IE Interfaces
Vol. 17, Special Edition, pp. 90-96, December 2004.

공급사슬에서 다품목 일괄구매 및 조달 일정계획에 관한 연구

차병철 · 문일경†

부산대학교 산업공학과

Joint Replenishment and Delivery Scheduling in a Supply Chain

Byung-Chul Cha, Il-Kyeong Moon

Department of Industrial Engineering, Pusan National University, Busan, 609-735

In this paper, we consider the joint replenishment problem of a one-warehouse, *n*-retailer system. We introduce a joint replenishment and delivery strategy of a warehouse and develop a mathematical model based on the proposed strategy. Two efficient algorithms are presented and compared using numerical examples. The proposed strategy is compared with the common cycle strategy for 1,600 randomly generated problems, and has been proven to be superior to the common cycle strategy.

Keywords: joint replenishment problem, one-warehouse *n*-retailer problem, RAND

1. 서 론

다수의 품목을 개별적으로 주문하기보다는 묶어서 한꺼번에 주문하는 경우 수송비용과 주문비용을 줄일 수 있으며, 같은 공급자에게서 구매하는 경우에는 가격할인까지 기대할 수 있다. 특히 다수의 품목을 다수의 해외 공급자를 통해 구매하고 있는 국내 물류센터의 경우에는 일괄구매를 통해 관련 비용을 상당히 줄일 수 있을 것으로 본다. 본 연구에서는 다품목 일괄구매 모형으로 잘 알려져 있는 Joint Replenishment Problem (JRP)을 확장하여 공급사슬, 특히 물류센터에서 적용할 수 있는 일괄구매 및 조달일정계획 모형을 개발하고 그 해법을 소개하고자 한다.

간단한 비용함수임에도 불구하고 Arkin *et al.*(1989)은 JRP가 NP-hard 문제임을 증명하였다. 이는 JRP에서 고려되는 품목 수 n이 증가하게 되면 한정된 시간 안에 최적해를 찾기가 거의 불가능하다는 것을 의미한다. 이 때문에 지난 수십여 년 간 근사 최적해를 찾기 위한 많은 발견적 기법(heuristic method)들이 연구되어졌다. Goyal(1973,1974)은 최적화 조건을 동시에 만족하는 기본주기 T와 각 품목의 발주시기 k_i를 열거법(enumeration approach)을 이용하여 구하였고, 자신의 발견적 기법이 항상 최적해를 구할 수 있음을 주장하였다. 그러나 이후 Van Eijs(1993)는 Goyal(1974)이 열거법을 통해 구한 해가 최적해를 보장할 수 없음을 보였다. Silver(1975,1976)는 일괄구매의 장 · 단점을 고찰하고 간단한 비반복적인 절차를 통해 해를 구하는 발견적 기법을 제시하였다. 그러나 closed-form으로 한번의 절차를 거쳐 해를 구할 수 있는 장점에도 불구하고 정수해 k_i를 구하기 위해 반올림 근사를 이용했기 때문에 다른 발견적 기법에 비해 성능이 우수하진 못했다. Kaspi and Rosenblatt(1983,1991)은 반복적인 절차를 통해 T와 k_i를 구하는 발견적 기법이 초기해에 따라 서로 다른 지역 최적해(local minimum)로 수렴함을 알고 m개의 다양한 초기해를 통해 개선된 근사 최적해를 찾도록 하는 발견적 기법을 개발하고 그 알고리듬을 RAND라고 명명하였다. 이 발견적 기법에서는 먼저 기본주기 T가 가질 수 있는 최소, 최대영역을 정의하고 이를 m등분하여 각 초기 T_j에 대해 지역 최적해를 구하기 위한 반복적인 발견적 기법을 적용하였다. 이를 통해 m개의 지역 최적해를 구하고 그 중 최소값

본 연구는 교육부에서 주관하는 차세대물류IT기술연구사업단에 의해 지원받은 연구임.
†연락저자 : 문일경 교수, 609-735 부산광역시 금정구 장전동 산 30번지 부산대학교 산업공학과, Fax : 051-512-7603,
E-mail : ikmoon@pusan.ac.kr

을 근사 최적해로 구하였다. 실험을 통해 이 알고리듬은 지금까지 개발된 많은 발견적 기법에 비해 상당히 우수한 해를 제공하는 것으로 알려져 있다.

One-warehouse, n-retailer 문제도 JRP와 같이 많은 연구가 이루어져 오고 있다 (Silver,1998). 특히, 최근 이슈가 되고 있는 공급사슬관리와 관련하여 다단계 의사결정으로서 그 중요성이 확대되고 있다. Schwarz(1973)는 one-warehouse, n-retailer 문제가 상당히 복잡한 문제임을 언급하면서, basic policy를 제안하였다. Basic policy는 이후 관련 연구의 핵심이 되는 정책으로 다음의 특성들을 가진다. "Warehouse의 주문은 warehouse의 재고가 0이거나 적어도 하나의 retailer의 재고가 0일 때만 일어난다. 또한 retailer에 대한 조달은 retailer의 재고가 0일 때만 일어난다. 그리고 각 retailer에 대해 일정한 횟수로 조달된 양과 warehouse의 주문량은 서로 같다." 그는 또한 one-warehouse, one-retailer에 대해 (n, Q) 정책을 제안하였다. 이 정책은 warehouse가 같은 물량을 동일한 시간간격으로 retailer에게 조달하는 정책이다. Graves (1979)는 JRP가 one-warehouse, n-retailer 문제와 매우 밀접한 연관이 있음을 보였으며, Lu and Posner(1994)는 one-warehouse, n-retailer 문제에 대한 두 가지 발견적 기법을 제안하며, 이를 JRP로 확장할 수 있음을 언급하였다. 최근에 Abdul-Jalbar *et al.* (2003)은 one-warehouse, n-retailer 문제에 대한 기존의 연구방법들을 정리하고 warehouse 중심의 의사결정인 중앙화(centralization)와 개별 retailer 중심의 의사결정인 분산화(decentralization) 전략 모형을 정의하고 그 효과에 대한 연구를 발표하였다.

본 연구는 앞선 연구와는 달리 one-warehouse, n-retailer 모형에 다품목의 일괄구매 상황을 고려하여 다품목을 취급하는 물류창고에서 이용할 수 있는 일괄구매 및 조달일정계획문제에 대한 모형을 개발, 소개하고자 한다. 해법을 위해서는 일반적인 일괄구매 모형(JRP)에서처럼 각 의사결정변수들의 최적성 조건들을 이용한 반복 알고리듬을 개발하고, 그 절차들을 수치예제를 통해 보여줄 것이다. 실험을 통해 개발된 알고리듬의 성능은 공통주기를 사용하는 경우와 비교, 평가될 것이다.

2. 물류창고에서의 일괄구매모형

일반적인 다품목 일괄구매 모형(JRP)은 각 품목의 수요(D_i)가 일정한 경우 다품목 일괄구매 모형에서는 단일품목의 경제적 발주량(EOQ) 모형에서와 같이 주문비용과 재고유지비용의 합으로 구성되어지는 다음과 같은 단위기간당 총비용을 최소로 하는 구매전략을 구하는 것이 목적이다.

$$TC(T,k_1,k_2,\cdots,k_n)=\frac{S+\sum_{i=1}^{n}\frac{s_i}{k_i}}{T}+\sum_{i=1}^{n}\frac{D_i k_i T h_i}{2}$$

고려되는 주문비용은 크게 구매 의사결정에 따라 공동으로 소요되는 major ordering cost(S)와 개별품목에 소요되는 minor ordering cost(s_i)로 나누어진다. 그리고 각 제품별 재고유지비용은 단위기간 동안 한 단위당 h_i가 소요된다. 다품목 일괄구매 모형의 가장 큰 특징은 모든 품목을 기본주기 T의 정수배인 k_iT 주기마다 발주함으로써 일괄구매에 따른 비용절감 효과를 얻고자 하는 것이다. 즉 이 문제는 총비용을 최소로 하는 기본주기 T와 각 품목의 발주주기를 결정하는 n개의 정수 k_i를 구하는 것으로 요약된다.

본 연구에서는 다품목 일괄구매 모형을 확장하여, 공급사슬상의 물류창고에서 일괄구매를 통해 구매한 다수의 품목을 각 품목이 필요한 생산자나 소매점에게 공급하는 모형을 개발하고자 한다. <그림 1>과 같이 다수의 품목을 해외로부터 구매하여 생산자나 소매점들에게 조달하는 물류창고를 고려해 보

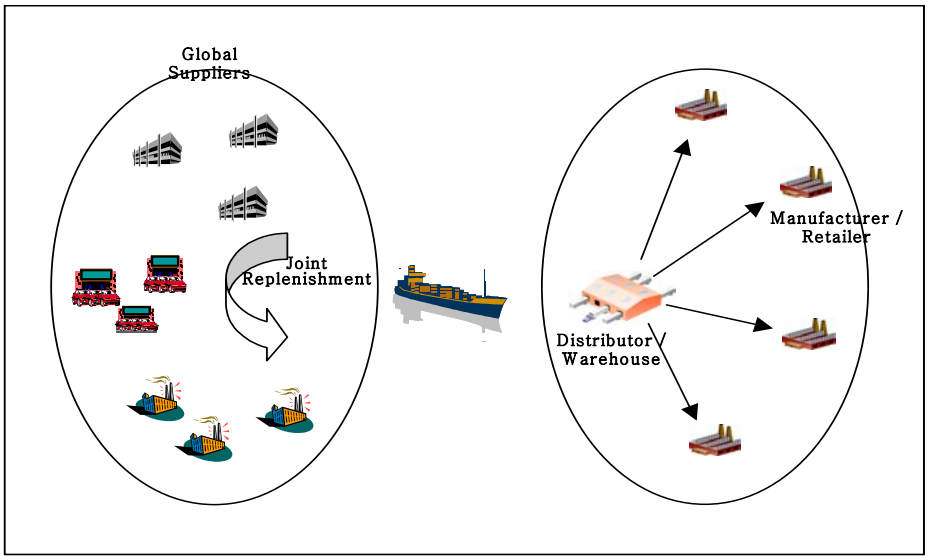

그림 1. 물류창고에서의 일괄구매 전략.

자. 물류창고의 구매환경에서 일반적인 일괄구매 모형(JRP)에서와 같이 구매에 따른 major ordering cost가 발생한다면, 기존의 one-warehouse, *n*-retailer 문제와는 다른 일괄구매 전략을 고려할 필요가 있다. 물류창고에서 각 소매점으로의 조달은 one-warehouse, one-retailer 전략에서 일반적으로 사용되어지고 있는 동일 시간간격 조달정책을 사용할 것이다.

본 연구모형에서 이용될 기호들은 다음과 같다.

i : 품목에 대한 index, i=1, 2, .., m
D_i : 품목 i에 대한 수요율
S^W : 물류창고의 major ordering cost
s_i^w : 물류창고의 품목 i에 대한 minor ordering cost
h_i^w : 물류창고의 품목 i에 대한 재고유지비용 (단위, 단위시간당)
s_i^R : 물류창고의 품목 i에 대한 소매점으로의 조달비용
h_i^R : 소매점의 품목 i에 대한 재고유지비용 (단위, 단위시간당)
T : 물류창고의 기본 주문주기(basic cycle time) - 의사결정 변수
k_i : 품목 i에 대한 구매주기를 결정하는 정수 - 의사결정변수
f_i : 품목 i에 대한 k_iT 주기 동안의 조달횟수 - 의사결정변수

관련 비용으로서 물류창고의 주문비용(major & minor)과 재고유지비용, 각 소매점으로의 조달비용과 각 소매점에서의 재고유지비용이 고려되어진다. 품목 i에 대해 물류창고에서 각 소매점으로의 동일 시간간격 조달정책에 따른 물류창고 및 소매점 i에 대한 재고수준의 변화가 <그림 2>에 나타나 있다.

<그림 2>에서와 같이 품목 i는 물류창고의 일괄구매 정책에 따라 k_iT 주기마다 발주가 이루어지고 f_i회 나누어서 소매점 i에 조달된다. 이에 따라 물류창고의 일괄구매 및 조달에 따른 단위기간당 총비용은 다음과 같이 표현된다.

$$TC(T,k_1,k_2,\cdots,k_n,f_1,f_2,\cdots,f_n) = \frac{S^W+\sum_{i=1}^{n}\frac{s_i^W}{k_i}}{T}+\sum_{i=1}^{n}\frac{(f_i-1)k_iTD_ih_i^W}{2f_i} + \sum_{i=1}^{n}\frac{f_is_i^R}{k_iT}+\sum_{i=1}^{n}\frac{k_iTD_ih_i^R}{2f_i} \quad (1)$$

본 모형에서 일반적인 다품목 일괄구매 모형(JRP)과 같이, 제시된 총비용 함수로부터 각 의사결정변수들의 최적성 조건을 구하면 다음과 같다.

첫째, 주어진 일련의 k_i와 f_i값들에 대하여, $TC(T)$는 convex 함수임이 명확하다. 곧 TC를 T에 대해 미분함으로써 다음과 같이 T에 대한 최적성 조건을 쉽게 구할 수 있다.

$$T^* = \sqrt{\frac{2\left(S^W+\sum_{i=1}^{n}\frac{s_i^W+f_is_i^R}{k_i}\right)}{\sum_{i=1}^{n}k_iD_i\left(h_i^W+\frac{h_i^R-h_i^W}{f_i}\right)}} \quad (2)$$

둘째, 주어진 일련의 f_i 값들과 T 에 대하여 $TC(k_i)\le TC(k_i+1)$ 와 $TC(k_i)\le TC(k_i-1)$ 를 만족하는 정수 k_i에 대한 최적성 조건은 다음과 같이 구해진다.

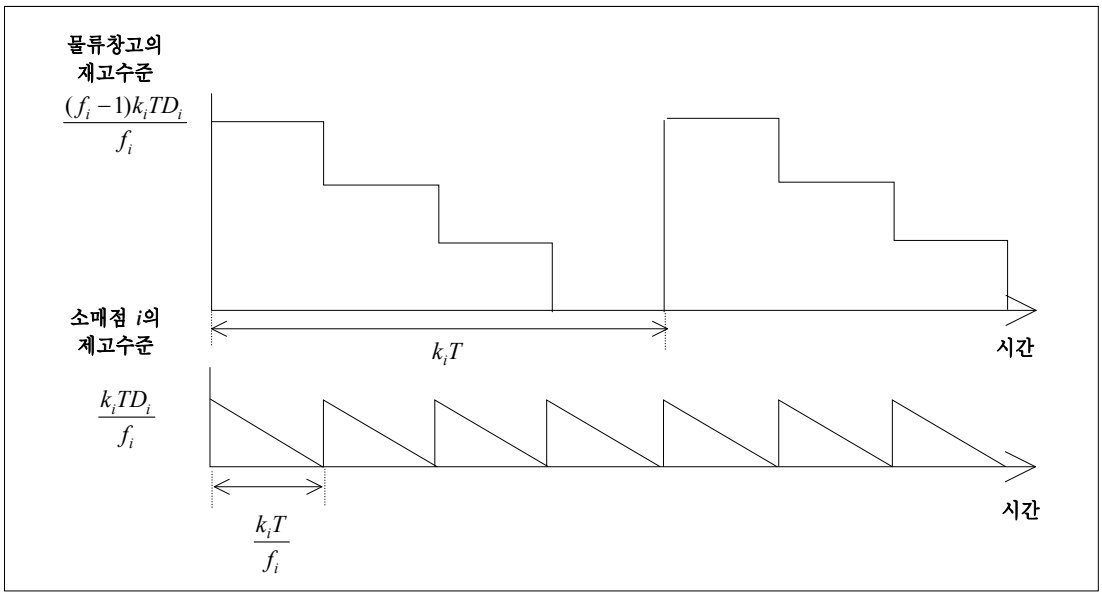

그림 2. 물류창고 및 소매점 i에 대한 재고수준의 변화.

$$k_i(k_i-1) \le \frac{2(s_i^w + f_i s_i^R)}{T^2 D_i \left(h_i^w + \frac{h_i^R - h_i^w}{f_i} \right)} \le k_i(k_i+1) \quad (3)$$

마지막으로, 주어진 일련의 k_i값들과 T에 대하여 $TC(f_i) \le TC(f_i+1)$ 와 $TC(f_i) \le TC(f_i-1)$ 를 만족하는 정수 f_i에 대한 최적성 조건은 다음과 같이 구해진다.

$$f_i(f_i-1) \le \frac{k_i^2 T^2 D_i (h_i^R - h_i^w)}{2 s_i^R} \le f_i(f_i+1) \quad (4)$$

단, 여기서 $h_i^R - h_i^w$이면 항상 f_i=1이 된다. 이것은 품목 i에 대한 물류창고의 재고유지비용이 소매점보다 클 경우 구매된 품목 i를 물류창고에 보관하지 않고 바로 소매점으로 조달함을 의미한다(crossdocking의 의미가 된다).

각 의사결정변수들의 최적성 조건인 식 (2)~(4)를 이용하여 다음과 같이 간단한 반복적 절차를 통해 근사 최적해를 구해낼 수 있다.

반복적 알고리듬

1단계: iteration number r=0. 모든 품목에 대하여 $k_i(r)$=1, $f_i(r)$=1로 둔다. $T(r)$=0

2단계: $r = r$+1

3단계: 주어진 $k_i(r$-1)과 $f_i(r$-1) 값들을 가지고 식 (2)를 이용하여 T의 최적해를 구한다.
$T(r)$=T. 만일 $T(r)$= $T(r$-1)이면, 알고리듬을 종료.
그렇지 않으면, 다음 4단계로 이동한다.

4단계: 주어진 $T(r)$과 $f_i(r$-1) 값들을 가지고 식 (3)을 이용하여 각 품목 i에 대한 k_i의 최적해를 구한다. 각 품목 i에 대하여 $k_i(r) = k_i$.

5단계: 주어진 $T(r)$과 $k_i(r)$ 값들을 가지고 식 (4)를 이용하여 각 품목 i에 대한 f_i의 최적해를 구한다. 각 품목 i에 대하여 $f_i(r) = f_i$. 2단계로 이동한다.

다음의 수치예제를 통해 제시된 알고리듬의 절차를 보이고자 한다. 본 예제에서는 6개의 품목을 취급하는 물류창고를 고려하고 있으며, 다음 <표 1>은 각 품목 i에 대한 수요율과 관련된 비용들을 나타내고 있다. 여기서 S^W=$200로 가정한다.

표 1. 수치예제에 대한 관련 비용

품목	1	2	3	4	5	6
D_i	10,000	5,000	3,000	1,000	600	200
s_i	45	46	47	44	45	47
h_i^w	1	1	1	1	1	1
s_i^R	5	5	5	5	5	5
ι	1.5	1.5	1.5	1.5	1.5	1.5

<표 2>는 반복적 알고리듬을 적용한 결과를 각 단계별로 보여주고 있다. 적은 횟수의 반복절차를 통해 지역 최적해를 구할 수 있음을 확인할 수 있다.

표 2. 반복적 알고리듬에 대한 절차 및 결과

r	T(r) 2단계	$ki(r)$ 3단계	$fi(r)$ 4단계	TC
0	0	1, 1, 1, 1, 1, 1	1, 1, 1, 1, 1, 1	
1	0.1842	1, 1, 1, 1, 2, 3	4, 3, 2, 1, 2, 2	
2	0.1973	1, 1, 1, 1, 2, 3	4, 3, 2, 1, 2, 2	
3	0.1973			$4850.39

3. 수정된 RAND 알고리듬

전술한 바와 같이 반복 알고리듬은 지역 최적해로 수렴할 가능성이 많다. 본 장에서는 일반적인 JRP 모형에 대한 RAND 알고리듬의 아이디어를 적용하여 반복적 알고리듬을 수정하고, 수치예제를 통해 그 결과를 비교해 보고자 한다. 수정된 알고리듬의 절차는 다음과 같다.

수정된 RAND 알고리듬

1단계: $T_{max} = \sqrt{2\left(S + \sum_{i=1}^{n} s_i \right) / \sum_{i=1}^{n} D_i h_i}$ 및

$T_{min} = min \sqrt{\frac{2s_i}{D_i h_i}}$ 을 계산한다.

2단계: [T_{min}, T_{max}]를 m개의 등구간으로 나눈다.
(T_1, ⋯, T_j, ⋯, T_m). j=0.

3단계: $j=j$+1, r=0.
$T_j(r)=T_j$, 모든 품목에 대하여 $f_i(r)$=1로 둔다.

4단계: $r=r$+1.

5단계: 주어진 $T(r$-1)과 $f_i(r$-1) 값들을 가지고 식 (3)을 이용하여 각 품목 i에 대한 k_i의 최적해를 구한다. 각 품목 i에 대하여 $k_i(r) = k_i$.

6단계: 주어진 $T(r$-1)과 $ki(r)$ 값들을 가지고 식 (4)를 이용하여 각 품목 i에 대한 f_i의 최적해를 구한다. 각 품목 i에 대하여 $f_i(r) = f_i$.

7단계: 주어진 $k_i(r)$과 $f_i(r)$ 값들을 가지고 식 (2)를 이용하여 T의 최적해를 구한다. $T(r)$=T.

8단계: 만일 $T(r) \neq T(r$-1)이면, 4단계로 이동.
그렇지 않으면, $T_j^* = T_j(r)$, 각 품목 i에 대하여
$k_{ij}^* = k_i^*(r)$, $f_{ij}^* = f_i^*(r)$.
$\left(T_j, K_{ij}^*, J_{ij}^* \right)$ 에 대하여 TC_j를 계산한다.

9단계: 만일 $j \neq m$이면, 3단계로 이동.
그렇지 않으면, 종료 후 최소의 TC_j를 갖는 $\left(T_j^*, k_{ij}^* s, f_{ij}^* s\right)$를 선택한다.

앞에서 사용한 동일한 수치예제를 가지고 수정된 RAND 알고리듬을 적용한 경우의 절차 및 결과가 <표 3>에 자세히 기술되어 있다.

<표 4>에서는 앞의 두 알고리듬에 대한 결과를 공통주기법(Common Cycle Method)을 사용한 경우와 비교하고 있다. 본 수치예제에 대하여 수정된 RAND 알고리듬은 반복적 알고리듬에 비해 좀더 나은 결과를 보여주고 있으며, 공통주기법을 사용한 경우에 비해서는 상당한 개선효과가 있다는 것을 확인할 수 있다.

4. 수치실험 및 결과

개발된 알고리듬들을 평가하기 위해서 <표 5>와 같이 각 관련 모수들을 일양분포로부터 랜덤하게 발생시켰다. 품목 수 n에 대한 네 가지 경우 (n=10, 20, 30, 50)와 major ordering cost S^W에 대한 네 가지 경우 (S^W=100, 200, 300, 400)를 조합하여 각 100문제씩, 총 1,600개의 문제를 생성하여 공통주기법, 반복적 알고리듬, 수정된 RAND 알고리듬에 대해 비교실험을 수행하였다. 또한, 수정된 RAND 알고리듬에 대해서는 적정한 초기해의 수 m을 결정하기 위해 총 네 가지 경우 (m=0.5n, n, 2n, 4n)를 고려하여 실험을 수행하였다.

표 5. 관련 모수의 데이터 범위

D_i	s_i^w	s_i^R	h_i^w	h_i^R
[500, 5000]	[30, 50]	[0.1 s_i^w, 0.3 s_i^w]	[0.5, 3.0]	[1.2 h_i^w, 2.0 h_i^w]

수행된 결과가 <표 6>과 <표 7>에 정리되어 있다. <표 6>은 품목 수 n과 S^W의 16개 조합에 대해 수행된 각 100회의 실험에서 각 알고리듬이 최소값을 찾은 횟수를 보여주고 있다. <표 6>으로

표 3. 수정된 RAND 알고리듬에 대한 절차 및 결과

T_j	r	$k_i(r)$ 5단계	$f_i(r)$ 6단계	$T(r)$ 7단계	TC_j
T_1=0.0949	1	1, 1, 2, 3, 4, 6	2, 2, 2, 2, 2, 2	0.1486	
	2	1, 1, 1, 2, 3, 5	3, 2, 2, 2, 2, 2	0.1763	
	3	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	
	4	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	\$4828.89
T_2=0.1259	1	1, 1, 1, 2, 3, 5	3, 2, 2, 2, 2, 2	0.1763	
	2	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	
	3	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	\$4828.89
T_3=0.1568	1	1, 1, 1, 2, 2, 4	4, 3, 2, 2, 2, 2	0.1870	
	2	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	
	3	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	\$4828.89
T_4=0.1878	1	1, 1, 1, 1, 2, 3	4, 3, 2, 1, 2, 2	0.1973	
	2	1, 1, 1, 1, 2, 3	4, 3, 2, 1, 2, 2	0.1973	\$4850.39
T_5=0.2188	1	1, 1, 1, 1, 2, 3	5, 3, 3, 2, 2, 2	0.2035	
	2	1, 1, 1, 2, 2, 4	5, 3, 3, 3, 2, 2	0.1940	
	3	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 3	0.1886	
	4	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	
	5	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	0.1881	\$4828.89

표 4. 알고리듬들에 대한 결과비교

구 분	T	k_i	f_i	TC	%
공통주기법	0.2215	1, 1, 1, 1, 1, 1	5, 4, 3, 2, 1, 1	\$5001.31	3.57
반복적 알고리듬	0.1973	1, 1, 1, 1, 2, 3	4, 3, 2, 1, 2, 2	\$4850.39	0.45
수정된 RAND 알고리듬	0.1881	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 2, 2	\$4828.89	-

표 6. 최소값을 찾은 횟수

n	S^W	공통주기법	반복적 알고리듬	수정된 RAND 알고리듬			
				$m=0.5n$	$m=n$	$m=2n$	$m=4n$
10	100	9	56	96	97	100	100
	200	16	68	97	98	100	100
	300	30	73	98	99	100	100
	400	35	84	98	99	100	100
20	100	0	21	90	95	97	100
	200	0	39	85	96	96	100
	300	1	43	93	99	99	100
	400	2	50	96	97	99	100
30	100	0	8	77	92	98	100
	200	0	15	81	92	96	100
	300	0	30	88	96	100	100
	400	1	37	93	99	99	100
50	100	0	2	73	91	95	100
	200	0	4	83	90	98	100
	300	0	5	84	97	99	100
	400	0	8	78	92	98	100
Max.		35	84	98	99	100	100
Avg.		6	34	88	96	98	100

표 7. 수정된 RAND 알고리듬 ($m=4n$)과의 비교 (%)

n	S^W	공통주기		반복적 알고리듬		수정된 RAND 알고리듬					
						$m=0.5n$		$m=n$		$m=2n$	
		Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
10	100	4.4981	1.3608	0.8432	0.0863	0.1546	0.0020	0.1545	0.0016	0.0000	0.0000
	200	2.7094	0.7450	0.6675	0.0338	0.1208	0.0012	0.1208	0.0012	0.0000	0.0000
	300	1.6622	0.4526	0.3317	0.0315	0.1521	0.0016	0.1521	0.0015	0.0000	0.0000
	400	1.3211	0.2845	0.2509	0.0206	0.0088	0.0001	0.0088	0.0001	0.0000	0.0000
20	100	4.5056	2.1163	0.9543	0.2253	0.0931	0.0027	0.0405	0.0012	0.0405	0.0006
	200	3.5416	1.4383	0.7488	0.0920	0.1430	0.0069	0.0401	0.0008	0.0401	0.0008
	300	2.8448	1.0351	0.4965	0.0579	0.1249	0.0027	0.0411	0.0004	0.0411	0.0004
	400	2.2595	0.7659	0.3078	0.0367	0.0836	0.0011	0.0068	0.0002	0.0051	0.0001
30	100	4.4798	2.4529	1.1259	0.2499	0.1763	0.0084	0.0764	0.0026	0.0103	0.0002
	200	3.4031	1.8641	0.8044	0.1311	0.0840	0.0055	0.0656	0.0020	0.0316	0.0008
	300	2.8020	1.4661	0.4842	0.0823	0.0369	0.0016	0.0215	0.0005	0.0000	0.0000
	400	2.3622	1.1716	0.4545	0.0553	0.0483	0.0012	0.0107	0.0001	0.0107	0.0001
50	100	4.7648	2.6441	1.1332	0.3654	0.0694	0.0055	0.0694	0.0022	0.0443	0.0009
	200	4.0959	2.1811	0.7170	0.2287	0.0605	0.0032	0.0510	0.0013	0.0187	0.0002
	300	3.6484	1.8279	0.4928	0.1463	0.0760	0.0037	0.0407	0.0007	0.0407	0.0004
	400	3.2255	1.5573	0.3691	0.0976	0.0647	0.0025	0.0216	0.0007	0.0055	0.0001
Max.		4.7648		1.1332		0.1763		0.1545		0.0443	
Avg.			1.4602		0.1213		0.0031		0.0011		0.0003

부터 수정된 RAND 알고리듬이 다른 두 알고리듬 (공통주기법, 반복적 알고리듬) 보다 항상 우수함을 확인할 수 있다. <표 7>은 가장 좋은 해를 제공하는 수정된 RAND 알고리듬 ($m=4n$)의 성능과 다른 알고리듬들을 비교하여 보여주고 있다. 특히, Kaspi and Rosenblatt (1991)의 실험에서처럼 수정된 RAND 알고리듬에 대해서는 m의 값이 증가할수록 알고리듬의 해가 개선됨을 확인할 수 있다. 비록 m의 크기에 대해 수행시간이 크게 문제될 것은 없지만, n=20, 30, 50에 대해서는 초기해 m의 수를 품목 수 n (m=20, 30, 50)만큼만 고려해도 좋은 해를 얻을 수 있었다 ($m=4n$와 비교하여 최대 0.0764%만큼만 오차가 발생). <표 7>에서 JRP를 이용한 수정된 RAND 알고리듬은 공통주기법보다 최대 4.7648%, 평균 1.4602%만큼 우수한 결과를 보임을 확인할 수 있다.

5. 결론 및 추후연구

본 연구는 one-warehouse, n-retailer 모형에 다품목의 일괄구매 상황을 고려하여 모형을 정의하고, 일반적인 다품목 일괄구매 모형(JRP)에 사용되어지는 반복 알고리듬의 개념을 적용하여 모형을 개발하였다. 단순한 반복 알고리듬이 지역 최적해에 수렴함을 고려하여, RAND 알고리듬의 아이디어를 이용한 수정된 RAND 알고리듬을 개발하였다. 수치예제를 통해 두 알고리듬의 절차 및 결과를 비교해 보았고, 1,600개의 문제를 랜덤하게 발생하여 비교실험을 수행하였다. 실험을 통해 물류창고의 JRP를 활용한 수정된 RAND 알고리듬이 공통주기법을 사용하는 경우보다 상당히 개선된 결과를 얻을 수 있음을 보였다. 본 연구에서는 물류창고에서 각 품목 i에 대해 동일 시간간격 조달하는 정책을 사용하고 있지만, 이러한 조달정책이 일반적으로 최적정책이 아님이 알려져 있다. 이와 관련, 최적정책에 관한 연구가 현재 진행중에 있다. 한편 자원에 대한 제약조건이 있는 보다 복잡한 문제에 대하여는 유전자 알고리듬(genetic algorithm)이 적용될 수 있으며 의미 있는 연구가 될 것으로 보인다.

참고문헌

Abdul-Jalbar, B., Gutierrez, J., Puerto, J. and Sicilia, J. (2003), Policies for inventory/distribution systems: The effect of centralization vs. decentralization, International Journal of Production Economics, **81-82**, 281~293.

Arkin, E., Joneja, D. and Roundy, R. (1989), Computational complexity of uncapacitated multi-echelon production planning problems, Operations Research Letters, **8**, 61-66.

Goyal, S. (1973), Determination of economic packaging frequency for items jointly replenished, Management Science, **20**, 232-238.

Goyal, S. (1974), Determination of optimum packaging frequency of items jointly replenished, Management Science, **23**, 436-443.

Graves, S. (1979), On the deterministic demand multi-product single machine lot scheduling problem, Management Science, **25**, 276-280.

Kaspi, M. and Rosenblatt, M. (1983), An improvement of Silver's algorithm for the joint replenishment problem, IIE Transactions, **15**, 264-269.

Kaspi, M. and Rosenblatt, M. (1991), On the economic ordering quantity for jointly replenished items, International Journal of Production Research, **29**, 107-114.

Lu, L and Posner, M. (1994), Approximation procedures for the one- warehouse multi-retailer system, Management Science, **40**, 1305-1316.

Schwarz, L. (1973), A simple continuous review deterministic one-warehouse n-retailer inventory problem, Management Science, **19**, 555-566.

Silver, E. (1975), Modifying the economic order quantity (EOQ) to handle coordinated replenishment of two or more items, Production & Inventory Management, **16**, 26-38.

Silver, E. (1976), A simple method of determining order quantities in jointly replenishments under deterministic demand, Management Science, **22**, 1351-1361.

Silver, E., Pyke, D. and Peterson, R. (1998), Inventory Management and Production Planning and Scheduling, 3rd edition, John Wiley & Sons, New York.

Van Eijs, M. (1993), A note on the joint replenishment problem under constant demand. Journal of the Operational Research Society, **44**, 185-191.

차 병 철
부산대학교 산업공학 학사
부산대학교 산업공학 석사
현재: 부산대학교 산업공학 박사과정
관심분야: Supply Chain Management, 시스템 분석 및 설계

문 일 경
서울대학교 산업공학 학사
서울대학교 산업공학 석사
미국 Columbia 대학교 산업공학 박사
현재: 부산대학교 산업공학과 교수 기술사 (공장관리)
관심분야: 생산일정계획 및 재고관리, Supply Chain Management, Reverse Logistics