수송및 준비시간을 고려한 Zero Wait 방안의 다품종 회분식 공정의 생산계획

문성득(정), 박선원(정), 이원국(정)
한국과학기술원 화학공학과

Scheduling of Multiproduct Batch Processes Under Zero Wait Policy with Transfer and Setup Times

Sungdeuk Moon, Sunwon Park, and Wonkook Lee.
Department of Chemical Engineering,
Korea Advanced Institute of Science and Technology.

1. Introduction

회분식 공정은 공정 및 운전상의 유연성을 모두 갖추고 있어서 다양한 원료와 수요를 충족시킬 수 있으며, 시장 동향에도 신속히 대응할 수 있으므로 다목적 공정의 특성을 가진 특수한 다품종 회분식에 관한 연구가 활발히 이루어지고 있다.

회분식 공정의 최적화는 생산할 제품과 이들이 거쳐야할 장치들과의 적절한 조합에 의하여 제품의 생산순서를 계획함으로써 정해진 제품들을 최단시간내에 생산하는 것이 그 목적이라 하겠다. 이러한 문제들의 일반적인 형태는 N 개의 제품, M 개의 장치 등과, 각각의 장치에 대하여 각각의 제품들이 거치면서 걸리는 시간들의 집합으로 구성되어 있다.

다목적 공정의 스케줄링 방법에는 중간저장 탱크를 사용하는 방안에 따라 여러가지로 구분될수 있는데, 그중에서 고분자나 정밀화학 분야 등에서와 같은 제품들은 반응이 계속적으로 진행되므로 장치간에 대기 시간이 있으면 제품의 변질이 우려되는 경우가 있으므로 이때에는 대기 시간 없이 바로 다음 장치로 이동시키는 방안의 ZW(Zero Wait) 스케줄링을 고려해야 한다. 이러한 조업의 생산계획의 예를 Fig. 1 의 Gannt chart 에 나타내었다. 기존의 수송및 준비시간이 포함된 ZW 방안에 대한 조업완료시간을 결정하는 수학적 최적화 기법에서는 어떤 장치에서 제품의 순서에 따라 영향을 받는 준비시간(setup time)을 고려함으로 MINLP 형태의 최적 생산계획의 해법들이 개발되어져 있다[1].

본 연구에서는 ZW 조업하에서 makespan을 최소화 하는 수학적 최적화 기법에서 수송및 준비 시간등이 포함되어 있는 회분식 공정의 최적생산계획을 푸는 방법을 MINLP의 형태가 아닌 MILP의 formulation 형태로 수식화시켜 최적해를 구하는 새로운 방법을 제시하고자 한다.

2. ZW Policy with Transfer and Setup Times

지금까지 연구된 수학적 최적화기법의 최적생산계획은 거의 모두가 수송및 준비시간을 고려치 않은 MILP formulation에 의한 최적생산계획이었다 [2]. 그러나 실제 공정의 조업은 수송시간이나 준비시간이 매우 중요한 요소이며 특정의 공정은 제품의 장치에서의 공정시간보다 더 큰 의미를 수송시간 혹은 준비시간이 갖는 경우도 있다. 그래서 조업 완료시간을 계산하는 알고리듬의 개발영역에서 이들 수송시간과 준비시간이 고려되는 연구가 활발히 진행되어져 왔다[1]. 수송시간, T_{ij} 는 제품 i 와 장치 j 간의 고정 변수이지만 준비시간은 장치 j 에서 조업되는 제품 i 및 i+1 의 순서에 의존하는 변수 이므로 수식화에 매우 까다로운 조건을 갖고 있다. 이를 sequence dependent set-up time 이라고 하며 그 변수를 $S_{i(i+1)j}$ 이라고 표시 할때, 그 의미는 제품 i 에 이어 제품 i+1 이 장치 j 에서 조업하기 위하여, 장치 j 가 필요로 하는 준비시간이다.

수송시간과 준비시간이 존재하는 ZW방안에서 최적생산계획의 계산법을 고려하기 위하여, C_{ij} 를 i 번째 제품이 j 번째 장치에서 조업이 완료되는 총시간이라 하고, P_{ij} 는 i 제품이 j 장치에서의 조업 시간(processing time)이라 하고, 이진변수 X_{ij} 를 다음과 같이 정의 한다.

$$X_{ij} = \begin{cases} 1 \text{ if product } i \text{ is in position } j \text{ in the sequence} \\ 0 \text{ otherwise} \end{cases}$$

결국 목적함수는 C_{NM} 을 최소화하는 것이며, 이에 대한 최적해를 얻기 위하여, 몇가지 제한 조건들을 설정하여야 한다. 우선, 생산계획순서의 하나의 위치는 반드시 하나의 제품이 할당되어야 한다는 제한 조건과 제품은 주어진 생산 조업순서에 하나씩 할당된다는 선형의

제한 조건등이 존재한다. 그리고, 제품 i 가 장치 j 까지 조업을 거치면서 조업이 완료되는 총시간 들에 대한 제한조건들은 다음과 같이 나타낼 수 있다.

$$C_{ij} \geq C_{(i-1)j} + \sum_{m=1}^{N}\sum_{k=j}^{M} P_{mk} X_{mi} + \sum_{m=1}^{N}\sum_{k=j-1}^{M} T_{mk} X_{mi} + \sum_{m=1}^{N}\sum_{k=1}^{M} S_{mkj} X_{m(i-1)} X_{ki} - \sum_{m=1}^{N}\sum_{k=j+1}^{M} (P_{mk} + T_{mk}) X_{m(i-1)} \tag{1}$$

식(1)에서 보면, 준비시간, S_{mkj} 이 포함되어 있는 항은 이진변수 X 의 2차항을 포함하므로 X 에 대하여 MINLP가 된다.

3. Idle Time Search(ITS) Method

ITS법의 기본적인 개념은 Fig. 1 에서와 같이 제품 i 가 장치 j 에서의 조업 시간, P_{ij} 는 일정하므로 makespan의 최소화에는 아무런 영향을 주지 못한다. 그러므로, makespan 의 최소화에는 모든 제품이 마지막 장치를 지나면서 이루는 지연시간등의 조합에 전적으로 의존한다고 생각할 수 있다. ZW방안에서 makespan에 영향을 주는 시간들로는, 첫번째 장치에서 조업되는 첫 제품이 마지막 장치를 제외한 모든 장치를 지나면서 소모되는 시간 (head time, h_i)과 제품 i 다음에 제품 k 가 처리될때 마지막 장치에서 고려할 수 있는 장치 지연시간(idle time, d_{ik}) 등 두가지만으로 구성되어 있음을 알 수 있다. 즉 makespan은 각 제품들이 마지막 장치를 나올 때 각 제품들의 생산은 끝나고 마지막 제품이 마지막 장치를 통과함으로 makespan이 결정되는 것이므로, 마지막 장치에서의 지연시간들의 조합만을 고려하는 것이 바람직할 것이다. 그러므로, 본 연구에서 makespan을 최소화하는 방안으로 이러한 개념을 도입하여 최적생산계획을 구하는 ITS법이라는 알고리듬을 제시하고자 한다.

Idle time search(ITS)법은 제일 먼저 제품 i 다음에 제품 k 가 조업되는 순서로 이루어진 제품들의 순서쌍 (i,k) 에 대한 지연시간, d_{ik} 들을 계산한다. 이때의 지연시간에는 준비시간이 포함되어 있으며 $N \times N$ 행렬을 이루게 된다. 다음 단계 에서는 이렇게 얻은 지연시간의 행렬을 MILP formulation의 해를 얻는 과정을 수행하여 최소의 지연시간을 갖는 순서쌍들의 집합을 찾는다. 여기에서 얻은 최적의 경로는 모든 제품이 연속적으로 이루어져 있는 순서쌍들의 집합을 이루므로 하나의 loop을 형성하게 될 것이다. 그러므로, 마지막 단계는 loop을 이루는 순서쌍 중에서 지연시간이 가장 큰 순서쌍을 제거하여 loop의 고리를 끊고 head time과 모든 제품의 수송및 조업시간을 합산하여 makespan을 계산하면 그 결과가 바로 최적의 생산계획의 해를 나타내는 것이다. 이러한 개념으로 makespan 을 최소화하는 목적함수, 즉 최소의 지연시간을 갖는 순서쌍들의 집합을 구하는 목적함수는 다음과 같이 나타낼 수 있다.

Objective Function :
$$\text{Min } C_d = \sum_{i=1}^{N}\sum_{k=1}^{N} d_{ik} X_{ik} \tag{2}$$

여기에서 (i,k) 는 제품 i 와 제품 k 가 이루는 순서쌍을 나타내고, 결정된 순서쌍의 i 제품이 인접한 뒤의 순서쌍에 다시 올 수 없으며, 똑같은 순서쌍이 다시 계산될 수 없다는 제한조건이 필요하고, 한 순서쌍에는 같은 제품이 배치될 수도 없다는 제한 조건도 필요하므로 다음과 같은 수식들로 나타낼 수 있다.

$$\sum_{k=1}^{N} X_{ik} = 1 \ (i = 1,2,\cdots,N), \quad \sum_{i=1}^{N} X_{ik} = 1 \ (k = 1,2,\cdots,N), \quad \sum_{i=1}^{N} X_{ii} = 0 \tag{3}$$

그리고, 순서쌍 (i,k)가 결정되면 (k,i)는 결정될 수 없다. 만일 이러한 제한조건이 없다면 두개의 제품이 포함된 순서쌍이 반복해서 결정될 수가 있게 될 것이다. 이것은 이진변수 X 들중에 N 개만이 1 의 값을 갖고 나머지는 0 인 값을 갖기 때문에, 제품 i 와 제품 k 가 한꺼번에 두번 선택이 되는 것이므로 최적의 조업순서를 결정하는데 다른 어떤 제품이 포함되지 않을 경우가 생긴다. 그러므로 이러한 제한조건을 고려하여 수학적으로 표현하면 다음과 같다.

$$X_{ik} + X_{ki} \leq 1 \tag{4}$$

위의 식 (2)에서부터 식 (4)까지는 기존의 전달시간및 준비시간을 고려한 MINLP법보다, 더 나아가서는 전달및 준비시간을 고려하지 않은 MILP법보다도 간단한 형태로 formulation 되었음을 알 수 있다. 이것은 MINLP formulation에 적용되어질 계산들을 먼저 수행한 후 그 결과를 이용하기 때문에 MILP의 형태로도 최적해를 구할 수 있는 것이다.

마지막 단계로 위의 단계에서 얻어진 순서쌍들의 loop에서 head time과 지연시간이 큰 것을 고려하여, 그값이 최소가 되는 순서쌍을 자름으로써 우리가 얻고자 하는 해를 구하게

되는 것이다. 예를 들어 위에서 구한 순서쌍에서 (3,2)의 제품쌍의 지연시간이 가장 크고 2번 제품의 head time이 가장 적다고 할때, (3,2)의 순서쌍을 자르게 되어 2번 제품이 제일 처음으로 처리되어야 하는 제품이 되는 것이며 3번 제품은 제일 마지막으로 처리되어야 하는 제품이 되는 것이다. 순서쌍 (j,k)를 자를때, 최종적으로 makespan을 최소화하는 계산과정을 수학적으로 표현하면,

$$\text{Min } C_{NM} = \sum_{i=1}^{N}\left(T_{i,M-1} + P_{i,M} + T_{i,M}\right) + \text{Min } C_d + \text{Min}(h_k - d_{jk}) \qquad (5)$$

과 같이 표현할 수 있다. 윗식의 마지막 항은 k제품의 head time과 자른 순서쌍 (j,k)의 지연시간 (d_{jk})과의 차가 최소가 되도록 한다는 것을 의미한다.

4.Results and Discussion

수송및 지연시간을 고려한 ZW 방안에 대한 8×4크기의 예제를 사용하여 기존의 MINLP에 의한 방법과 본 연구에서 제안한 ITS법으로 풀이하고 그 결과를 비교하여 보았다. 조업시간은 Table 1 에, 수송및 준비시간은 Table 2 와 3 에 각각 나타내었다. 지연시간을 계산하는 단계는 비교적 간단하므로 여기에서는 생략하였으며, 각 제품의 head time(h_i) 과 제품들의 순서쌍 (i,k)에 의한 지연시간, d_{ik} 들을 구한 결과는 Table 4 에 나타내었다. Table 4의 지연시간들은 모두 마지막 장치에 대한 준비 시간이 포함되어 있다. 그러므로, 모든 제품들로 구성된 제품생산계획을 최적화하는 것은 Table 4 의 각 제품들의 순서쌍에 의한 지연시간을 최소화 시키는 순서쌍들의 부분집합으로 배열하는 것이 최적의 생산계획을 수립하는 것이 된다.

본 연구에서 MINLP및 MILP의 수학적 formulation은 IBM RS/6000 workstation 에서 상용 소프트웨어인 GAMS를 이용하여 해를 구하였다. 먼저 MINLP의 방법에 대한 해는 GAMS DICOPT++를 사용하였고 그 결과는 P5-P4-P8-P7-P2-P1-P6-P3의 최적생산의 순서를 얻었고 그때의 최적생산시간은 198 unit time, 그리고 컴퓨터 계산시간(CPU time)은 201.9초 이었다.

ITS법을 이용한 계산은 지연시간을 최소화하는 식(2)에서 식(4)까지 적용하여 GAMS ZOOM을 사용하였고, 계산 결과로 얻어진 순서쌍들의 집합은 (8,5), (5,1), (1,6), (6,3), (3,7), (7,4), (4,2), (2,8)등으로 하나의 loop을 형성하고 있었다. 이때 지연 시간의 합(C_d)은 60 unit time이었고 컴퓨터 계산시간은 약 2초 정도 이었다. 다음 단계에서 마지막 장치 M을 지나는 모든 제품들의 조업시간과 앞뒤의 전달시간들의 합은 118이며, (8,5)의 제품쌍의 지연시간이 가장 큰 14이고 그 다음으로 큰 지연시간은 12로 (3,7)의 순서쌍이 갖고 있다는 것을 Table 4 에서 쉽게 알 수 있다. 그때 8과 5를 자르면 5번 제품이 30의 head time을 갖게되고 3과 7을 자른다면 7번 제품은 31의 head time을 갖게 된다. 식(5)의 마지막 항에 의하여 (8,5)를 자른 결과가 (3,7)을 자른 결과보다 1이 작으므로 5번이 처음 시작하는 제품이고 8번이 제일 마지막으로 처리되는 제품이 된다. 그러므로, 예제의 최적의 생산 순서는 P5-P1-P6-P3-P7-P4-P2-P8이며, 그때의 시간들을 식(5)에 대입하면, 118 + 60 + 30 - 14 = 194 unit time의 최적 생산시간을 얻을 수 있었다.

두가지의 방법으로 얻은 결과를 Table 5 에 정리하였다. Table 5 에 의하면 ITS 법이 MINLP 법보다 모든 면에서 뛰어난 결과를 가져 왔다. 여기에서 ITS법에 의한 CPU time은 모든 단계를 계산하는데 걸리는 시간을 나타내었다.

REFERENCES

1. Rajagopalan,D. and I.A. Karimi. Completion Times in Serial Mixed-Storage MultiProduct Processes with Transfer and Set-up Times, *Comput. Chem. Engng*, **13**, 175 (1989).
2. Birewar, D.B. and I.E. Grossmann, Incorporating Scheduling in the Optimal Design of Multiproduct Batch Plants, *Comput. Chem. Engng*, **13**, 141 (1989).

Acknowledgement
포항공대 지능자동화 연구센타를 통한 한국과학재단의 부분적 재정지원에 감사드립니다.

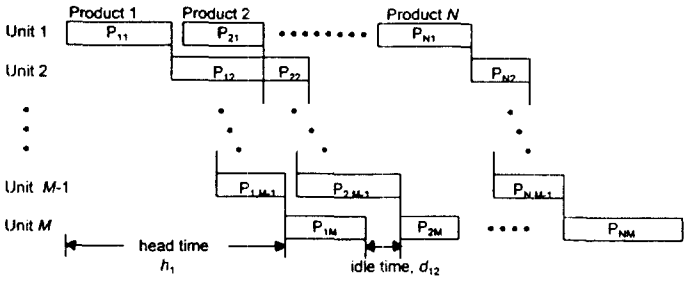

Fig. 1. General Gannt Chart of ZW Policy

Table 1. Processing Times for Example with 8 Products and 4 Units

	U1	U2	U3	U4
P1	10	20	5	30
P2	15	8	12	10
P3	20	7	9	5
P4	13	7	17	10
P5	8	3	16	7
P6	6	9	22	7
P7	7	5	15	12
P8	14	13	6	4

Table 2. Transfer Times for Example with 8 Products and 4 Units

	U0	U1	U2	U3	U4
P1	2	2	2	2	3
P2	3	3	3	3	1
P3	2	4	2	2	1
P4	2	2	1	4	2
P5	1	1	1	3	2
P6	3	3	2	2	2
P7	1	2	1	2	1
P8	2	4	1	1	2

Table 3. Setup Times of Units for Example.

Unit *i-j*	1	2	3	4	Unit *i-j*	1	2	3	4
S(1-2)	3	1	2	4	*S*(5-1)	2	2	1	4
S(1-3)	2	2	1	3	*S*(5-2)	2	2	3	2
S(1-4)	1	4	2	2	*S*(5-3)	2	3	3	3
S(1-5)	1	3	2	3	*S*(5-4)	1	1	3	4
S(1-6)	3	2	1	1	*S*(5-6)	4	3	3	1
S(1-7)	3	1	2	1	*S*(5-7)	2	3	2	3
S(1-8)	2	3	3	2	*S*(5-8)	4	3	2	2
S(2-1)	4	1	2	3	*S*(6-1)	1	2	1	2
S(2-3)	1	1	4	3	*S*(6-2)	3	3	2	3
S(2-4)	3	2	3	2	*S*(6-3)	4	1	1	2
S(2-5)	3	1	2	2	*S*(6-4)	2	3	4	1
S(2-6)	1	1	2	4	*S*(6-5)	2	1	1	4
S(2-7)	2	2	1	1	*S*(6-7)	2	2	3	2
S(2-8)	3	2	1	3	*S*(6-8)	1	3	3	3
S(3-1)	2	1	4	3	*S*(7-1)	2	3	2	1
S(3-2)	1	2	3	2	*S*(7-2)	1	1	1	1
S(3-4)	2	2	2	2	*S*(7-3)	2	1	2	2
S(3-5)	2	1	2	2	*S*(7-4)	1	2	1	3
S(3-6)	2	2	1	3	*S*(7-5)	3	2	2	2
S(3-7)	1	4	2	2	*S*(7-6)	1	2	4	1
S(3-8)	2	2	3	1	*S*(7-8)	2	1	1	1
S(4-1)	4	3	4	3	*S*(8-1)	1	1	2	1
S(4-2)	1	4	3	3	*S*(8-2)	4	1	4	4
S(4-3)	3	2	2	1	*S*(8-3)	2	1	3	2
S(4-5)	1	3	3	2	*S*(8-4)	3	4	1	1
S(4-6)	3	1	2	2	*S*(8-5)	3	1	3	2
S(4-7)	1	1	3	1	*S*(8-6)	3	3	1	1
S(4-8)	2	3	1	3	*S*(8-7)	2	3	1	1

Table 4. Head Times for Product *i* and Idle Times for Product Pairs (i,j).

j / *i*	h_i	P1	P2	P3	P4	P5	P6	P7	P8
P1	41	-	4	3	2	3	1	1	2
P2	44	8	-	8	10	8	15	6	6
P3	44	15	17	-	16	13	21	12	14
P4	42	4	6	6	-	8	14	7	3
P5	30	11	14	14	12	-	18	9	12
P6	45	2	8	4	13	9	-	10	3
P7	31	6	8	9	6	6	15	-	5
P8	40	17	19	17	18	14	26	13	-

Table 5. Comparison of Results obtained from MINLP and ITS Methods

	MINLP Formulation	Idle Time Search Method
Optimal sequence	P5-P4-P8-P7-P2-P1-P6-P3	P5-P1-P6-P3-P7-P4-P2-P8
Optimal unit time	198	194
CPU time	201.9 sec	about total 10 sec