

다단위 회분식 공정에 대한 제품 생산 계획의 최적화

김경숙 · 조영상

한국과학기술연구원 화학공정연구실
(1991년 11월 22일 접수, 1992년 3월 11일 채택)

Near Optimal Production Scheduling for Multi-Unit Batch Processes

Kyeong-Sook Kim and Young-Sang Cho

Chemical Process Laboratory, Korea Institute of Science and Technology, Seoul 136-791, Korea
(Received 22 November 1991; accepted 11 March 1992)

요 약

회분식 공정의 생산성과 효율에 매우 중요한 최적 생산 계획을 결정하는 새로운 알고리즘을 제안하였다. 본 연구에서 제안하는 알고리즘은 다단위 공정으로 이루어진 다품종 공정의 최적 생산 계획을 구하는 방법으로 기본 원리는 단위 공정이 두 개일 경우 최적 생산 계획을 찾아내는 Johnson 알고리즘과 동일하다. 본 연구에서 제안한 알고리즘을 다양한 예제에 적용한 결과 새로운 방법이 이전에 제안된 방법보다 우수하다는 것이 입증되었다. 또한 현재 연구가 진행 중인 표를 이용하여 총 공정 처리시간을 구하는 방법을 본 논문에서 소개하였다.

Abstract—The determination of a production sequence is an important problem in a batch process operation. In this paper, a new algorithm for a near optimal production sequence of N products in a M unit serial multiproduct batch process is proposed. The basic principle is the same as that of Johnson's algorithm for two-unit UIS system. The test results on a number of selected examples exhibit the superiority over the previously reported results. In addition, a tabulation technique is presented to calculate the makespan of a given sequence of production for all processing units under UIS mode.

1. 서 론

1950년대의 화학 공장이 안고 있던 과제는 구식의 회분식 공정을 현대적인 연속 공정으로 전환하는 것이었다. 그러나 최근의 조사에서 밝혀진 바에 의하면 현재에도 세계에서 생산되고 있는 화학 제품들 중의 상당량이 회분식 공정을 이용하는 공장에서 생산되고 있으며 이 비율은 앞으로도 감소하지 않을 것으로 전망되고 있다.

일반적으로 연속 공정은 소품종 대량 생산을 목적으로 하여 반응조건이 비교적 간단하고 일년 내내 수요가

있는 제품들에 적합하게 반하여 회분식 공정은 정밀성이 요구되는 다품종 소량 생산에 주로 이용되며 제품이 복잡한 반응을 거치거나 반응이 매우 느리게 진행되는 경우 및 계절 상품에 유용하다는 특성이 있다. 최근의 경향은 소비자의 기호가 고급화됨에 따라 고부가 가치 제품 즉 다품종 소량 생산을 통하여 만들어진 제품을 선호하여 다품종 공정 및 다목적 공정이 요구된다. 또한 원료와 수요가 다양화되어 공정의 유연성이 요구되고 시장동향에 신속히 대응하기 위하여 운전의 유연성도 필요하므로 결과적으로 이 요구들을 적절히 만족시켜 줄 수 있는 회분식 공정을 최적화하는 것은 매우 중요한

의미를 갖는다. 이러한 추세에 따라 현재 회분식 공정의 최적화에 대한 연구가 활발히 진행되고 있으며 본 연구도 국내에 산재된 회분식 공장들을 개선하여 생산성과 효율을 높이하고자 하는 취지로 시작되었다.

회분식 공정의 최적화란 일반적으로 생산할 제품들이 이들이 거쳐야 할 장치들과 적절히 조합하여 제품의 생산 순서를 계획함으로써 최단 시간으로 정해진 제품을 생산하는 것을 의미하여 이를 생산 계획이라 하기도 한다.

구체적으로 생산 계획이란 시간, 장치, 재고 및 제품의 시장 수요를 대처할 수 있는 노동력과 에너지 등의 유용한 자원들을 가장 경제적이고 효율적인 방법으로 적절히 배치하는 것을 의미하며 공정 조업상의 상세한 일정 계획은 전체 생산 계획의 일부이지만 다양한 공정들이 직면하는 문제들을 고려한 생산 일정을 결정하는 상세 일정 계획의 이론은 현재 활발히 연구가 진행되고 있다.

회분식 공정 최적화의 목적 함수로는 일반적으로 "makespan"이라 정의된 모든 제품을 생산하는데 소요되는 총 시간을 많이 채택하며 그 외에도 전 제품이 공정을 흐르는 평균시간, 최대 지연시간 등이 이용된다.

회분식 공정에 관한 연구가 시작된 것은 1960년대로 Ketner[1]와 Loonkar[2] 등이 단일 제품을 생산하는 회분식 공정의 자본 비용을 최소화하는 것에 대한 연구를 하였다. 회분식 공정을 최적화하는 방법이 구체적으로 연구되기 시작한 것은 70년대 중반 이후로 Sparrow[3] 등이 다품종 공장에 대한 경험 법칙을 제안하였으며 Grossmann[4] 등은 제품의 순환 시간 제약 조건을 경험 법칙에 삽입하였다. Takamatusu[5] 등과 Szwarc[6], Gupta[7] 및 Wiede[8] 등은 공정간의 저장 방침을 고려한 다품종 공정의 생산 계획 문제에 관하여 연구한 바 있으며 Ku와 Karimi[9]는 경험 법칙을 토대로 총 공정 처리 시간(makespan)을 구하는 식을 유도하였고 이를 기초로 Rajagopalan[10] 등이 MIS(Mixed Intermediate Storage) 방침을 사용하는 다품종 공정 문제의 총 공정 처리 시간을 구하는 방법을 정리하여 제안하였다. 또한 Ku[11]와 Yeh[12] 등은 MILP(Mixed Integer Linear Programming) 기법을 이용한 방법을 연구하였으며 최근에도 Malone[13]과 Das[14] 등이 MILP기법의 일종인 시뮬레이티드 아닐링(simulated annealing) 방법을 이용하여 생산 계획을 최적화하는 것에 대하여 소개하였다.

2. 다품종 공정의 생산 계획 이론

회분식 공정으로 이루어진 공장은 두 가지로 분류할

수 있는데 그 하나는 다품종 공장이며 또 하나는 다목적 공장으로 각각의 특징은 다품종 공장에서는 모든 제품들이 동일한 공정 단계를 거치며 다목적 공장에서는 서로 다른 경로를 거쳐 생산된다는 것이다. 일반적으로 다품종 공정의 생산 계획 문제를 flowshop문제라 하며 다목적 공장의 생산 계획 문제를 jopshop 문제라 부른다. 본 연구는 비교적 해석이 용이한 다품종 공정을 대상으로 한 것이다.

다품종 공정의 생산 계획 즉 flowshop문제는 다시 두 가지로 나눌 수 있는데 첫째는 생산하고자 하는 제품의 생산순서를 결정하는 것으로 이는 전체 공정 완료 시간에 상당한 영향을 미치며 둘째는 주어진 제품들의 생산순서가 결정된 후 각 단위 공정간의 저장 방침을 고려하여 총 공정 처리 시간을 최소화하는 작업이다. 이 후에서는 전자를 제품 생산 계획(sequencing), 후자를 상세 일정 계획(scheduling)이라 하겠다.

2-1. 문헌에 제안된 생산 계획 방법

다품종 공정의 생산 계획 문제 즉 flowshop 문제의 해법에는 경험 법칙을 이용하는 방법과 MILP법 및 BAB(Branch and Bound) 방법이 있는데 MILP법과 BAB 방법은 제품과 단위 공정의 수가 증가하면 그 적용에 한계가 있어 그러한 제한이 없는 경험 법칙을 기초로 하여 최적에 가장 가까운 해를 찾는 방법에 관한 연구가 계속되어 왔다. 일반적인 제품 생산 계획 문제에 사용되는 경험적 알고리즘은 두 가지로 분류할 수 있는데 첫째는 경험 법칙에 준하여 몇 개의 순서를 찾아낸 후 그 중에서 가장 좋은 해를 구하는 것이고, 둘째는 임의의 생산 계획에서 시작하여 연속적인 계산을 통해 더 이상 개선이 이루어지지 않을 때까지 순서를 바꾸어주는 것이다. 이 방법은 정확도는 높지만 계산 시간이 오래 걸리는 것이 단점이다. 현존하는 알고리즘 중 정확한 해를 도출해 내는 알고리즘은 Johnson 알고리즘이 유일한 것이며, 대부분의 경험적 알고리즘은 Johnson 알고리즘을 근간으로 하여 개발되었다.

Johnson 알고리즘을 사용하면 두 개의 단위 공정으로 이루어진 UIS(Unlimited Intermediate Storage) 방침의 flowshop 문제에 대해 최소 총 공정 처리 시간을 갖는 생산 계획을 구할 수 있으며 이 알고리즘의 적용과정은 다음과 같이 요약된다.

(1) N개의 제품을 첫 단위 공정에서의 처리 시간 보다 두 번째 단위 공정에서의 처리 시간이 더 긴 제품들의 그룹 P와 나머지 그룹 Q의 두 그룹으로 나눈다.

(2) P그룹에 속하는 제품들을 첫 단위 공정에서의 처리 시간이 짧은 것부터 차례로 나열한다.

(3) Q그룹에 속하는 제품들을 두 번째 단위 공정에서

Table 1. Example 2.1

Product	1	2	3	4	5	6
Processing time						
on Unit 1	10	5	11	3	7	9
on Unit 2	4	7	9	8	10	15

서의 처리 시간이 긴 것부터 차례로 나열한다.

(4) 2와 3에서 구한 순서들을 P-Q순으로 나열하면 그것이 최소 총 공정 처리 시간을 갖는 최적 순서이다.

예를 들어 제품이 6개이고 단위 공정이 두 개인 UIS flowshop 문제에 위의 Johnson의 알고리즘을 적용하면 다음과 같다.

위 예제에서 P그룹은 [2, 4, 5, 6]이고, Q그룹은 [1, 3]이므로 최적 생산 계획은 [4-2-5-6-3-1]이다.

위의 Johnson의 알고리즘은 단위 공정이 2개인 경우에만 최적해를 구할 수 있으므로 단위 공정이 여러 개인 실제 공정에 적용하는데 어려움이 있어 이후 Dannenbring[15]이 제안한 RAES(Rapid Access with Extensive Search) 알고리즘이 단위 공정이 여러 개인 경우에 많이 이용되고 있다. RAES 알고리즘은 이미 제안된 바 있는 유사한 알고리즘들 중에 가장 효과적인 방법으로 알려져 있다. RAES에서의 RA(Rapid Access) 단계에서는 최초의 생산 계획을 결정하며 ES(Extensive Search) 단계에서는 현재의 생산 계획을 개선하는 과정이 반복된다.

RAES 알고리즘은 다음 단계로 실행한다.

○1단계: RA(Rapid Access) 과정

(1) M개의 단위 공정으로 이루어진 flowshop문제가 주어졌을 때 다음의 인공적인 처리 시간 계산을 통해 가상의 두 개의 단위 공정문제로 바꾸어 준다.

$$a_i = \sum_{j=1}^M (M-j+1) \cdot t_{ij} \quad i=1, 2, \dots, N \quad (1)$$

$$b_i = \sum_{j=1}^M t_{ij} \quad i=1, 2, \dots, N \quad (2)$$

여기서 a_i 와 b_i 는 가상의 단위 공정 1과 2에서의 제품 i 의 인공적인 처리 시간을 나타낸다. 또한 N 과 M 은 각각 제품의 수와 단위 공정의 수를 나타낸다.

(2) 위에서 구한 가상의 두 단위 공정계에 Johnson의 법칙을 적용하여 최초의 생산 계획을 정한다.

○2단계: ES(Extensive Search) 과정

(1) 현재의 생산 계획과 이웃한 것 중에 더 짧은 총 공정 처리 시간을 가진 생산 계획이 존재하는지 검토한다. 여기서 이웃한 생산 계획이란 현재의 생산 계획

에서 서로 인접한 제품의 순서를 바꾸어줌으로서 생기는 $N-1$ 개의 생산 계획을 의미한다.

(2) 이웃한 생산 계획들 중 가장 짧은 총 공정 처리 시간을 갖는 생산 계획을 선택하여 1을 반복한다. 더 짧은 총 공정 처리 시간을 갖는 생산 계획이 없으면 그 생산 계획이 최종 생산 계획이다.

총 공정 처리 시간을 구하는 과정은 바로 상세 일정 계획을 의미하며 이는 Ku와 Karimi[9]가 제안한 식 즉, 주어진 저장 방침에 따라 각 제품의 순서대로 최대값 함수를 적용하여 총 공정 처리 시간을 구하는 방법을 이용한다.

또한 규모가 작은 문제에 많이 이용되는 MILP 방법은 목적함수와 조건이 모두 선형인 문제를 최적화하는 방법으로 일부 변수들이 정수 또는 한쌍의 정수값 (0-1)을 갖는다는 점에서 일반적인 LP문제와 차이가 있다. 한 쌍의 변수는 다음과 같이 정의된다.

$$X_{ij} = \begin{cases} 1: \text{단위공정 } j \text{에 제품 } i \text{가 채류하는 경우} \\ 0: \text{단위공정 } j \text{에 제품 } i \text{가 채류하지 않는 경우} \end{cases} \quad (3)$$

총 공정 처리 시간을 최소화하는 것이 우리의 목적 이므로 목적 함수는 제품의 수가 N 개이고 단위 공정의 수가 M 개인 회분식 공정의 전체 공정 완료 시간 C_{NM} 의 최소화이며 제약 조건은 다음과 같다.

첫째, 공정이 진행되는 동안에 하나의 제품이 전 공정 상에서 한 위치에만 지정되어 있어야 한다.

둘째, 공정상의 각 위치는 각각 하나의 제품이 차지 하고 있어야 한다.

셋째, 공정 완료 시간에 대한 것으로 이 조건에 의해 바로 공정 완료 시간을 계산할 수 있다. 식은 다음과 같다.

$$C_{ij} = \text{MAX}\{C_{i(j-1)}, C_{(i-1)j}, C_{(i-j-1)} - t_{kij}\} + t_{kij} \quad (4)$$

윗 식에서 $i \leq 0$, $j \leq 0$ 또는 $j > M$ 인 경우 $C_{ij} = 0$ 이다.

위의 수식화과정을 통해 얻어진 수식들은 이미 널리 알려진 LINDO(Linear Interactive Discrete Optimizer) 나 MPOS(Multipurpose Optimization System) 등의 package를 이용하여 해를 구할 수 있다. 그러나 이 방법은 상당한 기억용량을 필요로 하기 때문에 문제 크기가 커지면 적용에 한계가 있다.

이외에 BAB(Branch and Bound) 방법은 기본적으로 모든 가능한 제품 생산 계획을 부분적으로 열거하는 방법으로 문제를 좀더 전문적인 부문제들로 가치를 내려서 모든 가능한 순열들이, 제품 i 로 시작하는 모든 생산 계획을 포함한 부그룹 $i(i=1, N)$ 로 이어진 N 개의 부그룹들로 나누어지도록 한 것이다.

Table 2. Scheduling example 2.2

Product	1	2	3	4
unit 1	25	30	19	17
unit 2	10	15	21	23

Table 3. Makespan table for example 2.2

		Production sequence			
		1	2	3	4
$\alpha \cdots$	1	25	55	74	91
$\beta \cdots$	2		35	70	95
		118 = makespan			
		\dot{a}	\dot{b}	\dot{c}	\dot{d}

2-2. 본 연구에서 새로이 제안하는 알고리즘

본 연구에서 제안하고 있는 방법은 2 단위 공정계에만 적용이 가능한 Johnson의 알고리즘을 이용하기 위하여 다단위 공정계를 가상의 2 단위 공정계로 변환하되 개개의 단위 공정을 고려하여 전체 생산 계획을 결정하는 것이다.

우선 다음과 같은 전달시간과 준비시간(set-up time)이 무시된 간단한 생산 계획 문제를 생각해 보자. 만약 [1-2-3-4]의 생산 계획이 주어졌다면 UIS 방침을 사용한다고 할 때 이 생산 계획의 makespan은 Table 3을 이용하여 간단히 계산할 수 있다. Table 3에서 첫 줄은 각 제품들이 단위 공정 1을 완료하는 시간을 나타내고 있으며, 둘째줄은 각 제품의 공정 완료 시간을 의미한다. 즉 제품 1의 공정 완료 시간은 Table 3의 둘째줄에 나타나 있는 값 35이다. Table 3에서 제품 2의 공정 완료 시간은 공정 2에서 제품 1의 공정 완료 시간과 공정 2에서 제품 2의 처리 시간을 더해준 값이다.

Table 3에서 보면 β 와 c 가 만나는 곳에 들어갈 제품 2의 공정 완료 시간은 b 줄의 두 값 중 큰 값과 제품 2의 단위 공정 2에서 처리 시간을 더한 값이다. 이를 수식으로 표현하면 다음과 같다.

$$MS = T_1 + \sum_{i=2}^N L_i + \sum_{i=1}^{N-1} \text{pos}[F_{i+1} - (T_i - \sum_{k=1}^{i-1} F_k)] \quad (5)$$

여기서 T_i : 제품 i 의 총 공정 처리 시간

L_i : 제품 i 의 두 번째 공정 처리 시간

F_k : 제품 k 의 첫째 공정 처리 시간

MS: 총 공정 처리 시간

$$\text{pos}(a) = a \quad \text{if } a > 0 \\ = 0 \quad \text{if } a \leq 0$$

그러므로 식 (5)를 예제에 적용하면 다음과 같다.

$$MS = 35 + 59 + \text{pos}(30 - 35 + 25)$$

Table 4. Scheduling example 2.3

Prod.	1		2		3	
	P	T	P	T	P	T
		5		6		3
1	2	3	4	3	7	4
2	1	2	2	7	3	9
3	4	1	3	6	2	1

Table 5. Makespan table for example 2.3

Unit	Production sequence									
	1			2			3			
1	5	7	10	16	20	23	26	33	37	
2				10	11	13	23	25	32	37 40 49
3							13	17	18	32 35 41 50 52 53

$$+ \text{pos}(19 - 70 + 55) + \text{pos}(17 - 95 + 74) = 118$$

예제에 Johnson의 알고리즘을 적용하면 최적 생산 계획은 [4-3-2-1]이 되며 이 때 총 공정 처리 시간은 식을 이용하여 다음과 같이 구할 수 있다.

$$MS = 40 + 46 + \text{pos}(19 - 40 + 17)$$

$$+ \text{pos}(30 - 61 + 36) + \text{pos}(25 - 81 + 66) = 101$$

위의 결과는 식 (5)가 주어진 생산 계획에 대한 makespan을 계산할 수 있음을 보여준다. 식 (5)에서 첫째 항은 생산 계획상에서 처음으로 처리되는 제품의 전 공정 처리 시간을 나타내며 둘째 항은 위에서 지연이 생기지 않는 경우에 주어진 생산 계획의 예상되는 총 공정 처리 시간을 나타내고 있다. 또한 마지막 항은 첫 공정에서 지연이 요구될 경우 그 지연시간을 계산하는 항이다. 위에서 총 공정 처리 시간의 최소값은 마지막 항의 $(T_i - \sum F_k)$ 와 F_{i+1} 를 고려함으로써 구할 수 있음을 알 수 있다. 이는 Johnson의 알고리즘에서 F_{i+1} 값과 $(T_i - \sum F_k)$ 값을 조사하여 최적 생산 계획을 결정하는 것과 유사하다. 그러면 Table 4와 같은 좀더 일반적인 생산 계획 문제를 살펴보자. Table 4에서 P는 공정 처리 시간을 의미하며 T는 전달시간을 의미한다.

앞에서 [1-2-3]의 생산계획을 선택한다면 UIS 방침을 사용할 경우 이 생산계획의 총 공정 처리 시간은 역시 다음과 같은 표를 이용하여 계산할 수 있다.

Table 5에서 각 제품 밑에서 나누어진 세 세로칸들은 각각 해당된 단위 공정으로 제품이 전달되는 시간, 제품의 공정 처리 시간, 해당 단위 공정의 외부로 제품이 보내어지는 누적시간을 나타내며 각 칸들을 채우고 있는 숫자들은 맨 첫 제품이 첫 단위 공정에 전달되기 시작하는 시간부터 지정된 위치에 해당되는 일을 완료하기 까지 걸리는 시간을 의미한다.

만약 맨 마지막 단위 공정 전의 공정까지 총 공정 처리 시간에 영향을 미칠 수 있는 지연 시간이 발생하지 않는다고 가정한다면 예제 2.3의 가능한 최소 총 공정 처리 시간을 예제 2.2에서와 동일한 수학적 표현을 빌려 다음과 같은 식으로 나타낼 수 있다.

$$MS = T_1 + \sum_{i=2}^N L_i + \sum_{i=1}^{N-1} \text{pos}[P_{i+1} - (T_i - \sum_{k=1}^{i+1} F_k)] \quad (6)$$

여기서 T_i : 제품 i 의 각 단위 공정으로의 전달시간을 포함한 전체 공정 처리 시간

L_i : 제품 i 가 공정 전후의 전달 시간을 포함하여 맨 끝 단위 공정에 머무는 시간

F_k : 제품 k 의 공정 전후의 전달 시간을 포함한 첫 단위 공정에서의 체류 시간

P_i : 제품 i 가 마지막 단위 공정으로 전달되는 시간을 제외한 맨 끝 공정 이전의 공정까지의 공정 완료 시간

$$\begin{aligned} \text{pos}(a) &= a \quad \text{if } a > 0 \\ &= 0 \quad \text{if } a \leq 0 \end{aligned}$$

식 (6)은 식 (5)의 F_{i+1} 대신 P_{i+1} 을 넣어준 것을 제외하고는 그 형태가 식 (5)와 동일하다. 식 (5)에서 최적 생산 계획이 F_i 항들과 $(T_i - F_i)$ 항들로 이루어진 단순한 단위 공정 두 개의 생산 계획 문제에 Johnson의 알고리즘을 적용하여 구해질 수 있음을 상기할 때 공정이 여러 개인 경우에도 P_i 항과 $(T_i - F_i)$ 로 이루어진 가상의 단위 공정 2개의 생산 계획 문제에 Johnson의 법칙을 적용하면 최소 공정 처리 시간에 근접한 해를 구할 수 있을 것으로 추론된다.

위의 논리를 일반적인 생산 계획 문제의 초기 최적 생산 계획을 결정하는데 적용하였으며, 그 방법을 정리하면 다음과 같다.

- (1) 모든 제품에 대한 $P_i (i=1, 2, \dots, N)$ 를 계산한다.
- (2) 모든 제품에 대한 $(T_i - F_i)$ 를 계산한다($i=1, 2, \dots, N$).

(3) 주어진 문제를 가상의 단위 공정 2개의 최적 생산 문제로 바꾸어 준다. 즉 제품 i 의 첫 단위 공정에서의 공정 처리 시간을 P_i , 제품 i 의 두번째 단위 공정에서의 공정 처리 시간을 $(T_i - F_i)$ 라 놓는다.

(4) 3에서 변형된 가상의 예제에 Johnson의 알고리즘을 적용하여 생산 계획의 초기 가정을 하여 준다. 위에서 제안된 확장된 Johnson의 알고리즘을 예제 2.3에 적용하면 다음과 같다.

$$\begin{aligned} P_1 &= 11, P_2 = 15, P_3 = 17 \\ T_1 - F_1 &= 8, T_2 - F_2 = 18, T_3 - F_3 = 15 \end{aligned}$$

Table 6. Modified example 2.3

	1	2	3
1	11	15	17
2	8	18	15

Table 7. Makespan table for sequence [2-3-1]

Unit	Production sequence									
	1			2			3			
1	6	10	13	16	23	27	32	34	37	
2				13	15	22	27	30	39	42 43 45
3							22	25	31	40 42 43 45 49 50

위의 결과를 이용하여 가상된 단위 공정 2개의 생산 계획 문제를 만들면 Table 6과 같다.

Table 6의 문제에 Johnson의 알고리즘을 적용하면 최초의 최적 생산 계획은 [2-3-1]이 된다. 총 공정 처리 시간 계산표를 이용하여 [2-3-1]의 총 공정 처리 시간을 구하면 다음과 같다.

위의 Table 7에서 알 수 있듯이 마지막 단위 공정전까지 예기치 못한 지연 시간이 나타나지 않으므로 식 (6)을 사용하여도 동일한 결과를 얻을 수 있다.

$$\begin{aligned} MS &= T_1 + L_2 + L_3 + \text{pos}(P_2 - T_1 + \sum_{k=1}^2 F_k) \\ &\quad + \text{pos}(P_3 - T_2 + \sum_{k=1}^3 F_k) \\ &= 31 + 7 + 12 + \text{pos}(30 - 31) + \text{pos}(43 - 43) = 50 \end{aligned}$$

3. 예제 및 결과

4-3c

	1		2		3		4	
	P	T	P	T	P	T	P	T
		1		5		2		5
1	2	4	4	9	1	2	1	4
2	5	3	2	8	4	9	3	3
3	2	7	3	2	3	2	2	1

NA: 3-2-1-4 [62]

RAES: 4-3-2-1 [67]

*NA=New Algorithm의 약자

6-7a

	1		2		3		4		5		6	
	P	T	P	T	P	T	P	T	P	T	P	T
		1		3		3		4		2		1
1	1	2	4	2	1	2	3	6	1	5	3	1

Table 8. Results for examples

		NA	NA-1st	RAES	RAES-1st	
3-3	a	52	52	53	52	○
	b	50	50	57	54	○
	c	54	54	56	54	○
3-4	a	43	43	43	43	△
	b	52	52	52	52	△
	c	67	67	69	69	○
3-5	a	63	63	63	63	△
	b	71	70	73	71	○
	c	67	67	76	67	○
4-3	a	73	67	75	71	○
	b	64	62	68	63	○
	c	62	62	67	64	○
4-4	a	77	76	77	77	○
	b	30	30	30	30	△
	c	64	64	66	64	○
4-5	a	78	78	82	78	○
	b	84	84	88	84	○
	c	102	96	109	108	○
5-5	a	75	74	76	74	○
	b	105	101	98	98	×
	c	110	104	110	106	○
5-6	a	110	107	108	107	×
	b	111	108	107	107	×
	c	91	88	94	87	○
6-5	a	105	102	118	106	○
	b	112	112	118	114	○
	c	110	110	113	113	○
6-6	a	107	104	110	108	○
	b	118	118	123	116	○
	c	115	112	119	118	○
6-7	a	125	124	135	124	○
	b	165	159	154	151	×
	c	144	140	145	143	○
6-8	a	129	123	147	140	○
	b	157	150	156	150	×
7-6	a	133	127	143	137	○
	b	129	128	128	128	×
8-5	a	137	131	142	132	○
	b	130	128	129	129	×
8-8	a	160	160	164	157	○
	b	196	193	202	197	○
9-9	a	201	187	188	180	×
	b	221	201	228	215	○
10-10	a	241	241	424	424	○
	b	238	238	535	535	○

-1st : ES 과정을 한번 거친 후의 결과

○ : NA가 RAES 보다 우수한 경우

△ : NA가 RAES 결과가 동일한 경우

× : NA가 RAES 보다 나쁜 경우

2	2	3	8	1	5	7	2	3	2	4	4	2
3	6	4	2	8	4	3	2	7	7	7	5	3
4	3	2	3	3	3	8	8	6	8	2	2	4
5	2	4	1	5	2	2	5	6	6	5	8	1
6	8	2	1	2	6	5	2	2	3	1	6	2
7	6	1	7	4	1	4	4	3	2	4	3	4

NA : 1-6-2-3-5-4 [125]

RAES : 3-1-4-5-6-2 [135]

Table 8에서 볼 수 있듯이 전체 생산 계획에 RAES 알고리즘과 새로운 알고리즘을 적용한 결과 제품의 수가 5개 이하의 작은 갯수일 경우 새로운 알고리즘이 압도적으로 우수하였으며, 제품의 수가 많아지면 새로운 방법을 이용하여 구한 첫 생산 계획이 최단 총 공정 처리 시간에 접근하는 정도가 제품의 수가 작을 때에 비하여 상당히 효율이 떨어져 RAES 접근 방법의 접근 정도보다 약간 우세한 정도인 것으로 나타났다. 그러나 제품의 수가 많아질수록 RAES 방법으로는 찾을 수 없는 월 등하게 총 공정 처리 시간이 짧은 전체 생산 계획을 새로운 알고리즘이 찾아내는 것을 확인할 수 있었는데 이는 본 연구에서 제안된 새로운 알고리즘이 경향을 비교하여 전체 생산 계획을 제안하는 RAES 방법과는 달리 Johnson의 알고리즘과 동일한 이론적 근거를 토대로 제안된 알고리즘이기 때문인 것으로 생각된다. 또한 RAES 방법이 가상의 두 단위 공정을 정의할 때 각각의 공정시간에 일정한 웨이트 값을 곱하여 주는데 반하여 새로운 알고리즘은 단순 덧셈으로 가상의 두 공정을 정의하므로 두 방법 모두 동일한 ES 과정을 통하여 최적 생산 계획을 찾다고 할 때 계산 시간면에서도 새로운 알고리즘이 효율적이며 이는 제품의 수와 단위 공정의 수가 많아질수록 더욱 두드러질 것이다. 본 연구에서는 RAES 방법에서 사용하는 ES 방법을 새로운 알고리즘에도 적용하고 있으며 현재보다 새로운 알고리즘에 적합한 ES 방법에 관한 연구가 진행 중에 있다.

4. 결 론

본 연구의 목적은 회분식 공정의 최적화로 현재까지 연구된 회분식 공정의 최적화 방법을 고찰하고 이를 개선하여 국내에 많이 산재되어 있는 회분식 공장의 생산성과 효율을 높이하고자 하였다.

본 연구에서는 현재까지 알려진 최적화 방법들을 고찰한 결과 이 중 가장 적용범위가 넓고 유연성이 있는 경험적 접근을 통한 최적화 방법을 선택하여, 단위 공정이 두 개일 경우 최적 생산 계획을 구할 수 있는 John-

son 알고리즘의 원리를 토대로 다단위 회분식 공정에 까지 적용할 수 있는 새로운 알고리즘을 제안하였으며 상세 일정 계획 방법으로써는 아직 저장방법에 따른 개선이 요구되긴 하지만 이전의 각 경우의 최대치를 고려하는 상세 일정 계획 방법보다 훨씬 간단하고 편리한 표를 이용한 상세 일정 계획 방법을 제안하였다.

새로운 생산 계획 알고리즘의 우수성을 확인하기 위하여 현재 가장 널리 이용되고 있는 전체 생산 계획 방법인 RAES(Rapid Access Extensive Search) 알고리즘과 본 연구에서 새로이 제안한 알고리즘을 여러 예제들에 적용하여 그 결과들을 비교 검토하였는데 그 결과 종래의 경험적 접근에 의한 전체 생산 계획, 즉 RAES 알고리즘보다 본 연구에서 제안하는 새로운 알고리즘이 최적 생산 계획을 다방면에서 보다 효율적으로 찾아낼 수 있음이 확인되었다.

NOMENCLATURE

- a_i : the artificial processing time of product i on pseudo units 1
- b_i : the artificial processing time of product i on pseudo units 2
- C_{ij} : completion time of product i from unit 1 to unit j
- F_k : first unit staying time including the transfer time for product k
- L_i : last unit staying time including the transfer time for product i
- M : number of units
- MS : makespan
- N : number of products
- P : processing time
- P_i : the processing time upto the unit before the last unit excluding the transfer time to the last unit for product i

- t_{ij} : processing time of product i on unit j
- T : transfer time
- T_i : total processing time including the transfer time for product i
- X_{ij} : binary variable for MILP

REFERENCES

1. Ketner, S. E.: *Chem. Engng*, **67**, 121(1960).
2. Loonkar, Y. R. and Robinson, J. D.: *Ind. Engng Chem., Process Des. Dev.*, **9**, 625(1970).
3. Sparrow, R. E., Forder, G. J. and Rippin, D. W.: *The Chem. Engng*, **289**, 520(1975).
4. Grossmann, I. E. and Sargent, R. W. H.: *Ind. Engng Chem., Process Des. Dev.*, **18**, 343(1979).
5. Takamatsu, T. and Hishimoto, I.: *Ind. Engng Chem., Process Des. Dev.*, **21**, 431(1982).
6. Szwarc, W.: *Mgmt Sci.*, **29**, 477(1983).
7. Gupta, J. N. D.: *J. Opl. Res. Soc., Japan*, **29**, 206(1986).
8. Wiede, W. and Reklaitis, G. V.: *Comput. Chem. Engng*, **1**(4), 345(1987).
9. Ku, H., Rajagopalan, D. and Karimi, I.: *Chem. Engng Prog.*, August, 35(1987).
10. Rajagopalan, D. and Karimi, I.: *Comput. Chem. Engng*, **13**(1/2), 175(1989).
11. Ku, H. and Karimi, I.: *Ind. Eng. Chem. Res.*, **27**, 1840(1988).
12. Yeh, N. C. and Reklaitis, G. V.: *Comput. Chem. Engng*, **11**(6), 639(1987).
13. Malone, M. F.: "Batch Sequencing by Simulated Annealing", AIChE Annual Meeting, San Francisco, CA, paper 23f, 1989.
14. Das, H., Cummings, P. T., and LeVan, M. D.: *Comput. Chem. Engng*, **14**(12), 1351(1990).
15. Dannenbring, D. G.: *Manag. Sci.*, **23**, 1174(1977).