

로트 크기 결정 문제의 새로운 혼합정수계획법 모형 및 휴리스틱 알고리즘 개발

한정희¹ · 이영호^{2*} · 김성인² · 박은경²

¹강원대학교 경영학과 / ²고려대학교 정보경영공학전문대학원

An Alternative Modeling for Lot-sizing and Scheduling Problem with a Decomposition Based Heuristic Algorithm

Junghee Han¹ · Youngho Lee² · Seong-in Kim² · Eunkyung Park²

¹College of Business Administration, Kangwon National University

²Division of Information Management Engineering, College of Engineering, Korea University

In this paper, we consider a new lot-sizing and scheduling problem (LSSP) that minimizes the sum of production cost, setup cost and inventory cost. Setup carry-over and overlapping as well as demand splitting are considered. Also, maximum number of setups for each time period is not limited. For this LSSP, we have formulated a mixed integer programming (MIP) model, of which the size does not increase even if we divide a time period into a number of micro time periods. Also, we have developed an efficient heuristic algorithm by combining decomposition scheme with local search procedure. Test results show that the developed heuristic algorithm finds good quality (in practice, even better) feasible solutions using far less computation time compared with the CPLEX, a competitive MIP solver.

Keywords: Lot-sizing and Scheduling, Mixed Integer Programming, Heuristic

1. 서론

생산 비용과 재고 비용을 최소화하는 최적의 로트 크기와 생산 스케줄을 동시에 결정하는 문제를 로트 크기 및 스케줄 결정 문제(LSSP: Lot-Sizing and Scheduling Problem)라고 한다(Drexl and Hasse, 1995). LSSP 문제는 철강 생산 시스템(Zanoni and Zavanella, 2005), 석유 화학 제품 생산 시스템(Boctor and Poulin, 2005), 섬유 생산 및 가공 시스템(Silva and Magalhaes, 2006), 종이 생산 시스템(Gopalakrishnan, 2000) 등 다양한 형태의 생산 현장에서 생산 계획 수립을 위해 활용되고 있다(Gupta and Magnusson, 2005; Marinelli *et al.*, 2007; Pattloch *et al.*, 2001).

LSSP는 일반적으로 주어진 계획기간(planning horizon)내에

포함된 하나의 구간(time period)에서 생산할 수 있는 제품의 종류(Sox and Gao, 1999) 또는 셋업(setup)의 횟수(Suerie and Stadler, 2003)를 기준으로 작은 버킷(small bucket) 모델과 큰 버킷(big bucket) 모델로 구분한다. 작은 버킷 모델에서는 각 구간마다 한 종류의 제품만 생산하거나 한 번의 셋업을 실시할 수 있고, 큰 버킷 모델에서는 두 종류 이상의 제품을 생산하거나 두 번 이상의 셋업을 실시할 수 있다. 작은 버킷 문제에는 일단 생산을 시작하면 해당 구간의 최대 생산량까지 하나의 제품만을 생산하는 DLSP(Discrete Lot-sizing and Scheduling Problem), 각 구간마다 선택된 제품의 생산량을 임의로 결정할 수 있는 CSLP(Continuous Setup Lot-sizing Problem), 각 구간의 마지막 셋업 상태가 다음 구간까지 유지되는 PLSP(Proportional Lot-sizing and Scheduling Problem)가 있다(Drexl and Hasse,

이 연구에 참여한 연구자의 일부는 '2단계 BK21사업'의 지원비를 받았다.

* 연락저자 : 이영호, 136-701 서울시 성북구 안암동 5가 1 고려대학교 정보경영공학전문대학원, Fax : 02-929-5888, E-mail : yhle@korea.ac.kr
2007년 03월 접수; 2007년 05월 수정본 접수; 2007년 05월 게재 확정.

1995, 1996). 한편 한 구간에서 두 번 이상의 셋업을 실시할 수 있는 큰 버킷 문제로는 CLSP(Capacitated Lot-Sizing Problem)가 있다(Clark and Clark, 2000; Gopalakrishnan, 2000; Gupta and Magnusson, 2005; Fleischmann and Meyr, 1997; Meyr, 2000, 2002; Porkka *et al.*, 2003; Sox and Gao, 1999; Staggemeier and Clark (2001); Suerie and Stadtler, 2003).

CLSP 모형을 다룬 대부분의 연구에서 셋업 이월(setup carry-over)은 고려하였으나, 하나의 주문을 여러 기계에서 나누어 처리하는 분할 생산이나 연속된 두 구간에 걸쳐서 셋업을 실시하는 셋업 오버래핑(setup overlapping)은 고려하지 않는다(<Table 1> 참조). 셋업 이월을 허용하면 다음 구간에서 생산하는 첫번째 제품이 현재 구간에서 마지막으로 생산된 제품과 같을 때 추가적인 셋업을 생략할 수 있다. 다만 Clark and Clark (2000)가 분할 생산이 가능한 CLSP 모형을 제시하였고 Suerie (2006)가 셋업 오버래핑을 표현한 PLSP 모형을 개발하였다. 이 밖에도 Fleischmann and Meyr(1997)는 최소 생산량을 표현한 GLSP(General Lot-sizing and Scheduling Problem) 모형을 개발하였으나, 이 모형에서는 셋업 오버래핑을 고려하지 않았으며 기계가 한대일 경우에만 적용 가능하다. 이후 Meyr(2002)는 여러 대의 기계를 고려한 GLSP 모형을 개발하였으나, 여전히 셋업 오버래핑은 고려하지 않았으며, 셋업당 최소, 최대 생산량도 제한할 수 없다. <Table 1>에는 작은 버킷 문제에 속하는 PLSP와 큰 버킷 문제에 속하는 CLSP 및 GLSP 모형을 대상으로 8편의 대표적인 기존 연구의 특징이 요약되어 있다.

<Table 1>에서도 확인할 수 있듯이 분할 생산, 셋업의 이월 및 오버래핑이 가능하고 셋업당 최소 및 최대 생산량을 선택적으로 제한할 수 있는 모형을 기존 연구에서는 찾아볼 수 없다. 그러나 이와 같은 특징을 모두 관찰할 수 있는 생산 현장은 쉽게 찾아볼 수 있다. 가령 합금생산 공정에서는 여러 대의 주조로가 동시에 한 종류의 합금을 생산할 수 있다. 또한 합금생산 공정에서는 주조로의 용량에 따라 셋업당 최대 생산량이 결정되며, 배치(batch) 단위에 의해 셋업당 최소 생산량이 결정된다. 뿐만 아니라 동일한 합금을 장기간 생산할 경우에는 셋업이 이월되며, 셋업의 오버래핑은 위와 같은 연속생산 공정에서는 반드시 고려해야 할 사항이다. 왜냐하면 특별한 경우를 제외하면 주조로는 24시간 가동되므로 (주조로의 가동을 정지시키면 재가동에 매우 오랜 시간이 소요됨) 주조로의 셋업 변경은 날짜 변경 시점과 무관하기 때문이다.

따라서 이 논문에서는 다음과 같은 특징을 갖는 새로운 LSSP 모형을 개발한다.

- 계획기간 동안 최대 셋업 횟수를 임의의 값으로 설정 가능,
- 분할 생산(하나의 주문을 서로 다른 생산 능력을 갖는 다수의 기계에서 생산) 가능,
- 셋업의 이월 및 오버래핑 가능,
- 셋업당 최소 및 최대 생산량 표현 가능.

Table 1. Summary of recent research papers on LSSP

저자(년도)	기본 모형	기계 수/분할 생산	구간당 셋업 횟수/생산 순서 표현	셋업 이월	셋업 오버래핑
Sox and Gao(1999)	CLSP	1 /불가능	>2 /불가능	가능	불가능
Gopalakrishana (2000)	CLSP	1 /불가능	>2 /불가능	가능	불가능
Fleischmann and Meyr (2000)	GLSP	1 /불가능	>2 /가능	가능	불가능
Clark and Clark(2000)	CLSP	>1 /가능	>2 /가능	불가능	불가능
Meyr(2002)	GLSP	>1 /가능	>2 /가능	가능	불가능
Porkka <i>et al.</i> (2003)	CLSP	1 /불가능	>2 /불가능	가능	불가능
Suerie and Stadtler(2003)	CLSP	>1 /불가능	>2 /불가능	가능	불가능
Suerie(2006)	PSLP	1 /불가능	2 /가능	가능	가능

이 논문의 구성은 다음과 같다. 제 2장에서는 이 논문에서 제시한 LSSP 문제의 혼합정수계획법 모형을 제시하고, 제 3장에서 휴리스틱 알고리즘을 소개한다. 제 4장에서는 모의실험을 통해 수학 모형을 검증하고 휴리스틱 알고리즘의 수행도를 평가한다. 제 5장에서는 이 연구의 결론과 추후 연구 방향을 제시한다.

2. 수학 모형

이 논문에서 다루는 LSSP 문제를 수학 모형으로 표현하기 위해 다음과 같은 기호를 정의한다.

[집합 정의]

- N : 제품의 집합,
- K : 기계의 집합.

[파라미터 정의]

- T : 계획기간의 길이,
- L : 계획기간 동안 하나의 기계에서 허용되는 셋업 횟수,
- $M(i)$: 계획기간 동안 제품 $i \in N$ 의 주문(또는 납기)이 발생한 횟수,
- T_{im} : 제품 $i \in N$ 의 m 번째 납기 시점,
- D_{im} : 제품 $i \in N$ 의 m 번째 납기 시점까지 제품 $i \in N$ 의 누적 수요량,
- b_k^L : 기계 $k \in K$ 의 셋업당 최소 생산 시간,
- b_k^U : 기계 $k \in K$ 의 셋업당 최대 생산 시간,
- s_{ki} : 기계 $k \in K$ 에서 제품 $i \in N$ 의 생산 준비에 소요되

- p_{ki} : 기계 $k \in K$ 에서 제품 $i \in N$ 를 생산할 때의 단위시간당 생산량,
- α_{ki} : 기계 $k \in K$ 에서 제품 $i \in N$ 를 생산할 때의 단위시간당 생산 비용,
- β_{ki} : 기계 $k \in K$ 에서 제품 $i \in N$ 의 생산 준비에 소요되는 셋업 비용,
- π_{im} : 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 에 남은 재고의 단위 재고당 관리 비용.

[변수 정의]

- x_{kli} : 기계 $k \in K$ 에서 $l (= 1, \dots, L)$ 번째 셋업 후 제품 $i \in N$ 를 생산하면 1 (아니면 0),
- y_{kl}^S : 기계 $k \in K$ 에서 l 번째 셋업을 시작하는 시점,
- y_{kl}^E : 기계 $k \in K$ 에서 l 번째 셋업 후 생산을 종료하는 시점,
- f_{kli} : 기계 $k \in K$ 에서 l 번째 셋업 후 제품 $i \in N$ 를 생산하는 시간,
- u_{klim}^0 : 기계 $k \in K$ 에서 l 번째 셋업 후 생산하는 제품 $i \in N$ 의 생산 시작 시점($y_{kl}^S + s_{ki}$)이 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 보다 앞서면 1 (아니면 0),
- w_{klim}^1 : 기계 $k \in K$ 에서 l 번째 셋업 후 생산하는 제품 $i \in N$ 의 생산 종료 시점(y_{kl}^E)이 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 보다 앞서면 1 (아니면 0),
- w_{klim}^0 : 기계 $k \in K$ 에서 l 번째 셋업 후 생산하는 제품 $i \in N$ 의 생산 시작 시점($y_{kl}^S + s_{ki}$)부터 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 까지의 생산 시간 ($\leq f_{kli}$),
- w_{klim}^1 : 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 부터 기계 $k \in K$ 에서 l 번째 셋업 후 생산하는 제품 $i \in N$ 의 생산 시간($w_{klim}^0 + w_{klim}^1 = f_{kli}, \forall k, l, i, m$),
- v_{im} : 제품 $i \in N$ 의 m 번째 납기 시점 T_{im} 에 납기 후 남은 제품 $i \in N$ 의 재고량.

위에 정의한 기호를 이용하여 이 논문에서 다루는 LSSP 문제를 다음과 같은 혼합정수계획법(MIP: Mixed Integer Programming) 모형으로 표현할 수 있다.

$$\text{LSSP : Minimize } \sum_{k \in K} \sum_{l \leq L} \sum_{i \in N} (\alpha_{ki} f_{kli} + \beta_{ki} x_{kli}) + \sum_{i \in N} \sum_{m \leq M(i)} \pi_{im} v_{im}$$

Subject to

$$\sum_{i \in N} x_{kli} \leq 1 \quad k \in K, l \leq L, \quad (1)$$

$$x_{kli} \leq \sum_{i \in N} x_{k(l-1)j} \quad k \in K, l \leq L, i \in N, \quad (2)$$

$$y_{kl}^S \geq y_{k(l-1)}^E \quad k \in K, l \leq L, \quad (3)$$

$$y_{kl}^E = y_{kl}^S + \sum_{i \in N} (s_{ki} x_{kli} + f_{kli}) \quad k \in K, l \leq L, \quad (4)$$

$$f_{kli} \leq b_k^U x_{kli} \quad k \in K, l \leq L, i \in N, \quad (5)$$

$$f_{kli} \geq b_k^L x_{kli} \quad k \in K, l \leq L, i \in N, \quad (6)$$

$$T u_{klim}^0 - (T_{im} - s_{ki}) x_{kli} + y_{kl}^S \geq 0 \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (7)$$

$$T u_{klim}^0 - (T_{im} - s_{ki}) x_{kli} + y_{kl}^S \leq T \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (8)$$

$$T u_{klim}^1 - T_{im} x_{kli} + y_{kl}^E \geq 0 \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (9)$$

$$T u_{klim}^1 - T_{im} x_{kli} + y_{kl}^E \leq T \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (10)$$

$$u_{klim}^0, u_{klim}^1 \leq x_{kli} \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (11)$$

$$u_{klim}^1 \leq u_{klim}^0 \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (12)$$

$$w_{klim}^0 + u_{klim}^1 = f_{kli} \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (13)$$

$$w_{klim}^0 \leq b_k^U u_{klim}^0 \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (14)$$

$$w_{klim}^1 \leq b_k^L (1 - u_{klim}^1) \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (15)$$

$$y_{kl}^S + s_{ki} x_{kli} + w_{klim}^0 - T_{im} \leq T(1 - u_{klim}^0 + u_{klim}^1) \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (16)$$

$$y_{kl}^S + s_{ki} x_{kli} + w_{klim}^0 - T_{im} \geq -T(1 - u_{klim}^0 + u_{klim}^1) \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (17)$$

$$\sum_{k \in K} \sum_{l \leq L} p_{ki} w_{klim}^0 = D_{im} + v_{im} \quad i \in N, m \leq M(i) \quad (18)$$

$$x_{kli} \in \{0, 1\} \quad k \in K, l \leq L, i \in N, \quad (19)$$

$$y_{kl}^S, y_{kl}^E \geq 0 \quad k \in K, l \leq L, \quad (20)$$

$$f_{kli} \geq 0 \quad k \in K, l \leq L, i \in N, \quad (21)$$

$$u_{klim}^0, u_{klim}^1 \in \{0, 1\} \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (22)$$

$$w_{klim}^0, w_{klim}^1 \geq 0 \quad k \in K, l \leq L, i \in N, m \leq M(i), \quad (23)$$

$$v_{im} \geq 0 \quad i \in N, m \leq M(i). \quad (24)$$

목적함수는 생산 비용, 셋업 비용 및 재고 비용의 합을 최소화한다. 제약식 (1)은 제약식 (3), (4)와 함께 각 기계마다 임의의 시점에 서로 다른 제품을 동시에 생산할 수 없음을 나타낸다. 제약식 (2)는 각 기계마다 $(l-1)$ 번째 셋업이 이루어졌을 때 l 번째 셋업이 가능함을 의미한다. 이 식은 가능해(feasible solution)의 불필요한 중복을 제거하기 위한 일종의 대칭 현상 방지(symmetry breaking) 제약식이다. 제약식 (5), (6)은 셋업당 최소 및 최대 생산 시간을 표현한다. 제약식 (7), (8)은 기계 k 에서 l 번째로 생산하는 제품 i 의 생산 시작 시점($y_{kl}^S + s_{ki}$)과 제

품 i 의 m 번째 납기 시점 T_{im} 의 선후 관계(u_{klim}^0)를 나타내며, 제약식 (9), (10)은 생산 종료 시점(y_{klm}^E)과 납기 시점 T_{im} 의 선후 관계(u_{klim}^1)를 나타낸다. 제약식 (11)은 기계 k 에서 l 번째로 제품 i 를 생산할 경우에만 두 변수 u_{klim}^0 과 u_{klim}^1 이 값을 가질 수 있도록 하며, 제약식 (12)는 납기 시점과의 선후 관계를 나타내는 두 변수 u_{klim}^0 과 u_{klim}^1 사이의 관계를 나타낸다. 제약식 (13)-(15)는 기계 k 의 l 번째 셋업 후 생산하는 제품 i 의 생산 시간(f_{kli})은 제품 i 의 m 번째 납기 시점 이전의 생산 시간(w_{klim}^0)과 이후의 생산 시간(w_{klim}^1)의 합으로 표현할 수 있고, 이 값이 u_{klim}^0 과 u_{klim}^1 에 의해 제한됨을 의미한다. 제약식 (16), (17)은 $u_{klim}^0 = 1$ 이고 $u_{klim}^1 = 0$ 인 경우, 납기 시점 이전의 생산 시간(w_{klim}^0)을 계산할 수 있도록 한다. 제약식 (18)은 제품 i 의 m 번째 납기 시점까지의 누적 생산량은 납기 시점까지의 누적 수요량(D_{im})보다 크거나 같아야 하며, 클 경우에는 납기 후 재고(v_{im})가 발생함을 나타낸다.

서론에서도 언급하였듯이 이 논문에서 개발한 LSSP 모형은 분할 생산, 셋업의 이월 및 오버래핑, 셋업당 최소 및 최대 생산량을 모두 표현할 수 있다. 이 밖에도 이 논문에서 개발한 LSSP 모형의 주요한 특징은 입력 파라메타 값의 패턴이나 각 구간의 길이를 표현하는 시간 단위가 달라져도 모형의 크기가 변하지 않는다는 것이다. 즉, 시간을 표현하는 인덱스(t)가 사용되는 기존 모형(DLSP, CSLP, PLSP, CLSP 및 GLSP)에서는 각 제품의 납기 시점간 간격이 불규칙할 때 각 구간의 최소 시간단위는 전체 납기 시점간 간격의 최대 공약수로 표현된다. 가령 제품 A의 첫 번째 및 두 번째 납기 시점이 각각 3일째와 10일째이고 제품 B의 첫 번째 및 두 번째 납기 시점이 각각 5.5일째(즉, 5일 정오)와 7일째라고 하면, 계획기간 10일 동안 발생하는 네 번의 납기 시점간 간격의 최대 공약수는 0.5일이 된다. 따라서 하루를 기준으로 계획기간에 포함되는 구간수가 10에서 반일 기준으로 20으로 증가하게 된다. 즉, 시간을 나타내는 인덱스(t)를 사용하게 되면 제품의 종류($|N|$), 기계의 수($|K|$) 및 계획기간의 크기(T)가 고정되어 있어도 각 제품의 납기 시점 패턴에 따라 수학적 모형의 크기가 변하게 된다.

또한 Gopalakrishnan (2000)의 연구와 같이 생산 시작 시점이나 생산 시간을 정수값으로 표현하면 셋업 시간이나 셋업당 최소(또는 최대) 생산 시간과 같은 입력 값의 시간단위에 의해서도 수학적 모형의 크기가 달라질 수 있다. 가령 어떤 제품의 셋업 시간을 30분 단위로 표현하는 경우(예를 들면 2시간 및 2시간 30분), 생산 시작 시점과 생산 시간을 정수 값으로 표현하기 위해서는 1일 단위의 구간을 30분 단위로 48개의 세부 구간으로 다시 분할해야 한다. 따라서 매우 작은 크기의 문제도 납기 시점간 간격이 불규칙하거나 생산 시작 시점과 생산 시간을 일반 정수(general integer) 변수로 표현하면 수학적 모형의 크기가 지나치게 커지므로(최적)해를 구하는 것은 현실적으로 불가능하다. 이 같은 시간 분할의 개념을 기존 연구에서는 거시 시간 구간(macro time period)과 미시 시간 구간(micro time

period)이라는 용어로 표현한다(Fleischmann and Meyr, 1997; Meyr, 2000, 2002). 생산 시작 시점과 생산 시간을 실수(real) 변수로 표현하여도 납기 시점간 간격이 불규칙할 경우에는 수학적 모형의 크기가 매우 커지므로 기존 연구에서는 비교적 작은 크기의 문제만을 다루고 있다(가장 큰 문제 크기를 기준으로 Gupta and Magnusson (2005)의 연구에서는 $|N|=4$, $|K|=1$, $T=15$, Meyr (2002)의 연구에서는 $|N|=19$, $|K|=2$, $T=8$, Sox and Gao (1999)의 연구에서는 $|N|=8$, $|K|=1$, $T=8$). 만일 생산 시작 시점과 생산 시간은 정수 변수로 표현하면서 셋업 시간이나 셋업당 최소 및 최대 생산량과 같은 입력 값을 실수 값으로 표현하면, 기계마다 셋업 완료와 생산 시작 또는 생산 완료와 다음번 셋업 시작 시점 사이에 불필요한 유휴시간이 강제적으로 추가됨으로써 실제로 존재하는 가능해를 찾지 못하는 상황이 발생할 수 있다.

그러나 이 논문에서 제시한 모형 LSSP에서는 시간을 표현하는 인덱스(t) 대신 각 기계마다 셋업의 차수를 나타내는 인덱스(l)를 사용하므로 납기 시점의 패턴이나 기타 다른 입력 값의 시간 단위에 의해 모형의 크기가 변하지 않는다. 즉, LSSP 모형의 크기는 전적으로 제품의 종류($|N|$), 기계의 수($|K|$), 제품별 주문 횟수($M(i)$) 및 계획기간 동안 각 기계마다 허용되는 최대 셋업 횟수(L)에 의해서 결정된다. 앞서 다른 예제를 동일한 방법으로 LSSP 모형의 크기 변화를 살펴보자. 각 제품의 납기 시점을 다음과 같이 지정할 수 있다. $T_{A1}=3$ 일째, $T_{A2}=10$ 일째, $T_{B1}=5.5$ 일째, $T_{B2}=7$ 일째. 이 경우 LSSP 모형의 크기는 증가하지 않는다. 또한 셋업 시간이 2시간인 경우와 2시간 30분인 경우는 각각 셋업 시간을 30분 단위로 표현할 때 4와 5가 된다(단위 생략). 마찬가지로 납기 시점도 다음과 같이 30분 단위로 표현할 수 있다(단위 생략): $T_{A1}=3 \times 48$, $T_{A2}=10 \times 48$, $T_{B1}=5.5 \times 48$, $T_{B2}=7 \times 48$. 셋업당 최소 및 최대 생산 시간도 마찬가지로 30분 단위로 표현할 수 있으며, 만일 모든 입력 값을 표현하는 최소 시간 단위가 30분 이하라면 그 값을 이용하여 입력 값을 환산하여 모형에 표현하면 된다.

다만 주목할 점은 이 논문에서 다른 LSSP 문제가 NP-hard이며, 앞서 제시한 LSSP 모형의 복잡도(complexity)는 기계 대수, 제품의 종류 및 셋업 허용 회수가 증가함에 따라 지수적으로(exponentially) 증가한다는 것이다. 이 논문에서 제시하는 LSSP 문제가 NP-hard임은 다음과 같이 확인할 수 있다. Gupta and Magnusson (2005)에 따르면 1대의 기계에서 셋업의 이월을 고려한 CLSP 문제는 NP-hard이다. 따라서 이 논문에서 다루는 LSSP 문제는 CLSP와 동일한 목적함수를 표현하면서 다수의 기계에서 셋업의 이월을 고려하므로 NP-hard이다.

3. 휴리스틱 알고리즘

Gupta and Magnusson(2005) 및 Porkka *et al.*(2003)은 현대의 기계에서 셋업의 이월을 고려한 CLSP 모형을 대상으로 단순

한 형태의 지역 탐색(local search) 알고리즘을 개발하였으며, Suerie and Stadtler(2003)는 여러 대의 기계가 존재할 때 셋업 이월을 고려한 CLSP 문제를 해결하기 위해 내부 회전 스케줄링(internally rolling schedules with lot-sizing window) 알고리즘을 개발하였다. 또한 Fleischmann and Meyr(1997)는 현대의 기계에서 셋업 이월만을 고려한 GLSP 문제를 대상으로 시간의 역순으로(backward) 진행하면서 해를 구축하는 알고리즘을 개발하였으며, Drex1 and Haase(1995) 역시 PSLP 모형을 대상으로 시간의 역순으로 해를 구축하는 지역 탐색 알고리즘을 개발하였다. 시간의 역순으로 해를 구축하는 방법은 일반적으로 계획기간 내에 포함된 두 구간의 연결이 늦은 지점부터 셋업을 결정하므로 중첩되는 셋업을 효과적으로 줄일 수 있고 재고 비용을 최소화하는 생산 시간 결정에 유리하다. 그러나 이 방법은 존재하는 가능해를 찾지 못하는 상황이 자주 발생할 수 있다. 한편 Sox and Gao(1999)는 1대의 기계에서 셋업 이월만을 고려한 CLSP 모형을 대상으로 라그랑지안 듀얼(lagrangian dual) 알고리즘을 개발하였다. 이 밖에도 수 많은 휴리스틱 알

고리즘이 존재하지만, 셋업 오버래핑을 고려한 알고리즘은 찾아보기 어렵다.

이 논문에서 개발한 휴리스틱 알고리즘은 초기해 발견 알고리즘과 초기해 개선 알고리즘으로 구분할 수 있으며, LSSP 모형 기반의 분해(decomposition) 기법과 지역 탐색 기법을 결합하여 초기해 개선 알고리즘을 개발한다. 휴리스틱 알고리즘의 전반적인 동작 개요는 <Figure 1>에 나타낸다.

3.1 초기해 발견 알고리즘

단계 1 (기계 배정 순서 결정) : 납기 시점(T_{im})이 빠른 제품의 주문부터 기계를 배정하는 순서를 결정하여, 기계 배정 순서 목록을 생성한다. 즉, 제품 A의 납기 시점이 7, 10, 25이고 제품 B의 납기 시점이 5, 12, 18이면 기계 배정 순서는 제품 B/1→제품 A/1→제품 A/2→제품 B/2→제품 B/3→제품 A/3가 된다 (“/” 뒤의 숫자는 해당 제품의 주문 차수 m 을 표시한다).

단계 2 (생산 기계와 생산 시간 결정) : 현재까지 생산된 누적 생산량이 납기 시점까지 필요한 누적 수요량 D_{im} 보다 적고 기계 배정 순서가 가장 빠른 주문(im)부터 차례대로 누적 수요량에서 누적 생산량을 뺀 순 수요량을 생산하는데 드는 비용 (셋업 비용 + 생산 비용)을 최소화하는 기계를 선택하고 생산 시간을 결정한다. 모든 주문 im 에 대해서 해당 납기시점까지의 누적 생산량이 누적 수요량보가 크거나 같으면 초기해 발견 절차를 종료하고, 그렇지 않으면 이 과정을 반복한다. 즉, 모든 주문 im 에 대해 $\sum_{k \in K} \sum_{l \leq L} p_{ki} f_{kil} \geq D_{im}$ 을 만족하면 종료한다.

가령 im 의 배정 순서가 가장 빠르다고 하자. 납기 시점 (T_{im})까지 필요한 생산량을 d_{im} 으로 표시하면, 각 기계의 셋업당 최소 및 최대 생산 시간을 고려하여 현재 시점부터 T_{im} 까지 최소한 d_{im} 을 생산할 수 있는 기계 중에서 셋업 비용과 생산 비용의 합이 최소가 되는 기계를 선택하고 필요한 최소의 생산 시간을 할당한다. 만일 T_{im} 까지 d_{im} 을 하나의 기계에서 생산할 수 없는 경우에는 각 기계마다 T_{im} 까지의 여유 시간 동안만 생산하는 경우를 가정하고 가장 적은 비용이 소요되는 기계를 선택한다. 그리고 만족되지 못한 부족한 생산량을 다시 d_{im} 으로 설정한 다음 T_{im} 까지의 제품 i 의 누적 수요를 만족시킬 때까지 이 과정을 반복한다.

위에 기술한 초기해 발견 알고리즘은 납기 일정을 만족하는 가능해 발견에 우선 순위를 두고 있다. 즉, 납기 시점이 빠른 주문부터 순차적으로 생산 비용과 셋업 비용을 줄일 수 있는 기계와 생산 시간을 결정한다. 그러나 이와 같은 방법으로는 하나의 기계에서 연속적으로 동일한 제품을 생산할 때 중복되는 셋업을 효과적으로 줄이기 어렵다. 가령 어떤 기계의 셋업당 최대 생산 시간이 10시간, 첫 번째와 두 번째 셋업 후 생산하는 제품이 동일하며 생산 시간이 각각 3시간과 7시간이면

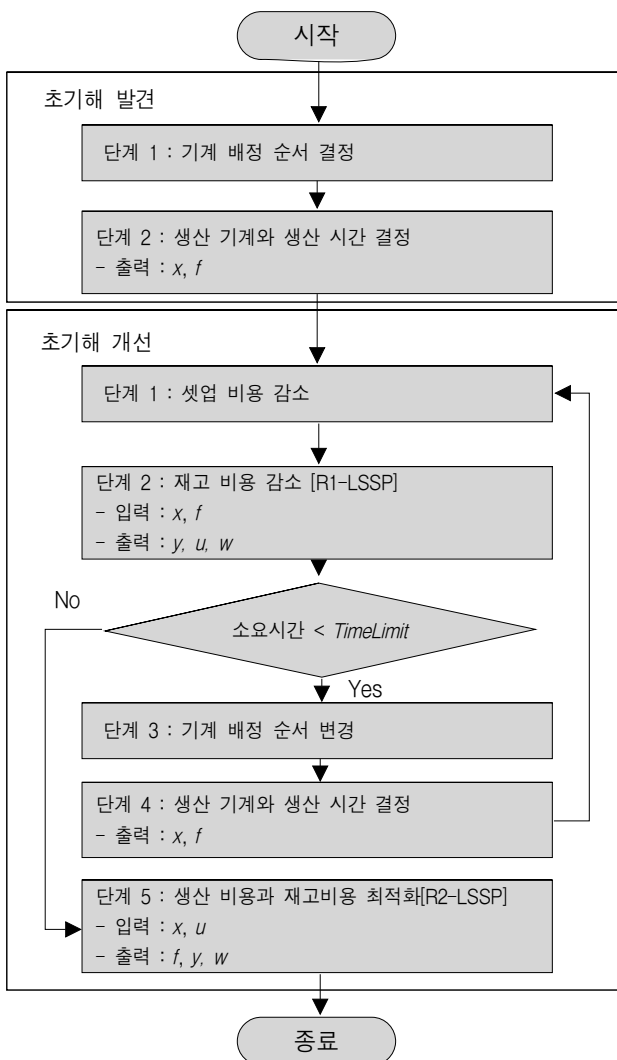


Figure 1. Flow chart of the proposed heuristic procedure

두 번째 셋업은 생략할 수 있으나, 초기해 발견 알고리즘에는 셋업 비용을 줄이기 위한 방법이 미흡하다고 여겨진다. 한편 단계 2에서 생산 시간을 결정할 때 수요를 만족하는 최소의 생산 시간만을 할당하도록 하고 있지만, 계획기간의 마지막 시점으로 갈수록 재고 비용이 증가할 수 있다. 즉, 초기해 발견 알고리즘에서는 어떤 제품을 생산할 기계가 선택되면 유휴시간(idle time) 없이 곧바로 생산을 시작하기 때문에 납기를 맞추지 못하는 경우는 줄일 수 있지만, 지금과 같이 납기 시점에 요구되는 수요를 가능한 미리 만족하도록 생산하면 재고 비용은 증가한다. 따라서 아래에는 앞서 설명한 문제점들을 고려하여 최소 비용의 가능해를 찾는 초기해 개선 알고리즘을 개발한다.

3.2 초기해 개선 알고리즘

단계 1 (셋업 비용 감소) : 하나의 기계에서 동일한 제품을 생산하기 위한 셋업이 2회 연속 발생할 때, 각각의 셋업과 연결된 생산 시간의 합이 해당 기계의 셋업당 최대 생산 시간보다 작으면 두 번째 셋업을 생략함으로써 셋업 비용을 줄인다. 더 이상 셋업 비용을 줄일 수 없을 때까지 이 과정을 반복한다. 즉, 다음 조건을 만족하는 기계 $k \in K$ 와 $l = 2, \dots, L$ 의 조합을 발견할 수 없으면 단계 2로 이동한다: $f_{k(l-1)i} + f_{kli} \leq b_k^L$.

단계 2 (재고 비용 감소) : 각 기계마다 결정된 제품의 생산 순서(x)와 생산 시간(f)을 유지하면서, 각 기계의 생산 순서마다 셋업 시작 시점(y^s)을 조정함으로써 재고 비용을 줄인다. 즉, 제2절에서 개발한 수학적 모형 LSSP에 현재 결정된 생산 순서(x)와 생산 시간(f)을 입력하면 축소된(restricted) 수학적 모형 R1-LSSP를 얻게 되며, R1-LSSP의 (최적)해를 구함으로써 재고 비용을 줄이는 셋업 시작 시점(y^s)을 결정할 수 있다 (이 논문에서는 셋업이 완료되면 바로 생산을 시작한다고 가정하므로, 생산 시작 시점은 셋업 시작 시점에 의해 결정된다).

여기서 주목할 점은 R1-LSSP에는 이진(binary) 변수 x 의 값이 결정되어 있으므로 u 변수만이 이진 변수로 남게 되며, 또한 각 기계마다 셋업당 생산 종료시점을 나타내는 y^E 값은 x 와 f 의 값이 이미 정해졌으므로 y^s 값에 의해 자동적으로 결정된다는 것이다. 즉, 로트 크기 결정문제와 스케줄링 문제의 복잡도가 모두 제거된 R1-LSSP는 비교적 최적해를 구하기 쉬운 형태의 혼합정수계획법 문제로 표현된다.

단계 3 (기계 배정 순서 변경) : 단계 2에서 구한 해에서 재고비용($\pi \times v$)이 가장 많이 발생하는 i/m 에 해당하는 기계 배정 순서 r 을 찾아 $(r+1)$ 번째 순서와 위치를 바꾸고 단계 4로 이동한다.

선택된 i/m 의 기계 배정 순서를 늦추는 이유는 해당 i/m 의 주문을 맞추기 위한 생산 시작 시점이 늦어질 가능성이 있고, 따라서 납기를 맞출 수 있다면 재고 비용 감소를 기대할 수 있기

때문이다. 또한 각 기계마다 제품의 생산 순서를 결정하고(단계 1) 셋업 상태의 이월을 시도할 때(단계 2) 셋업 비용 감소도 기대할 수 있다. 이 단계에서는 가장 최근에 생성된 $Max_PriorityList$ 개의 배정 순서와 현재 생성된 배정 순서를 비교함으로써 이전에 검토한 배정 순서가 다시 생성되지 않도록 하며, 새로운 배정 순서를 찾을 때까지 선택된 i/m 의 배정 순서를 한 단계씩 늦춘 다음 단계 4로 이동하게 된다. 만일 선택된 i/m 의 배정 순서를 더 이상 낮출 수 없는 경우에는(마지막 순서이거나 새로운 배정 순서가 생성되지 않는 경우) 재고 비용이 그 다음으로 많이 발생하는 i/m 을 선택하여 이 과정을 반복한다. 한편 재고 비용이 발생하는 i/m 이 존재하지 않으면, 임의의 i/m 을 선택하여 이에 해당하는 기계 배정 순서를 한 단계 앞당긴 후 단계 4로 이동한다.

단계 4 (생산 기계와 생산 시간 결정) : 개선 단계의 휴리스틱에 소요된 시간이 정해진 시간 제한 $TimeLimit$ 을 초과하였으면 단계 5로 이동하고, 그렇지 않으면 초기해 발견 절차의 단계 2와 동일한 방법으로 현재 결정된 배정 순서대로 생산 기계와 생산 시간을 결정하고 초기해 개선 절차의 단계 1로 이동한다.

단계 5 (생산 비용과 재고비용 최적화 및 종료) : 이 단계에서는 현재까지 구한 최선해(x, f, y, u, w)를 대상으로 각 제품의 셋업 시작 시점(y^s)과 생산 시간(f)을 조정함으로써 추가적인 비용 감소를 시도하고, 개선 절차를 종료한다. 즉, 현재까지 구한 최선해에서 이진 변수 x 와 u 의 값을 제2절에서 개발한 수학적 모형 LSSP에 대입함으로써 축소된 수학적 모형 R2-LSSP를 얻게 되며, R2-LSSP의 최적해를 구하여 생산 비용과 함께 재고 비용을 동시에 줄일 수 있다. 여기서 주목할 점은 LSSP의 모든 이진 변수의 값이 결정되었으므로 R2-LSSP가 선형계획법(linear programming) 문제로 표현된다는 것이다.

4. 모의 실험

모의 실험을 통하여 제 3장에서 제시한 휴리스틱 알고리즘의 수행도를 평가하며, 모의 실험 데이터는 다음과 같이 생성한다.

- 각 기계의 생산 능력 설정: $capa_k = 1 + U[1, 2]$, $U[a, b]$ 는 실수 a 와 b 사이의 균일분포(uniform distribution)를 나타낸다.
- 셋업 시간(1시간 단위) 및 셋업 비용:
 - 셋업 시간(s_{ki}) = $\lfloor capa_k \times 2 + U[3, 5] \rfloor$
 - 셋업 비용(β_{ki}) = $s_{ki} \times 300$
- 단위 시간당 생산량과 생산 비용:
 - 단위 시간당 생산량(p_{ki}) = $\lfloor capa_k \times 3 \rfloor$
 - 단위 시간당 생산 비용(α_{ki}) = $p_{ki} \times 10$.
- 셋업당 최대/최소 생산 시간(1시간 단위): $b_k^U = \lfloor U[5, 7] \rfloor$
 $b_k^L = \lfloor U[2, 3] \rfloor$.

- 각 제품의 주문 횟수: $|M(i)| = \lfloor U[T/2, T+1] \rfloor$
- 각 제품의 주문 차수별 납기 시점과 누적 수요량:
 납기 시점(1시간 단위)
 - $T_{i1} = \text{Min}\{s_{ki} + b_k^L : i \in N, k \in K\} \times 2 + \lfloor U[0, 1] \times T / \text{Max}\{|M(i)| : i \in N\}$
 - $T_{im} = T_{i(m-1)} + \text{Min}\{s_{ki} + b_k^L : i \in N, k \in K\} + 2 \times \lfloor U[0, 1] \times T / \text{Max}\{|M(i)| : i \in N\} \rfloor, m=2, \dots, M(i).$
 누적 수요량
 - $D_{i1} = U[10, 20],$
 - $D_{i(m-1)} = D_{im} + U[10, 20], \quad m = 2, \dots, M(i).$
- 재고 유지 비용 :
 - $\pi_{im} = (T_{i(m+1)} - T_{im}) \times 5, \quad m = 1, \dots, M(i) - 1,$
 - $\pi_{im} = 100, \quad m = M(i)$

휴리스틱 알고리즘은 Visual C++과 CPLEX(버전 9.0)를 연동하여 구현하였으며, 최적해는 CPLEX를 이용하여 구하였다.

Table 2. Computational results

No	T	L	K	N	M(i)		목적함수			수행시간(단위 : 초)		변경 횟수
					Min/Max	Avg	CPLEX	Heuristic	Gap	CPLEX	Heuristic	
1	3	12	2	3	2 / 3	2.3	13,700	13,700	1.00	18	360	2912
2	4	12	2	3	2 / 4	2.6	16,160	16,160	1.00	296	360	1945
3	4	12	2	3	3 / 4	3.3	20,550	20,550	1.00	34	360	2535
4	3	12	3	4	1 / 3	2	16,190	16,490	1.02	1,303	360	1741
5	5	15	2	3	2 / 3	2.3	12,330	13,320	1.08	125	360	2269
6	4	16	2	3	2 / 4	2.6	15,910	16,935	1.06	101	360	1496
7	5	15	2	3	3 / 4	3.6	20,590	22,060	1.07	9,990	360	1361
8	5	20	2	3	2 / 3	2.6	14,080	14,680	1.04	306	360	1534
9	5	20	2	3	3 / 5	3.6	20,025	20,990	1.05	7,861	360	1153
10	3	12	3	5	1 / 3	2	19,475	21,275	1.09	6,491	360	1562
11	4	12	3	4	3 / 5	4	25,800	27,110	1.05	10,000	360	1669
12	5	15	3	4	3 / 5	4.2	33,955	34,580	1.02	10,000	360	1184
13	4	16	3	4	3 / 4	4	19,225	19,885	1.03	10,000	360	690
14	3	12	4	5	1 / 3	2.8	30,480	26,785	0.88	10,000	360	1132
15	7	21	2	3	3 / 7	5.6	52,305	35,635	0.68	10,000	360	1363
16	4	12	3	5	3 / 4	3.6	40,025	37,525	0.94	10,000	360	1385
17	3	12	4	6	1 / 3	2.2	25,560	25,560	1.00	10,000	360	910
18	4	16	3	5	2 / 4	3.2	22,825	23,730	1.04	10,000	360	760
19	5	20	3	4	3 / 5	4	22,715	22,710	1.00	10,000	360	894
20	3	12	5	6	1 / 3	1.8	22,360	23,260	1.04	10,000	360	884
21	5	15	3	5	2 / 5	4	44,200	38,700	0.88	10,000	360	672
22	4	12	4	5	3 / 4	3.8	39,030	37,290	0.96	10,000	360	1009
23	4	16	4	5	2 / 5	3.4	23,475	24,065	1.03	10,000	360	814
24	7	21	3	4	4 / 6	5	67,890	42,610	0.63	10,000	360	662
25	5	15	4	5	3 / 5	3.8	40,415	39,025	0.97	10,000	360	560
26	5	20	3	5	3 / 5	3.8	30,360	29,720	0.98	10,000	360	554
27	3	12	5	7	1 / 3	2	27,315	26,755	0.98	10,000	360	788
28	4	12	4	6	3 / 5	4.1	48,525	46,100	0.95	10,000	360	516
29	4	16	4	6	2 / 4	3.3	36,010	31,700	0.88	10,000	360	500
30	7	21	3	5	4 / 7	5	-	53,765	-	10,000	360	536
31	5	20	4	5	2 / 5	4	-	27,990	-	10,000	360	396
32	4	12	5	6	3 / 5	3.8	48,715	47,865	0.98	10,000	360	519
33	5	15	4	6	4 / 5	4.3	-	55,435	-	10,000	360	422
34	5	15	5	6	3 / 4	3.5	40,365	35,610	0.88	10,000	360	363
35	4	12	5	7	2 / 5	3.4	-	54,205	-	10,000	360	396
36	5	20	5	6	3 / 5	4.1	63,715	31,240	0.49	10,000	360	274
37	7	21	4	5	3 / 7	4.8	-	47,035	-	10,000	360	536
38	4	16	5	6	3 / 4	3.8	-	47,865	-	10,000	360	394
39	5	20	4	6	2 / 5	3.8	-	31,390	-	10,000	360	383
40	4	16	5	7	2 / 5	3.4	-	54,205	-	10,000	360	348
41	5	15	5	7	3 / 5	4.1	-	58,610	-	10,000	360	307
42	5	20	5	7	2 / 5	3.3	-	33,630	-	10,000	360	525
43	7	21	4	6	4 / 7	5.5	-	68,930	-	10,000	360	390
44	7	21	5	6	3 / 7	5.5	-	66,335	-	10,000	360	311
45	7	21	5	7	5 / 7	5.8	-	78,390	-	10,000	360	240

모의 실험은 Pentium IV(CPU: 2.8GHz, RAM: 512MB)에서 수행하였으며, 실험 결과를 <Table 2>에 나타낸다. <Table 2>에 표시한 기호의 의미는 다음과 같다. T 는 일(day) 수로 표현된 계획기간의 길이를 나타내며, $M(i)$ 열(column)의 Min, Max 및 Avg는 제품별 주문 회수의 최소값, 최대값 및 평균값을 나타낸다. 휴리스틱 알고리즘의 제한 시간(TimeLimit)은 360초로 설정하였으며, CPLEX의 제한 시간은 10,000초로 설정하였다. CPLEX 수행시간 열에 나타낸 “-” 표시는 10,000초 동안 CPLEX가 초기해도 찾지 못한 경우를 나타낸다. Gap 열에는 (휴리스틱 Upper bound)/(최적해 및 CPLEX Upper bound) 값을 표시하고 마지막 열에는 휴리스틱이 제한 시간 동안 검사한 기계 배정 순서의 개수를 표시한다. 한편 Max_Priority List = 30으로 설정하였다.

CPLEX가 제한 시간 동안 최적해를 발견한 1~10번 문제에서 알 수 있듯이, 휴리스틱 알고리즘은 CPLEX가 구한 최적해와 비교하였을 때 큰 차이를 보이지 않는다(1~10번 문제의 평균Gap은 4.1%). 또한 10,000초 동안 CPLEX가 최적해를 찾지 못한 11~45번 문제 중에서 CPLEX 근사해가 휴리스틱 알고리즘이 발견한 근사해보다 우수한 경우는 6문제(11, 12, 13, 18, 20, 23번)에 불과하며, 이 때의 평균 Gap은 3.5%이다. 한편 휴리스틱 알고리즘이 360초 동안 240회 ~ 2912회의 배정 순서를 변경한 점을 고려하면, R1-LSSP의 최적해를 구하는 데 소요된 시간이 이 논문의 실험에 사용된 가장 큰 크기의 문제에서도 평균적으로 1.5초를 넘지 않음을 알 수 있다. 이와 같은 관찰 결과는 각 기계마다 셋업 차수 별로 생산할 제품이 정해진 상황에서 가능해를 구하기 위해 R1-LSSP의 최적해를 구하는 모형 분해 기법이 시간과 해의 품질 측면에서 매우 유효할 뿐만 아니라, 배정 순서를 변경하는 기법 역시 적절하였음을 나타낸다.

5. 결론 및 추후 연구과제

이 논문에서는 연속 생산 공정에서 생산 비용, 셋업 비용 및 재고 비용의 합을 최소화하는 새로운 형태의 LSSP 문제를 제시하였다. 이 논문에서 제시한 LSSP 문제에서는 분할 생산과 셋업의 이월 및 오버래핑을 고려하였으며, 셋업 횟수의 제한도 완화하였다. 또한 다기간(multi-period) LSSP 문제를 표현하기 위해 기계마다 셋업 차수를 나타내는 인덱스를 사용함으로써 시간 단위를 무한히 작게 표현하여도 수학적 모형의 크기가 증가하지 않는 새로운 형태의 혼합정수계획법 모형을 개발하였다. 모형 분해 기법과 지역 탐색 기법을 적절히 결합한 휴리스틱 알고리즘도 개발하였으며 모의실험 결과, 상용 최적화 소프트웨어(CPLEX)보다 우수한 품질의 해를 매우 빠른 시간 동안 발견함을 확인하였다.

추후 연구과제는 효과적인 절단평면(cutting plane) 개발을 통한 최적해 발견 기법을 개발하는 것과 이 논문에서 제시한 바와 같이 시간을 나타내는 인덱스 대신 셋업 차수를 표현하

는 인덱스만을 이용하여 셋업 순서에 의존적인(setup sequence dependent) 셋업 시간을 고려하는 모델 및 주문량을 충족하지 못하는 상황을 고려한 모델 등을 개발하는 것이다.

참고문헌

- Boctor, F. F. and Poulin, P. (2005), Heuristics for the N-product, M-stage, economic lot sizing and scheduling problem with dynamic demand, *International Journal of Production Research*, **43**(10), 2809-2828.
- Clark, A. R. and Clark, S. J. (2000), Rolling-horizon lot-sizing when set-up times are sequence-dependent, *International Journal of Production Research*, **38**(10), 2287-2307.
- Drexl, A. and Haase, K. (1995), Proportional lotsizing and scheduling, *International Journal of Production Economics*, **40**(1), 73-87.
- Drexl, A. and Haase, K. (1996), Sequential-analysis based randomized-regret-methods for lot-sizing and scheduling, *Journal of the Operational Research Society*, **47**(2), 251-265.
- Gopalakrishnan, M. (2000), A modified framework for modeling set-up carryover in the capacitated lotsizing problem, *International Journal of Production Research*, **38**(14), 3421-3424.
- Gupta, D. and Magnusson, T. (2005), The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and set-up times, *Computers & Operations Research*, **32**(4), 727-747.
- Fleischmann, B. and Meyr, H. (1997), The general lotsizing and scheduling problem, *OR Spektrum*, **19**, 11-21.
- Marinelli, F., Nenni, M. E. and Sforza, A. (2007), Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company, *Annals of Operations Research*, **150**(1), 177-192.
- Meyr, H. (2000), Simultaneous lotsizing and scheduling by combining local search with dual reoptimization, *European Journal of Operational Research*, **120**(2), 311-326.
- Meyr, H. (2002), Simultaneous lotsizing and scheduling on parallel machines, *European Journal of Operational Research*, **139**(2), 277-292.
- Pattloch, M., Schmidt, G. and Kovalyov, M. Y. (2001), Heuristic algorithms for lotsize scheduling with application in the tobacco industry, *Computers and Industrial Engineering*, **39**(3), 235-253.
- Porkka, P., Vepsäläinen, A. P. and Kuula, M. (2003), Multiperiod production planning carrying over set-up time, *International Journal of Production Research*, **41**(6), 1133-1148.
- Silva, C. and Magalhaes, J. M. (2006), Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry, *Computers and Industrial Engineering*, **50**(1), 76-89.
- Sox, C. R. and Gao, Y. (1999), The capacitated lot sizing problem with setup carry-over, *IIE Transactions*, **31**(2), 173-181.
- Staggemeier, A. T. and Clark, A. R. (2001), A survey of lot-sizing and scheduling models, *23rd Annual Symposium of the Brazilian Operational Research Society*, 938-947.
- Suerie, C. and Stadtler, H. (2003), The capacitated lot-sizing problem with linked lot sizes, *Management Science*, **49**(8), 1039-1054.
- Suerie, C. (2006), Modeling of period overlapping setup times, *European Journal of Operational Research*, **174**(2), 874-886.
- Zanoni, S. and Zavanella, L. (2005), Model and analysis of integrated production-inventory system: The case of steel production, *International Journal of Production Economics*, **93**(94), 197-205.