특히 연속형 시스템에 대한 선형 이차 최적 제어를 예로 들어 가격함수와 제어 반복법에 대해 설명했다. 이번 호에서는 마코프 의사 결정 모델의 대표적인 예로 컴퓨터, 통신, 제조 시스템 등에 흔히 나타나는 대기행렬 시스템에 최적화가 어떻게 적용되는지 알아본다. 특히 도착하는 고객이 다수의 서버를 두고 어디에서 서비스를 받을지 결정(제어)하는 routing 문제를 대상으로 한다. 주어진 가격함수를 최소화 하는 제어를 어떻게 결정할지 알아본다. 최적제어의 조건식을 유도하고 유한구간제어를 통해서 최적제어에 근사화된 제어를 구해본다. 본고의 routing 문제 설정은 [1]을 주로 참고했고, 유한구간제어에 대한 시뮬레이션 방법은 저자가 제안했다.

## 2. Routing 모델

그림 1과 같은 대기행렬 시스템을 생각하자. 고객은 큐 1 또는 큐 2를 선택해서 서비스를 받을 수 있다. 이런 큐의 선택을

routing 이라고 한다. routing의 가장 큰 목적은 공용으로 사용하는 한정된 자원(여기서는 두 개의 시스템)을 효율적으로 사용하는데 있다.

가장 간단한 경우를 생각해 보자. 고객들이 평균적으로 단위 시간당  $\lambda$ 명씩 포아송 분포를 따르며 도착한다. 도착한 고객은 평균적으로 단위시간당  $\mu_1$ ,  $\mu_2$ 명씩 지수 분포를 따르며 서비스를 제공하는 큐 중에서 하나를 선택한다. 일반적으로  $\mu_1 \neq \mu_2$ 이다. 고객의 도착 시간과 두 개의 서비스 과정은 독립이라고 하고, 고객이 도착할 때마다 두 개의 큐 길이는 알고 있다고 가정 하자.

지금부터는 위의 큐 모델을 마코프 의사 결정 모델로 묘사해 보자. 우리의 관심사항은 두 큐의 길이이기 때문에 상태 공간을  $X = (i, j) : i, j = 0, 1, \dots$  이라고 생각할 수 있다.  $i$ 와  $j$ 는 큐 1과 큐 2의 길이를 각각 나타낸다. 주어진 상태  $X(i, j)$ 에서 새로 도착한 고객을 어떤 큐로 보낼 것인가에 대한 제어  $u$ 가 정해지면 초기 조건  $(i, j)$ 에 대해 아래의 가격함수도 결정된다.

$$V(i, j) = E\left[\int_0^\infty e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) dt\right] \quad (1)$$

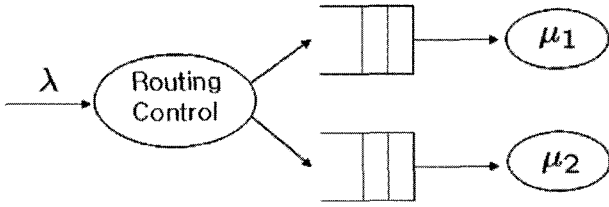


그림 1. 두 개의 시스템 사이의 라우팅 문제

여기서  $B_1$  과  $B_2$ 는 각각 큐 1과 큐 2에서의 고객에 대한 단위 시간당 비용이고  $\beta$ 는 주어진 감쇠 인자이다. 또한  $X_i(t)$ 는 시간  $t$ 에서의 큐  $i$ 의 길이를 나타낸다.

분석을 용이하게 하기 위해서 연속형 마코프 사슬 모델에 바탕을 둔 비용함수 (1)을 이산형 모델로 바꾸어 보도록 하자. 바꿀 때는 연속형에서의 어떤 정보도 잃지 않도록 근사화와 같은 방법은 사용하지 않는다.

상태 변화가 일어나는, 즉 새로운 고객이 도착하는 시간을  $T_0, T_1, \dots, T_k, \dots$ 라고 하면 비용 함수 (1)의 적분 구간을 나누어 아래와 같이 쓸 수 있다.

$$E\left[\int_0^\infty e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) dt\right] = \sum_{k=0}^{\infty} E\left[\int_{T_k}^{T_{k+1}} e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) dt\right] \quad (2)$$

$T_k$ 에서 새로운 상태  $X_k$ 가 되면(새로운 고객이 도착하면), 다음 상태의 변화가 있는  $T_{k+1}$ 까지는  $X_k$ 로 유지가 되므로,  $T_k \leq t < T_{k+1}$ 에서 다음과 같이 된다.

$$e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) = e^{-\beta t} (B_1 X_1(T_k) + B_2 X_2(T_k)) \quad (3)$$

수식 (2)에 (3)을 대입하면, 비용 함수는 아래와 같이 표현된다.

$$E\left[\int_0^\infty e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) dt\right] = \sum_{k=0}^{\infty} E\left[\int_{T_k}^{T_{k+1}} e^{-\beta t} dt\right] E[B_1 X_1(T_k) + B_2 X_2(T_k)] \quad (4)$$

수식 (4) 우변의 첫 번째 적분식의 기대값은 다음과 같이 계산된다.

$$E\left[\int_{T_k}^{T_{k+1}} e^{-\beta t} dt\right] = -\frac{1}{\beta} (E[e^{-\beta T_{k+1}}] - E[e^{-\beta T_k}]) \quad (5)$$

$k$ 번째 상태 변환 후 상태 유지 시간을  $V_{k+1}$ 이라고 하면  $T_{k+1} = T_k + V_{k+1}$ 이라고 쓸 수 있고, 수식 (5)는 아래와 같이 쓸 수 있다.

$$\begin{aligned} E\left[\int_{T_k}^{T_{k+1}} e^{-\beta t} dt\right] &= -\frac{1}{\beta} (E[e^{-\beta T_{k+1}}] - E[e^{-\beta T_k}]) \\ &= \frac{1}{\beta} E[e^{-\beta T_k}] (1 - E[e^{-\beta V_{k+1}}]) \end{aligned} \quad (6)$$

여기에서  $V_{k+1}$ 은  $T_k$ 와 독립이라는 사실을 이용했다. 확률변수  $V_{k+1}$ 가 평균이  $\frac{1}{\lambda}$ 인 지수 분포를 이룬다면 아래와 같이 계산된다.

$$E[e^{-\beta V_{k+1}}] = \int_0^\infty e^{-\beta t} \lambda e^{-\lambda t} dt = \frac{\lambda}{\beta + \lambda} \quad (7)$$

$T_k = V_1 + \dots + V_k$ 이고, 상태 유지 시간끼리는 상호 독립이라고 하면,  $E[e^{-\beta T_k}]$ 는 아래와 같이 계산된다.

$$\begin{aligned} E[e^{-\beta T_k}] &= E[e^{-\beta V_1}] E[e^{-\beta V_2}] \dots E[e^{-\beta V_k}] \\ &= \left(\frac{\lambda}{\beta + \lambda}\right)^k \end{aligned} \quad (8)$$

다음과 같은 치환을 하고

$$\alpha = \frac{\lambda}{\beta + \lambda}$$

수식 (7)과 (8)를 (6)에 대입하면 비용 함수 (1)은 최종적으로 다음과 같이 표시된다.

$$\begin{aligned} E\left[\int_0^\infty e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) dt\right] &= \sum_{k=0}^{\infty} \frac{\alpha^k (1 - \alpha)}{\beta} E[(B_1 X_1(T_k) + B_2 X_2(T_k))] \\ &= \frac{\lambda}{\beta + \lambda} \sum_{k=0}^{\infty} \alpha^k E[(B_1 X_1(T_k) + B_2 X_2(T_k))] \end{aligned} \quad (9)$$

다음 장에서는 비용함수 (9)를 최소화 하는 제어를 구하는 것에 대하여 논한다.

### 3. 최적해

그림 2에서는 상태  $(i, j)$ 에서 고객의 도착과 출발에 따른 모

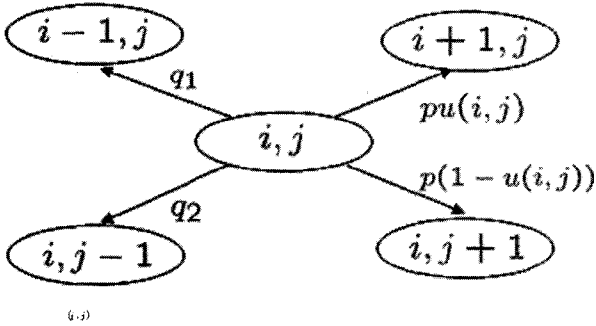


그림 2. 상태  $(i, j)$ 에서의 상태 전이 확률

든 가능한 상태 전이를 다음과 같은 제어  $u(i, j)$ 와 함께 나타내고 있다.

$$u(i, j) = \begin{cases} 1 & \text{if 도착한 고객이 큐 1로 갈 때} \\ 0 & \text{if 도착한 고객이 큐 2로 갈 때} \end{cases}$$

그림 2에 나타난 확률  $p, q_1, q_2$ 는  $\lambda, \mu_1, \mu_2$ 로부터 아래와 같이 구할 수 있다.

$$p = \frac{\lambda}{\gamma}, \quad q_1 = \frac{\mu_1}{\gamma}, \quad q_2 = \frac{\mu_2}{\gamma},$$

여기서  $\gamma = \lambda + \mu_1 + \mu_2$ 이다.

$u(i, j)$ 가 1인 경우 상태  $(i, j)$ 에서 새로 도착한 고객을 큐 1로  $p$ 라는 확률로 보내고, 큐 2로 보낼 확률은 0이 된다. 상태  $(0, j)$ 와  $(i, 0)$ 에서는 큐 1과 큐 2에서 각각 출발할 수 있는 고객이 없으므로, 확률  $q_1$ 과  $q_2$ 로 자신의 상태에 머무르게 된다.

지금부터는 세 가지 과정 (Step I, II, III)을 거쳐서 주어진 비용 함수 (9)에 대한 최적제어를 구해보자.

### 1 Step I

동적 프로그래밍을 이용해서 비용함수 (1)에 최적화 조건식을 다음과 같이 세울 수 있다.

$$V(i, j) = \min(b_1 i + b_2 j + \alpha p u(i, j) V(i+1, j) + \alpha p [1 - u(i, j)] V(i, j+1) + \alpha q_1 V([i-1]^+, j) + \alpha q_2 V(i, [j-1]^+)) \quad (10)$$

여기서  $b_1$ 과  $b_2$ 는 아래처럼 주어진다.

$$b_1 = \frac{B_1}{\beta + \gamma}, \quad b_2 = \frac{B_2}{\beta + \gamma}$$

또한,  $[i-1]^+$ 은  $i \geq 1$ 에서는  $i-1$ 이고,  $i=0$ 에서는 0을 의미한다. 이런 수학적 표현은 상태  $(0, j)$ 와  $(i, 0)$ 에서 자신의 상태 편이를 표현하기에 적합하다.

$u(i, j)$ 와 관련된 항만 분리하면 최적화 조건식 (10)은 다음과 같이 쓸 수 있다.

$$V(i, j) = b_1 i + b_2 j + \alpha q_1 V([i-1]^+, j) + \alpha q_2 V(i, [j-1]^+) + \alpha p \min[V(i+1, j), V(i, j+1)] \quad (11)$$

여기서  $\min[a, b]$ 는  $a$ 와  $b$  중 작은 수를 가리킨다.

다음 과정에서는 여기에서 소개한 최적화 조건식을 만족하는 최적제어에 대해서 소개한다.

### 2 Step II

최적화 조건식 (11)에서  $\min$ 과 관련된 항에서는 다음과 같이 두 가지 경우가 있다.

- 1)  $V(i+1, j) \leq V(i, j+1)$ :  $u(i, j) = 1$ : 인 경우  
고객이 큐 1로 갈 때 최소화된다.
- 2)  $V(i+1, j) > V(i, j+1)$ :  $u(i, j) = 0$ : 인 경우  
고객이 큐 2로 갈 때 최소화된다.

따라서 최적의 제어  $u^*(i, j)$ 는 아래와 같이 정리될 수 있다.

$$u(i, j) = \begin{cases} 1 & \text{if } V(i+1, j) \leq V(i, j+1) \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

(12)에서 보듯이 각 상태를 초기로 하는 최적의 비용함수들이 알려져 있다면 최적 제어  $u(i, j)$ 를 구할 수 있다.

다음 과정에서는 (12) 보다는 더 구현 용이한 최적 제어의 형태를 구해 본다.

### 3 Step III

Step II에서 구해진 최적제어에 대한 closed form을 구해 보자. 우선 다음과 같은 집합  $S$ 를 생각해 보자.

$$S = \{(i, j) : V(i+1, j) \leq V(i, j+1)\}$$

집합  $S$ 가 알려져 있다면 최적의 제어는 매우 간단하다. 새로운 고객이 도착했을 때 상태가  $S$ 에 속하면, 고객은 큐 1로 가서 서비스를 받으면 된다. 하지만  $S$ 를 구하기 위해서도 최적 비용

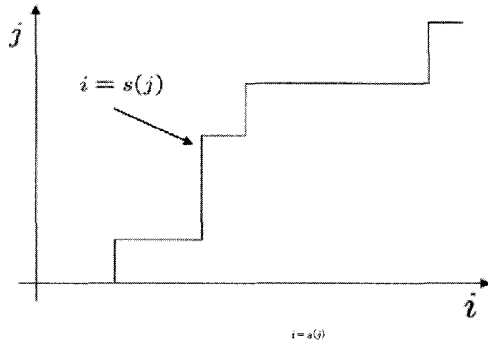


그림 3. 최적제어를 구하기 위한  $i = s(j)$  그래프

함수  $V(i, j)$ 를 구해야만 하는 어려움이 여전히 남아 있다.

최적의 해는  $j$ 에 대한 기준함수  $s(j)$ 를 바탕으로 표현될 수 있다. 따라서  $(i, j)$ 가  $i \leq s(j)$ 이면 최적 제어는 도착한 고객을 큐 1로 보낼 것이다. 최적해  $u^*(i, j)$ 는  $s(j)$ 를 이용하여 아래와 같이 쓸 수 있다.

$$u^*(i, j) = \begin{cases} 1 & \text{if } i \leq s(j) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

직관적으로 큐 2의 길이에 비해 큐 1의 길이가 충분히 작으면 새로 도착한 고객은 큐 1에서 서비스를 받는다. 상태  $(i, j)$ 에서 도착한 고객이 큐 1에서 서비스를 받아야 한다면 큐 1의 크기가  $i$ 보다 작은 경우 모두 큐 1에서 서비스를 받아야 한다. 이러한 사실로부터  $i = s(j)$ 는 그림 3과 같이 그려질 것이다. 예를 들면 (4,5)에서 도착한 고객이 큐 1으로 가는 것이 최적이라면 (3,5), (2,5), (1,5), (0,5)일 때도 큐 1으로 가는 것이 최적이다.

최적 제어는 그림 3과 같이 묘사된  $i = s(j)$ 의 그래프에 의해서 0과 1중에 하나를 선택하게 된다. 따라서 간혹  $i = s(j)$ 의 그래프를 스위칭 커브라고도 한다. 스위칭 커브가 주어지면 routing에 관한 최적제어는 두 개의 비용함수 인자  $\beta_1$ 과  $\beta_2$ 에 의존한다. 특히  $\beta_1 = \beta_2$ 이면 시스템은 대칭적이므로 스위칭 커브도  $i = j$ 가 된다. 즉 최적제어는 도착하는 고객이 큐의 길이가 짧은 쪽으로 가게 하면 된다. 하지만 대칭성이 깨지면 짧은 쪽으로 가는 것은 더 이상 최적의 해는 아니다.

사실 최적 제어  $u^*(i, j)$ 가 (12)와 같이 표시되는 데는 몇 가지 조건을 만족해야 한다. 우선  $\Delta V(i, j)$ 를 아래와 같이 정의하자.

$$\Delta V(i, j) = V(i+1, j) - V(i, j+1)$$

$\Delta V(i, j)$ 가 고정된  $j$ 에 대하여  $i$ 에 관한 증가함수이고, 또

한 고정된  $i$ 에 대하여  $j$ 에 관한 감소함수이면  $u^*(i, j)$ 가 (13)과 같이 표시될 수 있다. 주어진  $(i, j)$ 에 대하여  $\Delta V(i, j) \leq 0$ 이라고 가정할 때  $i$ 를  $\Delta V(i, j)$ 의 부호가 바뀔 때까지 증가시키면 바로 그 바뀌는 점  $(i^*(j), j)$ 가 그래프  $i = s(j)$  위에 있게 된다. 비슷한 방법으로  $j$ 를  $\Delta V(i, j)$ 의 부호가 바뀔 때까지 감소시키면 바로 그 바뀌는 점  $(i, j^*(j))$ 가 그래프  $i = s(j)$  위에 있게 된다. 위의 두 가지 단조성은 참고 도서 [1]의 admission problem을 참조하면 쉽게 증명할 수 있다.

#### 4. 유한구간제어를 통한 근사해

3장에서 소개한 최적화 조건식과 4장에서 소개한 최적제어의 표현은 계산 면에서 매우 불리하다. 이 장에서는 수치 계산을 위한 유한구간제어를 구해보기로 하자. 우선 유한구간제어가 최적해에 근사가능함을 보이자.

비용 함수 (9)를 두 구간으로 나누면 다음과 같다.

$$\begin{aligned} & \frac{\gamma}{\beta + \gamma} \sum_{k=0}^{\infty} \alpha^k E[(e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)))] \\ & + \frac{\gamma}{\beta + \gamma} \sum_{k=0}^{N-1} \alpha^k E[(e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)))] \quad (14) \\ & = \frac{\gamma}{\beta + \gamma} \sum_{k=N}^{\infty} \alpha^k E[(e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)))] \end{aligned}$$

모든 시간에서  $e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)) \leq K$ 을 만족하는  $K$ 가 있다면, (13)식으로부터 아래와 같은 부등식을 얻을 수 있다.

$$\begin{aligned} & \frac{\gamma}{\beta + \gamma} \sum_{k=0}^{\infty} \alpha^k E[(e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)))] \\ & \leq \frac{\gamma}{\beta + \gamma} \sum_{k=0}^{N-1} \alpha^k E[(e^{-\beta t} (B_1 X_1(t) + B_2 X_2(t)))] \quad (15) \\ & + K \sum_{k=N}^{\infty} \alpha^k \end{aligned}$$

마지막 항은 등비 급수로 다음과 같이 계산 된다.

$$K \sum_{k=N}^{\infty} \alpha^k = K \frac{\alpha^N}{1 - \alpha}$$

수식 (15)의 양변을 최소화하면 아래와 같은 무한구간과 유한구간의 부등식을 얻는다.

$$V^*(i) \leq V_N(i) + K \frac{\alpha^N}{1 - \alpha} \quad (16)$$

여기서  $V_N(i)$ 는  $N$ 구간에서 비용함수를 최적화했을 때의 값이다. 무한 구간과 유한 구간의 최소화라는 의미에서 쉽게  $V^*(i) \geq V_N(i)$ 임을 알 수 있고, 수식 (16)과 합치면 다음과 같이 된다.

$$V_N(i) \leq V^*(i) \leq V_N(i) + K \frac{\alpha^N}{1-\alpha} \quad (17)$$

$N$ 이 충분히 크면  $V_N(i)$ 가  $V^*(i)$ 에 충분히 가까이 갈 수 있음을 알 수 있다.

$V_N(i)$ 는 동적 프로그래밍을 통해서 다음과 같이 구할 수 있다.

$$\begin{aligned} V_{k+1}(i, j) = & b_1 i + b_2 j + \alpha q_1 V_k([i-1]^+, j) \\ & + \alpha q_2 V_k(i, [j-1]^+) \\ & + \alpha p \min[V_k(i+1, j), V_k(i, j+1)] \end{aligned} \quad (18)$$

동적 프로그램 (18)은 모든  $i$ 와  $j$ 에서 수행되어야 한다. 그런데 동적 프로그램 (18)을 수행하기 위해서는 무한대의  $i$ 와  $j$ 가 필요하다. 그래서 적절한 근사화가 필요한데 여기에서는 최대  $i$ 가 100 이라고 할 때  $V_k(101, j)$ 는 가장 가까운  $V_k(100, j)$ 으로 정하는 것으로 한다.

## 5. 시뮬레이션

4장에서 제안한 근사해를 시뮬레이션을 통해 직접 구해보도록 하자. 우선 시뮬레이션에 사용한 인자들은 다음과 같다.

$$\begin{aligned} \lambda = 0.5, \quad \mu_1 = 0.5, \quad \mu_2 = 0.5 \\ N = 100, \quad \beta = 1 \end{aligned}$$

비용함수에서  $b_1$ 과  $b_2$ 는 다음과 같이 설정한다.

$$\begin{aligned} b_1 = 2, \quad b_2 = 1 \\ b_1 = 1, \quad b_2 = 1 \\ b_1 = 2, \quad b_2 = 3 \end{aligned}$$

$b_1$ 이  $b_2$ 보다 큰 경우는 큐 1에 대한 비용이 상대적으로 크므로 고객을 큐 2 쪽으로 더 많이 보내려는 경향을 보일 것이고 반대는 큐 1 쪽으로 더 많이 보내려는 경향을 보일 것이다.

4장에서 언급했듯이 유한 구간의 시뮬레이션을 하기 위해서는 유한개의 상태를 생각해야 하는데, 여기서는 30개의 상태를 가정했다.

시뮬레이션은 CEMTool [3] 또는 MATLAB [4] 으로 가능하며,

사용한 파일은 다음과 같다.

```
%%% Optimal routing
state_num = 30;
horizon = 100;
lambda = 0.5;
mu_1 = 0.5;
mu_2 = 0.5;
beta = 0.1;
BB_1 = 1;
BB_2 = 1;

gamma = lambda + mu_1 + mu_2;
alpha = gamma/(beta + gamma);
q_1 = mu_1/gamma;
q_2 = mu_2/gamma;
p = lambda/gamma;
b_1 = BB_1/(beta + gamma);
b_2 = BB_2/(beta + gamma);

cost_before = zeros(state_num, state_num);
cost_after = zeros(state_num, state_num);
control = zeros(state_num, state_num);

for i=1:horizon
    for m=2:state_num
        mm = min(state_num-1, m);
        for n=2:state_num
            nn = min(state_num-1, n);
            temp = b_1*mm + b_2*nn + alpha*q_1*...
                cost_before(mm-1, nn) + alpha*q_2*...
                cost_before(mm, nn-1);
            if cost_before(mm+1, nn) <= ...
                cost_before(mm, nn+1)
                cost_after(mm, nn) = temp + ...
                    alpha*p*cost_before(mm+1, nn);
                control(mm, state_num-nn) = 1;
            else
                cost_after(mm, nn) = temp + ...
                    alpha*p*cost_before(mm, nn+1);
                control(mm, state_num-nn) = 0;
            end
        end
    end
    cost_before = cost_after;
end

switching = zeros(1, state_num);
for i=1:state_num
    switching(i) = size(find(control(:, i)), 1);
end
plot(switching(1:end-2));
```

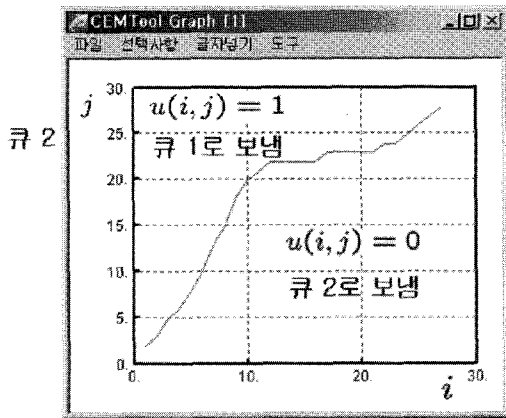


그림 4.  $b_1 = 2, b_2 = 1$ 인 경우

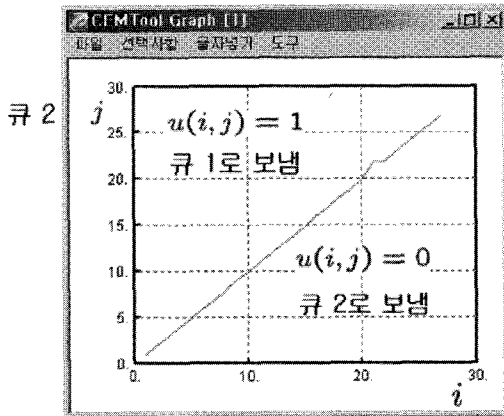


그림 5.  $b_1 = 1, b_2 = 1$ 인 경우

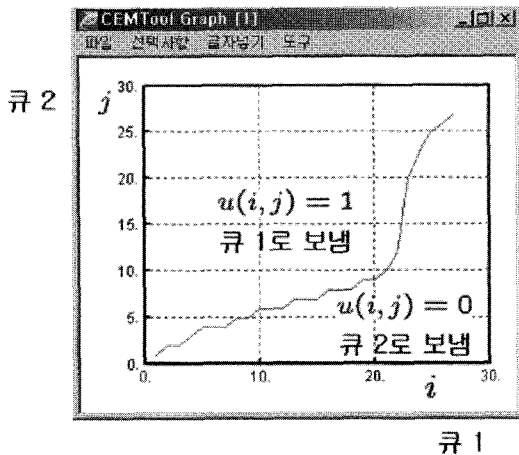


그림 6.  $b_1 = 2, b_2 = 3$ 인 경우

그림 4는 큐 1에 대한 비용이 크기 때문에 큐 2로 고객을 많이 보내는 경향이 있음을 말해주고 있다. 그림 5는 큐 1과 큐 2의 비용이 같기 때문에 도착한 고객은 둘 중에 짧은 큐로 이동하는 것이 최적임을 말해주고 있다. 그림 6은 그림 4의 경우와 반대 경우로 큐 2에 대한 비용이 크기 때문에 큐 1으로 고객을 더 보내려는 경향이 있음을 말해주고 있다.

## 7. 결론

본고에서는 확률 동적 자원 분배 문제의 한 예로 대기행렬 시스템의 최적화 문제를 다루었다. 간단한 두 개의 큐를 가진 시스템에서 주어진 가격함수를 최적화하는 제어가 만족해야 하는 최적화 조건식을 유도하고, 근사해를 구하기 위해 유한구간 제어를 소개했다.

## 참고문헌

- [1] C.G. Cassandras and Stephane Lafortune, Introduction to Discrete Event Systems, Kluwer Academic Publishers, 1999.
- [2] W. H. Kwon, and S. Han, Receding horizon control, Springer, 2005.
- [3] CEMTool : <http://cemtool.co.kr>
- [4] MATLAB : <http://www.mathworks.com>

### 저자약력



#### 한수희

- 1974년 8월 26일생.
- 1998년 2월 서울대학교 전기공학부(공학사).
- 2000년 2월 서울대학교 전기·컴퓨터공학부(공학석사).
- 2003년 8월 서울대학교 전기·컴퓨터공학부(공학박사).
- 2002년 11월~2003년 8월 서울대학교 제어계측신기술연구소 연구원.
- 2003년 9월~2005년 1월 서울대학교 BK 박사후 연구원.
- 2005년 1월~2005년 10월 스탠포드 방문연구원.
- 2005년 10월~2007년 4월 서울대학교 제어계측신기술연구소 선임 연구원.
- 2007년 5월~현재 서울대학교 기계·항공공학부 BK 연구 교수.
- 관심분야 : 최적 제어 및 확률 제어, 이동 구간 제어, 시간 지연 시스템, 불록형 최적화, 추정 이론, 컴퓨터 이용 제어 시스템 설계.