

GRASP 기법을 이용한 주문이월과 자원제약을 고려한 공급사슬 망에서의 생산계획 알고리즘

신현준[†] · 이영섭

상명대학교 경영공학과

A Production Planning Algorithm for a Supply Chain Network Considering Back-Order and Resource Capacity Using GRASP Method

Hyun Joon Shin[†] · Young Sup Lee

Department of Management Engineering, Sangmyung University

In an environment of global competition, the success of a manufacturing corporation is directly related to how it plans and executes production in particular as well as to the optimization level of its process in general. This paper proposes a production planning algorithm for the Multi-Level, multi-item Capacitated Lot Sizing Problem (MLCLSP) in supply chain network considering back-order. MLCLSP corresponds to a mixed integer programming (MIP) problem and is NP-hard. Therefore, this paper proposes an effective algorithm, GRHS (GRASP-based Rolling Horizon Search) that solves this problem within reasonable computational time and evaluates its performance under a variety of problem scenarios.

Keywords : Production Planning Algorithm, Back-Order, Resource Capacity, Multi-Level Multi-Item Capacitated Lot Sizing, Supply Chain Network

1. 서론

Higgins과 Slack에 의하면 생산계획은 생산을 계획하고 통제하는데 있어서 가장 중요한 활동이며, 제조 기업은 생산계획을 통해 시장의 수요와 기업의 내부 자원들을 효율적으로 조율할 수 있다[8, 19]. 더욱이 최근의 글로벌 제조 기업을 위한 생산계획 모델들은 공급사슬 망에서의 다양한 생산 활동들을 성공적으로 관리하기 위해 분산되어 있는 제조시설들을 총체적으로 고려하고 있다. 그간 제조시스템의 생산성과 효율성에 대해서는 집중적으로 많이 연구되어 온 반면, 다단계의 제조시설 또는 다단계의 기업을 아우르는 물류-생산 문제들에 대

한 폭 넓은 접근 방법은 부족했던 것이 사실이다 [23].

본 논문은 공급사슬 망에서 제조자원의 한정된 능력과 주문 이월을 고려한 생산계획을 제공하는 알고리즘 개발에 주안점을 둔다. 본 연구의 목적은 주문/셋업(setup)비용, 재고유지비용 및 주문 이월비용의 합을 최소화하는 것을 목적함수로 두고, 공급사슬 망에서 주문이월을 허용하는 MLCLSP(Multi-Level, multi-item Capacitated Lot Sizing Problem) 문제의 해를 효과적으로 구하는 GRHS (GRASP-based Rolling Horizon Search) 알고리즘을 개발하는 것이다.

일반적으로 로트(lot) 사이즈 결정 문제는 주문비용과 재고유지 비용만의 합을 최소화하는 것을 목적으로 하고, 특정 제조시설에서 특정 품목을 특정 기간에 셋업하

논문접수일 : 2009년 01월 15일 논문수정일 : 2009년 05월 25일 게재확정일 : 2009년 7월 23일

[†] 교신저자 hshin@smu.ac.kr

※ 본 연구는 2008년도 상명대학교 교내연구비 지원을 받아 이루어졌음.

고 생산할지 여부를 결정하는 것이다. 만약 자원 제약하에 고려한다면, 이것은 자원 제약하의 로트 사이즈 결정문제 (capacitated lot sizing problem : CLSP)가 된다. 본 논문이 사용하는 모델은 주문이월을 허용하는 자원 제약하의 다단계 다품목 로트 사이즈 결정 문제, 즉 MLCLSP이다. 본 논문에서 다루는 MLCLSP는 계획 수립기간 동안 수요와 자재명세서(BOM)가 주어졌을 때 재고 유지비용과 주문/셋업 비용 그리고 주문이월비용의 합을 최소화하기 위한 생산 계획을 결정하는 것이다. 이 때 최종 제품에 대한 BOM에서 구성 부품들 역시 공급망 상의 제조시설들에서 각각 생산될 수 있다. 여기서 고려해야 할 주요 제약 사항들은 다음과 같다. (1) 주문이월(back-order)이 허용된다; (2) 각 제조시설 내의 자원들의 용량은 한정되어 있다; (3) 재고가 없다고 가정했을 때 BOM상의 하위 품목의 생산이 모두 끝나야 상위 품목의 생산이 시작될 수 있다; (4) 전체 계획시평(planning horizon)은 일정한 시 구간(time period)들로 나뉜다; (5) 공급망 상의 모든 제조 시설들은 종속수요 외에 독립적인 외부 수요를 가질 수 있다.

다단계 로트 사이즈 및 스케줄링은 이 문제를 다루고 있는 기존 논문들에서 지적되어 있듯이 가장 어려운 단기 생산계획 문제에 속한다[16, 19, 23]. 일반적으로 CLSP 모델은 시구간이 셋업 시간에 비해 상대적으로 매우 큰 경우와, 매우 작아 하나의 시구간 내에 최대 1회의 셋업만 발생하는 경우로 나누어 볼 수 있다. 전자에 해당하는 모델의 경우, Eppen과 Martin 및 Stadler가 다양한 형태의 문제 모형을 제시하였고, Trigeiro 등 과 Tempelmeier와 Derstroff가 여러 개의 알고리즘을 제안하였다[7, 24-26, 29]. 후자에 해당하는 모델에 경우, 로트 사이즈 결정문제 뿐만 아니라 스케줄링 문제에 해당하는 세부 생산계획을 수립하는 수준의 문제가 된다. 이와 같은 이산 로트 사이즈 및 스케줄링 문제인 DLSP(Discrete Lot sizing and Scheduling Problem)는 하나의 시구간 당 최대 1제품의 생산만을 허용하고 시구간 전체에 걸쳐 생산을 하는 경우와 아니면 아예 하지 않는 경우로만 나누어진다. Fleischmann[11]은 셋업 시간이 0인 DLSP를 위해 분지한계법(branch and bound) 알고리즘을 제시하였다. 또한 연속 셋업 로트 사이즈 문제인 CSLP(Continuous Setup Lot-sizing Problem)는 하나의 시구간 당 한제품의 생산만 허용하지만 생산량의 제한은 두지 않는다. CSLP에 대해서 Karmarkar와 Schrage[15] 역시 분지한계법 알고리즘을 제안하였는데, 이 알고리즘은 라그랑지안 완화(Lagrangean relaxation) 기법을 이용해 전체 문제를 분해하고 하한(lower bound)을 얻어내어 사용하는 방식이었다. 반면에 비례 로트 사이즈 및 스케줄링 문제인 PLSP(Proportional Lot sizing and Scheduling Pro-

blem)는 가용한 자원을 공유한다는 전제하에 하나의 시구간에 최대 2개 제품의 생산을 허용한다. Drexl과 Haase [6]는 이 PLSP문제에 대한 해법으로 새로운 BRBR(backward-oriented regret-based biased random sampling) 기법을 제시하였다.

Bahl 등[5]은 CLSP 문제를 단일 단계와 다단계로 구분하였고 자원 제약하의 다단계 문제에 대한 접근이 가장 어렵다고 언급하였다. Bahl과 Ritzman[4]은 자재소요계획(MRP) 계산의 복잡성이 바로 MLCLSP 문제의 해를 구하는 것에 해당한다는 것을 보였다. Florian 등[11-13]은 한 가지 자원 유형에 셋업시간이 존재할 경우 단일품목 문제들의 일부는 NP-hard라는 것을 보였다. Tempelmeier와 Helber[27, 28]는 MLCLSP 문제를 위한 4종류의 2단계 휴리스틱 방법을 제시하였고, 그 후에 그들은 효과적인 Lagrangian relaxation 휴리스틱을 제안하였다. 이 방법론에서, 그들은 단일 품목의 자원 제약이 없는 로트 사이즈 결정 하위문제들을 얻기 위해 재고와 자원 용량 제약을 완화하였다. 이 하위문제에 대해서는 효율적인 $O(n \log n)$ 알고리즘이 존재하므로, 그들은 전체 문제에 대한 효율적인 탐색알고리즘을 개발할 수 있었다. 그들의 논문이 다루었던 문제는 위의 Florian 등이 증명하였던 문제보다 복잡하다. 따라서 이 문제의 최적해를 구하는 효율적인 알고리즘은 아직 존재하지 않으므로, 본 논문은 근접 최적해를 얻기 위한 방편으로 혼성 탐색 휴리스틱 알고리즘을 개발하였다. Kimms[17]는 다단계 복수기계를 갖는 PLSP문제에 대한 효율적인 해를 구하기 위해 유전자 알고리즘을 제안하고 그 성능을 타부 탐색 알고리즘과 비교하였다. Sahling 등[24]은 셋업비용과 재고유지비용 외에 잔업비용을 함께 고려하는 MLCLSP 문제에 대해 분할기법의 일종인 fix and optimize 휴리스틱을 제안하였고, Akartunali와 Miller[3]는 일반적인 재고비용만을 고려하는 MLCLSP 문제의 효과적인 솔루션을 얻기 위해 time window 기법을 적용하였다. 그들은 time window 내의 이진정수 변수인 셋업 변수들에 대해서는 정수형 변수로 유지하고, 반면 time window 밖의 셋업 변수들에 대해서는 실수형 변수로 이완하여 해를 구하는 방법을 고안하였다.

공급사슬관리 문제들에 대한 메타 휴리스틱적인 접근 방법론에 대해서는 지금까지 일부 소개되어 왔다. 그러나 그 중 많은 연구들이 이웃해 표현 기법과 가능해(feasible solution)를 지속적으로 유지하는 것이 어렵다는 이유로 위치문제 또는 할당문제로 변형된 모델들에 대해서만 초점을 맞춰왔다. Ross[22]는 데포(depot)와 운송수단의 운영비용을 최소화 하기 위해 기존의 분배망을 위한 재구성 문제에 대해 시뮬레이티드 어닐링 방법을 제시하였다. Jayaraman과 Ross[14]는 하나의 공장과 복수개의 분배센터

및 소매점으로 구성된 문제를 다루었고, 시뮬레이티드 어닐링 방법을 이용해 가장 좋은 분배센터의 집합을 선택하고 작업결정 프로세스를 통한 운송수단의 확정 방식으로 소매점들의 할당을 결정하였다. Vergara 등[30]은 공급자로부터의 재 보충 비용을 최소화하는 문제에 진화 알고리즘을 적용하였다. 그들은 이웃해를 생성하기 위해 염색체(chromosome)를 재 보충 순서로 고안하였다. 또한 Almeder [2]는 일반적인 재고비용 외에 잔업비용 최소화를 함께 고려하는 MLCLSP 문제에 Ant 탐색기법을 적용하는 알고리즘을 제안하였다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 공급사슬망 생산계획 문제의 모형을 제시하고, 제 3장은 제 2장에서 제시된 문제의 해를 구하기 위한 알고리즘을 설명한다. 제시된 알고리즘의 성능 분석은 제 4장에서 기술하며, 마지막으로 제 5장은 결론과 추후 연구방향에 대해 언급한다.

2. 문제 모형

본 논문은 공급사슬 망상에서 복수 개로 구성되는 제조 공장들을 대상으로 효율적인 생산계획을 수립하여 제공하는 것을 목적으로 한다. 각 제조시설들은 한정된 자원 용량을 갖고 있으며 복수개의 중간/최종제품을 취급한다. 제품의 생산은 각 시설의 자원을 공유할 수 있다. 본 논문의 대상이 되는 문제를 수학적으로 모형화하는데 사용되는 가정과 기호는 다음과 같다.

Assumptions:

- (1) 모든 주문에 대해서 주문 이월이 허용된다.
- (2) 각 제조시설은 종속수요 외에 독립적인 외부 수요를 가질 수 있다.
- (3) 각 자원은 주어진 생산능력 제한이 있다.
- (4) 한 부품은 몇 개의 상위 부/제품을 위해 공용부품으로 사용될 수 있다(일반화 BOM).
- (5) 계획 시평은 복수의 계획 기간들로 균등하게 나뉘어진다.
- (6) 한 제품의 생산은 해당 기간 동안에만 자원의 용량을 소모한다.
- (7) 재고 유지, 셋업, 주문 이월비용은 기간 t 에 걸쳐 동일하게 적용된다.
- (8) 한 제품의 생산을 위한 자원의 셋업은 다음 기간으로 연장되어 넘어가지 않는다.
- (9) 제조시설 간의 부품 운송시간은 가공시간에 포함되어 있다.

Notations :

$k = 1, \dots, K$ 중간/최종 제품의 인덱스

$t = 1, \dots, T$ 계획 기간의 인덱스

$j = 1, \dots, J$ 제조시설의 인덱스

$S(k)$ BOM상에서 제품 k 를 필요로 하는 상위 제품 인덱스들의 집합

q_{kjt} 시설 j 에서 제품 k 의 기간 t 동안의 생산량

z_{kjt} 시설 j 에서 제품 k 의 기간 t 말의 재고수준

bo_{kjt} 시설 j 에서 제품 k 의 기간 t 동안의 주문 이월량

c_{jt} 시설 j 의 기간 t 동안의 가용 자원용량

d_{kjt} 시설 j 에서 제품 k 의 기간 t 동안의 수요량

a_{ki} BOM상에서 상위 부/제품 i 를 한 단위 생산하는데 소요되는 부품 k 의 수량

h_{kj} 시설 j 에서 제품 k 의 재고 유지비용

s_{kj} 시설 j 에서 제품 k 의 셋업 비용

b_{kj} 시설 j 에서 제품 k 의 주문 이월비용

v_{kjt} 시설 j 에서 제품 k 한 단위를 기간 t 동안에 생산하는데 필요한 시간

u_{kjt} 시설 j 에서 제품 k 를 기간 t 에 셋업 하는데 소요되는 시간

l_k 부/제품 k 의 리드타임

M 매우 큰 값

y_{kjt} 기간 t 에 시설 j 에서 제품 k 의 셋업 여부. 이진 결정 변수이며 q_{kjt} 와의 관계는 다음과 같이 정의된다.

$$y_{kjt} = \begin{cases} 0 & \text{if } q_{kjt} = 0; \\ 1 & \text{otherwise.} \end{cases}$$

위의 가정과 기호들과 함께, 본 논문의 대상이 되는 공급사슬 망 생산계획 문제는 다음과 같은 MIP문제 (MLCLSP)로 모형화 될 수 있다.

MLCLSP:

$$\text{Minimize } \sum_{k=1}^K \sum_{j=1}^J \sum_{t=1}^T (s_{kj} y_{kjt} + h_{kj} z_{kjt} + b_{kj} bo_{kjt}) \quad (1)$$

subject to

$$q_{k,j,t-1} + z_{k,j,t-1} - d_{kjt} - bo_{k,j,t-1} - \sum_{i \in S(k)} a_{ki} q_{ijt} + bo_{kjt} = z_{kjt} \quad \forall k, j \text{ and } t, \quad (2)$$

$$\sum_{k=1}^K (v_{kjt} q_{kjt} + u_{kjt} y_{kjt}) \leq c_{jt} \quad \forall j \text{ and } t, \quad (3)$$

$$bo_{kjt} - bo_{k,j,t-1} \leq d_{kjt} \quad \forall k, j \text{ and } t, \quad (4)$$

$$q_{kjt} - My_{kjt} \leq 0 \quad \forall k, j \text{ and } t, \quad (5)$$

$$z_{kjt} = 0 \quad \forall k \text{ and } j, \quad (6)$$

$$z_{kjt} \geq 0, q_{kjt} \geq 0 \quad \forall k, j \text{ and } t, \quad (7-1)$$

$$bo_{kjt} \geq 0, y_{kjt} \in \{1, 0\} \quad \forall k, j \text{ and } t, \quad (7-2)$$

위의 공급사슬망 생산계획 모델은 전체 계획 시평에 걸쳐 각 계획 기간별로 수행된다. 모든 중간/최종 제품은 재고로 보관되는 동안 유지비용을 발생시키며, 구매/생산 주문이 발주됨과 동시에 셋업 비용을 발생시킨다. 또한 해당기간의 수요를 충족시키지 못하는 경우 주문이월 비용이 발생한다. 식 (1)의 목적함수는 셋업과 재고 유지 그리고 주문 이월비용의 합을 최소화하는 것이다. 식 (2)는 계획 시평 동안 각 제조시설에 존재하는 재고 균형에 대한 제약이다. 본 제약식은 일반적인 재고균형식의 좌변에 $-bo_{k,j,t-1}$ 과 bo_{kjt} 항을 추가함으로써 주문이월을 허용하도록 한다. 좌변의 $q_{k,j,t-1}$ 는 수요 d_{kjt} 를 충족시키기 위해 리드타임인 l_k 전에 생산/구매 지시되어 현재 구간인 t 에 가용하게 된 수량을 의미한다. 여기에 $z_{k,j,t-1}$ 는 이전 기간($t-1$)에서 이월된 재고 물량으로써 더해지고, $-d_{kjt}$ 는 현 기간(t) 동안의 제품 k 의 수요로써 차감된다. 또한 여기에 $-bo_{k,j,t-1}$ 항은 이전 기간에 발생한 주문이월로써 차감되고, $-\sum_{i \in S(k)} a_{ki} q_{ijt}$ 항은 BOM 상의 상위 제품들($i \in S(k)$)의 제품 k 에 대한 수요이므로 차감된다. 따라서 좌변식에서 수요를 충족시킬 가용재고가 부족하여 $q_{k,j,t-1} + z_{k,j,t-1} - d_{kjt} - bo_{k,j,t-1} - \sum_{i \in S(k)} a_{ki} q_{ijt}$ 부분이 음수가 된다면 그 수량만큼이 주문이월량인 bo_{kjt} 로 결정되고, 만약 0 또는 양수가 된다면 비용최소화 문제 특성상 주문 이월량은 0이 되며 양수인 부분이 우변의 재고수준 값인 z_{kjt} 가 된다. 식 (3)은 각 제조시설에 대해 주어진 가용 용량 이내에서 생산이 이뤄져야 한다는 제약이다. 식 (4)는 현 기간 t 에서 결정될 주문 이월량 (bo_{kjt})과 이전 기간 $t-1$ 에서 발생한 주문 이월량 ($bo_{k,j,t-1}$)의 차이가 현 기간의 수요 (d_{kjt})를 초과할 수 없게 함으로써 주문 이월량의 상한이 누적 주문 이월량과 해당 기간의 수요가 되게 한다. 식 (5)는 생산과 셋업 간의 관계식으로 생산을 위한 셋업이 없다면($y_{kjt} = 0$) 생산량이 존재할 수 없으며($q_{kjt} = 0$) 그 역도 마찬가지임을 뜻한다. 식 (6)은 마지막 시구간에서는 재고량 z_{kjt} 이 없음을 의미한다. 본 모델은 최종제품을 생산하는데 필요한 BOM 구조를 a_{ki} 로 표현하고 셋업 비용, 재고비용, 주문 이월비용 간의 절충을 고려함과 동시에 자원제약을 반영한다는 면에서 생산계획 문제의 주요한 특성들을 포함하고 있다고 할 수 있다.

3. 생산계획 알고리즘

MLCLSP와 같은 MIP 문제의 최적 해를 구하기 위해서는 일반적으로 분지한계법(brach-and-bound)과 같은 암묵

적 열거법(implicit enumeration method)들의 변형 알고리즘을 사용한다. 그러나 이러한 접근방법은 현실적인 크기의 문제에 직면하면 그 계산시간 면에서 실효성이 없기에 알고리즘 수행시간 대비 해의 우수성 면에서 효율적인 알고리즘 개발이 필수적이다. 따라서 본 연구는 근접 최적해를 찾는 알고리즘을 개발하는 것을 목표로 한다. 서론에서도 언급하였듯이, 본 논문에서 제안하는 GRHS (GRASP-based Rolling Horizon Search) 알고리즘은 GRASP (Greedy Randomized Adaptive Search Procedures) 탐색의 틀 안에서, MIP문제를 이완 (relaxation) 하여 이완된 LP 문제를 일정한 시간구역 (Time fence)을 변경해 가며, 특정 시간구역에 포함된 계획에 대해서만 반복적으로 풀어 나감으로써 효율적인 셋업 계획 및 생산량을 동시에 찾아 나가는 방식을 취한다.

```

procedure GRASP(maxitr, seed)
  Read Input();
  for  $k = 1, 2, \dots, \text{maxitr}$  do
    Solution  $\leftarrow$  Greedy Randomized Construction(seed),
    Solution  $\leftarrow$  Local Search(Solution),
    Update_Solution(Solution, Best_Solution),
  endfor,
  return Best_Solution,
end GRASP,
  
```

<그림 1> GRASP procedures

GRHS 알고리즘은 먼저 MLCLSP 모형 중 이진 정수형 셋업 변수인 y_{kjt} 의 값을 결정한다. 이것은 MIP로 정의되는 MLCLSP를 LP 문제로 완화시킨다. 다시 말해서 식 (3)에서 y_{kjt} 변수들의 값이 주어지면, 즉 전체 문제의 셋업 패턴이 결정되면 이것이 셋업시간 u_{kjt} 과 곱해지고 우변으로 이항되어 자원 용량 값 c_{jt} 과 함께 계산되면서 상수화가 된다. 이는 식 (1)과 (5)에도 마찬가지로 적용된다. 이와 같이 함으로써 전체 모델 MLCLSP는 정수형 변수가 존재하지 않게 되며 간단한 단체법(simplex)으로 최적해를 구할 수 있는 LP 문제로 변형된다. LP 문제의 최적해를 구해주는 소프트웨어들은 많이 있다. 본 연구는 XPRESS-MP(release 2007) 소프트웨어로 LP문제를 모델링하고 최적해를 구한다. GRHS 알고리즘을 통한 탐색은 지속적으로 더 좋은 정수 변수 해, 즉 효과적인 셋업패턴을 찾아 나가며 이것은 바로 이어서 LP 문제의 해를 구해봄(생산량 q_{kjt} 결정)으로써 그 성능이 평가될 수 있다. 그런 이유로 본 논문에서 제안하는 알고리즘은 LP 문제풀이와 탐색알고리즘 간의 협력을 바

```

evaluate ( $S, f_{XpressMP}, f_{Lingo}, V, improved, H$ )
 $S$  is the current setup pattern
 $f_{XpressMP}$  is the returned objective function value from Xpress-MP for the current solution  $S$ 
 $V$  is the best objective function value found yet
 $improved$  is returned as true if  $S$  improves  $V$ , else returned as false
 $H$  is all the improved solutions found during the course of the algorithm

begin
 $improved := false$ 
 $f_{XpressMP} := Xpress-MP(S)$  /* solve LP by Xpress-MP */
if  $f_{XpressMP} = infeasible$  then return /* infeasible */
if  $f_{XpressMP} < V$  then /* improved solution */
     $improved := true$  /* set improved */
     $V := f_{XpressMP}$  /* update  $V$  */
     $H := H \cup S$  /* add  $S$  to  $H$  */
end if
end

```

<그림 2> Evaluation procedures

```

Construction ( $S, f_{XpressMP}, Seed$ )
 $S$  is the setup pattern and consists of:
     $y_{kjt}$  binary integer variable about whether or not to produce for all  $k, j$  and  $t$ 

begin
for all  $k, j$  and  $t$  do /* convert a MIP into a LP */
    if  $rand(Seed) > 0.5$  then /* randomly decide the binary integer variable  $y_{kjt}$  */
         $y_{kjt} := 1$ 
    else
         $y_{kjt} := 0$ 
    end if
end for
 $f_{XpressMP} := Xpress-MP(S)$  /* solve the reduced LP by Xpress-MP and return objective function value */
end

```

<그림 2> Construction procedures

탕으로 하는 혼성알고리즘(hybrid algorithm)이라 할 수 있다. 탐색알고리즘의 자세한 설명은 다음과 같다.

3.1 GRASP

GRASP는 유전자알고리즘, 시뮬레이티드 어닐링 및 타부탐색과 함께 조합최적화 문제에서 근접 최적해를 찾는 데 적합하도록 설계된 메타 휴리스틱으로 잘 알려져 있다[1, 9, 21]. GRASP는 기존 문헌에서 잘 기술되어 있으므로 본 논문에서는 기본적인 작동원리를 중심으로 설명하기로 한다.

GRASP는 다중 재시작(multi-start) 또는 반복과정을 통해 탐색을 수행하며, 각 반복은 2 단계로 구성된다. 첫 번째 단계는 구성단계로써 하나의 가능해를 생성하고 두 번째 단계는 첫 번째 단계에서 제공된 가능해로부터 이웃해들을 생성하여 그 중에서 지역 최적해를 찾는 단계이다. 최소화 문제에 대해 최대 $maxitr$ 회 반복하도록 설

정된 GRASP는 다음 <그림 1>의 의사코드(pseudo-code)와 같은 형태로 표현될 수 있다.

3.2 GRHS 알고리즘

GRHS 알고리즘은 $maxitr$ 회의 다중 재시작(multi-start)을 통해 복수 개의 초기해를 중심으로 탐색을 시도하는 전역 탐색(global search)의 성격을 갖는다. <그림 2>는 매 회의 반복에서 초기해를 선정하는 construction 절차다. 이 절차에서는 난수를 발생하여 임의로 셋업패턴을 생성하고 완료된 LP문제의 해와 목적함수를 구한다. 그러나 MLCLSP 문제의 크기가 커짐에 따라 하나의 초기해로부터 생성되는 이웃해는 급격하게 증가하므로, 아무리 LP문제를 빠른 시간 내에 푼다고 하더라도 반복적인 탐색과정의 모든 이웃해를 평가(<그림 3> 참고)하기에는 알고리즘 수행속도에 많은 부하를 주게 된다.

따라서 본 연구에서는 효율적인 탐색을 위해 일정한

```

GRHS (maxitr, TBase, TFence, Seed)
   $S^*$       is the current setup pattern and consists of:
              $y_{kjt}$  binary integer variable about whether or not to produce for all  $k, j$  and  $t$ 
   $S^{**}$     is the best neighbor of  $S^*$ 
   $V^{**}$     the objective function value for  $S^{**}$ 

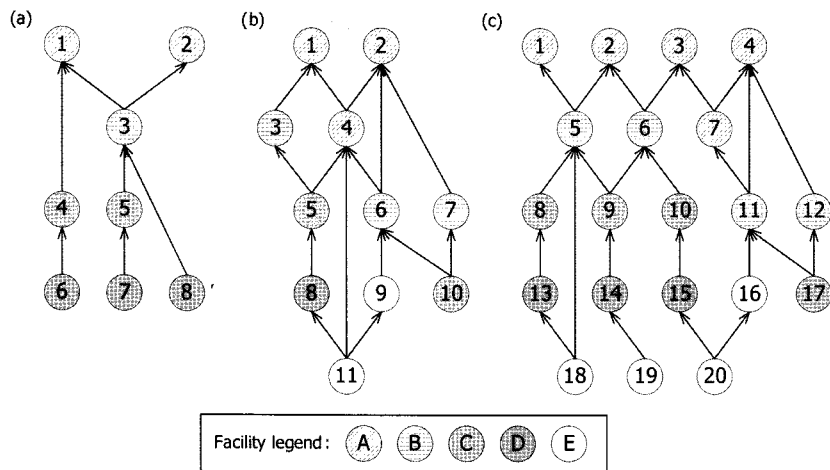
begin
  STBase := TBase                                /* initialize STBase and STFence */
  STFence := TFence
  for itr := 1 to maxitr do                        /* iterate for maxitr */
    Construction ( $S^*$ ,  $f_{XpressMP}$ , Seed)           /* construction phase */
    while TBase < T do                            /* local search phase - iterate until TBase reaches T */
       $V^{**}$  :=  $\infty$                                 /* initialize  $V^{**}$  */
      for all  $k, j$  and TBase ≤ t < TFence do /* search only for period between TBase and TFence */
        if  $y_{kjt} = 0$  then                        /* switch setup pattern  $y_{kjt}$ ; 0→1 or 1→0 for particular  $k, j$  and  $t$  */
           $y_{kjt} := 1$  and flag:=1
        else
           $y_{kjt} := 0$  and flag:=0
        end if
        evaluate ( $S^*$ ,  $f_{XpressMP}$ ,  $V^{**}$ , improved, H) /* evaluate the changed setup pattern */
        if improved then                          /* if improved */
           $V^{**} := f_{XpressMP}$                        /* update  $V^{**}$  with  $f_{XpressMP}$  */
           $S^{**} := S^*$                                /* update  $S^{**}$  with  $S^*$  */
        else
          if flag = 1 then                          /* then restore the changed setup pattern to the previous value */
             $y_{kjt} := 0$ 
          else
             $y_{kjt} := 1$ 
          end if
        end if
      end for
       $S^* := S^{**}$                                   /* change the current setup pattern  $S^*$  with  $S^{**}$  */
      TBase := TBase + STBase                       /* increase TBase by STBase */
      TFence := TFence + STFence                   /* increase TFence by STFence */
    end while
    TBase := STBase                                /* reset the increased TBase to the initial TBase value */
    TFence := STFence                              /* reset the increased TFence to the initial TFence value */
  end for
end

```

〈그림 3〉 GRHS algorithm procedures

시구간(time window)을 두고 이 시구간을 이동 (rolling horizon)하면서 각 구간 내에 속하는 셋업변수(y_{kjt})만을 이웃해 생성의 대상으로 삼도록 한다. 각 시구간은 GRHS 알고리즘의 모수인 *TBase*와 *TFence*값에 의해 결정되며 한 구간의 시작 시점은 *TBase*이고 종료 시점은 *TFence*까지이다. 이러한 시구간은 rolling horizon 방식을 따라 계획 시평의 앞에서 뒤로 이동하는데, 현재해(current solution)의 모든 이웃해 들을 생성 및 평가하는 하나의 탐색이 완료 될 때마다 결정 모수인 *TBase*와 *TFence*값 만큼씩 앞으로 전진한다. GRHS 알고리즘에서 사용되는 각 결정모수 들이 알고리즘 성능에 미치는 영향은 제 4장에서 실험 을 통해 분석하기로 한다.

〈그림 3〉에 설명되어 있듯이 GRHS 알고리즘의 이웃 해 생성 방안은 다음과 같다. 주어진 해에서 정수형 변수 집합 y_{kjt} 중 하나의 변수를 0에서 1 또는 1에서 0으로 변화시킨다. 원래 MLCLSP 모형에는 전체 $K \times J \times T$ 개의 y_{kjt} 정수형 변수들이 존재하나, *TBase*와 *TFence*로 결정되는 한 시구간 내에서는 현재해로부터 최대 $K \times J \times (TFence - TBase)$ 개의 이웃해가 생성되고 평가되며, 이러한 이웃해들 중 가장 좋은 해로 이동(move)이 일어남으로써 일회의 탐색이 완료된다. 이 때 결정모수 *TBase*와 *TFence*값이 각각 0과 *T*로 주어지면 GRHS 알고리즘은 $K \times J \times T$ 개의 전체 y_{kjt} 변수들을 이웃해 생성의 대상으로 하는 GRASP의 기본형태가 된다.



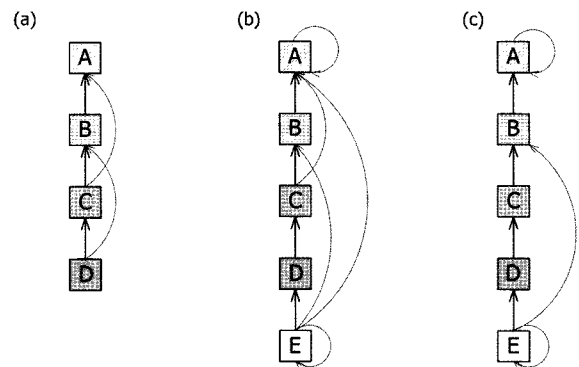
<그림 4> BOM structures for each problem type : (a) Small, (b) Medium and (c) Large size

4. 실험 및 분석

4.1 실험계획

제조환경의 복잡도 고저에 따라 본 논문에서 제안한 알고리즘의 성능을 객관적으로 평가하기 위해, 3가지 형태의 제조환경(small, medium and large size)을 갖도록 문제유형을 고안하였다. 그리고 각 문제 유형에 대하여 각각 10개의 문제 데이터를 임의로 생성하였다. <표 1>은 문제 데이터를 생성하는데 사용된 모수들을 정리하고 있다. 3가지 문제 형태 중 small size 유형은 6개의, medium 및 large size는 18개의 계획기간을 각각 갖는다. Small size 유형 문제는 네 개의 제조시설(A, B, C, D)과 8개의 제/부품으로 구성되어 있고, medium 및 large size 유형 문제는 다섯 개의 제조시설(A, B, C, D, E)과 각각 11개와 20개의 제/부품으로 구성되어 있다. 또한 small과 medium 문제는 2개의 최종제품을, large 문제에서는 4개의 최종제품을 생산한다. 각 문제유형에 대해 제품의 BOM구조와 각 제조시설에서 생산되는 제품의 관계도를 <그림 4>에서 정리하고 있다. 한 예로, <그림 4>와 <그림 5>의 (c)는 16개의 중간부품들(5에서 20)과 4개의 최종제품들(1에서 4)이 5곳의 제조시설(A, B, C, D, E)에서 생산된다는 것을 보여준다. 즉, {1, 2, 3, 4, 7}, {5, 6, 11, 12}, {8, 9, 10, 17}, {13, 14, 15} 그리고 {16, 18, 19, 20} 제/부품들은 각각 제조시설 A, B, C, D 그리고 E에서 생산된다. 실험과 분석을 보다 단순하고 명료하게 하기 위해, 모든 리드타임과 BOM 상하위 요구량 모수를 1로 설정하였다. 또한 기간별 각 제조시설의 생산능력은 다음과 같은 방식으로 산정하였다.

$$C_{jt} = Avg(d_{kjt}) \times (Avg(v_{kit}) + Avg(u_{kjt})) \quad (8)$$



<그림 5> Facility structures for each problem type : (a) Small, (b) Medium and (c) Large size

즉, 각 제조시설의 생산능력은 <표 1>과 같이 주어진 해당 제조시설의 기간별 수요평균에 평균 가공시간 및 평균 셋업시간의 합을 곱하여 산출한다.

4.1.1 알고리즘 비교 대안

GRHS 알고리즘의 성능을 비교 평가하기 위하여, 본 논문에서는 제 2장에서 모형화한 MLCLSP 모델 (MIP 문제)의 해를 small, medium, large 유형별로 구한다. 여기서, MIP 문제의 해는 Xpress-MP(Optimizer 버전 18.1) 소프트웨어를 이용하여 구하며, small 및 medium 문제에 대해서는 최적해를, large 문제에 대해서는 24시간이 경과했을 때의 가능해를 구하도록 한다. 성능 평가는 Xpress-MP 소프트웨어가 산출한 목적함수 값 및 수행시간 대비 GRHS의 결과값을 비교하도록 한다.

4.1.2 GRHS 알고리즘 실험 모수

GRHS 알고리즘의 성능에 주요한 영향을 미치는 결정 모수는 1) 전체 반복회수를 결정하는 *maxitr*, 각 시구간의

폭과 시구간의 이동 속도를 결정하는 2) *TBase*과 3) *TFence* 이다. 본 실험에서는 이들 결정모수가 GRHS 알고리즘에 미치는 영향을 분석하고, 모수들의 효과적인 적정 값을 제안하기 위하여 small, medium, large 유형별로 <표 2>와 같이 결정 모수 값들을 설계하였다.

<표 1> Experimental design

Parameters	Values
계획기간의 수(<i>T</i>)	6 / 18
제조시설의 수(<i>J</i>)	4(A, B, C, D) / 5(A, B, C, D, E)
리드타임	1
BOM 상하위 요구량(<i>a_{kj}</i>)	1
단위제품당 가공시간(<i>v_{kjt}</i>)	Uniform[0.5, 2.0]
셋업시간(<i>u_{kjt}</i>)	Uniform[5, 12]
셋업비용(<i>s_{kj}</i>)	Uniform[10, 20]
재고유지비용(<i>h_{kj}</i>)	Uniform[1, 10]
주문이월비용(<i>b_{kj}</i>)	Uniform[20, 30]
기간별 수요량(<i>d_{kjt}</i>)	Uniform[5, 30]
부/제품 수(<i>K</i>)	8, 11, 20
문제 수	10

<표 2> Decision parameter design

Parameters Types	Parameters		
	<i>maxitr</i>	<i>TBase</i>	<i>TFence</i>
Small	1, 3, 5	2, 4	3, 6
Medium	1, 3, 5	2, 4, 6	3, 6, 12, 18
Large	1, 3, 5	2, 4, 6	3, 6, 12, 18

4.2 알고리즘 평가 및 분석

문제크기 및 복잡도와 같은 각 문제들의 특성들이 주어진 임의의 생성 데이터에 따라 좌우되므로, 본 논문은 성능 평가 척도로서 다음과 같은 상대오차값 *REO*(relative error on objective value)와 *RET*(relative error on time)을 각각 사용한다.

$$REO = \frac{S_{GRHS} - S_{MP}}{S_{MP}}, \tag{9}$$

여기서,
S_{GRHS} = GRHS 알고리즘에 의해 얻은 총 비용,
S_{MP} = MLCLSP(MIP 문제)를 XPRESS-MP로 구했을 때 얻는 총 비용. 단, 여기서 small 및 medium 문제의 경우는 최

적해, large 문제의 경우는 24시간 수행 후 얻은 가능해임.

$$RET = \frac{T_{GRHS} - T_{MP}}{T_{MP}} \tag{10}$$

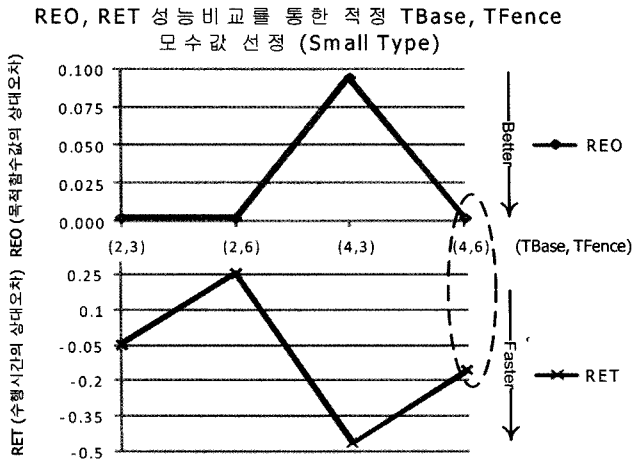
여기서,

T_{GRHS} = GRHS 알고리즘에 의한 수행시간,

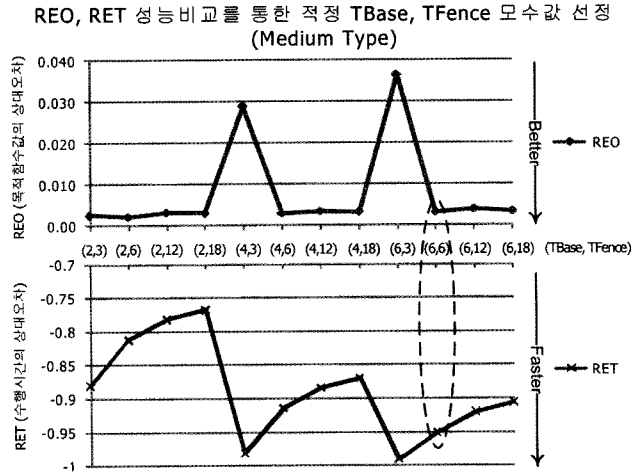
T_{MP} = MLCLSP(MIP 문제)를 XPRESS-MP로 구했을 때 수행시간. 단, 여기서 small 및 medium 문제의 경우는 최적해를 얻는데 실제 소요된 시간, large 문제의 경우는 가능해를 목적으로 수행한 24시간임.

GRHS 알고리즘의 주요 결정모수 들인 (*TBase*, *TFence*) 값의 변화에 따른 실험 결과는 <그림 6~8>에 정리되어 있다. 이 실험 결과를 통해(*TBase*, *TFence*) 값이 MLCLSP 문제의 해를 구해내는데 소요된 시간대비 목적함수 값 (총 비용)의 효율에 미치는 영향을 알 수 있다. 문제 유형 medium과 large에 대해서는 *TBase*와 *TFence* 값이 각각 (6,6) 일 때 시간대비 총비용 효율 면에서 매우 우수하였다. 그 반면(4, 3) 및 (6, 3) 값들에 대해서는 수행속도 면에서는 좋은 성능을 보였으나, 시구간의 폭을 결정하는 *TFence* 값이 작아짐에 따라 탐색구간 충분하지 않아 상대적으로 열등한 해를 산출한 것으로 판단된다. 그 이외에 큰 *TFence* 값에 의해 큰 폭의 시구간을 갖게 되는 (2, 18), (4, 18), (6, 18)의 경우를 보면 총 비용 면에서는 만족할 만한 성능을 보였으나, 시구간 전진 속도를 결정하는 *TFence* 값이 2일 때는 수행속도 면에서 많은 시간을 소모하였고 이 값이 4와 6으로 커짐에 따라 수행속도의 부담이 줄어드는 현상을 관찰할 수 있었다. 같은 방식으로 문제 유형 small에 대해서는 (4, 6)에 해당하는 결정모수의 쌍이 효율이 높은 것을 알 수 있다.

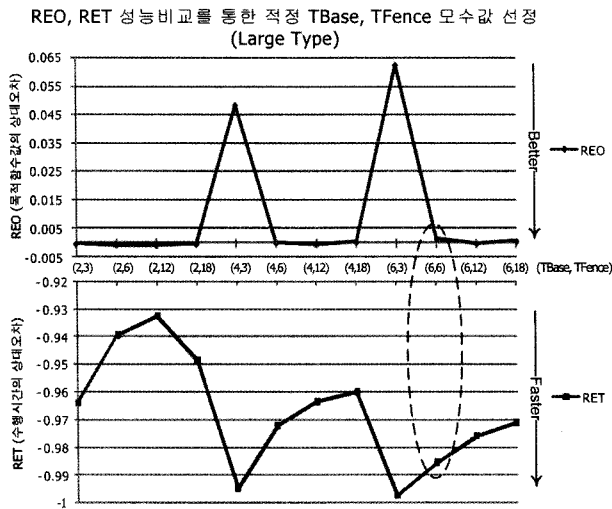
문제 유형별로 *maxitr* 값에 따른 평균상태오차 (average relative error : *ARE*)와 평균수행속도 (average computation times : *ACT*)의 변화를 확인하기 위한 실험을 수행한 결과, 전체 탐색의 반복회수를 결정하는 모수인 *maxitr* 값이 1, 3, 5로 증가함에 있어 그 해(총 비용)의 개선 정도의 차이가 크다는 것을 알 수 있었다. 특히 문제 유형 large의 경우는 총 비용에 있어 *maxitr* 값에 따른 변화가 큰데 이것은 문제의 복잡도가 증가함에 따라 여러 번의 반복 탐색을 통해 최적해에 가까운 해를 찾아내기 때문이라고 파악된다. 문제 유형 medium과 large에 있어 각 problem 들에서 가장 좋은 best 해를 찾아보면, medium의 경우는 20% (*maxitr* = 1), 50% (*maxitr* = 3), 30% (*maxitr* = 5) 이고 large의 경우는 10% (*maxitr* = 1), 30% (*maxitr* = 3), 50% (*maxitr* = 5)의 비율로 발견이 되었다. 이것은 *maxitr* 값이 증가함에 따라 *ARE*에 있어 큰 향상을 보이지는 않더라도



〈그림 6〉 Performance comparison for small problems type according to TBase and TFence



〈그림 7〉 Performance comparison for medium problems type according to TBase and TFence



〈그림 8〉 Performance comparison for large problems type according to TBase and TFence

정밀한 탐색을 가능하게 한다는 것을 의미한다. 여기서 알고리즘 수행속도는 $maxitr$ 값에 따라 선형적으로 증가하므로 $maxitr$ 값을 일정 값(= 3) 이상 증가시키는 것은 본 실험에서 경제적이지 못하다고 할 수 있다. 결론적으로 GRHS 알고리즘은 탐색 반복회수를 결정하는 모수인 $maxitr$ 보다는 시구간의 폭과 이동속도를 조절하는 모수들인 $TBase$ 와 $TFence$ 에 의해 더 큰 영향을 받는다고 할 수 있다.

〈표 3〉은 문제유형 small, medium 및 large에 대해서, GRHS 알고리즘에 의해 산출된 S_{GRHS} 와 XPRESS-MP에 의해 산출된 S_{MIP} 값과의 비교 결과를 전체 결정모수들의 경우(All parameters), 제안하는 모수들을 사용하였을 때의 경우(Proposed parameters), 그리고 각 problems 별로 가장 우수한 해를 제공한 경우(The best case)로 나누어서 정리

하고 있다. 전체 결정모수들의 경우 대비 제안하는 모수들을 사용하였을 때의 결과는 small, medium, large 문제 유형별로 각각 90.12%, 54.43%, 98.89% 향상되는 것을 알 수 있다. 또한 XPRESS-MP에 의해 S_{MIP} 해를 구하는 데 소요되는 시간 대비 GRHS 알고리즘의 수행시간은 medium과 large 문제유형에 있어 각각 95.14%와 99.33%로 단축되는 현격한 수행속도 절감을 관찰할 수 있다. 따라서 본 실험결과는 GRHS 알고리즘이 제안한 결정모수들을 사용하였을 경우, 매우 짧은 시간 내에 효율적인 해를 산출할 수 있다는 것을 보여준다. 마지막으로 ‘The best case’에 나타나 있듯이, S_{MIP} 과 비교하여 small과 medium의 경우 결정모수 값에 따라 최적해에 매우 근접한 해를 얻을 수 있었고 large의 경우 더 좋은 해를 얻을 수 있었다.

본 실험을 통해 일정한 크기의 문제에 대해 LP 솔루션을 1회 푸는데 걸리는 시간이 정해져 있다면, 본 논문에서 제안하는 Rolling Horizon 방식의 $TBase$ 와 $TFence$ 모수들의 값 조정을 통해 GRHS 알고리즘의 수행속도를 제어

〈표 3〉 Summary of computational results

Types	All parameters		Proposed parameters ^a		The best case		ACT ^b for S_{MIP}
	ARE	ACT ^b	ARE	ACT ^b	ARE	ACT ^b	
Small	0.0253 (0.0057) ^c	0.13 (0)	0.0025 (0.0043)	0.13 (0)	0 (0)	0.15 (0.09)	0.03 (0.01)
Mediu-m	0.0079 (0.0121)	5.74 (5.1)	0.0036 (0.0025)	2.5 (0.12)	0.0013 (0.0011)	4.55 (2.17)	51.47 (34.22)
Large	0.009 (0.0214)	23.72 (22.47)	0.0001 (0.0025)	9.66 (0.09)	-0.0022 (0.0029)	27.09 (10.64)	1440 (0)

주) ^a: 제안 모수 값, $maxitr = 3$, small 유형 : ($TBase, TFence$) = (4, 6), medium과 large 유형 : (6, 6).

^b: 시간 단위(분).

^c: 표준편차.

할 수 있다는 것 역시 확인할 수 있었다.

5. 결론 및 추후연구

본 논문에서는 공급사슬망에서 주문이월과 자원제약을 고려하는 단단계 다품목 로트사이즈 결정 문제, MLCLSP를 풀기 위한 휴리스틱 탐색알고리즘을 제안하였다. MLCLSP는 MIP문제에 해당하므로 최적해를 효율적으로 구하는 알고리즘이 존재하지 않는다. 따라서 본 논문에서는 GRASP 탐색알고리즘에 시공간 전진방식을 접목시키는 GRHS 알고리즘을 제시하였다. GRHS 알고리즘은 셋업변수를 상수로 이완하여 생긴 LP 문제를 단체법으로 풀어나감으로써 효율적인 셋업패턴을 찾아내는 방법을 취한다. 다양한 시나리오의 실험을 통해 GRHS 알고리즘의 주요 결정모수들의 효율적인 값을 제시하였고, 비교실험을 통해 최적화 소프트웨어 패키지인 XPRESS-MP(release 2007)보다 매우 짧은 시간 내에 훨씬 좋은 해를 찾아 내는 것을 보였으며 현실적인 문제에의 적용가능성을 입증하였다. 향후에 수요가 불확실하게 주어질 경우 및 정보가 부분적으로 공유되는 공급사슬망 환경으로의 GRHS 알고리즘의 확장 가능성에 대한 추가 연구가 계획되어 있다. 또한 수천단위의 최종 제품수를 가지는 제조업체의 경우와 같이 Large-scale 문제에 대해 분할(decomposition) 기법 등의 방법론을 도입하는 연구와 어느 정도 큰 사이즈의 문제까지 GRHS 휴리스틱이 타당한 결과를 도출해낼 수 있을지 등의 스트레스트테스트와 관련된 실험 및 분석에 대한 추후연구가 필요하다.

참고문헌

- [1] Aiex, R. M., Resende, M. G. C., and Ribeiro, C. C.; "Probability distribution of solution time in GRASP: An experimental investigation," *Journal of Heuristics*, 8 : 343-373, 2002.
- [2] Almeder, C.; "A hybrid optimization approach for multi-level capacitated lot-sizing problems," *European Journal of Operational Research*, 2009, In press.
- [3] Akartunali, K. and Miller, A.; "A heuristic approach for big bucket multi-level production planning problems," *European Journal of Operational Research*, 193 : 394-411, 2009.
- [4] Bahl, H. C. and Ritzman, L. P.; "An Integrated Model for Master Scheduling, Lot Sizing and Capacity Requirements Planning," *Journal of the Operational Research Society*, 35 : 389-399, 1984.
- [5] Bahl, H. C., Ritzman, L. P., and Gupta, J. N. D.; "Determining Lot Sizes and Resource Requirement: a Review," *Operations Research*, 35 : 329-345, 1987.
- [6] Drexl, A. and Haase, K.; "Proportional lot sizing and scheduling," *International Journal of Production Economics*, 40 : 73-87, 1995.
- [7] Eppen, G. D. and Martin, R. K.; "Solving multi-item capacitated lot-sizing problems using variable redefinition," *Operations Research*, 35 : 832-848, 1987.
- [8] Ertogral, K. and Wu, S. D.; "Auction-theoretic coordination of production planning in the supply chain," *IIE Transactions*, 32 : 931-940, 2000.
- [9] Feo, T. A. and Resende, M. G. C.; "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, 6 : 109-133, 1995.
- [10] Fleischmann, B.; "The discrete lot-sizing and scheduling problem," *European Journal of Operational Research*, 44 : 337-348, 1990.
- [11] Florian, M., Lenstra, J. K., and Rinnooy Kan, A. H. G.; "Deterministic Production Planning Algorithms and Complexity," *Management Science*, 26 : 669-679, 1980.
- [12] Hung, Y. F. and Chien, K. L.; "A Multi-Class Multi-Level Capacitated Lot Sizing Model," *Journal of the Operational Research Society*, 51(11) : 1309-1318, 2000.
- [13] Hung, Y. F. and Hu, Y. C.; "Solving Mixed Integer Programming Production Planning Problems with Setups by Shadow Price Information," *Computers and Operations Research*, 25 : 1027-1042, 1998.
- [14] Jayaraman, V. and Ross, A.; "A Simulated Annealing Methodology to Distribution Network Design and Management," *European Journal of Operational Research*, 144 : 629-645, 2003.
- [15] Karmarkar, U. and Schrage, L.; "The deterministic dynamic product cycling problem," *Operations Research*, 33 : 326-345, 1985.
- [16] Kimms, A.; "Competitive Methods for Multi-level Lot Sizing and Scheduling : Tabu search and Randomized Regrets," *International Journal of Production Research*, 34(8) : 2279-2298, 1996.
- [17] Kimms, A.; "A genetic algorithm for multi-level, multi-machine lot sizing and scheduling," *Computers and Operations Research*, 26(8) : 829-848, 1999.
- [18] Kuik, R. and Salomon, M.; "Batching Decisions : Structure and Models," *European Journal of Operational Research*, 75 : 243-260, 1994.
- [19] Kuik, R., Salomon, M. Van Wassenhove, L. N., and Maes, J.; "Linear Programming, Simulated Annealing and Tabu Search Heuristics for Lotsizing in Bottleneck Assembly Systems," *IIE Transactions*, 25(1) : 62-72, 1993.
- [20] Meas, J., McClain, J. O., and Van Wassenhove, L. N.; "Multi-level Capacitated Lotsizing Complexity and LP-based Heuristics," *European Journal of Operational Research*, 53 :

- 131-148, 1991.
- [21] Rojanasoonthon, S. and Bard, J.; "A GRASP for parallel machine scheduling with time windows," *INFORMS Journal on Computing*, 13(1) : 32-51, 2005.
- [22] Ross, A. D.; A Two-Phsed Approach to the Supply Network Reconfiguration Problem. *European Journal of Operational Research*, 122 : 18-30, 2000.
- [23] Roundy, R. O.; "Efficient, Effective Lot Sizing for Multi stage Production Systems," *Operations Research*, 41 : 371-385, 1993.
- [24] Sahling, F., Buschkuhl, L., Tempelmeier, H., and Helber, S.; "Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic," *Computers and Operations Research*, 36 : 2546-2553, 2009.
- [25] Stadler, H.; "Mixed Integer Programming Model Formulations for Dynamic Multi-Item, Multi-Level Capacitated Lotsizing," *European Journal of Operational Research*, 94 : 561-581, 1996.
- [26] Stadler, H.; "Improved rolling scheduling for the dynamic single-level lot-sizing problem," *Management Science*, 46(2) : 318-326, 2000.
- [27] Tempelmeier, H, and Derstroff, M.; "A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times," *Management Science*, 425 : 738-757, 1996.
- [28] Tempelmeier, H. and Helber, S.; "A Heuristic for Dynamic Multi-Item Multi-Level Capacitated Lotsizing for General Product Structures," *European Journal of Operational Research*, 75 : 296-311, 1994.
- [29] Trigeiro, W. W., Thomas, L. J. and McClain, J. O.; "Capacitated lot-sizing with setup times," *Management Science*, 35 : 353-366, 1989.
- [30] Vergana, F. E., Khouja, M., and Michalewicz, Z.; "An Evolutionary Algorithm for Optimizing Material Flow in Supply Chain," *Computers and Industrial Engineering*, 43 : 407-421, 2002.