

# Rolling Horizon 환경하에서 다단계 재고 모형의 분배계획 수립에 관한 연구

안재성 · 권익현\* · 김성식

고려대학교 산업시스템정보공학과

## Distribution Planning in a Multi-Echelon Inventory Model under Rolling Horizon Environment

Jae-Sung Ahn · Ick-Hyun Kwon · Sung-Shick Kim

Department of Industrial Systems and Information Engineering, Korea University, Seoul, 136-701

In this paper we propose a distribution planning method aiming the use in the real-life situations. The assumed form of the distribution network is arborescence. At every node in the distribution network, orders are placed periodically. At each renewal of planning horizon, demand informations of periods in the horizon are updated. The objective of the problem is to minimize the total cost, which is the sum of holding and backorder costs of all sites during planning horizon. For such a situation, this study addressed an effective distribution plan when demands for demand-sites are provided for a given planning horizon.

**Keywords:** supply chain management, multi-echelon, distribution planning, rolling horizon

### 1. 서론

기업 환경의 세계화가 진행되면서 많은 기업들이 더 넓은 범주에서 외부 생산, 유통, 공급업자들과 긴밀한 관계를 맺게 되면서 공급사슬경영(supply chain management)에 대한 관심이 날로 증가되고 있다(Chopra and Meindi, 2001). 따라서 기존의 한 기업 내부에서 효율성을 추구하던 것이 이제는 공급사슬 전체에 걸친 효율성을 추구하고자 하는 시도들로 발전하고 있다. 또한 과거에는 어려웠던 기업 간의 정보교환과 의사소통이 최근에 인터넷과 같은 정보기술의 급속한 발달로 인하여 가능해지고 있다는 점도 공급사슬이 대두되고 있는 주요 원인으로 간주할 수 있다. 그러나 기업 활동이 세계화됨에 따라 생산 및 물류관리가 더욱 복잡해지고 이에 따르는 불확실성 또한 보다 증가하고 있어 공급사슬 전체를 효과적으로 운영·관리하는 것에는 많은 어려움이 따르고 있다(Axsäter, 2000; Erenguc *et al.*,

1999).

지난 수십년 동안 공급사슬 내의 재고 문제는 상당한 주목을 받아왔다. Wagner와 Whitin(1958)이 단일계층(single-echelon) 재고 모형에 대한 로트 크기 결정 문제(lot-sizing problem)를 동적계획법(dynamic programming)을 이용하여 다룬 이래로 많은 연구들이 단일계층 로트 크기 결정 문제에 관해 진행되어 오고 있다(Baker, 1989; Silver, 1973). Zangwill(1966)의 경우 Wagner와 Whitin의 알고리즘을 보다 일반화하여 재고이월(backorder)이 가능한 경우로 확장시켰다. 다단계(multi-echelon) 재고 모형에 관한 최초의 연구는 Clark과 Scarf(1960)에 의해 시도되었는데, 이들은 여러 노드들이 연속적으로 이루어진 시스템(serial system)에서 주기적 재고조사(periodic review) 주문정책을 이용하는 방법에 대해 다루었다. 다단계 재고 모형에 대한 대표적인 모형으로는 METRIC(Sherbrooke, 1968)과 MOD-METRIC(Muckstadt, 1973)을 들 수 있으며, 이들은 수율이 낮은 제품에

본 연구는 한국과학재단 목적기초연구(R01-2000-000-00404-0)의 지원으로 수행되었음.

\*연락처 : 권익현, 136-701 서울시 성북구 안암동 5가 고려대학교 산업시스템정보공학과,

Fax : 02-929-5888, E-mail : queens@korea.ac.kr

2003년 6월 접수, 2회 수정 후 2003년 10월 게재 확정.

대하여 (S-1, S) 주문정책을 적용하였다. 이후 다단계 분배 모형에 관한 많은 연구가 현재까지 활발히 진행되어 오고 있다 (Afentakis, *et al.*, 1984; Blackburn and Millen, 1982; Diks and De Kok, 1999; Federgruen, 1993).

본 논문에서는 다단계 분배형 공급사슬(multi-echelon distribution supply chain)을 대상으로 효과적인 주문정책 문제를 다루고자 한다. 분배형 공급사슬(distribution supply chain)이란 일반적인 제조회사에서 흔히 나타나는 형태로 생산된 제품들을 고객에게 효율적으로 공급하기 위한 목적으로 전체 공급사슬 네트워크를 구성하는 창고와 물류센터, 중간 창고, 소매점 등의 재고량을 결정하는 것을 말한다. 분배형 공급사슬에 있어서 대부분의 연구는 하나의 창고와 여러 개의 소매점으로 구성된 2단계 분배형 공급사슬인 단일 창고 복수 소매점 시스템(one-warehouse multi-retailer system)에 집중되어 왔다. 그러나 보다 현실적이고 일반적인 공급사슬의 형태라고 할 수 있는 다단계 구조를 갖는 트리 형태(arborescence)의 모델을 다룬 사례는 거의 없다고 할 수 있다. 이러한 이유로는 대부분의 연구가 공급 사슬을 구성하는 개별 노드들에 대한 로트 크기를 결정하는 문제(dot-sizing problem)로 국한시켰기 때문이다. 로트 크기 결정 문제는 각 노드 마다 재고유지 비용(holding cost)과 재고이월 비용(backorder cost) 외에 주문 비용(ordering cost)이 존재하므로 매기간 필요한 양만큼을 주문하는 것이 아니라 주문 비용과 재고 비용(재고유지 비용과 재고이월 비용)의 관계를 고려하여 어느 시점에 얼마만큼의 양(batch order quantity)을 주문할지를 결정하는 것이다. 이러한 로트 크기 결정 문제는 공급사슬 네트워크의 단계 수가 높아지거나 전체 공급사슬을 구성하는 노드의 수가 많아질 경우 모델링 및 해법에 소요되는 시간에 제약이 따를 수밖에 없다. 그러나 전체 공급사슬이 하나의 제조회사로 구성된 경우의 예에서처럼 주문 비용은 크게 중요

하지 않은 대신에 매 기간마다 요구되는 고객의 수요를 적시에 만족시키는 것에 초점이 맞추어진 상황이라면 매 기간 주문을 통해 제품을 공급하는 간단하면서 실행 가능한 주문 정책이 효과적일 것이다. 본 연구에서는 이와 같이 주문 비용을 고려하지 않기 때문에 비교적 간단하지만 보다 현실적이고 일반적인 형태의 공급사슬 모형에 적용 가능한 주문정책의 결정 방법에 관해 연구하고자 한다.

이와 함께 이러한 분배계획의 보다 현실적인 적용을 위해서는 변화하는 고객의 수요정보를 반영할 수 있는 rolling horizon 환경 하에서의 계획수립이 가능해야 할 것이다. 실제적인 공급사슬 분배 네트워크에 있어서 생산된 제품이 운반되는 과정은 끊임없이 계속되어지고 있으며, 대부분의 제조회사가 궁극적으로 바라는 것 또한 이처럼 수요가 새롭게 갱신되는 환경 하에서 변화하는 수요를 만족시키기 위한 적절한 분배계획이 지속적으로 수립되어지는 것이다. 본 연구에서는 이와 같은 상황을 반영할 수 있는 계획수립 방법을 고안하여 주어진 계획기간마다 트리 모형의 아크(arc)를 따라 흐르는 제품의 수량을 결정하고 공급사슬 네트워크 전체를 효율적으로 운용하는 문제를 해결하려 한다.

### 2. 문제 정의

본 논문의 대상이 되는 공급사슬의 구조는 <그림 1>과 같은 트리 형태를 나타낸다. 최상위 노드는 외부의 무한 용량을 갖는 공급자(supplier)에게서 제품을 공급받아 하위 노드(successor node)에게 분배하며 최하위 노드는 소매점(retailer)으로서 고객의 직접적인 수요가 발생하는 부분이다. 중간 노드는 제품을 임시 보관하는 중간 창고의 기능과 더불어 최하위 노드처럼

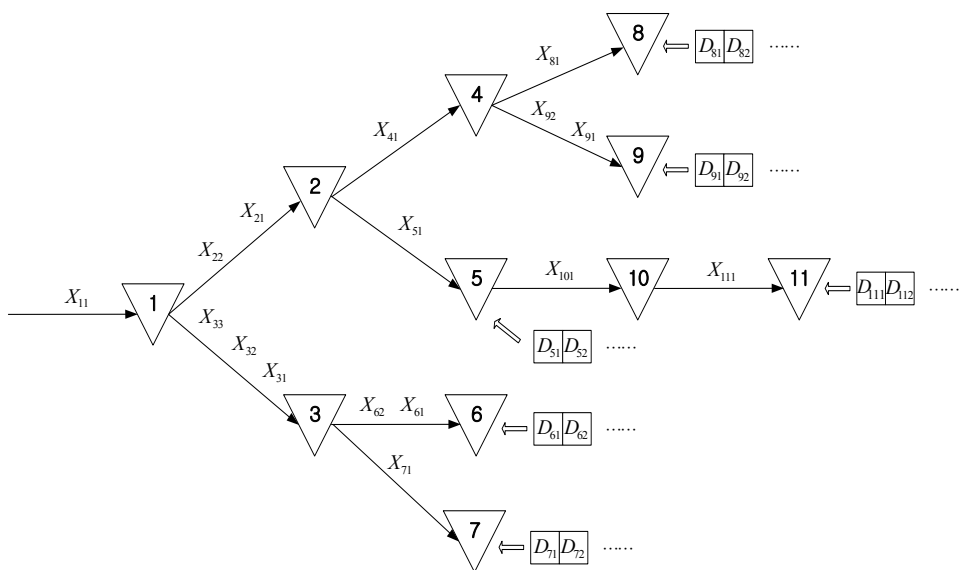


그림 1. 공급사슬 네트워크 구조.

고객의 직접적인 수요가 발생하는 소매상으로서의 역할을 할 수 있다. 각각의 노드들은 서로 다른 조달시간(leadtime)을 가지며 이러한 조달시간은 단위 계획 기간(time bucket)의 정수배(multiple integer) 형태로서 확정적(deterministic)으로 주어진다고 가정한다. 재고비용의 경우 재고이월 비용이 재고유지 비용보다 크고 재고유지 비용은 네트워크 상에서 하위 단계에 있는 노드일수록 상위에 있는 노드들보다 상대적으로 큰 값을 갖는 것으로 가정한다. 수요가 발생하는 노드는 계획 기간(planning horizon) 동안의 수요 예측(demand forecasting)에 기반하는 확정적인 수요 정보를 가지고 있고 이러한 수요 정보는 시간이 흐름에 따라 수정되어 변경된다. 또한 이러한 수요 정보가 의미 있는 분배계획 수립에 반영되기 위해서는 각 노드들은 최초의 공급자에서부터 출발해서 해당 노드까지 도착하는데 소요되는 누적 조달시간(cumulative leadtime) 이상의 기간에 대한 수요 정보를 알고 있어야 한다.

이러한 환경 하에서 제품이 최종 고객에게 공급되기 위해서는 최상위 노드에서 실제 수요가 발생하는 노드까지 여러 중간 노드들을 거쳐야만 한다. 그러므로 실제 고객의 수요가 발생하는 노드에서는 가까운 시점에 필요로 하는 수요라 하더라도 다단계 재고모형의 구조와 각 노드 사이의 조달시간 등을 고려한다면 해당 노드에 제품을 공급하는 상위 노드들은 이보다 이전 시간에 분배계획이 수립되어야만 한다. 공급사슬 네트워크를 구성하는 모든 아크(arc) 위에는 이처럼 앞으로 발생할 수요 정보를 기반으로 미리 수립된 분배계획에 의한 수송중(in-transit)인 물량들이 존재하게 된다. <그림 1>의 모든 아크 상에는 이와 같은 수송중인 물량을 나타내고 있다. 수송중인 물량은 각 노드들이 필요한 물량을 매 기간 주문을 통해 보충하는 주문정책으로 인해 해당 노드의 조달시간 만큼의 수가 존재하게 된다. 이러한 수송중인 물량은 이전 시점에서 예측된 수요정보에 바탕을 둔 분배계획의 결과로 현 시점에서 변경할 수 없으며 새로이 갱신된 수요 정보와는 다를 수 있다. 따라서 각 아크 상에 존재하는 수송중인 물량은 현 시점에서 새로운 상황을 반영하는 분배계획을 수립하는데 있어 계약을 주게 되는데, 이러한 제약조건 하에서 전체 공급사슬 네트워크 상에서 발생하는 재고비용을 최소화할 수 있는 효율적인 분배 알고리즘이 필요하다.

본 연구에서는 분배계획과 관련된 모든 상황이 매 기간 초에 발생하고, 전체 네트워크를 구성하는 모든 노드들에서 발생하는 일련의 사건들은 다음과 같은 순서로 진행되어진다고 가정한다.

- ① 현재 기간의 수요를 만족시키기 위한 수송물량 도착
- ② 지난 기간의 수요를 만족시키지 못해 발생한 재고이월(backorder)이 있을 경우 이를 보충
- ③ 실제 수요 발생 및 수요 정보 갱신
- ④ 갱신된 수요 정보를 바탕으로 새로운 분배계획 수립 및 이로 인한 물량 이동

#### ⑤ 분배계획의 결과로 발생한 재고비용 계산

여기서 주목할 점은 현실적인 측면에서 볼 때 수요예측에 의해 예상되는 예측 수요와 실제로 발생하는 수요는 반드시 일치하지 않는다는 것이며 이러한 현상은 재고비용 발생의 한 원인이 된다. 따라서 보다 효과적인 분배계획을 위해서는 계획 수립시 사용되는 수요 정보에 대해서 보다 현실적인 상황을 반영할 수 있는 연구가 필요한데 이에 대한 보다 자세한 내용은 본 논문의 결론에서 추후 연구과제로 언급하였다.

앞서 언급한 상황과 가정하에서 공급사슬 네트워크를 모델링하기 위해서는 다음과 같은 기호가 사용된다.

<기호>

- $N$  : 노드 개수
- $T$  : Rolling horizon 횟수
- $D_{it}$  : 노드  $i$ 의 기간  $t$ 에서의 수요
- $X_{it}$  : 노드  $i$ 로 기간  $t$ 에 도착하기 위해 수송중인 물량
- $I_{it}$  : 노드  $i$ 의 기간  $t$ 에서의 재고량
- $s_i$  : 노드  $i$ 를 통해 직접적으로 제품을 공급받는 하위 노드(successor)들의 집합
- $h_i$  : 노드  $i$ 의 재고유지 비용(holding cost)
- $b_i$  : 노드  $i$ 의 재고이월 비용(backorder cost)
- $L_i$  : 상위 노드에서 노드  $i$ 로의 조달시간
- $CL_i$  : 최상위 공급자로부터 노드  $i$ 까지의 누적조달시간(cumulative leadtime)
- $DN$  : 수요발생 노드

정의된 기호를 바탕으로 공급사슬 네트워크를 모델링하면 다음과 같다.

$$\min \sum_{t=1}^T \sum_{i=1}^N [h_i \max(I_{it}, 0) + b_i \max(-I_{it}, 0)] \quad (1)$$

s. t

$$I_{it} = I_{i(t-1)} + X_{it} - D_{it} - \sum_{j \in s_i} X_{j(t+L_j)}, \quad (2)$$

$$i=1, \dots, N, \quad t=1, 2, \dots, T+CL_i$$

$$I_{it} \geq 0, \text{ if } i \notin DN \quad (3)$$

$$X_{it} \geq 0, \quad i=1, 2, \dots, N, \quad t=1, 2, \dots, T+CL_i \quad (4)$$

분배계획은 장시간에 걸쳐 모든 노드에서 발생하는 재고유지 비용과 재고이월 비용의 합을 최소화시키는 데 목적이 있다. 제약식으로는 해당 노드에 대한  $t$ 기간의 재고량은  $(t-1)$ 기간의 재고량과  $t$ 기간에 새로이 도착하는 물량의 합에서 해당 노드의  $t$ 기간의 수요와 하위 노드들의 요청에 의해 보내어지는 물량의 합을 뺀 양과 같아야 한다는 제약, 수요가 발생하지 않는 노드의 경우 재고이월이 발생할 수 없다는 제약, 그리

고 비음수 제약식으로 이루어져 있다.

위의 모델은 기본적으로 선형계획(linear programming)의 형태를 나타내지만 공급사슬 네트워크의 구조가 복잡해지거나 네트워크를 구성하는 노드의 수가 많아진다면 분배계획을 수립하는 데 많은 시간이 소요된다는 문제점이 있어 현실적인 적용에 많은 제약이 따른다. 또한 실제적인 현장에서는 rolling horizon을 기반으로 수요가 계속적으로 갱신되는 환경 하에서 반복적인 계산을 제한된 시간(polynomial time) 내에 이루어지도록 하는 분배계획이 수립되어야 하므로 보다 신속하고 효과적인 분배 알고리즘이 필요하다.

본 연구에서는 모델링된 공급사슬 네트워크에 대한 효율적인 운용을 위한 분배 알고리즘을 개발하였다. 주어진 공급사슬 네트워크에 대해 체계적으로 표현하고 개발된 알고리즘의 절차를 보다 명확하게 설명하기 위해 다음과 같은 기호를 추가적으로 정의하였고 이에 대한 이해를 돕기 위하여 <그림 2>를 통해 구체적인 예와 함께 표현하였다.

<추가 기호>

$K$  : 단계 수 (예 :  $K=4$ )

$J_k$  : 단계  $k$  에 속하는 모든 노드들의 집합(예 :  $J_2 = \{5, 6, 7\}$ )

$S_i$  : 노드  $i$  가 포함하는 모든 하위 노드들의 집합

(예 :  $S_3 = \{5, 6, 8, 9\}$ )

$M_i$  : 노드  $i$  에 속하는 하위 노드들 중 가장 하위 단계

(예 :  $M_3 = 1, M_4 = 2$ )

$L_{ij}$  : 노드  $i$  에서 노드  $j$  로의 누적 조달시간

(예 :  $L_{19} = L_3 + L_5 + L_9$ )

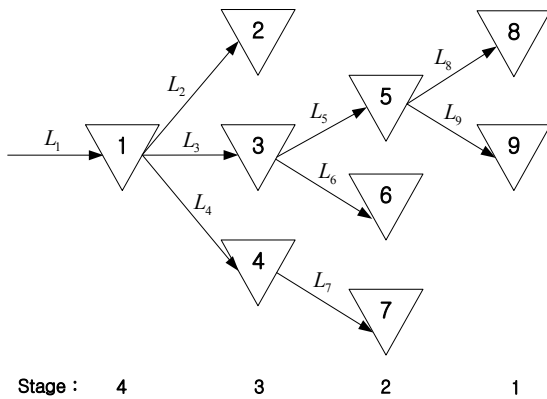


그림 2. 4단계 분배 네트워크의 표현 예.

### 3. 분배 알고리즘

분배계획은 매 계획 기간마다 새로이 갱신된 수요 정보를 바탕으로 수립되어지는데 분배계획의 수립 결과로 실제 네트워크 상에 반영되는 값들은 각 노드의 아크 상에 새로이 추가되

는 수송중인 물량들이다. 즉, 각 노드의 아크 상에는 해당 노드의 조달시간만큼에 해당하는 수송중인 물량이 존재하게 되는데, 이러한 수송중인 물량은 각 노드들이 필요한 물량을 매 기간 주문을 통해 보충하는 주문정책 때문이다. 하나의 계획 기간이 경과하면 각 노드의 아크 상의 수송중인 물량들 가운데 현 시점의 수요를 만족시키기 위한 물량은 해당 노드에 도착하게 되고 나머지 수송중인 물량도 마찬가지로 한 기간만큼 이동하게 된다. 이와 함께 새로이 갱신된 수요 정보를 만족시키기 위한 주문계획을 반영하는 수송중인 물량이 각 아크 상에 추가되고 이 결과 각 노드의 아크 상에는 매 계획 기간 동안 동일한 수의 수송중인 물량이 존재하게 된다.

분배계획의 수립 절차는 각 아크 상의 수송중인 물량을 제약으로 하여 하위 단계에서 상위 단계의 순서로 진행된다. 모든 노드의 아크 상을 따라 흐르는 수송중인 물량은 동일한 시점에 존재하지만, 하위 단계의 노드인 경우 가까운 시점의 수요를 위한 것이라면 상위 단계로 올라갈수록 이러한 수송중인 물량은 보다 먼 시점에 대한 수요를 위해 필요한 것이다. 그러므로 네트워크 아래 부분의 아크 상에 존재하는 수송중인 물량을 제약으로 하여 이를 통해 물품을 공급받는 하위 노드들에 대한 분배계획을 먼저 수립하고 이러한 계획 결과를 바탕으로 보다 상위 단계의 노드에 대한 분배계획을 수립하는 방식으로 전개되어야 한다.

이와 함께 네트워크 상의 모든 노드들은 해당 노드의 재고 부족 또는 과다로 생기는 재고비용의 발생을 줄이기 위해서 매 기간 필요한 양만큼을 주문하고자 하는데, 이 경우 상위 노드의 가용물량이 하위 노드에서 필요로 하는 물량보다 많다면 필요로 하는 물량만큼만 할당하고 나머지 물량은 상위 노드에서 보관하는 것이 재고유지 비용의 최소화 측면에서 합당하다. 반대로 상위 노드의 가용물량이 하위 노드에서 필요로 하는 물량보다 적을 경우에는 하위 노드들 중에서 재고이월 비용이 큰 노드에서 필요로 하는 물량부터 우선적으로 할당하는 것이 재고이월 비용의 최소화 측면에서 바람직하다.

<알고리즘 절차>

Step 1 :  $k=1$  로 설정

Step 2 :  $\forall i \in J_k$  에 대해 Step 2.1 반복

2.1 If  $S_i = \emptyset$ , 식 (2)에서  $I_{it}$  ( $t=1, 2, \dots, L_i$ ) 계산

Otherwise,  $t=1, 2, \dots, L_i$  동안 Step 2.2에서 2.4 반복

2.2  $i$ 에 속하는 모든 하위 노드들에 대해 가장 마지막 단계( $M_i$ )에 속하는 노드부터 바로 하위 단계인 단계( $k-1$ )에 속하는 노드 순으로 식 (2)에서  $I_{jt}$  ( $j \in S_i, t = t + L_{ij}$ )를 0으로 만드는 임시  $X_{jt}$  계산

2.3 Step 2.2에서 계산된 단계 ( $k-1$ )의 모든  $X_{jt}$  ( $j \in S_i, t = t + L_j$ )를 가지고, 식 (2)에서  $I_{it}$  계산

2.4 If  $I_{it} \geq 0$  일 경우,  $X_{jt}$  ( $j \in S_i, t = t + L_{ij}$ )는 Step 2.2에서 구한 모든 임시  $X_{jt}$  값과 동일한 값으로 확정

Otherwise, 노드  $i$ 의 이전 기간의 재고량( $I_{i,t-1}$ )과 노드  $i$ 로 도착될 양( $X_{jt}$ )의 합( $I_{i,t-1} + X_{jt}$ )을 노드  $i$ 를 포함하여 재고이월 비용이 가장 큰 하위 노드 순서로 할당하면서  $X_{jt}$  ( $j \in S_i, t = t + L_{ij}$ ) 계산

- Step 3:  $k = k + 1$ 로 설정 후,  
 If  $k \leq K$ 이면 Step 2로 이동  
 Otherwise, Step 4로 이동
- Step 4: 네트워크에 있는 모든 노드들에 대해 단계 1에서 단계  $K$ 에 속하는 노드 순으로 식 (2)에서  $I_{it}$  ( $i = 1, 2, \dots, t = CL_i + 1$ )를 0으로 만드는  $X_{it}$  계산

Step 1에서 먼저  $k = 1$ 로 설정하여 가장 하위 단계에 존재하는 노드들에 대해 알고리즘을 적용한다. Step 2.1에서 단계  $k$ 에 속하는 노드  $i$ 의 하위 노드가 존재하지 않을 경우 식 (2)에서 하위 노드로 보내주어야 하는 물량에 해당하는 결정변수는 존재하지 않고, 이전 기간의 재고량( $I_{i,t-1}$ )과 새로이 도착하는 물량( $X_{jt}$ )에서 그 기간의 수요량( $D_{it}$ )만으로 새로운 기간의 재고량( $I_{it}$ )을 계산한다. 그러나 해당 노드의 하위 노드가 존재할 경우에는 하위 노드로 보내 주어야 할 양을 결정하여야 하는데, 이러한 경우 Step 2.2를 통해 우선 해당 노드로부터 물품을 공급받는 모든 노드에 대해서, 이중 가장 하위 단계에 속하는 노드부터 상위 단계에 속하는 노드 순으로 순차적으로 각 노드에서 필요로 하는 물량을 임시적으로 계산한 후 Step 2.3에서 식 (2)를 이용하여 해당 노드의 가용물량(이전 기간의 재고량과 도착물량의 합)과 필요로 하는 물량(해당 노드의 수요와 Step 2.2에서 임시로 구한 하위 노드들의 필요 물량)의 차이를 비교한다. Step 2.4에서  $I_{it} \geq 0$ 일 경우, 해당 노드는 하위 노드에서 필요한 양을 충분하게 만족시킬 수 있으므로 이전 단계에서 구한 값과 동일하게 하위 노드에서 필요로 하는 양만큼을 할당하는 분배계획을 수립한다.  $I_{it} < 0$ 일 경우, 하위 노드에서 필요로 하는 양을 만족시키지 못하므로 이로 인하여 발생하는 재고이월 비용을 최소화시키기 위하여 재고이월 비용이 큰 노드에서 필요로 하는 물량부터 우선적으로 할당하는 분배계획을 수립한다. Step 3에서는 이러한 절차를 네트워크를 구성하는 모든 노드들에 대해 적용시키기 위한 과정이다. 마지막으로 Step 4는 외부의 공급자로부터 최상위 노드에 물량을 할당하는 단계로, 공급자는 무한의 용량을 가지고 있으므로 이 경우 Step 2에서처럼 임시 물량의 계산을 통한 비교절차는 생략하고 하위 노드에서 필요로 하는 물량을 구한 다음 이러한 물량을 만족시키는 양만큼을 할당하는 분배계획을 수립한다.

위의 알고리즘은 특정한 한 계획시점에서의 분배계획 수립에 관한 것으로 rolling horizon 환경하에 적용하기 위해서는 새로이 갱신되는 수요 정보를 바탕으로 매 계획 시점마다 이러한 알고리즘을 동일한 절차를 통하여 반복적으로 적용하여야 한다. 제안된 알고리즘에 대한 이해를 돕기 위해 다음 절에 수치 예제를 제시하였으며, 알고리즘에 관한 보다 자세한 절차

는 부록을 통해 수도 코드(pseudo code) 형태로 나타내었다.

### 4. 수치 예제 및 결과 분석

본 절에서는 <그림 3>과 같이 5개의 노드로 구성된 3단계 네트워크 모형에 대하여 본 연구에서 제안된 알고리즘을 적용하여 분배계획을 수립하는 과정을 살펴보고자 한다. 각 노드들은 서로 다른 조달시간(leadtime)을 갖으며 각 노드에 대한 재고비용과 초기 재고, 기간별 수요량, 조달시간 및 수송중인 물량에 대한 정보는 <표 1>에 자세하게 나타내었다.

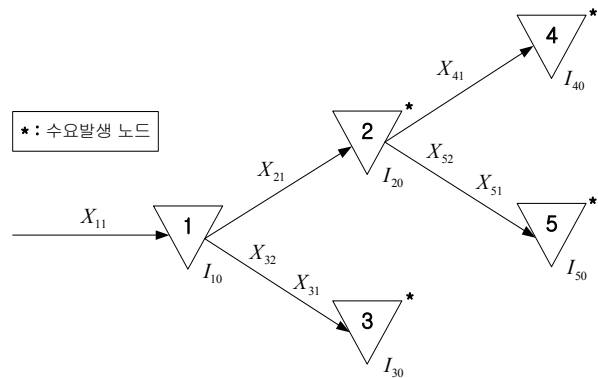


그림 3. 네트워크 분배계획 예제.

표 1. 각 노드의 기본 정보

노드 $i$	$h_i$	$b_i$	$I_{i0}$	$D_{it}$					$X_{it}$	
				1	2	3	4	5	1	2
1	1	-	4	-	-	-	-	-	16	-
2	2	7	2	3	3	5	-	-	8	-
3	2	8	5	6	6	4	3	-	3	6
4	4	5	-2	4	4	5	5	-	3	-
5	3	10	10	6	4	7	6	3	2	3

대상 네트워크는 3단계 구조를 갖고 있으므로  $K = 3$ 으로 설정한 후 분배 알고리즘을 실행하면 다음과 같다.

- Step 1:  $k = 1$ 로 설정
- Step 2:  $J_1 = \{4, 5\}$ 
  - 2.1  $S_4 = \emptyset, L_4 = 1$ 이므로  $t = 1$ 일 경우 식 (2)에서  $I_{41} = I_{40} + X_{41} - D_{41} = -2 + 3 - 4 = -3$  계산
  - 2.1  $S_5 = \emptyset, L_5 = 2$ 이므로  $t = 1, 2$ 일 경우 각각 식 (2)에서  $I_{51} = I_{50} + X_{51} - D_{51} = 10 + 2 - 6 = 6,$   
 $I_{52} = I_{51} + X_{52} - D_{52} = 6 + 3 - 4 = 5$  계산
- Step 3:  $k = 2, k \leq K$ 이므로 Step 2로 이동
- Step 2:  $J_2 = \{2, 3\}$

- 2.1  $S_2 = \{4, 5\}$ ,  $L_2 = 1$
- 2.2  $t = 1$ ,  $j = 4$ ,  $t' = 2$ 로 설정 후, 식 (2)에서  $I_{42} = I_{41} + X_{42} - D_{42} = -3 + X_{42} - 4 = 0$ 으로 놓고  $X_{42} = 7$  계산
- 2.2  $t = 1$ ,  $j = 5$ ,  $t' = 3$ 으로 설정 후, 식 (2)에서  $I_{53} = I_{52} + X_{53} - D_{53} = 5 + X_{53} - 7 = 0$ 으로 놓고  $X_{53} = 2$  계산
- 2.3  $I_{21} = I_{20} + X_{21} - D_{21} - (X_{42} + X_{53}) = 2 + 8 - 3 - (7 + 2) = -2 < 0$
- 2.4 할당 가능량( $I_{20} + X_{21}$ )은 10이고 재고이월 비용은  $b_4 < b_2 < b_5$ 의 순이므로 노드 5, 2, 4 순으로 할당하면 노드 5에 2, 노드 2에 3 그리고 노드 4에 5를 할당하게 되고 그 결과  $X_{42} = 5$ ,  $X_{53} = 2$ ,  $I_{21} = 0$ ,  $I_{42} = -2$ ,  $I_{53} = 0$  계산

- 2.1  $S_3 = \emptyset$ ,  $L_3 = 1$ 이므로  $t = 1, 2$ 일 경우 각각 식 (2)에서  $I_{31} = I_{30} + X_{31} - D_{31} = 5 + 3 - 6 = 2$ ,  $I_{32} = I_{31} + X_{32} - D_{32} = 2 + 6 - 6 = 2$  계산

Step 3:  $k = 3$ ,  $k \leq K$ 이므로 Step 2로 이동

Step 2:  $J_1 = \{1\}$

- 2.1  $S_1 = \{2, 3, 4, 5\}$ ,  $L_1 = 1$ 이므로 노드 1에 속하는 하위 노드들 중 단계 1에 속하는 노드 4, 5와 단계 2에 속하는 노드 2, 3의 순으로 Step 2.2 적용
- 2.2  $t = 1$ ,  $j = 4$ ,  $t' = 3$ 으로 설정 후, 식 (2)에서  $I_{43} = I_{42} + X_{43} - D_{43} = -2 + X_{43} - 5 = 0$ 으로 놓고  $X_{43} = 7$  계산
- 2.2  $t = 1$ ,  $j = 5$ ,  $t' = 4$ 로 설정 후, 식 (2)에서  $I_{54} = I_{53} + X_{54} - D_{54} = 0 + X_{54} - 6 = 0$ 으로 놓고  $X_{54} = 6$  계산
- 2.2  $t = 1$ ,  $j = 2$ ,  $t' = 2$ 로 설정 후, 식 (2)에서  $I_{22} = I_{21} + X_{22} - D_{22} - (X_{43} + X_{54}) = 0 + X_{22} - 3 - (7 + 6) = 0$ 으로 놓고  $X_{22} = 16$  계산
- 2.2  $t = 1$ ,  $j = 3$ ,  $t' = 3$ 으로 설정 후, 식 (2)에서  $I_{33} = I_{32} + X_{33} - D_{33} = 2 + X_{33} - 4 = 0$ 으로 놓고  $X_{33} = 2$  계산
- 2.3  $I_{11} = I_{10} + X_{11} - (X_{22} + X_{33}) = 4 + 16 - (16 + 2) = 2$  계산
- 2.4  $I_{11} \geq 0$ 이므로  $X_{43} = 7$ ,  $X_{54} = 6$ ,  $X_{22} = 16$ ,  $X_{33} = 2$ ,  $I_{11} = 2$ ,  $I_{22} = I_{33} = I_{43} = I_{54} = 0$  계산

Step 3:  $k = 3$ ,  $k > K$ 이므로 Step 4로 이동

Step 4:

- $J_1 \in \{4, 5\}$ ,  $CL_4 = 3$ ,  $CL_5 = 4$ ,
- $I_{44} = I_{43} + X_{44} - D_{44} = 0 + X_{44} - 5 = 0$ ,  $X_{44} = 5$
- $I_{55} = I_{54} + X_{55} - D_{55} = 0 - X_{55} - 3 = 0$ ,  $X_{55} = 3$
- $J_2 \in \{2, 3\}$ ,  $CL_2 = 2$ ,  $CL_3 = 3$
- $I_{23} = I_{22} + X_{23} - D_{23} - (X_{44} + X_{55}) = 0 + X_{23} - 5 - (5 + 3) = 0$ ,  $X_{23} = 13$
- $I_{34} = I_{33} + X_{34} - D_{34} = 0 + X_{34} - 3 = 0$ ,  $X_{34} = 3$
- $J_3 \in \{1\}$ ,  $CL_1 = 1$
- $I_{12} = I_{11} + X_{12} - (X_{23} + X_{34}) = 2 + X_{12} - (13 + 3) = 0$ ,

$$X_{12} = 14$$

분배계획으로 다음 기간에 각 노드의 아크 상으로 보내어지는 물량과 이로 인한 기간 말의 재고 현황은 <표 2>에 나타내었다. 이와 함께 분배계획의 결과로 이번 기간 동안 네트워크 전체에서 발생하는 재고 비용의 합은  $39(\times = I_{11} \times h_1 + I_{31} \times h_3 + (-I_{41}) \times b_4 + I_{51} \times h_5)$ 이다.

표 2. 분배계획 수립 결과 및 각 노드별 기간 말의 재고 현황

$X_{12}$	$X_{22}$	$X_{33}$	$X_{42}$	$X_{53}$
14	16	2	5	2
$I_{11}$	$I_{21}$	$I_{31}$	$I_{41}$	$I_{51}$
2	0	2	-3	6

본 연구에서 제시된 모형은 선형계획(linear programming)의 형태를 나타내므로 제안된 분배 알고리즘의 타당성을 검증하기 위해서 선형계획법과 본 연구의 알고리즘을 적용한 결과를 비교하여 실험해 보았다. 제안된 알고리즘은 C언어를 사용하여 프로그래밍하였고 선형계획 프로그램인 LINGO 8.0을 이용하였으며 <표 3>과 같은 실험 조건 하에서 다양한 형태의 네트워크 모형을 임의로 만든 다음 비교 실험을 수행하였다. 비교 실험을 통해 제안된 알고리즘과 선형계획법이 항상 동일한 결과를 얻음을 확인하였고 이로써 제안된 알고리즘의 타당성을 입증할 수 있었다. 또한 이러한 비교 실험을 통해 선형계획법의 경우 많은 노드로 구성된 대단위 문제(large scale problem)나 계속적으로 수요 정보가 변경되는 rolling horizon 환경에 적용하기에 많은 제약이 있는 반면 본 연구에서 제안한 알고리즘은 이보다 쉽고 간단하게 적용 가능함을 알 수 있다.

표 3. 실험 조건

항 목	사용값
노드 수	5, 10, 15, 20, 30
노드별 조달시간	1~5
노드별 재고유지 비용	1~10
노드별 재고이월 비용	20~100

이와 함께 본 연구에서는 제안된 알고리즘이 수요 정보가 주기적으로 갱신되는 rolling horizon 환경 하에서 어느 정도의 수행 시간을 소요하는지 알아보기 위해 노드의 수를 광범위하게 변화시켜가면서 실험을 수행하였고 이 결과를 <그림 4>에 나타내었다. 실험은 펜티엄4의 1.8GHz CPU, 256 RAM을 사용하는 PC에서 수행되었으며 하나의 대상 문제마다 10,000번의 분배계획을 수립하였다. 그림에서 보듯이 노드의 수가 늘어남에 따라 수행 시간 또한 증가하였지만 그 증가의 정도가 완만

함을 알 수 있다. 본 연구에서는 이와 같은 실험을 통하여 제안한 분배 알고리즘은 쉽고 간단하게 여러 형태의 대단위 네트워크 문제에 적용 가능하고 수행 시간의 측면에서도 현실적임을 확인하였다.

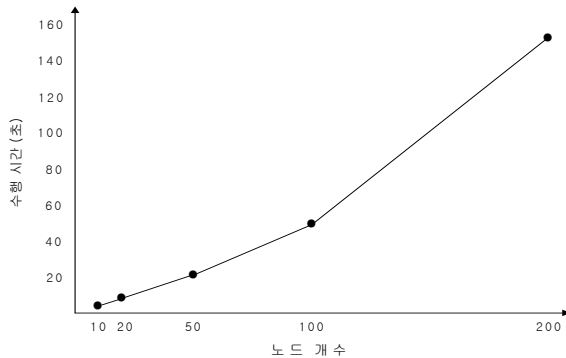


그림 4. 노드 개수에 따른 수행 시간.

## 5. 결론

본 연구에서는 현실적이면서 일반적인 네트워크 모형인 트리 형태의 네트워크 모델에 대해 모델링하고 이에 대한 효율적인 분배계획의 수립 방법을 제시하였으며 이를 rolling horizon 환경 하에서 적용하였다. 또한 선형계획(linear programming)의 결과와의 비교 실험을 통하여 제안된 알고리즘의 타당성과 효율성을 입증하였다.

본 연구에서는 수요정보에 대한 기본적인 가정으로 수요예

측을 통해 얻어지는 확정적인 값을 예측 수요로 사용하는 것이다. 그러나 실제적인 면에서 정확한 수요예측 기법을 통해 매 기간마다 새로이 갱신된 예측 수요를 바탕으로 분배계획을 수립한다 하더라도 예측된 수요는 실제로 발생하는 수요와는 항상 일치하지 않는다. 이런 예측 수요와 실제 수요의 차이는 제안된 알고리즘의 효율성을 저해시키고 이로 인해 전체적인 재고 비용의 합은 증가하게 된다. 따라서 보다 실제적인 면을 반영하고 알고리즘의 효율성을 높이기 위해서는 예측 수요를 확정적(deterministic)으로 주어지는 값이 아니라 확률적(stochastic)인 형태를 나타내는 것으로 가정하여 접근하는 방법이 필요하겠다. 즉, 매 기간마다의 수요 정보는 특정한 분포를 나타내는 것으로 가정하고 이러한 분포 내에 존재하는 여러 값들 중에서 어떠한 값을 예측 수요로 선택하는 것이 재고 비용의 최소화 측면이나 고객에 대한 적절한 서비스 수준(service level)을 맞추어 주는 데 있어 효과적인지에 대한 연구가 추가적으로 진행되어야 한다. 또한 이러한 결과가 rolling horizon 환경 하에서도 효율적인지를 규명하는 실험이 함께 병행되어야 한다 (Chand, 1982; Sung and Park, 1987). 이와 함께 본 연구에서 제시한 모형이 보다 현실적이기 위해서는 네트워크에 물품을 공급하는 공급자의 생산 용량에 대한 제약과 각 노드의 수송 용량이나 물품을 보관하는 창고 용량에 대해 제한을 두는 것이 필요하다. 또한, 단일 제품(single item)이 아닌 복수 제품(multiple item)을 다루는 것으로, 하위 노드는 특정한 하나가 아닌 여러 개의 상위 노드로부터 물품을 공급받는 것이 가능하도록 하는 보다 일반적인 네트워크 모형으로의 확장에 관한 연구 등이 필요하리라 생각된다.

### <부 록>

#### <Solution Procedure>

FOR  $k = 1, 2, \dots, K$

FOR  $i \in J_k$

IF  $S_i = \emptyset$ , THEN compute  $I_{it}$  from  $I_{it} = I_{i(t-1)} + X_{it} - D_{it}$  for  $t = 1, 2, \dots, L_i$ . END IF

IF  $S_i \neq \emptyset$ , THEN

FOR  $t = 1, 2, \dots, L_i$

FOR  $k' = M_i, M_i + 1, \dots, k - 1$

FOR  $j \in S_i$  and  $j \in J_{k'}$

Set  $t' = t + L_{ij}$

IF  $I_{j(t-1)} \geq 0$ , THEN

Set  $X_{jt'}^* = D_{jt'}$  and  $\hat{I}_{j(t-1)} = I_{j(t-1)}$ .

Sort  $X_{jt'}^*$  by descending order of backorder costs for all  $j' \in S_j$  and node  $j$ .

Move first  $X_{jt'}^*$ .

WHILE  $(\hat{I}_{j(t-1)} \neq 0)$  DO

IF  $X_{jt}^* \geq \hat{I}_{j(t-1)}$ , THEN set  $X_{jt}^* = X_{jt}^* - \hat{I}_{j(t-1)}$  and  $\hat{I}_{j(t-1)} = 0$ . END IF  
 IF  $X_{jt}^* < \hat{I}_{j(t-1)}$ , THEN set  $\hat{I}_{j(t-1)} = \hat{I}_{j(t-1)} - X_{jt}^*$  and  $X_{jt}^* = 0$ . END IF  
 Move next  $X_{jt}^*$ .  
 END WHILE  
 END IF  
 IF  $I_{j(t-1)} < 0$ , THEN set  $X_{jt}^* = D_{jt} - I_{j(t-1)}$ . END IF  
 END FOR  $j$   
 Set  $\hat{X}_{jt} = X_{jt}^* + \sum_{j \in S_i} X_{j(t+L_i)}^*$ .  
 END FOR  $k'$   
 Compute the  $I_{it}$  from  $I_{it} = I_{i(t-1)} + X_{it} - D_{it} - \sum_{j \in S_i} \hat{X}_{j(t+L_i)}$ .  
 IF  $I_{it} \geq 0$ , THEN set  $X_{jt} = \hat{X}_{jt}$  for all  $j \in S_i$ . END IF  
 IF  $I_{it} < 0$ , THEN  
 IF  $I_{i(t-1)} \geq 0$ , THEN set  $X_{it}^* = D_{it}$ . END IF  
 IF  $I_{i(t-1)} < 0$ , THEN set  $X_{it}^* = D_{it} - I_{i(t-1)}$ . END IF  
 Sort  $X_{jt}^*$  by descending order of backorder costs for all  $j \in S_i$  and node  $i$ .  
 Set  $\hat{I}_{i(t-1)} = I_{i(t-1)} + X_{it}$  and move first  $X_{jt}^*$ .  
 WHILE ( $\hat{I}_{i(t-1)} \neq 0$ ) DO  
 IF  $X_{jt}^* \geq \hat{I}_{i(t-1)}$ , THEN set  $X_{jt}^* = \hat{I}_{i(t-1)}$  and  $\hat{I}_{i(t-1)} = 0$ . END IF  
 IF  $X_{jt}^* < \hat{I}_{i(t-1)}$ , THEN set  $X_{jt}^* = X_{jt}^*$  and  $\hat{I}_{i(t-1)} = \hat{I}_{i(t-1)} - X_{jt}^*$ . END IF  
 Move next  $X_{jt}^*$ .  
 END WHILE  
 Set  $X_{jt}^* = 0$  for remaining  $X_{jt}^*$ .  
 Compute  $X_{jt}$  from  $X_{jt} = X_{jt}^* + \sum_{j \in S_i} X_{j(t+L_i)}^*$  for  $j \in S_i$ .  
 END IF  
 Compute  $I_{it}$  from  $I_{it} = I_{i(t-1)} + X_{it} - D_{it} - \sum_{j \in S_i} X_{j(t+L_i)}$ .  
 Compute  $I_{jt}$  from  $I_{jt} = I_{j(t-1)} + X_{jt} - D_{jt} - \sum_{j \in S_i} X_{j(t+L_i)}$  for all  $j \in S_i$ .  
 END FOR  $t$   
 END IF  
 END FOR  $j$   
 END FOR  $k$   
 FOR  $k = 1, 2, \dots, K$   
 FOR  $i \in J_k$   
 Set  $t = CL_i + 1$ .  
 Compute  $X_{it}$  from  $X_{it} = \max(D_{it} + \sum_{j \in S_i} X_{j(t+L_i)} - I_{i(t-1)}, 0)$ .  
 Compute  $I_{it}$  from  $I_{it} = I_{i(t-1)} + X_{it} - D_{it} - \sum_{j \in S_i} X_{j(t+L_i)}$ .  
 END FOR  $i$   
 END FOR  $k$



## 참고문헌

- Afentakis, P., B. Gavish and U. S. Karmarkar (1984), Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multi-Stage Assembly Systems, *Management Science*, 30, 222-239.
- Axsäter, S. (2000), *Inventory Control*, Kluwer Academic Publishers.
- Baker, K. R. (1989), Lot-Sizing Procedures and a Standard Data Set: A Reconciliation of the Literature, *Journal of Manufacturing and Operations Management*, 2, 199-221.
- Blackburn, J. D. and R. A. Millen (1982), Improved Heuristic for Multi-Echelon Requirements Planning Systems, *Management Science*, 28, 44-56.
- Chand, S. (1982), A Note on Dynamic Lot Sizing in a Rolling-Horizon Environment, *Decision Sciences*, 13, 113-119.
- Chopra, S. and Meindi, P. (2001), *Supply Chain Management: Strategy, Planning, and Operation*, Prentice Hall, Upper Saddle River, New Jersey.
- Clark, A. J. and Scarf, H. (1960), Optimal Policies for a Multi-Echelon Inventory Problem, *Management Science*, 6, 475-490.
- Diks, E. B. and De Kok, A. G. (1999), Computational Results for the Control of a Divergent N-Echelon Inventory System, *International Journal of Production Economics*, 59, 327-336.
- Erenguc, S. S., Simpson, N. C. and Vakharia, A. J. (1999), Integrated Production/Distribution Planning in Supply Chains: An Invited Review, *European Journal of Operational Research*, 115, 219-236.
- Federgruen, A. (1993), Centralized Planning Models for Multi-Echelon Inventory Systems under Uncertainty, in Graves, S.C., Rinooy Kan, A.H.G., and Zipkin, P.H. (Eds.), *Logistics of Production and Inventory*, North-Holland, Amsterdam, 133-174.
- Muckstadt, J. A. (1973), A Model for a Multi-Item, Multi-Echelon, Multi-Identure Inventory System, *Management Science*, 20, 472-481.
- Sherbrooke, C. C. (1968), METRIC: A Multi-Echelon Technique for Recoverable Item Control, *Operations Research*, 16, 122-141.
- Silver, E. A. and H. C. Meal (1973), A Heuristic for Selecting Lot Size Requirements for the Case of a Deterministic Time-Varying Demand Rate and Discrete Opportunities for Replenishment, *Production and Inventory Management*, 14, 64-74.
- Sung, C. S. and Park, Y. S. (1987), Dynamic Lot Sizing for a Single-Facility Multi-Product Problem in a Rolling-Horizon Environment, *Decision Sciences*, 18, 266-278.
- Wagner, H. M. and T. M. Whitin (1958), Dynamic Version of the Economic Lot Size Model, *Management Science*, 5, 89-96.
- Zangwill, W. I. (1966), A Deterministic Multi-Period Production Scheduling Model with Backlogging, *Management Science*, 13, 105-119.



## 안재성

고려대학교 산업공학 학사  
 현재: 고려대학교 산업시스템정보공학 석사  
 과정  
 관심분야: SCM, 생산계획



## 김성식

고려대학교 기계공학 학사  
 고려대학교 산업공학 석사  
 Southern Methodist University 산업공학 석사  
 Southern Methodist University 산업공학 박사  
 현재: 고려대학교 산업시스템정보공학과  
 교수  
 관심분야: Queuing Theory, 생산 및 재고관리  
 시스템, CIM/ERP/SCM



## 권익현

고려대학교 산업공학 학사  
 고려대학교 산업공학 석사  
 현재: 고려대학교 산업시스템정보공학 박사  
 과정  
 관심분야: 생산 및 물류관리, 시스템 최적화,  
 e-Business