

최적 설비 할당을 위한 정수계획 모형

강 장 하[†]

조선대학교, 산업공학과

Integer Programming Models for Optimal Resource Allocation

Jangha Kang[†]

[†]Department of Industrial Engineering, Chosun University, Gwangju, Korea

(Received : Jan. 24, 2020, Revised : Feb. 07, 2020, Accepted : Mar. 30, 2020)

Abstract : This article considers an operational capacity planning problem that occurs in a High-tech electronic manufacturing company. In the problem, we do not consider any capacity expansion but determine the allocation configuration between items and parallel facilities. A challenge of the problem is to control inventory holding days. The objective function of the problem is to minimize the total cost of inventory, production and resource allocation costs. We propose two integer programming models for the problem and test them with real world problem instances.

Keyword : Operational capacity planning, Resource Allocation, Due date, Integer Programming, Optimal Algorithm

1. 서 론

용량계획(Rough cut capacity planning:RCCP)은 작업환경변화를 통한 생산용량 변경이 쉽지 않은 대부분의 장치산업에서 원활한 설비 운용을 위해 필수적으로 수행하는 작업이다. 그림 1은 공급망 전반을 효율적으로 관리하기 위한 주요 계획들을 기간별, 기능별로 구분한 도표이다. 이 중 용량계획(capacity planning)은 주일정계획(Master production scheduling)과 함께 제조 공정의 중기계획에 해당하며, 주일정계획이 차질 없이 수행되도록 설비별 생산 품목을 결정하고, 설비의 계획정비 일정 및 휴무 계획 등을 결정한다.

반도체, TFT-LCD와 같은 첨단 전자산업의 제조공정은 다품목을 생산하는 복수의 병렬 설비들을 운용한다. 병렬 설비를 운용하는 생산 환경에서 설비별 생산 품목의 할당에 따라 생산용량 자체가 크게 변하게 된다. 즉, 설비에 할당된 품목이 적을 때, 작업환경변화

를 위한 작업 준비시간은 감소하여 효율적인 생산이 가능하고 결과적으로 설비의 생산용량이 증가하는 효과를 얻을 수 있다. 반대로, 설비에 할당된 품목이 많을 때, 잦은 작업환경변화에 따른 작업 준비시간의 증가로 인해, 설비의 생산용량이 감소하게 되지만, 고객 수요 변동에 대한 대응 수준은 높아질 수 있다. 이렇듯, 첨단 전자산업의 용량계획에 있어서, 설비별 생산 품목의 수준에 따라, 생산량으로 표현되는 생산용량과 납기로 표현되는 고객 수요에 대한 대응 수준이 서로 충돌되는 현상이 나타난다.

제조자 우선의 시장 환경에서는 생산량의 증가가 더욱 중시되어 고객의 요구가 무시되는 경향이 발생하였으나, 최근과 같은 무한 경쟁의 수요자 중심의 시장 환경이 조성된 경우, 고객이 희망하는 납기의 중요성이 더욱 증가한다. 이런 이유로, 주문의 납기에 맞춰, 설비별 생산 품목을 결정하는 운영수준의 용량계획이 요구되고 있다.

반도체, TFT-LCD 등과 같은 첨단 전자산업은 고가의 장비를 사용하는 자동화된 공장을 운영하기 때문에 장치산업이라고도 부른다. 이들 산업은 용량증설을 위해, 신규 공장을 짓거나 장치를 추가하여 용량을 늘리는 작업이 큰 비용을 발생시키고, 긴 lead time을 요구한다. 이런 이유로 오랫동안 여러 관점에서 용량 계획을 연구하여 왔다(Wu 등[2], Geng과 Jiang[3])

[†] Corresponding Author

성명 : 강 장 하

소속 : 조선대학교 산업공학과

주소 : 광주 동구 필문대로 309 조선대학교

전화 : 062-230-7126

E-mail : janghakang@chosun.ac.kr

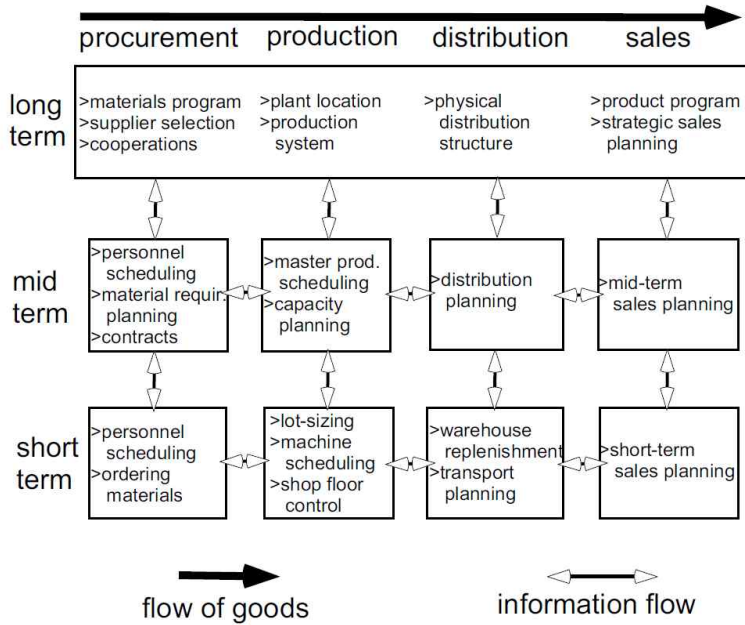


그림 1. The Supply Chain Planning Matrix (Fleischmann 등[1]).

참조). 용량계획은 계획 기간과 범위에 따라, 전략수준의 용량계획(strategic), 전술수준의 용량계획(tactical), 운영수준의 용량계획(operational)으로 나뉜다. 타 산업과 달리 첨단 전자산업에서는 운영수준의 용량계획을 고려할 때, 기존 설비의 운영 효율화를 통한 생산량 최대화에 집중했다.

Bermon과 Hood[4]는 IBM의 wafer fab을 위한 의사결정 문제를 연구하였다. 그 의사결정 문제는 기존 설비를 사용하여 수익을 최대화하는 생산 품목의 수량을 결정하는 것을 목적으로 한다. 그들은 문제를 위해 선형계획모형을 수립하였다. 또한, 그들은 수요를 주문의 관점으로 보지 않고 예상되는 판매 가능한 수요로 판단하여, 납기에 대한 고려 없이 판매 예측치 내에서는 생산하는 만큼 판매가 가능한 것으로 보았다. 그리고, 생산 품목을 병렬 생산 설비 중 어느 곳에 할당할지에 관심을 두었으며, 재고를 고려하지 않았다. Catay 등[5]은 Bermon and Hood의 문제에 재고를 도입하여, 재고 비용, 설비 획득비용, 설비 운영비용을 최소화하는 문제를 구성하고 이를 위해 Lagrangian 완화식에 기반을 둔 발견적 해법을 제안하였다. 이 밖에 많은 연구는 수요의 변동성에 관심을 두어, 확률적 관점의 접근 방법을 사용하였다(Swaminathan [6], Barahona 등[7], Zhang 등[85]).

용량계획에 관한 기존의 연구에서 충분히 다루고 있지 않지만, 현장에서 용량계획 시 주의해야 할 것으로 “과도한 선행생산”의 문제가 있다. 예를 들어, 납기 보다 1주일 이상 선행하여 생산하도록 계획한다면, 설비의 용량이 충분하다고 오인할 수 있기 때문이다. 이에

본 연구에서는 운영수준의 용량계획문제로 주문의 납기에 맞춰, 선행생산과 지연을 방지하여, 설비별 생산 품목을 결정하는 문제를 고려한다. 이를 위해 2개의 선형 정수계획 모형을 제시하고, 전산 실험을 통해 이들의 효율성을 검증한다.

본 논문의 구성은 다음과 같다. 다음 장에서는 문제의 상세 설명이 소개된다. 3장에서는 이 문제들을 해결하기 위한 두 가지 정수계획 모형이 제안된다. 4장에서는 실제 현장 데이터를 이용한 모형의 계산 실험과 이의 결과가 소개된다. 마지막 장에서는 본 논문의 결론과 추후 연구 방향이 제시된다.

2. 문제 설명

이 장에서는 본 연구의 대상이 되는 문제의 상세한 조건들을 설명하고자 한다.

2.1 입력 자료

- 계획기간: 1부터 T 까지를(주로 1개월) 대상으로 하며, 설비의 가동시간, 생산량, 수요 등은 모두 일자별 수량으로 정의한다. 각 일자를 나타낼 때 index s, t, u 를 사용한다.
- 생산 품목의 집합 P : 개별 품목을 나타낼 때 index i 를 사용한다.
- 생산 설비의 집합 R : 개별 설비를 나타낼 때 index r 를 사용한다.
- 설비 r 에서 생산 가능한 품목의 집합 $A_r (\subseteq P)$.

- 품목 i 를 생산할 수 있는 설비의 집합 $R_i (\subseteq R)$.
- 설비 r 에서 품목 i 단위당 표준 생산시간 w_{ir} (> 0): 실제 환경에서는 신제품의 표준 생산시간은 작업개선을 통해 지속적으로 감소한다. 그러나 이러한 개선작업은 미리 예상할 수 없는 부분이므로, 계획기간 동안 변하지 않는다고 가정한다.
- 설비 r 의 일자 t 에서의 가동시간 (또는 생산용량) $d_r^t (\geq 0)$.
- 설비 r 에서 품목 i 의 한 단위 생산비용 p_{ir} : 품목별 선호/비선호 설비를 지정하기 위한 관리적 관점의 penalty를 반영한다.
- 납기가 t 인 품목 i 의 수요 총량 d_i^t : 수요는 납기 이후에 생산할 수 없다. 품목 i 에 대해, 수요가 0보다 큰 일자들의 집합을 나타낼 때, $T_i \equiv \{t=1, \dots, T | d_i^t > 0\}$ 를 사용한다.
- 품목 i 의 선행생산 허용 일자 τ_i : 납기가 t 인 품목 i 는 $t - \tau_i$ 이전에 생산할 수 없다.
- 품목 i 의 단위당 재고보유비용 h_i : 선행생산을 최소화하기 위해 관리적 관점의 재고 보유 penalty를 반영한다.
- 품목 i 를 설비 r 에 할당 시 발생 비용 (할당비용) α_{ir} : 설비에서 가공하기 위해서 품목별로 요구되는 추가작업들이 있다. 예를 들어, 작업공구(jig), 고정구(fixture), 전선(cable)의 설치가 그것이다. 특정 설비에 현재 할당되어 생산 중인 품목의 경우 추가적인 비용이 발생하지 않으나, 새로이 할당되는 품목의 경우, 상당한 관리비용이 발생하게 된다. 따라서, 할당비용은 생산 수량과 관계없이 할당 여부에 따라 계획 전체에서 한 번만 발생한다고 가정한다.

2.2 비용항목

문제에서 고려하는 비용들은 표 1과 같으며, 총비용의 합을 최소화하는 것이 문제의 목적이다.

표 1. 문제에서 고려하는 비용항목들

기호	이름	단위
p_{ir}	생산비용	1/EA
α_{ir}	할당비용	1
h_i	재고보유비용	1/(EA·Days)

2.3 제약 항목

본 연구에서 고려하는 제약들은 다음과 같다.

- 수요 제약: 모든 수요는 차질 없이 전량 만족되어야 한다.
- 납기 제약: 모든 수요는 지연 없이 납기 이전에 생산되어야 한다.
- 선행 방지 제약: 납기가 t 인 품목 i 의 수요는 $t - \tau_i$ 이후에 생산할 수 있다. 그림 2는 선행생산 허용 일자가 3일 때, $t=5$ 시점의 수요를 위한 생산 가능

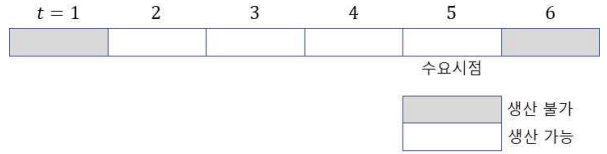


그림 2. 예제: 선행생산 허용 일자가 3일 때, $t=5$ 시점의 수요를 위한 생산 가능 시점

시점들이 2, 3, 4, 5임을 나타낸다.

- 용량 제약: 설비의 가용시간을 초과하여 생산할 수 없다.
- 할당 제약: 설비는 할당된 품목들만 생산할 수 있다.

3. 정수계획 모형

선행 방지 제약을 구현하는 두 가지 방법을 먼저 소개한 후, 각 방법을 활용하는 정수계획모형들 제안하겠다.

3.1 선행생산 방지 구현

선행 방지 제약을 구현하는 두 가지 방법을 제안하고자 한다. 하나는 재고변수를 도입하여 반영하는 방법이고 다른 하나는 납기 일자별로 생산 수량을 결정하는 방법이다.

첫째, 재고변수를 사용하는 방법을 살펴보자. t 번째 일자의 끝나는 시점에 보유하는 품목 i 의 재고를 I_i^t 라 할 때, 재고량 I_i^t 는 이후 τ_i 기간 동안 소진이 되어야 한다. 따라서 재고변수의 상한을 다음과 같이 제약하여, 납기보다 τ_i 일 선행하는 생산을 방지할 수 있다.

$$I_i^t \leq \sum_{s=t+1}^{t+\tau_i} d_i^s$$

그림 3은 선행생산 허용 일자가 3인 품목에 대한 보유 재고 상한의 예를 보여준다. $t=1$ 시점의 보유 재고 상한은 이후 3일 동안의 수요의 합 $40 (= 10 + 0 + 30)$ 으로 계산할 수 있다. 같은 방법으로 $t=2$ 시점의 경우, 이후 3일 동안의 수요의 합 $80 (= 0 + 30 + 50)$ 이다.

	$t=1$	2	3	4	5
생산량	50	0	0	0	50
재고	40	30	30	0	0
수요	10	10	0	30	50
재고 허용상한	40	80	80	50	0

그림 3. 예제: 수요재고 상한 계산 (선행생산 허용 일자가 3이다)

다음으로 생산 일자와 납기 일자가 지정된 생산변수를 통해 선행생산을 방지하는 방법을 살펴보자. 설비 r 에서 납기가 t 인 품목 i 의 수요를 위해 s 시점에 생산하는 수량을 변수 x_{ir}^{st} 라 하면, 다음 제약을 통해 납기보다 τ_i 일 선행하는 생산을 방지할 수 있다. (이하 이 변수를 흐름 변수(flow variable)라 지칭하겠다.)

$$\sum_{r \in R_i} \sum_{s = \max\{t - \tau_i, 1\}}^t x_{ir}^{st} = d_i^t$$

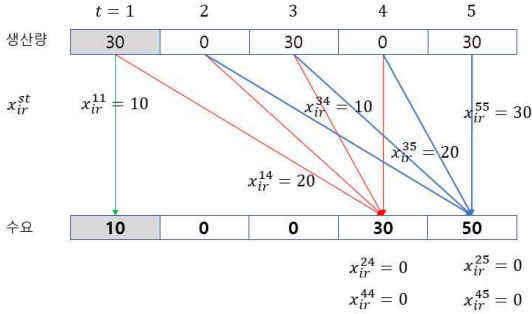


그림 4. 예제: 흐름 변수를 사용한 생산량 결정 (선행생산 허용 일자가 3이다)

그림 4는 선행생산 허용 일자가 3인 품목 i 가 설비 r 에서만 생산 가능하다고 할 때, 흐름 변수를 사용한 생산계획의 예제이다. $t=1$ 시점은, 이보다 선행하는 시점이 계획 구간을 벗어나므로 흐름 변수는 x_{ir}^{11} 하나이다. $t=2,3$ 의 경우는, 수요가 없으므로 흐름 변수가 정의되지 않는다. $t=4$ 시점의 수요 30은, 4개의 흐름 변수 $x_{ir}^{14}, x_{ir}^{24}, x_{ir}^{34}, x_{ir}^{44}$ 를 통해 공급된다: 즉, $x_{ir}^{14} = 20, x_{ir}^{24} = 0, x_{ir}^{34} = 0, x_{ir}^{44} = 10$.

3.2 재고변수를 사용한 정수계획모형: P_{INV}

앞서 설명한 재고변수를 통한 선형 정수계획 모형을 소개한다. 먼저, 사용되는 변수들은 다음과 같다.

- 재고변수: $I_i^t \geq 0$ for $t=1, \dots, T, r \in R$; t 번째 일자의 끝나는 시점에 보유하는 품목 i 의 재고량으로, 총 $|P| \times T$ 개가 정의된다.
- 생산변수: $x_{ir}^t \geq 0$ for $t=1, \dots, T, r \in R, i \in A_r$; t 번째 일자에 설비 r 에서 품목 i 의 생산량을 나타내고, $T \times \sum_{r \in R} |A_r|$ 개가 정의된다.
- 할당변수: $y_{ir} \in \{0, 1\}$ for $r \in R, i \in A_r$; 전체 계획 기간($t=1, \dots, T$) 중 설비 r 에서 품목 i 를 생산하지 않을 때 0; 생산할 경우 1의 값을 가지며, 총 $\sum_{r \in R} |A_r|$ 개가 정의된다.

재고변수를 사용한 정수계획모형은 다음과 같다. 이하 P_{INV} 로 지칭한다.

(P_{INV})

$$\begin{aligned} & \text{Minimize } \sum_{i \in P} \sum_{t=1}^T h_i I_i^t + \\ & \sum_{r \in R} \sum_{i \in A_r} \sum_{t=1}^T p_{ir} x_{ir}^t + \sum_{r \in R} \sum_{i \in A_r} \alpha_{ir} y_{ir} \end{aligned}$$

subject to

$$\sum_{i \in A_r} w_{ir} x_{ir}^t \leq c_r^t, \quad \text{for } t=1, \dots, T, r \in R \quad (1)$$

$$I_i^{t-1} + \sum_{r \in R_i} x_{ir}^t = d_i^t + I_i^t, \quad \text{for } t=1, \dots, T, i \in P \quad (2)$$

$$w_{ir} x_{ir}^t \leq c_r^t y_{ir}, \quad \text{for } t=1, \dots, T, r \in R, i \in A_r \quad (3)$$

$$\sum_{t=1}^T x_{ir}^t \leq y_{ir} \times \sum_{t=1}^T d_i^t, \quad \text{for } r \in R, i \in A_r \quad (4)$$

$$I_i^t \leq \sum_{u=t+1}^{t+\tau_i} d_i^u, \quad \text{for } i \in P, t=1, \dots, T - \tau_i \quad (5)$$

and $I_i^t \geq 0$ for $t=1, \dots, T, r \in R,$

$$x_{ir}^t \geq 0 \text{ for } t=1, \dots, T, r \in R, i \in A_r,$$

$$y_{ir} \in \{0, 1\} \text{ for } r \in R, i \in A_r,$$

$$I_i^0 = 0 \text{ for } i \in P.$$

목적식은 재고보유비용, 생산비용 및 할당비용 합 의 최소화이다. 식 (1)은 $|R| \times T$ 개가 정의되며, 각 설비의 일자별 가동시간이 가용시간을 초과할 수 없도록 한다. 식 (2)는 $|P| \times T$ 개가 정의되며, 전일 재고와 당일 생산량의 합이 당일의 수요와 당일 재고의 합과 같다는 균형방정식이다. 여기에서 생산량은 설비별로 발생하기 때문에 생산이 가능한, 즉 R_i 에 속한 설비 전체에서의 생산량을 합산한다. 식 (3)은 $T \times \sum_{r \in R} |A_r|$ 개가 정의된다. 할당이 발생한 상태($y_{ir} = 1$)에서 설비 r 에서 품목 i 의 생산을 설비의 가용시간 내에서 허용하며, 할당이 발생하지 않는 경우($y_{ir} = 0$)는 생산하지 못하도록 제약한다($w_{ir} x_{ir}^t \leq 0$). 식 (4)는 $\sum_{r \in R} |A_r|$ 개가 정의된다. 필수적이지는 않지만, 개선된 선형완화식(linear relaxation)을 얻기 위해 추가된다. 선형완화식에 대한 추가적인 정보는 Nemhauser와 Wolsey[9]를 참고하라. 식 (5)는 $|P| \times T$ 개가 정의되고, 재고변수의 상한 제약으로 앞서 소개한 대로 선행 생산을 제약한다.

P_{INV} 에 정의된 변수는 총 $(T+1) \times \sum_{r \in R} |A_r| + |P| \times T$ 개 이고, 이 중 $\sum_{r \in R} |A_r|$ 개의 이진 정수변수를 포함한다. 변수 중 가장 큰 부분을 차지하는 것은 생

산변수 x_{ir}^t 이며, 계획 구간의 크기 T 와 설비별 생산 가능한 품목의 총 수량 $\sum_{r \in R} |A_r|$ 이 변수의 개수에 큰 영향을 준다. P_{INV} 에 정의된 제약식은 총 $\sum_{r \in R} |A_r| + T \times (|R| + 2|P| + \sum_{r \in R} |A_r|)$ 개로, 점근적으로 $T \times \sum_{r \in R} |A_r|$ 에 수렴한다. 따라서, 제약식의 개수는 변수와 마찬가지로 계획 구간의 크기 T 와 설비별 생산 가능한 품목의 총 수량 $\sum_{r \in R} |A_r|$ 에 큰 영향을 받는다.

3.3 흐름변수를 사용한 정수계획모형: P_{FLOW}

이 모형에서는 흐름변수가 앞 모형의 재고변수와 생산변수를 대체한다. 사용하는 변수는 다음과 같다.

- 흐름변수: $x_{ir}^{st} \geq 0$ for $i \in P, r \in R_i, t \in T_i, s = \max\{1, t - \tau_i, \dots, t\}$: 설비 r 에서 납기가 t 인 품목 i 의 수요를 위해 s 시점에 생산하는 수량을 나타낸다. 변수의 개수는 점근적으로 $\sum_{i \in P} \tau_i |R_i| |T_i|$ 에 수렴한다.
- 할당변수: $y_{ir} \in \{0, 1\}$ for $r \in R, i \in A_r$: P_{INV} 의 할당변수와 같다.

흐름변수를 사용한 정수계획모형은 다음과 같다. 이 하 P_{FLOW} 로 지칭한다.

(P_{FLOW})

$$\begin{aligned} & \text{Minimize } \sum_{r \in R} \sum_{i \in A_r} h_i \times \left(\sum_{t \in T_i, s = \max\{t - \tau_i, 1\}}^{t-1} (t-s)x_{ir}^{st} \right) \\ & + \sum_{r \in R} \sum_{i \in A_r} p_{ir} \times \left(\sum_{t \in T_i, s = \max\{t - \tau_i, 1\}}^t x_{ir}^{st} \right) + \sum_{r \in R} \sum_{i \in A_r} \alpha_{ir} y_{ir} \end{aligned}$$

subject to

$$\sum_{i \in A_r} w_{ir} \sum_{u=t}^{\min\{t+\tau_i, T\}} x_{ir}^{tu} \leq c_r^t, \text{ for } t = 1, \dots, T, r \in R \quad (6)$$

$$\sum_{r \in R_i, s = \max\{t - \tau_i, 1\}}^t x_{ir}^{st} = d_i^t, \text{ for } i \in P, t \in T_i \quad (7)$$

$$\sum_{s = \max\{t - \tau_i, 1\}}^t x_{ir}^{st} \leq d_{ir}^t y_{ir}, \text{ for } i \in P, r \in R_i, t \in T_i \quad (8)$$

and $y_{ir} \in \{0, 1\}$ for $r \in R, i \in A_r$,

$$x_{ir}^{st} \geq 0 \text{ for } r \in R, i \in A_r, t \in T_i,$$

$$s = \max\{t - \tau_i, 1\}, \dots, t.$$

재고변수 없이, 흐름변수를 통해 재고보유비용을 계산하기 위해서는 각 흐름변수에 따라 발생하는 재고보유 기간을 구해야 한다. 납기가 t 인 수요를 s 시점에 생산할 경우 $(t-s)$ 일 동안 재고로 보유되기 때문에

재고보유 비용은 $h \times (t-s) \times x_{ir}^{st}$ 이 된다. P_{FLOW} 의 목적식에서는 이를 이용하여 재고 비용을 반영한다.

식 (6)은 $|R| \times T$ 개가 정의되며, 각 설비의 일차별 작동시간이 가용시간을 초과할 수 없도록 한다. 식 (7)은 $\sum_{i \in P} |T_i|$ 개가 정의되며, 흐름변수를 통해 수요가 만족되도록 제약한다. 식 (8)은 $\sum_{i \in P} |R_i| \times |T_i|$ 개가 정의되며, 할당이 발생한 상태($y_{ir} = 1$)에서 설비 r 에서 품목 i 의 생산을 허용한다. 할당이 발생하지 않는 경우($y_{ir} = 0$)는 생산을 불허한다.

P_{FLOW} 에 $\sum_{i \in P} \tau_i |R_i| |T_i|$ 의 실수변수와 $\sum_{r \in R} |A_r|$ 개의 이진 정수변수가 사용된다. 제약식의 개수는 $|R| \times T + \sum_{i \in P} |T_i| + \sum_{i \in P} |R_i| \times |T_i|$ 로, 점근적으로 $\sum_{i \in P} |R_i| |T_i|$ 에 수렴한다.

3.4 모델 비교

$\sum_{i \in P} |R_i| = \sum_{r \in R} |A_r|$ 이므로, 두 모형의 변수, 제약식의 수를 표 2와 같이 정리할 수 있다. 이를 분석하면, 수요가 등성등성 발생하는 경우 P_{FLOW} 가 P_{INV} 보다 현저히 적은 개수의 제약식을 가질 것이고, 수요가 발생하는 시간 간격이 평균적으로 τ_i 보다 긴 경우, P_{FLOW} 가 더 적은 변수를 갖는 것을 예상할 수 있다. 그런데, 이는 상당히 일반적인 경우로, 대부분의 첨단 전자업체에서 동일 품목이 주 2회 이상 주문되는 경우는 흔하지 않다. 따라서 현실적으로 문제의 크기 면에서, P_{FLOW} 가 보다 타당한 모형이라 볼 수 있다.

표 2. 두 모델의 크기 비교

모델	실수변수	정수변수	제약식
P_{INV}	$O\left(T \sum_{i \in P} R_i \right)$	$\sum_{i \in P} R_i $	$O\left(T \sum_{i \in P} R_i \right)$
P_{FLOW}	$O\left(\tau \sum_{i \in P} R_i T_i \right)$	$\sum_{i \in P} R_i $	$O\left(\sum_{i \in P} R_i T_i \right)$

4. 전산실험

두 모델의 실질적인 성능을 확인하기 위해서, 전산 실험을 실행하였다. 이 장에서는 test instance들의 출처와 특성들을 설명하고, 실험결과를 소개한다.

4.1 실험데이터

제조공장이 전 세계적으로 산재한 국내 굴지의 첨단 전자회사의 4개월간의 제조 실적을 확보하여 공장 단위로 데이터를 정리하였다. 용량계획의 단위가 1개월

이므로, 공장별, 월별로 구분하여 최종 조립공정을 대상으로 총 47개의 실험 대상을 선정하였다. 또한, 선행생산 허용 기간 τ_i 를 모든 품목에 3 또는 7로 부여하여 최종적으로 94개의 test instance를 확보하였다. 선행생산 허용 기간이 짧은 경우 주어진 수요를 맞추지 못하는 경우가 발생하므로, 문제 제약을 고려하여, 수요를 수정하는 전처리 과정을 수행하였다. 표 3은 이러한 instance들의 개략적인 크기를 소개하고 있다. nT 열은 각 문제의 T 를 의미하고, $nResources$, $nAssign$, $nDemand$ 는 각각 $|R_i|$, $\sum_{i \in P} |R_i|$, $\sum_{i \in P} |T_i|$ 를 의미한다. T 는 30일 또는 31일 이고, $|R_i|$ 는 평균 10, $\sum_{i \in P} |R_i|$ 는 평균 250, $\sum_{i \in P} |T_i|$ 는 평균 450 정도이다. 그러나, T 를 제외한 다른 지표들의 표준편차가 상당히 높다. 이는 공장별로 상황이 서로 달라, 문제의 크기가 균일하지 못하다는 것을 의미한다.

표 3. 실험데이터의 통계값

	nT	nResources	nAssign	nDemand
count	94	94	94	94
mean	30.7	10.4	249.6	405.1
std	0.5	11.0	336.5	529.4
min	30	1	1	1
25%	30	2	34	61.5
50%	31	9.5	70.5	189
75%	31	13	382	437.75
max	31	51	1272	2291

비용 상수값은 실험을 위해 다음과 같이 임의로 지정하여 사용하였다. 생산비용(p_{ir})은 $\frac{w_{ir}}{100}$, 할당비용(α_{ir})은 내부적으로 관리하는 할당 선호도(0,1,...)에 $100 \times T$ 를 곱한 값, 재고보유비용(h_i)은 0.1을 사용하였다.

4.2 실험환경

제시된 선형정수계획 모형의 최적해를 얻기 위하여 상용 LP/MIP Solver를 사용하였다. 사용한 solver는 64bit Gurobi Optimizer version 9.0.0로, Python interface를 사용하여 구현하고 구동하였다. 실험 컴퓨터는 메인메모리 128Gbit가 설치된 Windows 2012 Server로, 설치된 CPU는 Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz이고, 총 20개의 코어를 모두 사용하는 병렬 최적화 mode를 사용하였다. 각 실험은 5분(300초)을 초과하지 못하도록 시간 한계를 부여하였다. 이는 해당 업체의 현장 작업자가 받아들일 수 있는 시간의 한계이다.

4.3 실험 문제 분류

실험결과를 효율적으로 설명하기 위해, 전체 실험 문제 94개를 아래와 같이 3종류로 분류하였다.

- S1: P_{INV} 모형으로 1초 이내에 최적해가 얻어진 문제들. 총 57개.
- S2: P_{INV} 모형으로 1초 이상 300초 이하의 시간동안 최적해가 얻어진 문제들. 총 13개.
- S3: P_{INV} 모형으로 300초 이내에 최적해를 얻지 못한 문제들. 총 24개.

표 4, 5, 6에 위 세 종류의 test instance의 통계값이 열거되어 있다. S1의 문제들은 설비가 16개 이하, 설비당 생산 가능한 품목의 수가 약 10개 이하인, 생산 품목의 수와 복잡도가 낮은 공장의 문제들로 판단된다. S2의 문제들은 설비가 8~47개, 설비당 생산 가능한 품목의 수가 약 20개 정도이다. S3은 설비는 10~50개 정도로 S2와 큰 차이를 보이지 않지만, 설비당 생산 가능한 품목의 수가 약 30개 정도로 좀 더 복잡한 공장의 문제들로 판단된다.

표 4. S1 문제의 통계값

	nT	nResources	nAssign	nDemand
count	57	57	57	57
mean	30.6	5.3	46.7	121.0
std	0.5	5.1	39.3	122.8
min	30	1	1	1
25%	30	2	15	41
50%	31	2	41	79
75%	31	10	63	179
max	31	16	156	570

표 5. S2 문제의 통계값

	nT	nResources	nAssign	nDemand
count	13	13	13	13
mean	30.8	13	283.5	738
std	0.4	10.6	163.7	711.6
min	30	8	114	259
25%	31	8	123	348
50%	31	10	240	408
75%	31	10	382	427
max	31	47	679	2048

표 6. S3 문제의 통계값

	nT	nResources	nAssign	nDemand
count	24	24	24	24
mean	30.8	21	713.1	899.4
std	0.4	13.5	336.5	561.1
min	30	11	287	333
25%	30.75	12	530.25	506.25
50%	31	15	600.5	773.5
75%	31	21.5	935.5	983.5
max	31	51	1272	2291

4.4 결과 분석

그림 5, 6은 최적해를 얻은 S1, S2에 속한 문제들의 수행시간을 나타낸 도표이다. Runtime_flow는 P_{FLOW} 모형의 수행시간이고 Runtime_inv는 P_{INV} 모형의 수행시간이다. 그림 5는 S1 문제들에 대해 P_{INV} 모형의 수행시간이 더 짧은 것을 보여준다. 반대로 그림 6은 S2 문제들에서 P_{FLOW} 의 수행시간이 더 짧은 것을 보여준다. 표 7, 8은 S1, S2 문제의 생성된 변수, 제약식의 크기와 수행시간을 보여준다. 표 7은 S1문제에서 변수, 제약식 모두 P_{FLOW} 가 P_{INV} 보다 작지만, 수행시간은 P_{INV} 가 더 짧은 것을 보여준다. 사실 1초 이내에 해결될 정도의 문제의 경우, 무시할 만한 시간이므로, 큰 의미를 주는 차이는 아니다. 반면, 표 8은 S2 문제들에 대해서 P_{FLOW} 가 P_{INV} 보다 평균 3배 빠르다는 것을 보여준다.

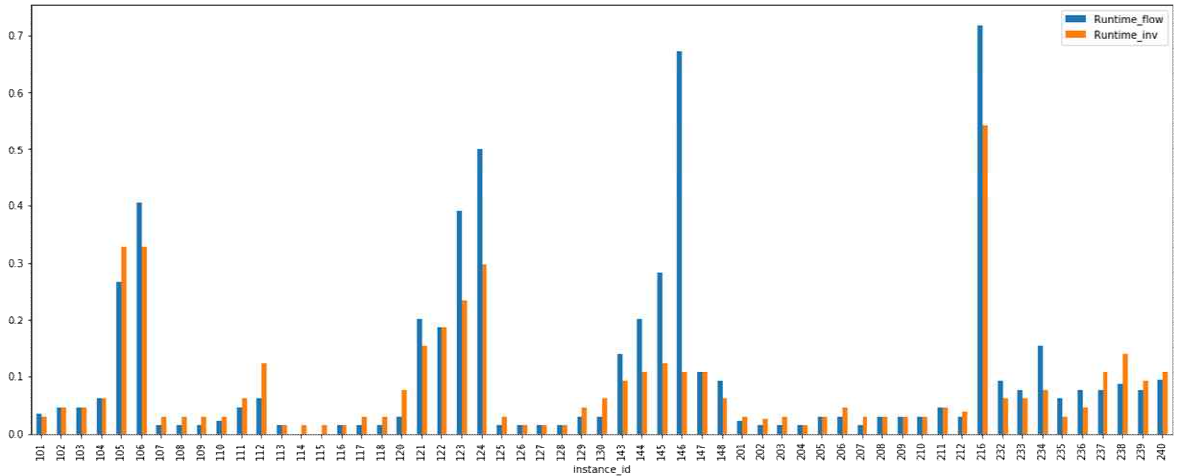


그림 5. S1 문제들의 수행시간 결과

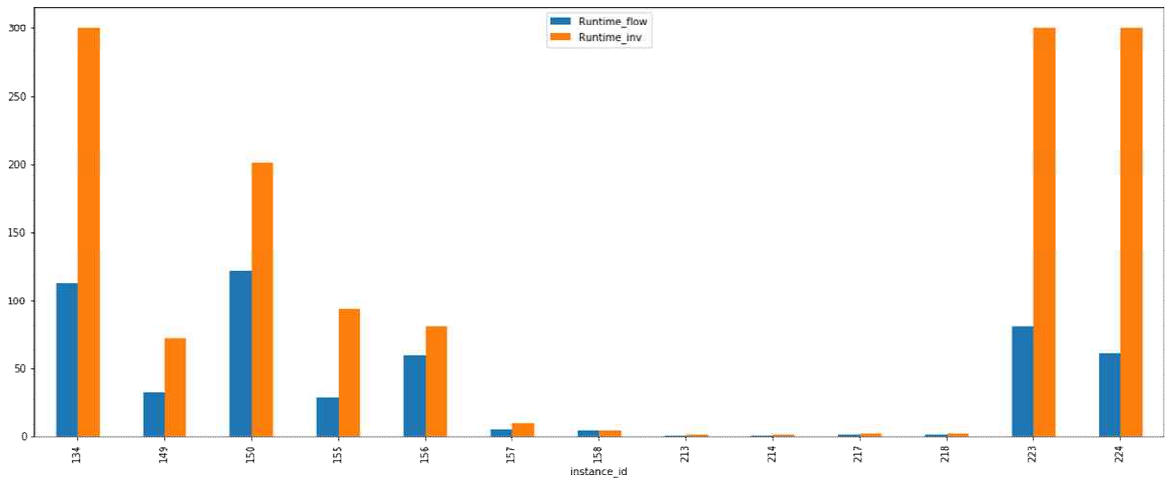


그림 6. S2 문제들의 수행시간 결과

S3 문제들은 300초(5분) 이내에 최적해를 찾지 못한 경우가므로, 수행시간보다 최종해의 품질을 평가할 필요가 있다. 그림 7은 최종해의 optimality gap을 비교하여 보여준다. 여기서 optimality gap이란 LP solver가 종료된 시점에 보유하는 정수해(objVal)와 선형완화식의 한계값(lpBound)의 상대 차이를 의미하며, 다음과 같이 계산한다.

$$\text{Optimality gap}(\%) = \frac{\text{objVal} - \text{lpBound}}{\text{objVal}} \times 100$$

Gap이 0%에 가까울수록 최적해에 가까운 정수해를 얻었다고 판단할 수 있다. 그림 7은 P_{FLOW} 가 P_{INV} 보다 작은 optimality gap을 갖는다는 것을 보여준다. 그림 8은 S3 문제들의 node_count를 보여준다. 여기서 node_count란 300초 동안 계산한 분지한계마디(nodes of the Branch-and-bound tree)의 수를 나타낸다. 최적해를 얻지 못한 경우, node_count가 작다는 것은 마디 각각을 해결하는데 걸리는 시간이 길어서 충분히 많은 마디를 계산하지 못했다고 해석할 수 있다. 그림 8은 대부분의 instance에서 P_{FLOW} 모형이 더 많은 마디를 계산했다는 것을 보여준다.

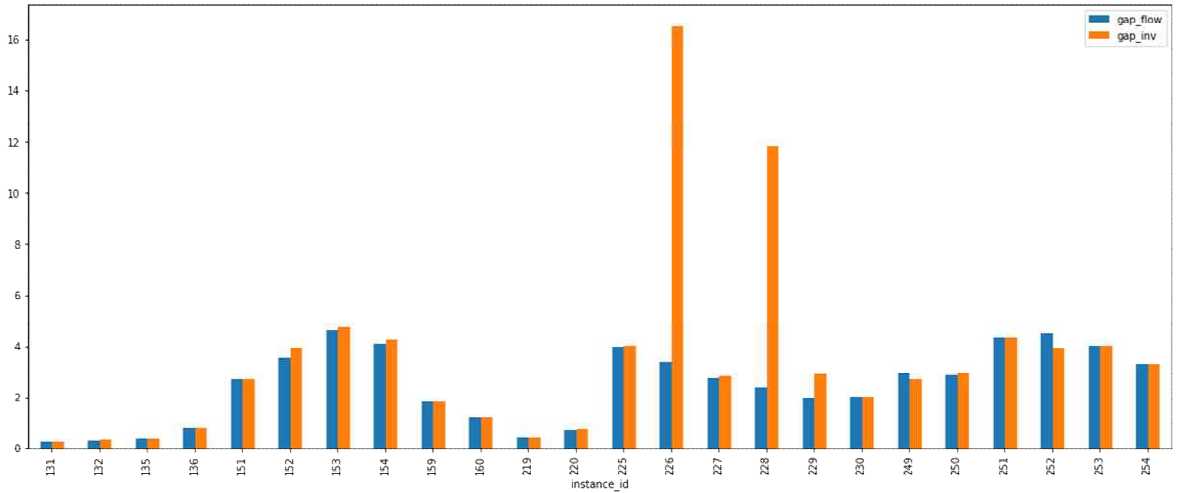


그림 7. S3 문제들의 Optimality Gap(%) 비교

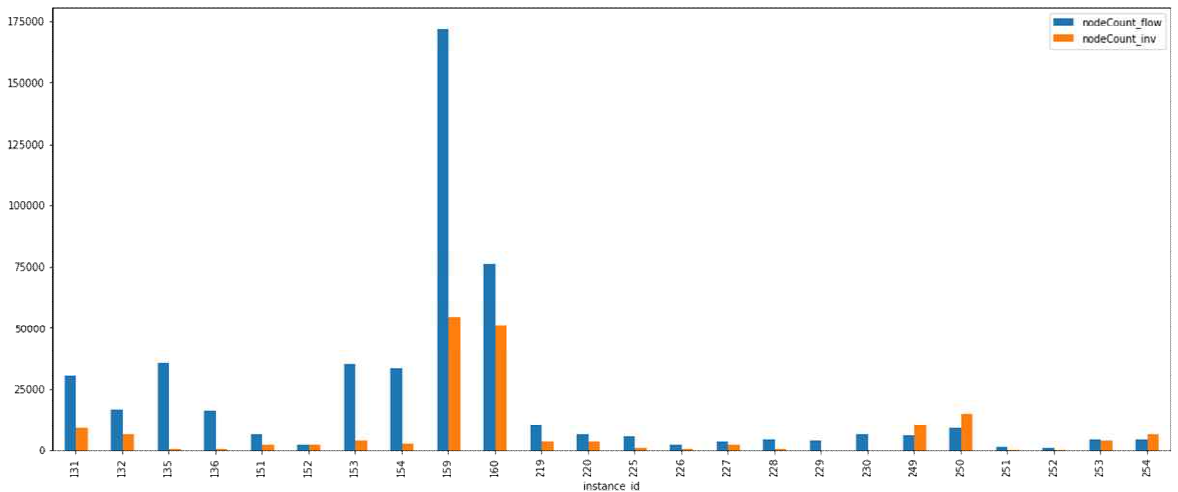


그림 8. S3 문제들의 node count 비교

표 9는 그림 7, 8이 보여주는 정보를 통계치로 정리하여 보여준다. 평균적으로 P_{FLOW} 가 P_{INV} 대비 71% 수준의 optimality gap을 보인다(즉, 더 좋은 해를 제시한다). 이는 변수의 수, 제약식의 수가 90%, 19% 정도로 모형이 작아, 약 3배 많은 마디를 계산하기 때문에 이해할 수 있다.

결과적으로 두 모델 모두 현장에서 사용할 수 있는 성능을 보여준다. 60% 실험 문제에 대해서 1초 이내에 최적해를 구했으며, 75% 문제에 대해 5분 이내에 최적해를 구했다. 5분 동안 최적해를 구하지 못한 나머지 보다 복잡한 문제에 대해서도 흐름변수 모형과 재고변수 모형 모두 평균적으로 각각 2.5%, 3.5%의 optimality gap을 보여준다. 이는 현장에서 만족할 수준의 결과이다. 안정적인 계획 품질을 선호하는 현장의 입장에서는 어려운 문제에 대해서 더 빠르고, 더 낮은 optimality gap을 제공하는 흐름변수 모형(P_{FLOW})이 더 우월하다 할 수 있겠다.

4. 결론

본 연구에서는 용량계획 단계에서 주문의 납기를 고려하여, 설비별 생산 품목을 결정하는 할당문제를 다루었다. 이 문제에서 선행생산과 지연을 방지하여 납기의 관리한다. 선행생산을 제약하기 위해 재고변수를 도입하는 방법과 흐름변수를 도입하는 방법을 제시하였고, 이들 방법을 각각 사용하는, 두 개의 선행 정수 계획 모형을 소개하였다.

현장의 실제 문제를 사용한 전산 실험을 통해 모형의 효율성을 검증하였다. 실험결과 대부분 (75%)의 평균적인 크기의 문제들에 대해 5분 이내에 최적해를 구할 수 있었다. 최적해를 구하지 못한 나머지, 상대적으로 복잡한 문제들의 경우에도 optimality gap이 평균적으로 2.5%, 3.5% 수준인 해를 얻을 수 있었다. 두 모델 모두 현장에서 사용 가능하며, 이중 흐름변수 모형이 용량계획수립 시스템에 더 적절하다고 판단된다.

본 연구에서는 설비별 생산 품목의 수를 최소화하도록 할당비용을 고려하였다. 그러나, 동일 설비에 할당되는 품목들 사이에서 발생할 수 있는 효율의 상승 또는 충돌을 고려하지는 못하였다. 향후 동일 설비에 할당되는 품목 간의 관계를 고려하는 연구를 기대한다.

표 7. S1 문제의 크기 및 수행시간

	변수		제약식		수행시간	
	flow	inv	flow	inv	flow	inv
count	57	57	57	57	57	57
mean	1,841	2,237	596	2,873	0.103	0.082
std	2,311	2,112	608	3,114	0.155	0.096
min	4	63	37	125	0.000	0.016
25%	405	635	183	692	0.016	0.031
50%	965	1688	265	2032	0.037	0.047
75%	2295	2422	959	3039	0.094	0.109
max	9257	8316	2094	12096	0.718	0.542

표 8. S2 문제의 크기 및 수행시간

	변수		제약식		수행시간	
	flow	inv	flow	inv	flow	inv
count	13	13	13	13	13	13
mean	15388	11860	3688	14509	39.3	105.4
std	18792	6276	3789	7736	44.1	125.0
min	2012	5694	1007	8050	0.7	1.0
25%	4223	5920	1505	8088	1.4	2.0
50%	7499	10160	1973	12403	28.6	72.6
75%	13399	15014	2480	15778	60.6	201.7
max	58994	27769	11679	35899	122.3	300.3

표 9. S3 문제의 크기 및 수행결과

	변수		제약식		Gap		nodeCount	
	flow	inv	flow	inv	flow	inv	flow	inv
count	24	24	24	24	24	24	24	24
mean	24898	27609	5895	31233	2.5	3.5	20580	7605
std	15641	12966	3172	14420	1.5	3.7	36570	14434
min	9408	11540	3066	14392	0.3	0.3	968	1
25%	15055	19402	3844	20366	1.1	1.1	4279	772
50%	19753	22231	4758	25822	2.7	2.9	6405	2638
75%	30394	37680	6349	43466	3.7	4.0	19877	6345
max	76598	49027	14283	58668	4.6	16.5	171847	54507

참고문헌

1. B. Fleischmann, H. Meyr, and M. Wagner, Supply Chain management and Advanced Planning: Concepts, Models, Software and Case Studies, ed. H. Stadtler and C. Kilger, 71-96. Springer-Verlag, New York, USA (2002).
2. Wu, S.D., Erkoc, M., and Karabuk, S., The Engineering Economist, 50, 125-158 (2005).
3. Geng, N. and Jiang, Z., International Journal of Production Research, 47, 3639-3655 (2009).
4. Bermon, S. and S.J. Hood, Interfaces, 29(5), 31-51 (1999).
5. Catay, B., S.S. Erenguc, and A.J. Vakharia, Computers and Operations Research, 30(9), 1349-1366 (2003).
6. Swaminathan, J.M., European Journal of Operational Research, 120(3), 545-558 (2000).
7. Barahona, F.S., Bermon, O., Gunluk, and S. Hood, Tech. Report RC22196, IBM, T.J. Watson Research Center, New York, USA, 2003.
8. Zhang, F., R. Roundy, M. Cakanyildirim, and W.T. Huh, IIE Transactions, 36(1), 23-37 (2004).
9. Nemhauser, G.L. and L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, Hoboken, New Jersey, USA (1988).