The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 21, No. 2, pp.169-175, Apr. 30, 2021. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2021.21.2.169
JIIBC 2021-2-24

로트 크기 문제의 비축 효율성 알고리즘

Stock Efficiency Algorithm for Lot Sizing Problem

이상운*

Sang-Un, Lee*

요 약 로트 크기 문제(LSP)는 다항시간으로 최적 해를 찾을 수 있는 알고리즘이 알려져 있지 않은 NP-완전의 난제이다. LSP에 대해 다항시간으로 해를 구할 수 있는 W-W 알고리즘이 알려져 있지만, 이 알고리즘은 너무나 복잡하여 이해와 적용에 어려움이 있어 S-M의 휴리스틱 근사 알고리즘이 제안되었다. 본 논문에서는 LSP의 근사 해가 아닌 최적 해를 찾을 수 있는 간단한 공식을 가진 $O(n)$의 선형 복잡도 알고리즘을 제안하였다. 제안된 알고리즘은 t시점에서의 로트 크기(생산량) X_t^*은 비축 비가 절차 비를 초과하지 않는 $t+k$ 시점을 결정하여 $[t,t+k]$ 구간의 요구량 합으로 단순히 결정하였다. 제안된 알고리즘을 다양한 실험 데이터에 적용한 결과 모든 데이터에 대해 최적 해를 찾았다.

Abstract The lot sizing problem(LSP) is a hard problem that classified as non-deterministic(NP)-complete because of the polynomial-time optimal solution algorithm is unknown yet. The well-known W-W algorithm can be obtain the solution within polynomial-time, but this algorithm is a very complex, therefore the heuristic approximated S-M algorithm is suggested. This paper suggests $O(n)$ linear-time complexity algorithm that can be find not the approximated but optimal solution. This algorithm determines the lot size X_t^* in period t to the sum of the demands of interval $[t,t+k]$, the period $t+k$ is determined by the holding cost will not exceed setup cost of $t+k$ period. As a result of various experimental data, this algorithm finds the optimal solution about whole data.

Key Words : Lot size, Holding cost, Setup cost, Stock efficiency, Economic production

Ⅰ. 서 론

로트 크기 문제(lot sizing problem, LSP)는 각 기간 $t=1,2,\cdots,n$에 판매 예상 량(또는 요구량)에 맞도록 생산하지 않고, 특정 기간에 과다 생산하여 비축된 량을 이후 기간에 생산하지 않고 판매하는 방식으로 소요비용을 최소로 하여 이득을 최대화시키는 경제적 생산 수량 조절 방식을 말한다.[1]

LSP는 다항시간으로 최적 해를 찾는 알고리즘이 알려져 있지 않은 NP-완전(complete)로 분류된 난제이다.[14]

이에 따라 지금까지 수많은 연구가 진행되고 있으며[1,3-5], 다항시간으로 근사 해를 구하는 탐욕 알고리즘들이 제안되었으며, 대표적인 방법으로는 L4L(Lot-for-lot), EOQ (Economic order quantity), POQ(Period order quantity), LUC (Least unit cost), LTC(Least total

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2021년 1월 9일, 수정완료 2021년 3월 13일
게재확정일자 2021년 4월 9일

Received: 9 January, 2021 / Revised: 13 March, 2021 / Accepted: 9 April, 2021
*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

cost), LPC(Least period cost), Wagner-Whitin(W-W)[6], Silver-Meal(S-M)[7] 등이 있다.

Wagner-Whitin(W-W)[6] 알고리즘은 동적계획법(dynamic programming)을 적용하여 절차비와 비축비를 최소화시키도록 현재 기간 t의 생산량을 결정하는 방법으로 매우 복잡하여 Silver-Meal(S-M)[7]의 휴리스틱 기법이 제안되었으며, 또한, Sadjadi et al.[8]은 W-W 알고리즘을 보다 간단히 수행하는 개선된 알고리즘을 제안하였으며, Wagelmans과 Hoesel[9]는 W-W를 $O(n\log n)$수행 복잡도로 계산하는 선형시간 알고리즘을 제안하였다. 이외에도 Gaafar와 Choueiki[10]은 다항시간 알고리즘이 존재하지 않는다고 가정하고, 메타휴리스틱 기법의 일종인 신경망(neural network, NN)을 적용하여 근사 해를 구하기도 하였다.

본 논문에서는 가장 쉽게 기간 t의 생산량인 로트 크기를 경제적으로 결정하는 공식을 제안한다. 2장에서는 LSP의 정의와 예제 데이터를 통해 LSP의 로트 크기를 결정하는 방법을 이해한다. 3장에서는 비축 효율성 알고리즘을 제안하고, 4장에서는 실험 데이터를 대상으로 본 논문에서 제안된 경제적 생산 알고리즘의 적합성을 검증하여 본다.

II. 로트 크기 문제

경제적이고 효율적으로 생산 활동을 수행하기 위하여 그룹화 된 제품, 부품 또는 원재료 등이 하나로 통합된 것을 로트(lot)라고 하며, 로트 크기(lot size)는 단위 로트의 수량을 뜻한다. 로트 크기 문제(lot sizing problem, LSP)의 최대 수익을 얻기 위한 공식은 식 (1)로, 최소 소요비용 공식은 식 (2)로 정의된다.[1] 즉, 매출액-(절차비+생산비+비축비)로 총 매출액에서 생산준비와 마무리에 소요되는 간접비용, 생산에 소요되는 직접비용과 잉여물품 저장비용을 뺀 순수익으로 정의된다.[1,11,12]

- T:총 기간(period) 수
- s_t :기간 t에서 연속 생산하는 경우 준비 및 마무리에 소요되는 비용인 생산 또는 구매 절차비용(고정비)
- p_t :기간 t에서 단가(단위 판매비용)
- c_t :기간 t에서 단위 생산(구매)비용(변동비)
- h_t :기간 t에서 단위 재고비용(t 종료시점에서 부과)
- d_t :기간 t에서 요구량
- X_t :기간 t에서 생산(구매) 수량
- I_t :기간 t 종료시점의 재고수준
- L_t :기간 t에서의 부족량
- y_t :기간 t에서 생산(구매) 활동의 지시 변수

$$\max.\,\Pi = \sum_{t=1}^{T} p_t(d_t - L_t) - \sum_{t=1}^{T}(s_t y_t + c_t X_t + h_t I_t) \quad (1)$$
$$= \sum_{t=1}^{T} p_t d_t - \sum_{t=1}^{T}(s_t y_t + c_t X_t + h_t I_t + p_t L_t)$$

$$\min.\,\Pi' = \sum_{t=1}^{T}(s_t y_t + c_t X_t + h_t I_t + p_t L_t) \quad (2)$$

$$\text{s.t} \quad I_{t-1} + X_t - (d_t - L_t) = I_t : i = 1,2,\cdots,T$$
$$l_t \le d_t : i = 1,2,\cdots,T$$
$$y_t = \begin{cases} 1 \text{ if } X_t > 0 \\ 0\, otherwise \end{cases} : t = 1,2,\cdots,T$$
$$I_0 = 0$$
$$X_t \ge 0, I_t \ge 0, L_t \ge 0 : t = 1,2,\cdots,T$$
$$y_t INleft0,1\} : t = 1,2,\cdots,T$$

표 1의 예제 데이터 (a)는 특정 회사의 X년 4분기를 기준으로 제시된 데이터이다. 이에 대한 생산, 비축과 판매에 대한 최적 해는 (b)에 제시되어 있다.[1] 즉, 1/4분기에는 생산과 판매도 하지 않으며, 2/4분기에 2/4분기와 4/4분기 판매량을 생산하여, 2/4분기에 판매하고 남은 수량을 3/4분기를 거쳐 4/4분기까지 비축하였다가 4/4분기에 판매하는 경우 순익은 $112,250이 됨을 알 수 있다.

표 1. lsp-1 예제 데이터
Table 1. Example data of lsp-1

(a) Problem

Period t	p_t	c_t	h_t	s_t	Demand d_t
1(Spring)	15	10	3	25,000	3,000
2(Summer)	21	10	3	25,000	11,750
3(Fall)	12	10	3	25,000	2,000
4(Winter)	18	10	3	25,000	4,000

(b) Optimal solution

Period t	I_{t-1}^*	X_t^*	d_t	L_t^*	I_t^*
1(Spring)	0	0	3,000	3,000	0
2(Summer)	0	15,750	11,750	0	4,000
3(Fall)	4,000	0	2,000	2,000	4,000
4(Winter)	4,000	0	4,000	0	0
Net profit	$112,250				

The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 21, No. 2, pp.169-175, Apr. 30, 2021. pISSN 2289-0238, eISSN 2289-0246

LSP를 풀기 위해 다음과 같이 다양한 방법들이 제안되고 있다.

- L4L : 매 기간에서 요구하는 수량만큼 생산.
- EOQ : 고정된 생산량 법칙.
- POQ : EOQ/기간 당 평균 요구량.
- LUC : 단위당 평균비용을 최소화 시키는 이후의 'n' 기간 요구량을 충족하는 생산량 결정.
- LTC : 운송과 생산비용이 가장 가까운 이후의 'n' 기간 요구량을 충족하는 생산량 결정.
- LPC : 단위 시간 당 평균비용을 최소화 시키는 이후의 'n' 기간 요구량을 충족하는 생산량 결정.
- W-W : 동적계획법 (회귀 절차)으로 해를 구하는 법.
- S-M : 동적계획법으로 해를 구하는 법.

표 2의 lsp-2 예제 데이터에 W-W 알고리즘을 적용하여 해를 구하면 $X_1=200$, $X_3=172$, $X_5=196$을 생산하여 $\Pi'=\$344$를 얻는다.

표 2. lsp-2 예제 데이터
Table 2. Example data of lsp-2

t	M1	M2	M3	M4	M5	M6
d_t	120	80	94	78	86	110
s_t	70	70	70	70	70	70
h_t	0.5	0.5	0.5	0.5	0.5	0.5

$n=6$(M6)
$Q_6=110$
$\underline{f_6(110)=70}$
$n=5$(M5/M6)
$Q_5=86,(86+110)=196$
$f_5(86)=70+70=140$
$\underline{f_5(196)=70+0.5(110)=125}$
$n=4$(M4/M5/M6)
$Q_4=78,\ (78+86)=164,\ (78+86+110)=274$
$f_4(78)=70+125=195$
$\underline{f_4(164)=70+0.5(86)+70=183}$
$f_4(274)=70+0.5(86)+1.00(110)=223$
$n=3$(M3/M4/M5/M6)
$Q_3=94,\ (94+78)=172,\ (94+78+86)=278,$
$(94+78+86+110)=368$
$f_3(94)=70+183=253$
$\underline{f_3(172)=70+0.5(78)+125=234}$
$f_3(278)=70+0.5(78)+1.00(86)+70=265$
$f_3(368)=70+0.5(78)+1.00(86)+1.5(110)=360$
$n=2$(M2/M3/M4/M5/M6)
$Q_2=80,\ (80+94)=174,\ (80+94+78)=252,$
$(80+94+78+86)=338,$
$(80+94+78+86+110)=448$
$f_2(80)=70+234=304$
$\underline{f_2(174)=70+0.5(94)+183=300}$
$f_2(252)=70+0.5(94)+1.00(78)+125=320$
$f_2(338)=70+0.5(94)+1.00(78)+1.5(86)+70=394$
$f_2(448)=$
$70+0.5(94)+1.00(78)+1.5(86)+2.0(110)=544$
$n=1$(M1/M2/M3/M4/M5/M6)
$Q_1=120,\ (120+80)=200,\ (120+80+94)=294,$
$(120+80+94+78)=372,$
$(120+80+94+78+86)=458,$
$(120+80+94+78+86+110)=568$
$f_1(120)=70+300=370$
$\underline{f_1(200)=70+0.5(80)+234=344}$
$f_1(294)=70+0.5(80)+1.00(94)+183=387$
$f_1(372)=70+0.5(80)+1.00(94)+1.5(78)+125=446$
$f_1(458)=$
$70+0.5(80)+1.00(94)+1.5(78)+2.0(86)+70=563$
$f_1(568)=$
$70+0.5(80)+1.00(94)+1.5(78)+2.0(86)+2.5(110)=768$

$f_1(200)=\underline{70+0.5(80)}+234=344$
$f_3(172)=\underline{70+0.5(78)}+125=234$
$f_5(196)=\underline{70+0.5(110)}=125$

$$Q_1=200, Q_3=172, Q_5=196, \Pi'=344$$

Ⅲ. 비축 효율성 알고리즘

본 장에서는 s_t, p_t, c_t, h_t를 고려하는 문제와, 단지 s_t, h_t만 고려하는 문제에 대해 최소 투자-최대 수익을 얻는 비축 효율성 알고리즘(stock efficiency algorithm, SEA)을 제안한다.

(1) s_t, p_t, c_t, h_t 문제 : 다음의 3가지 판단 기준을 적용하여 생산량, 판매량, 비축량을 결정한다.

- 생산 기준 : $(p_t - c_t)d_t - s_t > h_{t-1}d_t$
- 판매 기준 : $(p_t - c_t) > |(p_{t+1} - c_{t+1} - h_t)|$
- 비축 기준 : $h_t d_{t+1} < (c_{t+1}d_{t+1}) + s_{t+1}$

(2) s_t, h_t=불변(동일) 문제 : t에서 신규 생산하여 $t+k$ 기간까지 비축비가 신규 생산에 소요되는 절차 비 s_{t+k}를 초과하지 않는 $d_{t+k}\sum_{i=1}^{k} h_{t+i} - s_{t+k} > 0$가 되는 k를 결정하여, $[t, t+k-1]$기간 동안의 요구량 총합을 t에서 생산하는 로트 크기 X_t^*로 결정한다.

$t = 1$
until $t \geq n$ do
  for $k = 1$ to $n - t$
    if $d_{t+k}\sum_{i=0}^{k-1} h_{t+i} - s_{t+k} < 0$ then $k \leftarrow k+1$
    else if $d_{t+k}\sum_{i=1}^{k} h_{t+i} - s_{t+k} > 0$ then k 결정,
      exit
    endif
  end
  $X_t^* = \sum_{i=0}^{k-1} d_{t+i}$, $t \leftarrow t+k$
end

(3) s_t, h_t=변동(상이) 문제 : $d_{t+k}\sum_{i=1}^{k} h_{t+i} - s_{t+k} > 0$가 되는 k를 결정하고, $s_t \rangle\ s_{t+i,} i < k$가 존재하면 $t \leftarrow t+i$로 증가시키고, 이 구간은 분할하여 2회 생산한다.

$t = 1$
until $t \geq n$ do
  for $k = 1$ to $n - t$
    if $d_{t+k}\sum_{i=0}^{k-1} h_{t+i} - s_{t+k} < 0$ then $k \leftarrow k+1$
    else if $d_{t+k}\sum_{i=1}^{k} h_{t+i} - s_{t+k} > 0$ then k 결정,
      exit
    endif
  end
  $t \leftarrow t+k$: $d_{t+k}\sum_{i=1}^{k} h_{t+i} - s_{t+k} > 0$인 $t+k$ 구간 결정.
  if $\exists (s_t \rangle\ s_{t+i,} i < k)$ then $l \leftarrow k, k \leftarrow k+i$
    $X_t^* = \sum_{i=0}^{l-1} d_{t+i}$, $X_l^* = \sum_{i=0}^{k-1} d_{l+i}$
  else $X_t^* = \sum_{i=0}^{k-1} d_{t+i}$
  endif
  $t \leftarrow t+k$
end

s_t, p_t, c_t, h_t 문제인 lsp-1에 L4L과 제안된 SEA를 적용한 결과는 그림 1에 제시되어 있다. L4L은 $X_t = d_t$, $t = 1,2,3,4$를 생산하여 각 기간에 모두 판매한 경우로 Π =80,250의 수익을 얻는다. 반면에, SEA는 $X_1 = 0$, $X_2 = 15,750$, $X_3 = 0$, $X_4 = 0$을 생산하여 $t = 2, t = 4$에서 d_2, d_4를 판매하여 Π =112,250인 표 1에 제시된 최적 해를 얻을 수 있었다.

Ⅳ. 알고리즘 적용 및 결과 분석

본 장에서는 s_t, h_t 문제인 표 3의 5개 데이터에 대해 본 논문에서 제안된 SEA를 적용하여 알고리즘 적합성을 검증하여 본다.

표 2와 표 3의 6개 데이터에 대해 SEA를 적용하여 얻은 X_t와 Π′은 표 4에 제시하였다.

표 3. 실험 데이터
Table 3. Experimental data

lsp-3 (a)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
d_t	42	42	32	12	26	112	45	14	76	38
s_t	132	132	132	132	132	132	132	132	132	132
h_t	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6

lsp-4 (b)	M1	M2	M3	M4	M5	M6
d_t	20	30	23	19	32	28
s_t	90	90	90	90	90	90
h_t	1.2	1.2	1.2	1.2	1.2	1.2

lsp-5 (c)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
d_t	20	50	10	50	50	10	20	40	20	30
s_t	100	100	100	100	100	100	100	100	100	100
h_t	1	1	1	1	1	1	1	1	1	1

lsp-6 (d)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
d_t	10	62	12	130	154	129	88	52	124	160	238	41
s_t	54	54	54	54	54	54	54	54	54	54	54	54
h_t	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

lsp-7 (e)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
d_t	69	29	36	61	61	26	34	67	45	67	79	56
s_t	85	102	102	101	98	114	105	86	119	110	98	114
h_t	1.1	1	1	1	1	1	1	1.1	1.2	1.2	1.2	1.2

기간 (t)	판매단가 (p_t)	생산단가 (c_t)	비축단가 (h_t)	절차 비 (s_t)	요구량 (d_t)	재고량 (I_{t-1})	생산량 (X_t)	판매량	부족량 (L_t)	비축량 (I_t)	수익 (Π)
1(Spring)	15	10	3	25,000	3,000	0	3,000	3,000	0	0	-10,000
2(Summer)	21	10	3	25,000	11,750	0	11,750	11,750	0	0	104,250
3(Fall)	12	10	3	25,000	2,000	0	2,000	2,000	0	0	-21,000
4(Winter)	18	10	3	25,000	4,000	0	4,000	4,000	0	0	7,000
					20,750		20,750	20,750			80,250

(a) 정량 생산-판매(L4L)

- 생산 판단 기준 : $(p_t - c_t)d_t - s_t > h_{t-1}d_t$
  - Spring$(t=1)$: (15-10)x 3,000-25,000=-10,000 〈 0x 3,000= 0 ⇒ 생산 안함(0)
  - Summer$(t=2)$: (21-10)x11,750-25,000=104,250 〉 3x11,750=35,250 ⇒ 생산(1)
  - Fall$(t=3)$: (12-10)x 2,000-25,000=-21,000 〈 3x 2,000= 6,000 ⇒ 생산 안함(0)
  - Winter$(t=4)$: (18-10)x 4,000-25,000= 7,000 〈 3x 4,000= 12,000 ⇒ 생산 안함(0)
- 판매 판단 기준 : $(p_t - c_t) > |(p_{t+1} - c_{t+1} - h_t)|$
  - Spring$(t=1)$: (15-10)= 5 〈 |(21-10-3)|=8 ⇒ 판매 안함(0)
  - Summer$(t=2)$: (21-10)=11 〉 |(12-10-3)|=1 ⇒ 판매(1)
  - Fall$(t=3)$: (12-10)= 2 〈 |(18-10-3)|=5 ⇒ 판매 안함(0)
  - Winter$(t=4)$: (18-10)= 8 〉 |(0- 0-3)|=3 ⇒ 판매(1)
- 비축 판단 기준 : $h_t d_{t+1} < (c_{t+1}d_{t+1}) + s_{t+1}$
  - Spring$(t=1)$: -
  - Summer$(t=2)$: 3x2,000= 6,000〈10x2,000+25,000=45,000 ⇒ 비축(1)
  - Fall$(t=3)$: 3x4,000=12,000〈10x2,000+25,000=45,000 ⇒ 비축(1)
  - Winter$(t=4)$: -

기간	생산	판매	비축
1(Spring)	0	0	0
2(Summer)	1	1	1
3(Fall)	0	0	1
4(Winter)	0	1	-

기간 (t)	판매단가 (p_t)	생산단가 (c_t)	비축단가 (h_t)	절차 비 (s_t)	요구량 (d_t)	재고량 (I_{t-1})	생산량 (X_t)	판매량	부족량 (L_t)	비축량 (I_t)	수익 (Π)
1(Spring)	15	10	3	25,000	3,000	0	0	0	3,000	0	0
2(Summer)	21	10	3	25,000	11,750	0	15,750	11,750	0	4,000	52,250
3(Fall)	12	10	3	25,000	2,000	4,000	0	0	2,000	4,000	-12,000
4(Winter)	18	10	3	25,000	4,000	4,000	0	4,000	0	0	72,000
					20,750		15,750	15,750			112,250

(b) 비축 효율성 알고리즘(SEA)

그림 1. lsp-1 데이터의 로트 크기 X_t 결정
Fig. 1. Lot size X_t decision for lsp-1

표 5에는 실험 데이터에 대해 적용된 알고리즘들의 X_t와 Π'를 비교하여 제시하였다. W-W 결과를 알 수 없는 lsp-3과 lsp-4에 대해서는 SEA가 최적 해를 구할 수 있음을 보였다. lsp-4에서 S-M은 W-W를 개선한 방식임에도 불구하고 최적 해를 구하지 못하였는데 반해, SEA는 단순한 공식으로 최적 해를 구할 수 있었다.

W-W 결과가 알려진 lsp-2, lsp-5, lsp-6과 lsp-7에 대해, lsp-2, lsp-5, lsp-7은 SEA와 W-W가 동일한 최적 해를 얻었지만, lsp-6에 대해서는 SEA가 W-W 보다 개선된 최적 해를 얻었음을 알 수 있다.

표 4. 실험 데이터에 대한 SEA의 해
Table 4. Solution of SEA for experimental data

문제	비축효율성	X_t^*	Π'
lsp-2	$t=1:t=2(-30),t=3(24)$ $t=3:t=4(-31),t=5(16)$ $t=5:t=6(-15),-$	$X_1=d_1(120)+d_2(80)=200$ $X_3=d_3(94)+d_4(78)=172$ $X_5=d_5(86)+d_6(110)=196$	344.0
lsp-3	$t=1:t=5(-69.6),t=6(204)$ $t=6:t=8(-115.2),t=9(4.8)$ $t=9:t=10(-109.2),-$	$X_1=d_1(42)+d_2(42)+d_3(32)+d_4(12)+d_5(26)=154$ $X_6=d_6(112)+d_7(45)+d_8(14)=171$ $X_9=d_9(76)+d_{10}(38)=114$	610.2
lsp-4	$t=1:t=4(-21.6),t=5(63.6)$ $t=5:t=6(-56.4),-$	$X_1=d_1(20)+d_2(30)+d_3(23)+d_4(19)=92$ $X_5=d_5(32)+d_6(28)=60$	373.2
lsp-5	$t=1:t=3(-80),t=4(50)$ $t=4:t=7(-40),t=8(60)$ $t=8:t=10(-40),-$	$X_1=d_1(20)+d_2(50)+d_3(10)=80$ $X_4=d_4(50)+d_5(50)+d_6(10)+d_7(20)=130$ $X_8=d_8(40)+d_9(20)+d_{10}(30)=90$	580.0
lsp-6	$t=1:t=3(-44.4),t=4(102)$ $t=4:-,t=5(7.6)$ $t=5:t=6(-2.4),t=7(16.4)$ $t=7:t=8(-33.2),t=9(45.2)$ $t=9:-,t=10(10)$ $t=10:-,t=11(41.2)$ $t=11:t=12(-37.6),-$	$X_1=d_1(10)+d_2(62)+d_3(12)=84$ $X_4=d_4(150)=150$ $X_5=d_5(154)+d_6(129)=283$ $X_7=d_7(88)+d_9(52)=140$ $X_9=d_9(124)=124$ $X_{10}=d_{10}(160)=160$ $X_{11}=d_{11}(238)+d_{12}(41)=279$	501.2
lsp-7	$t=1:t=3(-26.4),t=4(88.1)$ $s_4=101,s_5=98$ $t=5:t=7(-3),t=8(182)$ $t=8:t=9(-69.5),t=10(44.1)$ $s_{10}=110,s_{11}=98$ $t=11:t=12(-46.8),-$	$X_1=d_1(69)+d_2(29)=98,X_3=d_3(36)+d_4(61)=97$ $X_5=d_5(51)+d_6(26)+d_7(34)=121$ $X_8=d_8(67)+d_9(45)=112,X_{10}=d_{10}(67)=67$ $X_{11}=d-11(79)+d_{12}(56)=135$	882.6

표 5. 실험 데이터에 대한 알고리즘 성능 비교
Table 5. Compare with algorithm performance for experimental data

문제	알고리즘	X_t												Π'
		1	2	3	4	5	6	7	8	9	10	11	12	
lsp-2	SEA	200		172		196								344.0
	L4L	120	80	94	78	86	110							420.0
	W-W	200		172		196								344.0
lsp-3	SEA	154					171			114				610.2
	L4L	42	42	32	12	26	112	45	14	76	38			1320.0
	EOQ	139				139		139			139			849.6
	m=2 FPD	84		44		138		59		114				790.8
	m=5 FPD	154					285							683.8
	m=n FPD	439												1406.4
	POQ	116			150			135			38			841.2
	PPB	154					247				38			724.2
lsp-4	SEA	92				60								373.2
	L4L	20	30	23	19	32	28							540.0
	S-M	73			51		28							399.6
	PPB	73			79									376.8
lsp-5	SEA	80			130				90					580.0
	L4L	20	50	10	50	50	10	20	40	20	30			1000.0
	FOQ	100			100			100						700.0
	FOP	70		60		60		60		50				680.0
	W-W	80			130				90					580.0
lsp-6	SEA	84			130	283		140		124	160	279		501.2
	L4L	10	62	12	130	154	129	88	52	124	160	238	41	648.0
	W-W	84			130	154	129	140		124	160	279		503.6
lsp-7	SEA	98		97		121			112		67	135		882.6
	L4L	69	29	36	61	61	26	34	67	45	67	79	56	1234.0
	W-W	98		97		121			112		67	135		882.6

V. 결 론

본 논문에서는 $t=1,2,\cdots,n$기간 각각에서 판매(또는 요구)량 d_t를 충족시키기 위해 특정 시점 t에서 d_t를 초과하여 생산하고, 잉여량을 차기 기간에 판매하는 경우 소요되는 비축 비가 추가 생산 준비에 소요되는 절차 비에 비해 절약할 수 있다면 특정 시점 t의 생산량 $X_t > d_t$로 결정하는 로트 크기 문제(LSP)를 다루었다.

LSP를 푸는 다양한 방법들이 제안되었지만 그 중에서도 가장 정확한 방법이 W-W 알고리즘으로 알려져 있다. 그러나 이 방법은 계산 과정이 복잡하여 이해하기 힘들어 적용에 어려움이 있어 보다 간단한 S-M 알고리즘이 제안되기도 하였다.

본 논문에서는 특정 시점 t를 기준으로 $t+k$ 시점의 요구량 d_{t+k}를 비축하는데 소요되는 비용이 신규로 생산하는데 소요되는 절차 비 s_{t+k}를 초과하지 않는 비축 효율성을 계산하여 t 시점에서의 생산량인 로트 크기 X_t^*를 $[t,t+k]$ 구간의 요구량 합으로 단순히 결정하는 알고리즘을 제안하였다.

제안된 알고리즘은 W-W나 S-M에 비해 단순한 공식임에도 불구하고 다양한 실험 데이터에 적용한 결과 모든 데이터에 대해 최소 투자-최대 수익을 얻는 가장 경제적인 생산량(로트 크기)을 결정할 수 있음을 보였다.

References

[1] C. H. Glock, E. H. Grosse, and J. M. Ries, "The Lot Sizing Problem: A Tertiary Study", International Journal of Production Economics, Vol. 155, pp. 39-51, Sep. 2014, https://doi.org/10.1016/j.ijpe. 2013.12.009

[2] A. Akbalik and C. Rapine, "The Single Item Uncapacitated Lot-Sizing Problem with Time-Dependent Batch Sizes: NP-hard and Polynomial Cases", European Journal of Operational Research, Vol. 229, No. 2, pp. 353-363, Sep. 2013, https://doi.org/ 10.1016/ j.ejor.2013.02. 052

[3] B. Karimi, S. M. T. F. Gihomi, and J. M. Wilson, "The Capacitated Lot Sizing Problem: A Review of Models and Algorithms", Omega, Vol. 31, No. 5, pp. 365-378, Oct. 2003, https://doi.org/ 10.1016/s0305-0483(03)00059-8

[4] R. Jans and Z. Degraeve, "Modeling Industrial Lot Sizing Problems: A Review", International Journal of Production Research, Vol. 46, No. 6, pp. 1619-1643, Nov. 2007, https://doi.org/10.1080/ 002075406009 02262

[5] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli, "Single-Item Dynamic Lot-Sizing Problems: An Updated Survey", European Journal of Operational Research, Vol. 263, No. 3, pp. 838-863, Dec. 2017, https://doi.org/10.1016/j.ejor.2017.05.008

[6] H. M. Wagner and T. M. Whitin, "Dynamic Version of the Economic Lot Size Model", Management Science, Vol. 5, No. 1, pp. 89–96, Oct. 1958, https://doi.org/10.1287/ mnsc. 5.1.89

[7] E. A. Silver and H. C. Meal, "A Heuristic for Selecting Lot Size Quantities for the Case of a Deterministic Time-Varying Demand Rate and Discrete Opportunities for Replenishment", Production and Inventory Management: Journal of the American Production and Inventory Control Society, Vol. 14, No. 1, pp. 64-74, Feb. 1973.

[8] S. J. Sadjadi, M. B. G. Aryanczhad, and H. A. Sadeghi, "An Improved Wagner-Whitin Algorithm", International Journal of Industrial Engineering & Production Research, Vol. 20, No. 3, pp. 117-123, Dec. 2009.

[9] A. Wagelmans and S. V. Hoesel, "Economic Lot Sizing: An O(n log n) Algorithm that Runs in Linear Time in the Wagner-Whitin Case", Oprtations Research, Vol. 40, No. 1, pp. S145-S156, Feb. 1992, https://doi.org/10.1287/ opre.40.1.S145

[10] L. K. Gaafar and M. H. Choueiki, "A Neural Network Model for Solving the Lot-Sizing Problem", Omega, Vol. 28, No. 2, pp. 175-184, Apr. 2000, https://doi.org/10.1016/S0305-0483(99) 00035-3

[11] J. C. Lang and W. Domschke, "Efficient Reformulations for Dynamic Lot-Sizing Problems with Product Substitution", OR Spectrum, Vol. 32, No. 2, pp. 263-291, Apr. 2010, https://doi.org/ 10.1007/ s00291-008-0148-1

[12] N. Brahimi, N. Absi, S. Dauzère-Pérès, N. M. Najid, and A. Nordli, "Single-Item Lot-Sizing Problems", European Journal of Operational Research, Vol. 168, No. 1, pp. 1-16, Jan. 2006, https:// doi.org/10.1016/ j.ejor.2004.01.054

저 자 소 개

이 상 운(정회원)

- 1987년: 한국항공대학교 항공전자공학과 (학사)
- 1997년: 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr