韓國經營科學會誌
第20卷 第2號
1995年 8月

Kanban함수를 이용한 Kanban 시스팀의 운영에 대한 연구

이상복* · 강석호**

A Study of Operation of Kanban Systems by using Function of Kanban numbers

Sang-bok Ree* · Suk-Ho Kang**

Abstract

Kanban 시스팀의 초기 Kanban 갯수 결정문제는 Kanban 시스팀의 최적 운영 문제 중 하나이다. 생산 공정 구조가 일반 형태를 갖는 Kanban 시스팀에서는 초기 최적 Kanban 갯수 결정 문제는 수학적으로는 정수계획법으로 모델화 되기 때문에 발견적 해법으로 가능해만 제시해 줄 수 있다. 실제 Kanban 시스팀을 사용하는 현장에서는 대부분 경험적으로 초기 Kanban 갯수를 결정한다. 본 논문에서는 현장 경험을 이용하여 각 생산 단계 사이에 생산량, 재고, 고갈이 Kanban 갯수에 따라 관계성을 찾을 수 있다면, 일반 생산 형태를 갖는 Kanban 시스팀에 Kanban 갯수 함수를 적용하여 초기 Kanban 갯수 결정하는 기법을 제시했다.

1. 서론

지난 수년 동안 일본 기업들의 생산성과 품질 향상으로 여러 나라에서 일본의 생산관리 기법에 많은 관심을 보여왔다. 일본 생산관리 기법은 적시시스팀(Just-In-Time JIT)과 전사적 품질관리(Total Quality Control TQC)로 특징 지어진다. TQC는 전 사원이 전 분야에서 품질향상을 위해, 조금의 불량도 허용하지 않는 마음 자세부터 시작한다. 이는 사원들의 자발적인 참여가 필요하기 때문에 회사에서는 자발적인 참여가 이루어지게 분위기를 조성하고 교육을 실시해야 한다. 이점은 구미 특히 미국기업이 전문가로 이루어진 스탭 중심의 운영과 차이를 보이고 있는 부분이다. JIT를 실제 현장에 구현한 토요타 자동차의 Kanban 시스팀은 일본기업이 성공한 일본식 생산방식으로 알려지고

* 서경대학교 산업공학과
** 서울대학교 산업공학과

있다. 또한 많은 학자들이 Kanban 시스템을 수학적으로 해석하고 많은 방법으로 해법을 제시하고 있다. Kanban 시스템은 Kanban의 갯수로 재고와 생산 속도를 통제하는 관리 기법이다. 생산 현장에서는 생산 라인의 전문가가 그날의 생산계획에 맞춰서 Kanban 갯수를 결정하고 있다. 지금까지 많은 연구는 생산 현장에서 경험자가 결정하는 초기 Kanban 갯수를 수학적으로 정해주고 있다. 생산공정 구조가 단순공정 형식에서는 쉽게 해를 준다. 그러나 생산 구조가 복잡해지면 모델식이 복잡한 정수계획법으로 표현되고 수학적으로 최적해를 보장할 수 없다. 최적해를 보장 못하는 경우는 가능해만 시뮬레이션 기법이나 발견적 기법으로 제시하고 있다. 본 논문에서는 복잡한 정수계획법으로 표현되는 모델식을 Kanban 갯수함수를 도입하여 기존의 기법대신 간단한 해법을 제시하려고 시도했다.

1-1. Kanban 시스팀 개요

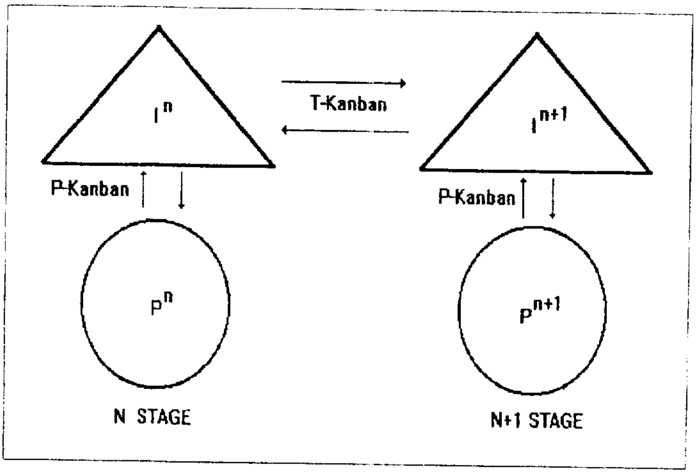
Kanban 시스팀은 JIT 시스팀을 토요타 자동차 생산 현장에서 구현한 생산통제 시스팀이다. Kanban 시스팀이라면 JIT를 구현한 일본식 생산 방식을 뜻하는 것으로 사용된다. Kanban은 카드를 뜻하며 이 카드는 컨테이너(Bin)에 붙인다. 이 카드에는 컨테이너에 담을 부품명, 생산량, 생산장소, 이동장소, 시간 등 정보가 기록되어 있다. Kanban 시스팀은 Kanban의 수로 생산속도와 재고를 통제한다. Kanban 갯수가 많으면 생산은 빨라지나 재고가 많아지고, Kanban 갯수가 너무 적으면 재고 고갈로 다음 공정에서 작업이 멈춰질 수 있다. 현장에서는 현장 전문가가 처음엔 적당한 Kanban 수로 운영하다가 차차 Kanban 갯수를 줄이는 방식으로 운영하고 있다. Kanban 시스팀 운영은 단일 Kanban 시스팀과 복수 Kanban 시스팀이 있다. 원리는 똑 같으나 복수 Kanban 시스팀은 생산과 이동이 다른 Kanban으로 운영되고 단일 Kanban 시스팀은 하나의 Kanban으로 생산과 이동을 통제 한다. 현 일본에서는 70% 이상 복수 Kanban 시스팀을 사용한다.

[그림 1]은 복수 Kanban 시스팀의 Kanban 흐름도이다. 부품은 n-stage에서 n+1 stage으로 흐른다고 하자. 1) n+1 stage에서 부품이 필요하면 빈 컨테이너에 이동 Kanban (T-Kanban)을 부착하여 n-stage I^n로 보낸다. 2) I^n에 부품이 가득들은 컨테이너가 있으면 부품 컨테이너에 붙은 생산 Kanban (P-Kanban)을 떼어 놓고, n+1 stage에서 온 이동 Kanban을 붙여 n+1 stage의 I^{n+1}로 보낸다. 3) n-stage에서는 떨어진 생산 Kanban(P-Kanban)은 빈 컨테이너에 붙여서 P^n으로 보내면, P^n에서 보내온 생산 Kanban 수 만큼 생산해서 컨테이너에 생산 Kanban를 붙이고 I^n으로 보낸다. 4) n+1 stage에서는 이동 Kanban이 붙은 컨테이너를 I^{n+1}에 보관한다. P^{n+1}에서 사용할 때는 이동 Kanban을 떼어놓고 생산 Kanban을 붙여서 P^{n+1}으로 이동한다. 떨어진 이동 Kanban은 다시 I^{n+1}에서 I^n으로 보낸다.

1-2. 연구 배경

지금까지 발표된 많은 논문은 MRP와 Kanban 시스팀의 비교[4], Kanban 시스팀의 시뮬레이션 연구[3], 수학적 모델로 최적해 및 가능해 접근[1,2,5]의 방식이 많았다. 생산 구조가 단순한 선형 조립 라인에서는 최적해를 보장한

[그림 1] 복수 Kanban 시스템



다. 그러나 생산 구조가 조립나무 구조나 일반 구조는 복잡한 정수문제(Complex Integer Problem)이거나 비선형으로 표현된다[1,5]. 이들 문제에선 Kanban 시스템의 Kanban 흐름과 생산이 확정적인 조건 하에서 문제를 풀고 있다.

2. Kanban 갯수 함수를 이용한 Kanban 시스템

지금까지 Kanban 시스템의 운영에 대한 연구는 Kanban 흐름과 생산이 확정적인 조건 하에서 문제를 연구했다. 확정적인 경우는 Kanban의 이동시간을 무시했으며, 예측못한 사고는 고려하지 않았다. 이런 가정에도 아래 [그림 2]와 같은 일반 구조의 Kanban 시스템은 정수계획법 해법으로 최적해를 줄 수 없다.

현실에 맞게 확정적인 가정을 빼면 위의 해법엔 확률적 함수를 이용하는 것이 적합하다. 현실이 불명확하고 애매하지만 이것은 확률식으로 표현 하여 Kanban 갯수에 따른 각 생산, 재고, 재고 고갈 함수를 찾을 수 있다면 확정적인 경우의 복잡한 해법보다 간단한 해법을 찾을 수 있고, 해가 현실에 더 적합하다.

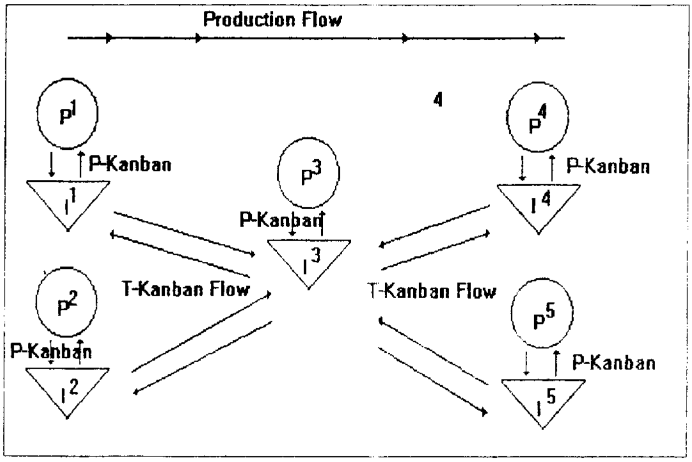
본 논문은, 수요가 이산적이고 생산 공정 구조가 일반 형태인 Kanban 시스템에서 경험적인 확률적 관계성을 이용하여 해를 구하는게 목적이다. 확정적이란 확률과 상치되는 개념이지만 관계성을 구현하는데 확정적인 요소가 많으면 많을 수록 좋다.

각 단계에 Kanban이 증가하면 생산량을 증가 시킨다. Kanban이 전후 단계로 충분하다면 연속 생산으로 생각할 수 있다. Kanban 갯수에 따른 시스템의 함수에 대해 살펴보자.

2-1. 생산 함수

각 생산 단계에서 생산하는 최적의 Kanban 갯수는 정확히 모르고 경험적으로 각 생산 단계별로 생산 함수를 갖는다. Kanban 갯수에 따른 생산 함수를 찾아야 한다. [그림 3]과 같이 일반 구조에서 각 생산 단계에서 Kanban

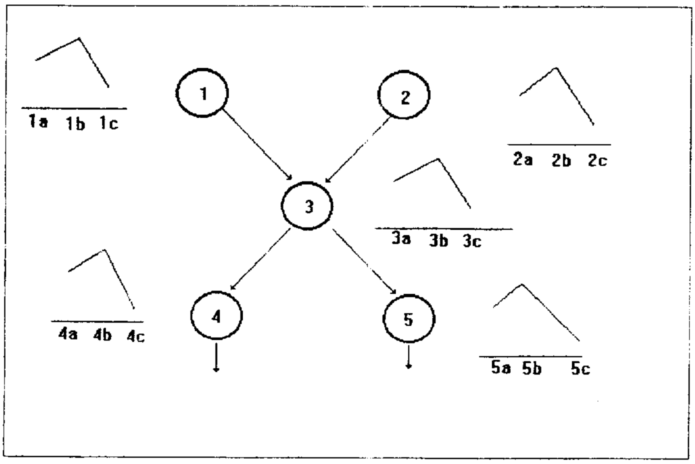
[그림 2] 일반 구조의 Kanban 시스팀



갯수에 따른 생산 함수를 나타낼 수 있다. 각 생산 단계에서 생산 함수는 Kanban 갯수에 따른 생산량을 측정함으로서 얻을 수 있다.

경험적으로 각 생산 단계에서 Kanban 갯수가 적을 때는 생산이 적고 Kanban 갯수가 증가함에 따라 생산이 점차적으로 증가한다. 그러나 Kanban 갯수가 더 증가하면 전 단계의 부품 부족과 다음 단계에서 가져가지 않으므로 생기는 재고 때문에 다시 생산이 떨어진다. 생산이 떨어지는 속도는 Kanban갯수가 적을 때보다 더 많이 떨어진다. [그림 3]에서 각 생산 단계에서 이 함수를 선형이라 가정하면 다음과 같이 나타낼수 있다.

[그림 3] 일반 구조의 생산 함수



[제안 1] 생산함수

f : *Kanban* 갯수→생산 함수 값

$$f(x_s) = \begin{cases} m_{s1}x_s + n_{s1} & a \le x_s \le b \\ m_{s2}x_s + n_{s2} & b \le x_s \le c \end{cases} \quad (2.1)$$

x_s = 단계 s에서 *Kanban*의 갯수 a, b, c는 Kanban의 갯수의 범위, m_{s1}, n_{s1}, m_{s2}, n_{s2}는 생산 s단계의 생산 함수가 갖는 고유 상수이다.

Kanban 갯수에 따라 생산율(생산량)이 선형이라면, $a \le x_s \le b$인 범위에서는 함수 $f(x_s) = m_{s1}x_s + n_{s1}$는 증가함을 뜻한다. Kanban 수가 증가함에 따라서 생산이 증가한다. 또 범위 $b \le x_s \le c$에서는 함수 $f(x_s) = m_{s2}x_s + n_{s2}$는 감소하는 선형식이다. Kanban 수가 어느 정도 이상 증가하면 앞 뒤 단계와 부품 재고 문제로 도리어 생산이 둔화된다. $m_{s1} > 0$, $m_{s2} < 0$을 뜻한다. m_{s1}, n_{s1}, m_{s2}, n_{s2}는 각 생산 단계에서 실험을 통해서 정할 수 있다.

[그림 4]와 같이 각 단계마다 생산 함수는 다르다. 각 단계에서 앞 뒤 단계를 고려해서 최적 Kanban 갯수를 구해야 한다. 합성 공정에선 전 공정의 최소 생산량에 의존한다. 각 단계 간의 생산 관계를 $RF(j, k)$ = 단계 k에서 단계 j 사이의 생산 관계라 하면 다음과 같이 제안한다.

[제안 2]

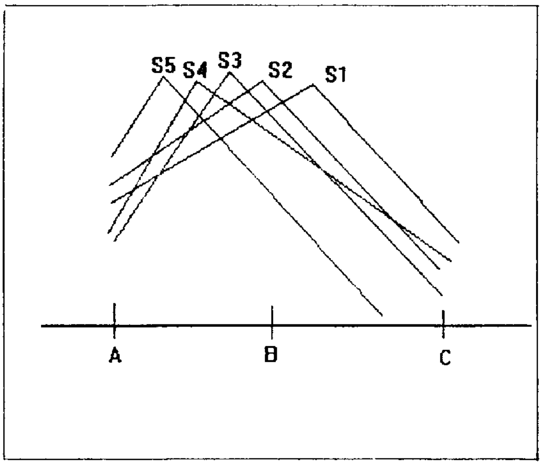
$$RF(j, k) = \begin{cases} f(x_k), & j\text{의 전단계가 없을 때} \\ \min[RF(i,j), f(x_k)], & i \in \{j\text{'s predecessors}\} \end{cases} \quad (2.2)$$

$RF(j,k)$ = 단계k에서 단계 j사이의 생산 관계는 단계 k에 연결된 모든 전 단계들의 생산량의 최소치에 언제나 지배 받게 된다. 이를 식으로 나타내면 식(2.2)와 같다.

각 생산 단계의 관계를 RF로 나타내면, 각 단계에선

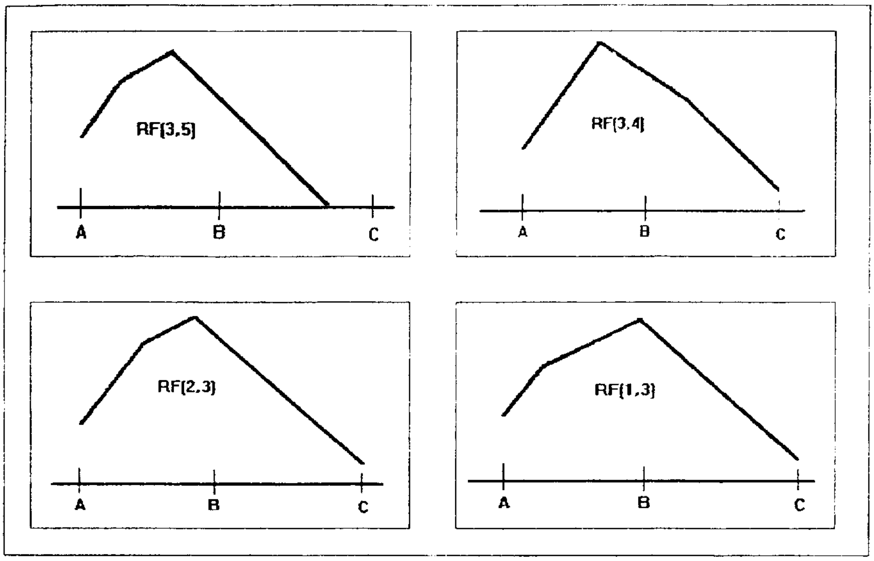
$RF(1, 3) = \min\{f(x_1), f(x_3)\}$,
$RF(2, 3) = \min\{f(x_2), f(x_3)\}$,
$RF(3, 4) = \min\{RF(1, 3), Rf(2, 3), f(x_4)\}$,
$RF(3, 5) = \min\{RF(1, 3), RF(2, 3)$,
$f(x_5)]$이다.

[그림 4] Kanban 갯수에 따른 생산 함수



이를 그림으로 다시 그리면 아래 [그림 5]와 같다. 그림 속에서 $RF[1, 3]$, $RF[3, 4]$, $RF[3, 5]$는 처음엔 같은 선을 나타낸다. 생산 측면만 보면 생산 함수의 최소값만 취하면 되지만 Kanban 갯수에 따라서 재고가 생길 수도 있고, 재고 고갈이 생길 수도 있어 최적 Kanban 갯수는 재고 와 재고 고갈을 고려한 비용이 최소가 되게 하는 Kanban 갯수를 찾아야 한다.

[그림 5] 생산단계들의 생산관계 함수



2-2. 재고 함수

Kanban 시스템에선 재고를 인정하지 않지만 Kanban 갯수에 따른 재고는 생긴다. Kanban 시스템에선 운영에 필요한 최소한의 Kanban 갯수만큼 재고를 갖는다. Kanban 시스템을 운영하는 데는 Kanban 갯수에 따라 생기는 재고를 고려해야 최적 Kanban 갯수를 구할 수 있다. 재고 함수도 Kanban 갯수에 따라 선형이라 가정하면 다음과 같이 나타낼 수 있다.

[제안 3]

재고 함수 g : *Kanban* 갯수→재고 함수 값

$$g(x_s) = \begin{cases} n_{s3} & a \le x_s \le b \\ m_{s4} x_s + n_{s4} & b \le x_s \le c \end{cases} \qquad (2.3)$$

x_s=단계 s에서 *Kanban*의 갯수, a, b, c는 Kanban의 갯수 범위, m_{s4}, n_{s3}, n_{s4}, 생산 s단계의 재고 함수가 갖는 고유 상수이다.

각 단계에서 Kanban의 갯수가 적을 때에는 재고가 일정하다. 즉 Kanban 수가 적을 때는 일정 수준 이하의 재고만 가지게 되고, 또 전후 단계와 적절히 유지되므로 일정한 재고만 유지된다. Kanban의 갯수가 증가 되면, 생산은 많아지나 전후 단계에서 이동은 일정하므로 곧 재고가 쌓이게 되므로 식 (2.3)과 같이 나타낸다.

곧 각 단계별로 재고 함수를 나타내면 아래 [그림 6]과 같다.

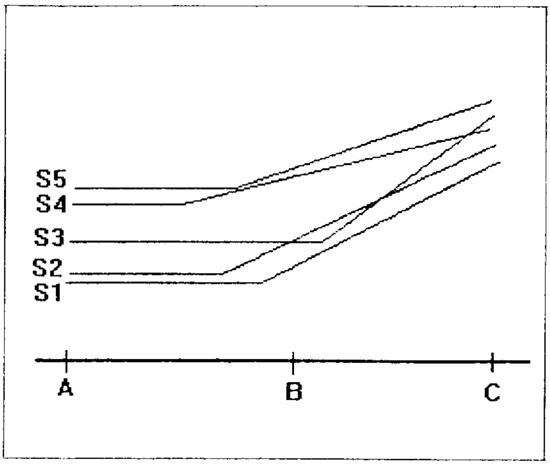
각 단계 간의 재고 관계를 $RG(j, k)$ =단계 k에서 단계 j 사이의 재고 관계라 하면 다음과 같이 제안한다.

[제안 4]

$$RG(j, k) = g(x_k) \; j \in \{k\text{'s } predecessors\} \qquad (2.4)$$

$RG(j, k)$=단계 k에서 단계 j사이의 재고 관계는 단계 s에 연결된 모든 전단계들의 재고량엔 직접적인 영향은 받지 않는다. 단지 단계 s만 의존한다. 이를 식으로 나타내면 식(2. 4)와 같다.

[그림 6] 각 단계들의 재고 함수



2-3. 고갈 함수

Kanban 시스템에선 재고 고갈은 인정하지 않지만 Kanban 갯수에 따른 고갈이 생길 수도 있다. Kanban 시스템을 운영하는 데는 Kanban 갯수에 따라 생기는 고갈도 고려해야 최적 Kanban 갯수를 구할 수 있다. 고갈 함수도 Kanban 갯수에 따라 선형이라 가정하면 다음과 같이 나타낼 수 있다.

[제안 5]

고갈 함수 h : *Kanban* 갯수→고갈 함수 값

$$h(x_s) = \begin{cases} m_{s5}x_s + n_{s5} & a \le x_s \le b \\ n_{s6} & b \le x_s \le c \end{cases}$$

x_s = 단계 s에서 *Kanban*의 갯수, a, b, c는 Kanban의 갯수의 범위, m_{s5}, n_{s5}, n_{s6}는 생산 s 단계의 고갈 함수가 갖는 고유 상수이다.

각 단계에서 Kanban의 갯수가 적을 때에는 앞 단계에서 부품을 요구할 때, 생산이 없어 고갈이 발생한다. Kanban의 갯수가 많아지면 고갈은 없어진다.

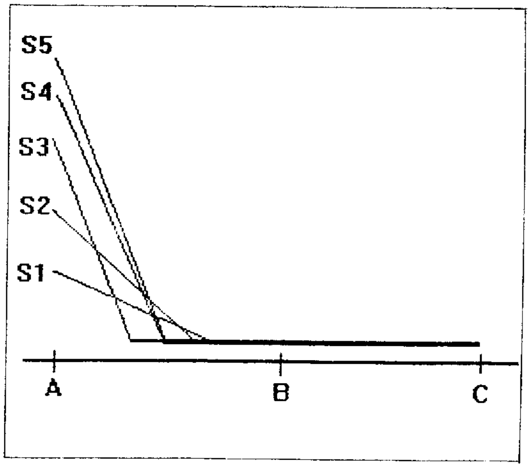
각 단계별로 고갈 함수를 나타내면 아래 [그림 7]과 같다. 고갈 함수는 전 단계의 고갈 함수에 직접적인 영향은 받는다. 각 단계 간의 고갈 관계를 $RH(j,k)$ = 단계 k에서 단계 j 사이의 고갈 관계라 하면 다음과 같이 제안한다.

[제안 6]

$$RH(j,\ k) = \begin{cases} h(x_k),\ j\text{의 전단계가 없을 때,} \\ \max[RH(i,j),\ h(x_k)], \\ i \in \{j\text{'s predecessors}\} \end{cases} \quad (2.6)$$

$RH(j,k)$ = 단계 k에서 단계 j사이의 고갈 관계는 단계 s에 연결된 모든 전단계들의 고갈량의 최대치에 언제나 지배 받게 된다. 이를 식으로 나타내면 식(2.6)와 같다.

[그림 7] 각 단계들의 고갈 함수



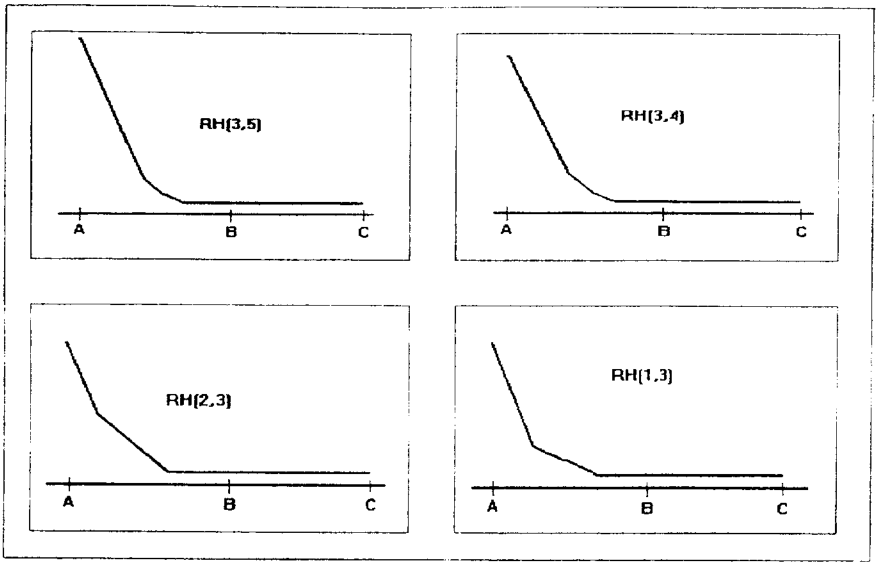
고갈 관계함수로 각 생산 단계의 고갈 관계를 나타내면, 각 단계에선 다음과 같다.

$RH(1,3)$ = $\max[h(x_1),\ h(x_3)]$,

$RH(2,3)$ = $\max[h(x_2),\ h(x_3)]$,

$RH(3,4) = \max[RH(1,3),\ RH(2,3),\ h(x_4)]$, $RH(3,5) = \max[RH(1,3),\ RH(2,3),\ h(x_5)]$이다. 이를 그림으로 다시 그리면 아래 [그림 8]와 같다. 그림 속에서 $RH[1,3]$, $RH[3,4]$, $RH[3,5]$는 Kanban 수가 많아지면 같은 선을 나타낸다. Kanban 시스팀 운영의 최적 Kanban 갯수는 이상에서 살펴본 생산 함수와 재고 함수 그리고 고갈 함수를 모두 고려 해서 구해야 한다.

[그림 8] 생산 단계들의 고갈 관계 함수



3. 최적 Kanban 시스팀 운영

최적 Kanban 시스팀 운영은 가장 높은 생산 효율로 생산하여, 수요량을 충족시키면서 재고 비용과 고갈 비용을 최소화 시키는 문제이다. 단계 s에서 비용을 식으로 나타내면 다음과 같다.

$$Min.SC(s-1, s) = [(\{Max\ RF(s-1,\ s)\} - RF(s-1,\ s))(\text{생산비용})] + (\text{재고 비용})\ RG(s-1,\ s) + (\text{고갈비용})\ RH(s-1,\ s) \quad (3.1)$$

"$[\{Max\ RF(s-1,\ s)\} - RF(s-1,\ s)$(생산비용)]"식은 생산이 최고의 효율로 생산되고 있을 때는 생산에 드는 비용은 적고, 생산이 비효율로 생산될 수록 생산비용은 증가 함을 뜻 한다. 생산 공정의 모든 단계별 비용의 합이 최소가 되게 하는 것이 전체 Kanban 시스팀의 최적 운영이다. 이를 식으로 나타내면,

$$Min.\ \sum_{\forall (i,s)\ \text{is immediate predecessor of}\ s} SC(i,\ s),\ s = \text{각 단계} \quad (3.2)$$

윗 식을 만족하는 각 단계별 최적 Kanban 갯수를 찾는 것이 본 논문의 핵심이다. 이 문제를 이산 최적 기법으로 최적 해를 찾는 해법이 아직 개발되지 않았다. 이 문제의 해를 확률 기법을 이용하면, 모든 단계의 비용 합계

함수를 구하고, 최종 단계의 확률 값을 최적화 시키는 점을 찾으면 된다. 최종 단계가 구해지면 역으로 최적점을 차례로 구해 가면 모든 단계의 최적점을 구할 수 있다. 관계로는

$FT(s-1,\ s)=[\{Max\ \ RF(s-1,\ s)\}-RF(s-1,\ s)]$(생산비용),

$GT(s-1,\ s)=$(재고비용) $RG(s-1,\ s)$,

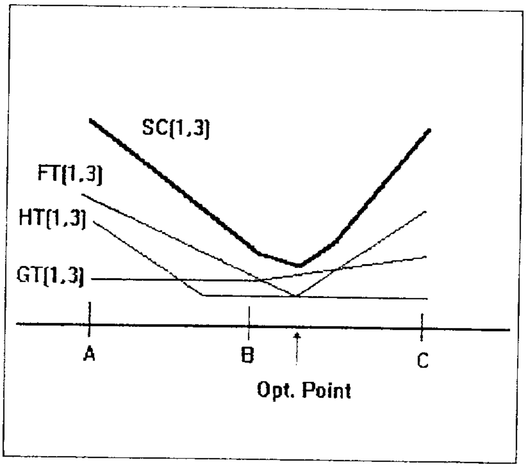
$HT(s-1,\ s)=$(고갈비용) $RH(s-1,\ s)$이라 하면,

$$SC(s-1,\ s)=FT(s-1,\ s)+GT(s-1,\ s)+HT(s-1,\ s) \tag{3.3}$$

식 (3.3)이 성립한다. 이 관계는 각 부분간의 덧셈만 해주면 된다.

단계 1에서만 비용 합계 함수를 구하면 아래 [그림 9]과 같다. 각 모수값 생산비 : 재고비 : 고갈비 = 10 : 5 : 3 으로 적당한 수치를 주었을 때, 단계 1의 총비용은 $SC(1,3)$ = $FT(1,3)+$ $GT(1,3)+$ $HT(1,3)$이다.

[그림 9] 단계1 에서 비용 합계 함수



위 [그림 9]와 같이, 마지막 단계 5는 $SC(3,5)$ = $FT(3,5)+$ $GT(3,5)+$ $HT(3,5)$의 관계식을 그릴 수 있다.

이상에서 제시된 방법을 "F-Kanban 초기 갯수 정하는 기법" 이라면, 이를 정리하면 다음과 같다.

F-Kanban 초기 갯수 정하는 기법

단계 Ⅰ : 각 생산 단계의 식 (2.1)(2.3)(2.5)를 정한다.

즉 $f(x_s)$, $g(x_s)$, $h(x_s)$, $s\ =\ 1,\cdots,N$

단계 Ⅱ : 각 생산 단계간의 관계식 (2.2)(2.4)(2.6)를 계산한다.

즉 $RF(s\text{-}1,s)$, $RG(s\text{-}1,s)$, $RH(s\text{-}1,s)$, $s\ =\ 2,\cdots,N$

단계 Ⅲ : 마지막 단계부터 FT(s-1,s), GT(s-1,s), HT(s-1,s) 및 SC(s-1,s)를 계산하여, Kanban 갯수를 정한다. 생산 단계 1까지 모두 계산한다.

4. 예제

앞에 있는 [그림 3]을 실제 수치를 대입하여 풀어보자. 각 생산 단계에서 생산 함수, 재고 함수, 고갈 함수는 다음 표1, 2, 3과 같다. 생산비용 = 10, 재고유지비용 = 5 그리고 고갈비용 = 10 으로 가정했다.

〈표 1〉 각 단계별 생산 함수

생산단계	생 산 함 수	
$f(x_1)$	$0.84x+5.7$	$0\le x\le 5.7$
	$-1.73x+20.349$	$5.7\le x\le 10$
$f(x_2)$	$x+6.6$	$0\le x\le 4.6$
	$-1.73x+19.158$	$4.6\le x\le 10$
$f(x_3)$	$2.15x+3.8$	$0\le x\le 3.7$
	$-1.6x+17.675$	$3.7\le x\le 10$
$f(x_4)$	$2.5x+4.6$	$0\le x\le 2.6$
	$-x+13.7$	$2.6\le x\le 10$
$f(x_5)$	$2.6x+6.72$	$0\le x\le 1.6$
	$-1.7x+13.6$	$1.6\le x\le 8$
	0	$8\le x\le 10$

〈표 2〉 각 단계별 재고 함수

생산단계	재 고 함 수	
$g(x_1)$	3.6	$0\le x\le 5.9$
	$0.78x+1$	$5.9\le x\le 10$
$g(x_2)$	3.82	$0\le x\le 4.6$
	$-0.7x+0.6$	$4.6\le x\le 10$
$g(x_3)$	5.38	$0\le x\le 7.8$
	$1.1x-3.2$	$7.8\le x\le 10$
$g(x_4)$	7.3	$0\le x\le 3.5$
	$0.4x+5.9$	$3.5\le x\le 10$
$g(x_5)$	7.85	$0\le x\le 5.1$
	$0.5x+5.3$	$5.1\le x\le 10$

〈표 3〉 각 단계별 고갈 함수

생산단계	고 갈 함 수	
$h(x_1)$	$-0.7x+3.22$	$0\le x\le 4.6$
	0	$4.6\le x\le 10$
$h(x_2)$	$-1.4x+5.74$	$0\le x\le 4.1$
	0	$4.1\le x\le 10$
$h(x_3)$	$-4.3x+9.89$	$0\le x\le 2.3$
	0	$2.3\le x\le 10$
$h(x_4)$	$-3.5x+11.2$	$0\le x\le 3.2$
	$-4x+12.8$	$0\le x\le 3.2$
$h(x_5)$	0	$3.2\le x\le 10$

위의 데이타를 가지고 마지막 단계부터 역으로 최적값을 구하면,

$$FT(3,5)=\begin{cases}-21.5x+45.12 & 0\le x\le 1.45\\ -8.4x+26.12 & 1.45\le x\le 3.1\\ 17x-52.88 & 3.11\le x\le 8\\ 83.12 & 8\le x\le 10\end{cases}$$

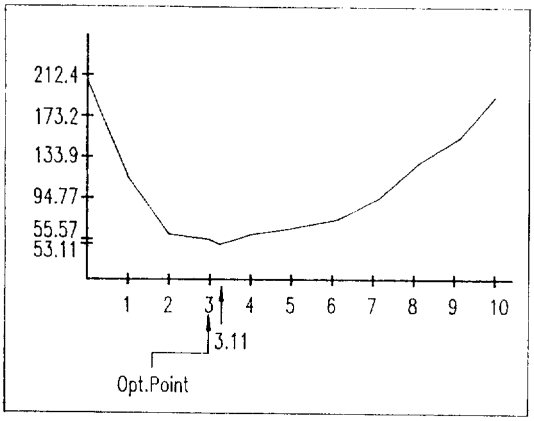
$$GT(3,5)=\begin{cases}39.15 & 0\le x\le 5.1\\ 2.5x+26.5 & 5.1\le x\le 10\end{cases}$$

$$HT(3,5)=\begin{cases}-40x+128 & 0\le x\le 2.716\\ -14x+57.4 & 2.716\le x\le 3.6\\ -7x+32.2 & 3.6\le x\le 4.6\\ 0 & 4.6\le x\le 10\end{cases}$$

$$SC(3,5)=\begin{cases}-61.5x+212.37 & 0\le x\le 1.45\\ -48.4x+192.37 & 1.45\le x\le 2.716\\ -22.4x+122.77 & 2.716\le x\le 3.11\\ 3x+43.77 & 3.11\le x\le 3.6\\ 10x+18.57 & 3.6\le x\le 4.6\\ 17x-13.63 & 4.6\le x\le 5.1\\ 19.5x-26.38 & 5.1\le x\le 8\\ 2.5x+109.62 & 8\le x\le 10\end{cases}$$

이를 그림으로 그리면 아래 [그림 10]과 같다. 이 때 최소치는 $x=3$일 때, 최소값 55.57이다. Kanban은 정수이므로 정수치에서 최소값을 찾았다.

[그림 10] 단계 5에서 비용 합계 함수



같은 방법으로 나머지 단계에서 $SC(j, k)$ 값을 구하면,

$$SC(3,5)=\begin{cases} -5.65x+204.02 & 0\le x\le 1.45 \\ -43.4x+185.02 & 1.45\le x\le 2.6 \\ -22.4x+130.42 & 2.6\le x\le 3.5 \\ -20.4x+123.42 & 3.5\le x\le 3.6 \\ -13.4x+98.22 & 3.6\le x\le 4.348 \\ 5x+18.22 & 4.348\le x\le 4.6 \\ 12x-13.98 & 4.6\le x\le 6.625 \\ 18x-53.73 & 6.625\le x\le 10 \end{cases}$$

$x=4$일 때, 최소값 44.6을 갖는다.

$$SC(2,3)=\begin{cases} -64.5x+196.4 & 0\le x\le 1.431 \\ -35.5x+154.9 & 1.431\le x\le 2.435 \\ -24x+126.9 & 2.435\le x\le 4.1 \\ -10x+69.5 & 4.1\le x\le 4.26 \\ 16x-41.25 & 4.26\le x\le 7.8 \\ 21.5x-84.15 & 7.8\le x\le 10 \end{cases}$$

$x=3$일 때, 최소값 28을 갖는다.

$$SC(1,3)=\begin{cases} -6.45x+186.03 & 0\le x\le 1.45 \\ -51.4x+167.03 & 1.45\le x\le 1.853 \\ -15.4x+100.33 & 1.853\le x\le 4.6 \\ -8.4x+68.13 & 4.6\le x\le 4.908 \\ 16x-51.62 & 4.908\le x\le 7.8 \\ 21.5x-94.52 & 7.8\le x\le 10 \end{cases}$$

$x=5$일때, 최소값 28.38을 갖는다.

이를 정리하면 초기 Kanban 갯수는 아래와 같다.

생산단계	Kanban 갯수
1↔3	5
2↔3	3
2↔3	4
3↔5	3

위의 예제가 간단한 공정이라는 한계점이 있지만, 위에서 제시한 함수 기법으로 구한 초기 Kanban 갯수는 정수계획법 보단 쉽게 구했다.

5. 결론 및 추후 연구

본 논문에서 Kanban 시스템 운영의 최적화를 푸는 방법으로 확률적 함수 이론을 도입했다. 초기 Kanban 갯수 구하는 문제는 복잡한 정수 계획법 문제로 NP-hard 문제로 알려졌고 [1,5], 제시된 발견적 기법도 어렵고 많은 계산 과정을 거치게 된다. 이에 비해 위에 제시된 기법은 계산이 많은 단점이 있지만, 해법이 간단하고 해가 명확하게 제시된다. 이는 현장에서 사용하는데 많은 도움이 된다. 이 이론을 적용할 수 있는 이유는 각 생산 단계 사이에 생산

능력, 재고, 고갈을 생산 단계간에 영향을 미치는 정도를 간단한 수식으로 표시할 수 있다는데 있다. 영향의 정도를 정확히 알기 어렵고 대충 알려져 있을 때 확률적인 이론 적용이 좋다.

추후 연구로 본 함수 이론을 좀 더 현실에 맞는 관계 함수를 찾아서 실제 현장에 적용하는 시스템을 개발해야 한다. 본 논문에서 제시된 해법과 정수계획법으로 모델화된 발견적 기법으로 제시된 해법과 비교 분석이 이루어져야 한다.

참고문헌

1. Bitran, C. R. and L. Chang, "A Mathematical Programming Approach to a Deterministic KANBAN System", *Management Science,* 33, 4 (1987), 427-439.
2. KIMURA, O and H. RERADA, "Design and Analysis of Pull System, a Method of Multi-stage Production Control", *International Journal of Production Research,* 19, 3 (1981), 241-253
3. KRAJEWSKI, L. J., B. E. KING, L. P. RITMAN and D. S. WONG, "Kanban, MRP and Shaping the Production Environment", *Management Science,* 33 1 (1987), 39-57.
4. SCHONBERGER, R. J., *Japanese Manufacturing Techniques : Nine hidden lessons in Simplicity,* the Free Press, New York, 1982
5. Kang, S. H. and S. B. REE, "A Study on Operations in Single-Card KANBAN System with a general-Type-Structure Production Process", *Journal of the Korean Institute of Indutrial Engineers,* 16 2 (1990), 109-117.