

## 다목적 회분공정의 최적생산계획을 위한 조업완료시간의 순환수학적표현법

박상대 · 정재학<sup>†</sup>

영남대학교 화학공학및공업화학부  
(1998년 12월 11일 접수, 1999년 1월 28일 채택)

### A Recursive Formulation of Completion Times Algorithm for Optimal Scheduling of Multi-purpose Batch Process

Sangdae Park and Jae Hak Jung<sup>†</sup>

School of Chemical Engineering and Technology, Yeungnam University, Kyongsan 712-749, Korea  
(Received 11 December 1998; accepted 28 January 1999)

#### 요 약

최근 소비자의 기호와 수요가 짧은 시간 내에 변화하고 제품의 수명이 짧아짐에 따라 반연속 및 회분공정에 대한 연구가 늘어나고 있다. 회분공정은 효율 면에서 연속 공정보다 낮다. 그러한 이유로 대부분의 회분공정에 대한 연구는 생산성 증가에 대한 부분으로 초점이 맞추어졌다. 회분식 공정의 한 종류인 다품종 회분식 공정의 조업연구가 활발하였던 것에 비해 다목적 공정에 대해 발표된 논문은 적었다. 그 이유는 다목적 회분식 공정이 다품종 회분식 공정에 비해 복잡하기 때문이다. 그러나 최근 각광받는 전자, 의약, 정밀화학 등의 분야에서의 다품종 소량 생산의 유형이 다목적 공정 조업화 되어감에 따라 다목적 회분공정의 연구분야는 많은 연구가 필연적인 분야가 되었다. 이런 연구에 대해 조업완료시간 알고리즘은 필연적으로 요구된다. 본 논문에서는 수송시간과 준비시간을 고려한 비순차(non-sequential) 다목적 회분식 공정의 조업완료시간을 순환수식(recursive formulation)으로 표현하고 그 수식을 여러 예제들을 통해 검증해 보았다. 그리고 수식의 순서도 및 계산 결과에 대한 Gantt Chart를 나타내었다. 처리시간, 준비시간 및 수송시간은 난수 발생기에 의해 생성시켰다.

**Abstract** – Recently, as the taste of consumers and the demand of products are changed within short period and the life cycle of products is shortened, the operation of semicontinuous and batch processes have been studied more frequently over the past decade. In general, the productivity of batch process is lower than that of continuous process. For these reason most of researches in batch process are focused on increasing the productivity. The study of operation of batch processes used to be classified as two category; multi-product batch process and multi-purpose batch process. Between them, multi-product processes were actively studied, but only a few papers about the multi-purpose batch plants were presented. Because the operation of multi-purpose batch process is more complicated than that of multi-product batch plant. In this paper we presented completion times algorithm of non-sequential multi-purpose batch process[13] with non-zero transportation times and set-up times as a recursive formulation. We also described the formulation by computer program and flowchart and the results of calculation by gantt chart and processing map. For testing the proposed formulation, we applied them to several problems which are composed of randomly generated processing times, set-up times and transfer times.

**Key words:** Completion Time Algorithm, Multi-Purpose Batch Plant

#### 1. 서 론

최근 소비자의 기호와 요구가 짧은 시간내에 변동함에 따라 수요자의 요구에 신속히 대처할 수 있는 고부가가치의 다품종 소량 생산 시스템들이 각광을 받고 있다. 그래서 최근 화학공업산업에서는 다품종 소량 생산체제에 큰 효과를 나타낼 수 있는 반연속식 공정(semi-continuous process)과 회분식 공정(batch process)의 중요성이 의약품, 정밀 화학, 농약, 페인트, 촉매 등에서 더욱 부각되고 있다. 또한 식품, 석유화학제품, 섬유 등을 생산하던 기존의 연속식 공정을 반연속식 또는 회분식 공정으로 변환하고자 하는 노력도 시도되어지고 있

다. 회분식 공정의 조업은 연속식 공정과는 달리 많은 노동력의 요구, 낮은 효율, 다음 조업을 위한 준비 등에 있어서 연속식 공정의 조업보다 비용이 비싸다. 그래서 회분식 공정의 조업에 있어서 경비절감을 위한 많은 연구가 진행 중에 있다. 이러한 연구들 중에서 가장 먼저 연구된 것이 생산계획(production scheduling)이다. 생산계획이란 장치들을 가장 효율적으로 사용할 수 있는 제품의 생산 순서를 찾는 것이다. 이런 최적화 문제에 있어서 많은 목적함수가 있겠으나 그 중에서 최종 제품이 마지막 장치에서의 조업완료시간(completion time)을 의미하는 makespan의 최소화에 대한 연구가 가장 활발히 이루어졌다.

회분식 공정을 조업의 측면에서 크게 두 가지 형태로 나눌 수 있다. 하나는 생산될 제품들의 처리경로가 비슷하거나 동일한 제품들을 생산하는 다품종용 회분식 공정(multi-product batch process)이고, 다른

<sup>†</sup>E-mail : jhjung@yucc.yeungnam.ac.kr

하나의 그 제품들의 경로가 일정치 않아 장치들을 다목적으로 사용하는 다목적용 회분식 공정(multi-purpose batch process)이다. 다목적용 회분식 공정에서는 모든 제품의 처리경로가 일정치 않고 이로 인해 각 장치에서의 제품순서 또한 일정치 않게 된다. 그러므로 다목적용 회분식 공정에 비해 다목적용 회분식 공정은 많은 복잡성을 띠고 있다. 회분식 공정은 그 공정 고유의 특성상 제품의 생산효율이 연속식 공정에 비해 낮다. 이러한 생산성의 저하를 극복하기 위한 많은 연구 중 하나가 각 장치와 장치들 사이에 중간저장조를 설치하여 장치들 사이에서 발생하는 병목현상(bottleneck)을 완화시킴으로써 생산성을 증가시키는 방법이다. 지금까지 연구되어진 중간저장조 방안에는 장치들 사이에 충분히 많은 탱크를 두는 UIS(Unlimited Intermediate Storage) 방안, 전혀 중간저장탱크를 설치하지 않는 NIS(No Intermediate Storage) 방안, 병목현상이 발생할 것으로 인지되는 장치들 사이에 유한한 개수의 저장조를 설치하는 FIS(Finite Intermediate Storage) 방안, 여러 가지 중간저장조 방안을 사용하는 MIS(Mixed Intermediate Storage) 방안, 하나의 중간저장조를 모든 장치들이 공통으로 사용하는 CIS(Common Intermediate Storage) 방안 그리고 화학제품의 가장 큰 특징 중에 하나인 대기시간(holding time)을 가질 경우 제품의 성분이 변하는 공정을 위한 ZW(Zero Wait) 방안 등이 연구되어졌다. 그리고 이러한 중간저장조 방안에 대한 각각의 조업완료시간 또한 많은 논문을 통해 발표되어졌다[1, 3-6, 14-18].

지금까지 이러한 다목적용 회분식 공정에 대한 많은 연구에 비해 다목적용 회분식 공정에 대한 연구는 극히 미미하였다고 볼 수 있다. 그 이유는 앞서 말한 바와 같이 다목적용 공정에 비해 상당히 복잡한 특성을 갖고 있기 때문이다. 이러한 이유로 다목적용 회분식 공정에서의 조업완료시간 결정 알고리즘에 관한 연구는 많으나 다목적 회분식 공정에 대한 조업완료시간을 수식으로 표현한 논문이 거의 없었다. 단지 하나의 대상공정에 적용할 수 있는 조업완료시간들이 발표되었을 뿐이었다[9].

본 논문에서는 이러한 다목적 회분식 공정 중 비순차 다목적 회분식 공정[13]에 대한 조업완료시간을 순환적 수식(recursive formulation)으로 표현하여 그 수식을 여러 개의 대상공정을 통해 검증해 보았다. 본 연구에서는 다목적 회분식 공정의 조업완료시간 결정 수식화를 위해 다목적 회분식 공정이 다목적 회분식 공정에 대해 어떤 다른 특징이 있는가를 살펴보고, 그 특징들을 변수로 도입하였고 그 변수들을 이용하여 조업완료시간 알고리즘을 유도하였다. 본 연구에서 제안한 순환적 알고리즘을 나타내는 컴퓨터 언어로 코딩된 프로그램을 부록 I에 나타내었다.

## 2. 조업완료시간 결정 알고리즘

본 논문에서는 단일 장치로 구성된 다목적 회분식 공정의 조업완료시간 알고리즘을 개발하기 위해 다음과 같은 과정을 택하였다. 첫째, 간단한 하나의 대상공정을 선택해서 그 공정이 다목적 회분식 공정과 다른 점을 분석하고, 둘째, 그 대상공정에 대한 수송시간과 준비시간을 고려하지 않은 조업완료시간 알고리즘을 개발한 다음 그 알고리즘을 단일 장치로 구성된 다른 예제 공정에 적용하였다. 마지막으로 수송시간과 준비시간을 고려한 알고리즘을 유도하였다.

### 2-1. 대상공정

앞서 설명한 바와 같이 회분식 공정의 조업연구들에서는 병목현상을 완화시키기 위해 중간 저장탱크를 사용하였다. 그 방법 이외에도 병목현상이 발생하는 장치에 그 장치를 하나 더 추가함으로써 병목현상을 줄일 수 있다. 본 논문에서는 단일 장치로 구성된 비순차 다목적 회분식 공정을 대상공정으로 하였다. 그리고 다목적 회분식 공정에 있어서 제품의 처리경로에 따라 3가지의 제품으로 분류하였다. 하나는

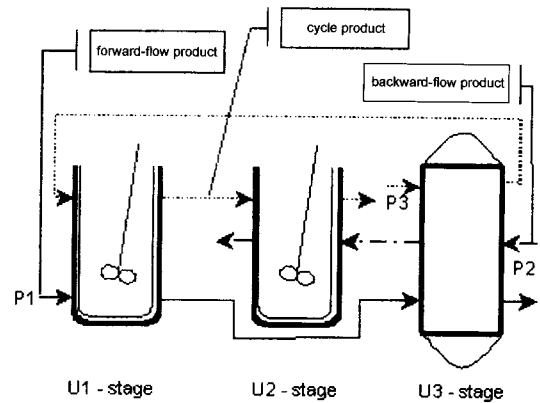


Fig. 1. Classification of products for non-sequential plants.

제품의 처리순서가 장치번호와 같은 제품들을 '정방향 제품(forward-flow product)'으로 정의하였다. 물론 정방향 제품들조차도 모든 장치를 거치지 않을 수도 있다. 이와는 방향이 반대인 제품들을 '역방향 제품(backward-flow product)'으로 정의하였다. 마지막으로 제품의 흐름이 정방향과 역방향을 모두 가지는 제품을 '순환 제품(cycle product)'이라고 분류했다. 위의 내용을 Fig. 1에 나타내었으며 이때  $P_i$ 는 제품을 의미하고  $U_j$ 는 장치번호를 의미한다. 그리고, 화살표는 제품의 경로를 의미한다.

### 2-2. 다목적용 회분식 공정과 다목적 회분식 공정의 조업완료시간결정 비교

다목적용 회분식 공정에서 모든 제품이 모든 장치에서 처리되어지는 반면 다목적 회분식 공정에서는 모든 제품이 모든 장치를 거치지 않을 수도 있다. 또, 단일 장치로 구성된 다목적 회분식 공정에서는 각 제품의 각 장치에서의 순서는 장치의 놓인 순서에 의해 결정되었지만 다목적 회분식 공정에서는 장치의 놓인 순서가 아닌 제품의 경로에 의존한다. 다목적 회분식 공정에 있어서는 모든 제품의 경로가 동일하고 각 장치에서의 처리되는 제품의 순서가 동일하다. 그래서, 다목적 공정의 제품순서를 행렬로 표현하면 1차원 행렬로 표현될 수 있다. 그러나 다목적 회분식 공정에서는 각 장치에서의 제품의 순서를 모두 고려한 즉, 2차원 행렬로 표현되어진다. 이러한 차이점들로부터 다목적 회분식 공정의 특징을 다음의 세 변수들로 표현하였다.  $j$ 장치에서  $k$ 번째로 처리되는 제품을 의미하는  $P_{jk}$ ,  $i$ 제품이  $m$ 번째로 처리되는 장치를 의미하는  $U_{im}$  그리고  $i$ 제품이  $j$ 장치에서의 순서를 의미하는  $O_{ij}$ 가 그 변수들이다. 위의 세 변수를 2차원 행렬로 표시했을 때  $P_{jk}$ 의 요소는 제품,  $U_{im}$ 의 요소는 장치 그리고,  $O_{ij}$ 의 요소는 순서를 나타낸다. 앞서 언급한 바와 같이 다목적 회분식 공정의 제품순서는 1차원 행렬로 다목적 회분식 공정의 제품순서는 2차원 행렬로 표현되어지므로 발생할 수 있는 총 조업순서의 경우의 수에 있어서도 차이점을 나타낸다. 다목적 회분식 공정의 경우에 있어서 총 조업순서 경우의 수는  $N!$ 이고 다목적 회분식 공정의 경우는  $(N_1!) \cdots (N_M!)$ 이다. 여기서  $N$ 은 생산해야할 총 제품수이고,  $N_j$ 는  $j$ 장치에서 처리되는 제품수이다. Fig. 2는 Yamalidou 등[19]에 의해 다루어진 다목적 공정의 예이다. Fig. 3은 Fig. 2의 예제 공정을 위의 세 변수를 사용하여 나타낸 것이고, Fig. 4는 총 조업순서의 수에 있어서 다목적 회분식 공정과 다목적 회분식 공정을 비교한 것이다.

### 2-3. 유도과정

다목적 회분식 공정의 조업완료시간 알고리즘의 개발에 앞서 수송시간과 준비시간을 고려하지 않고 UIS정책을 사용한 단일 장치로 구성된 다목적 회분식 공정의 조업완료시간 알고리즘을 살펴보면 다음

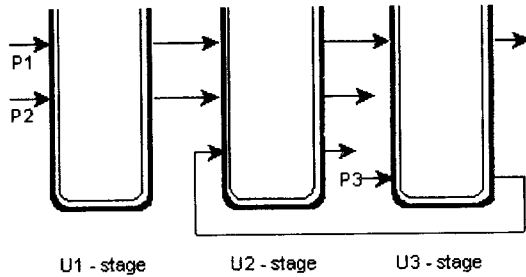


Fig. 2. Example process from Yamalidou et al.[19].

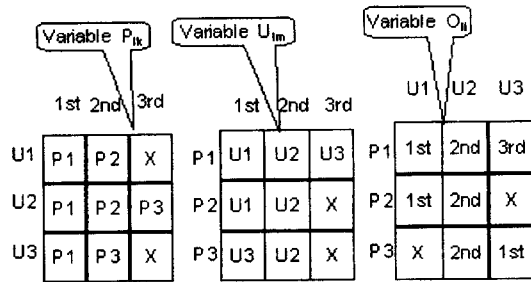


Fig. 3. Representation of three variables( $P_{jk}$ ,  $U_{im}$ ,  $O_{ij}$ ) for example process at Fig. 2.

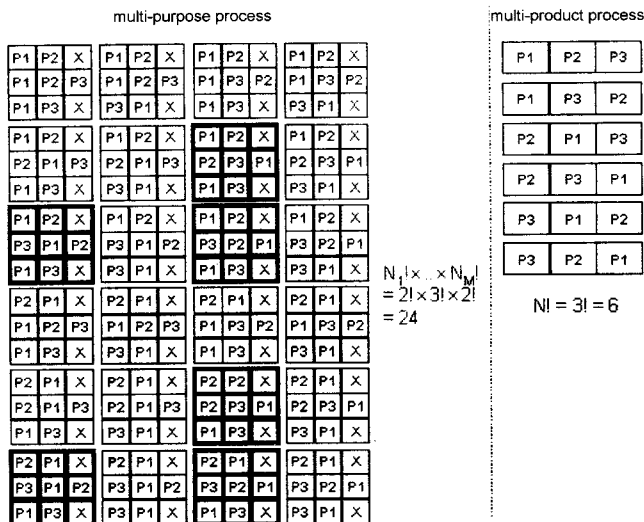


Fig. 4. Difference in total number of operation sequences between multi-product and multi-purpose operation with process at Fig. 2.

과 같다.

$$C_{ij} = \max(C_{i-1,j}, C_{i,j-1}) + t_{ij} \quad (1)$$

이다. 여기서  $i$ 는 제품,  $j$ 는 장치를 의미한다. 또,  $C_{ij}$ 는  $i$ 번째 제품이  $j$ 장치에서의 조업완료시간을 의미하고  $C_{i,j-1}$ 은  $i$ 제품이  $j$ 장치로 전송되기 바로 전 장치에서의 조업완료시간을 의미한다.  $C_{i-1,j}$ 는  $j$ 장치에서의 제품의 처리순서 중  $i$ 제품에 앞서 처리된 제품을 의미하고,  $t_{ij}$ 는  $i$ 제품이  $j$ 장치에서의 처리시간을 나타낸다. 만약  $C_{i,j-1}$ 이  $C_{i-1,j}$ 보다 클 경우는  $j$ 장치가 현재 사용 중이 아니라는 것을 의미한다. 또,  $C_{i,j-1}$ 이  $C_{i-1,j}$ 보다 작을 경우는 비록  $i$ 제품이  $j-1$ 장치에서 작업을 완료하였지만 현재  $j$ 장치가  $i-1$ 제품을 처리 중이라는 것을 의미한다. 즉 다시 말해서 병목현상이 발생하였다는 것을 의미한다. 다목적 회분식 공정에 있어서는 각 장치에서 제품의 순서가 일정하지 않으므로 제품의 순서에 대한 변수를 사용하지 않을 수 없었다. 그래서 식 (1)에서  $i$ 변수와  $j$ 변

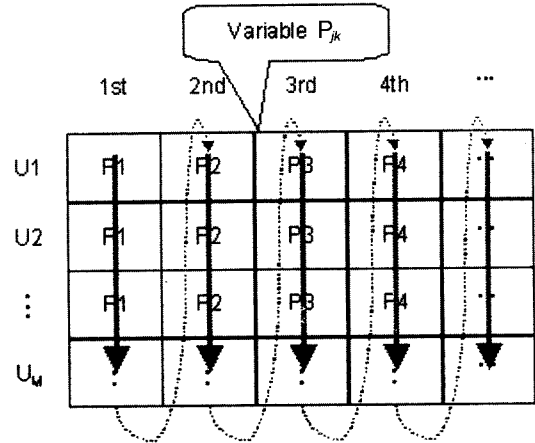


Fig. 5. Handling procedure of recursive formulation.

수 대신 제품을 의미하는  $P_{jk}$  변수와 장치를 의미하는  $U_{im}$  변수를 사용하였다. 그러므로  $C_{ij}$ 는  $C_{P_{jk},j}$ 가 되고,  $C_{i-1,j}$ 는  $C_{P_{jk-1},j}$ 로 됨을 알 수 있다. 또,  $C_{i,j-1}$ 은  $C_{P_{jk},O_{P_{jk},j-1}}$ 이 된다. 여기서  $P_{jk}$ 를  $\alpha$ 로  $P_{j,k-1}$ 을  $\beta$ 로  $O_{P_{jk},j-1}$ 을  $\delta$ 로 치환하면,

$$C_{\alpha j} = \max(C_{\beta j}, C_{\alpha U_{\delta \delta}}) + t_{\alpha j} \quad (2)$$

가 된다. 여기서  $U_{\delta \delta}$ 를  $\gamma$ 로 치환하면

$$C_{\alpha j} = \max(C_{\beta j}, C_{\alpha \gamma}) + t_{\alpha j} \quad (3)$$

가 된다. 결국 식 (3)은  $j$ 와  $k$ 변수로 이루어짐을 알 수 있다. Fig. 5에서처럼 식 (3)을 화살표 방향으로 진행하면서 적용한다. 그러나 다 품종 회분식 공정의 알고리즘과의 가장 큰 차이점은 한 번의 반복수행과정으로 makespan을 계산할 수 없다는 것이다. 왜냐하면, 다목적 공정의 가장 큰 특징인 제품의 처리순서가 일정하지 않다는 이유 때문이다. 그래서 몇 번의 반복을 거쳐야 makespan을 계산할 수 있다. 물론 식 (3)의  $C_{ij}$ 변수에 있어서  $C_{0,j}$ ,  $C_{i,0}$ 는 0의 값을 가진다.

## 2.4. 컴퓨터 프로그램 코딩

식 (3)을 컴퓨터 프로그램화하기 위해서는 다품종 회분식 공정보다 더 복잡한 과정을 필요로 한다. 왜냐하면, 한 번의 반복으로 makespan을 계산할 수 없기 때문이다. 본 논문에서는 식 (3)을 코딩하기 위해 다음의 3단계를 선택했다.

첫째, 대상공정이 주어지면 그 공정에 해당하는  $P_{jk}$ ,  $U_{im}$ ,  $O_{ij}$ ,  $t_{ij}$  값을 입력한다

둘째, 한 번의 반복 수행 작업에서 계산할 수 있는 값들을 찾아내고, 찾은 값들이 다음의 반복 수행때 변화되지 않게 고정시킨다.

셋째, 적절한 종료조건(TC: Terminal Condition)을 선택한다. 본 논문에서는  $t_{ij}$  변수의 항목 중 0의 값을 가지지 않는 변수의 개수 즉 각 장치에서 처리해야할 제품수의 합을 종료조건으로 선택했다.

$$TC = \sum_{j=1}^M N_j \quad (4)$$

우리는 위의 과정을 Fig. 6의 Flow-Chart에 나타내었고 Turbo C++ [MS-DOS Version]을 사용하여 식 (3)을 코딩하였다. 부록 I의 Source 화일의 대상공정은 Fig. 2의 공정이다.

부록 I의 Source 화일을 compile하면서 알게 된 다목적 회분식 공정의 가장 큰 특징은 이론상으로도 불가능한 조업순서가 존재한다는 것이었다. 불가능한 조업순서는 제품의 흐름방향이 다른 제품들이 처리

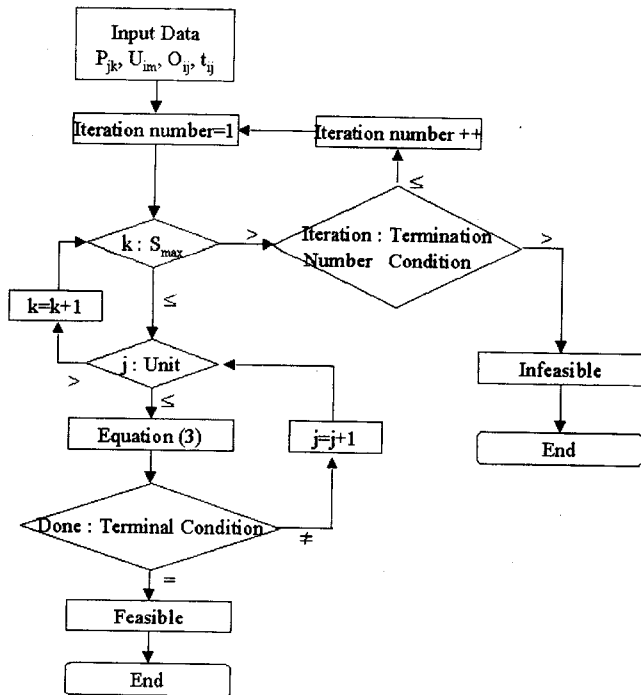


Fig. 6. Flowchart of recursive formulation.

되는 장치에서는 그 제품들의 처리순서가 같지 않은 경우에 발생했다. 그러나 이런 경우에도 그 장치에서 그 제품이 처음 처리되는 장치일 경우에는 예외가 된다. Fig. 4에서 다목적 회분식 공정의 조업순서 중에서 짙게 테두리된 것들이 불가능한 조업순서들이다.

### 2-5. 수송시간과 준비시간을 고려한 조업완료시간 알고리즘

본 논문에서는 수송시간을  $A_{ip}$ , 준비시간을  $S_{i,i+1,j}$  변수로 나타내었다.  $A_{ip}$ 는 i제품이 p번째 장치에서 p+1번째 장치로 전송할 때의 시간을 의미하고,  $S_{i,i+1,j}$ 는 j장치에서 i제품과 i+1제품사이의 준비시간이다. 여기서  $p=0,1,\dots,M$ 이다. 식 (3)으로부터 다음의 식 (4)를 유도하였다.

$$C_{\alpha,j} = \max(C_{\beta,j} + S_{\beta,\alpha,j} + A_{\alpha,\beta}, C_{\alpha,\gamma} + t_{\alpha,j} + A_{\alpha,\delta+1}) \quad (5)$$

위에서 제시한 순서도에서 입력 값에  $A_{ip}$ 와  $S_{i,i+1,j}$  변수를 추가하고 식 (3) 대신 식 (5)를 사용함으로써 수송시간과 준비시간을 고려한 순서도를 얻을 수 있다. 또,  $A_{ip}$ 와  $S_{i,i+1,j}$  변수를 추가함으로써 수송시간과 준비시간을 고려한 컴퓨터 프로그램을 얻을 수 있다.

## 3. 적용 예제

본 논문에서는 제시된 다목적 회분식 공정의 조업완료시간 결정 알고리즘을 검증하기 위해 처리시간, 수송시간 및 준비시간들이 무작위로 생성된 3개의 예제 공정 데이터를 적용하였고, 제한한 조업완료시간 알고리즘을 적용하여 얻은 결과값을 Turbo C++의 Graphic을 이용하여 Gantt Chart로 나타내었다.

### 3-1. 예제 1

Fig. 2는 Yamalidou[19]의 논문에서 다루어진 다목적 회분식 공정의 예로서 제품수 3개와 장치수 3개로 이루어진 공정이다. Fig. 7은 주어진 조업순서( $P_{jk}$ )들에 따른 조업완료시간( $C_{ij}$ )과 Gantt Chart를 나타낸 것이다. 그림에서  $t_{ij}$ 는 처리시간을 나타낸다.

### 3-2. 예제 2

화학공학 제37권 제3호 1999년 6월

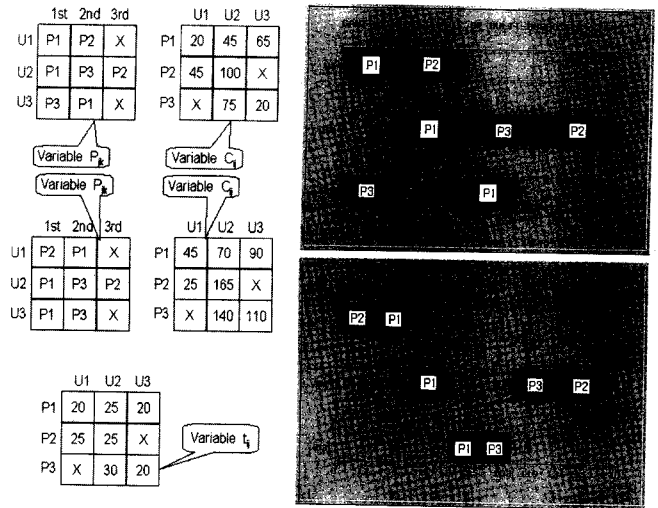


Fig. 7. Results of completion times calculation for example process 1.

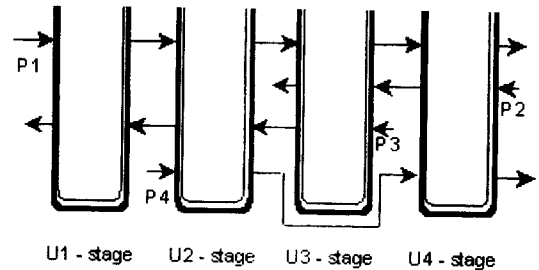


Fig. 8. Schematic diagram of example process 2.

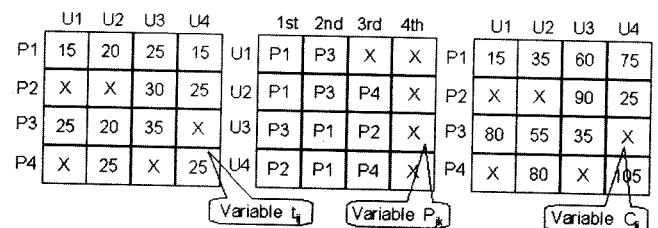
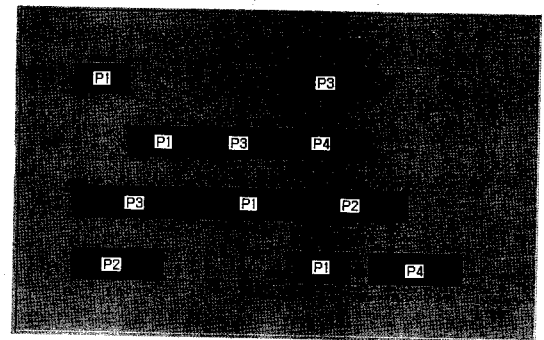


Fig. 9. Results of completion times calculation for example process 2.

Fig. 8은 임의로 만든 비순차 다목적 회분식 공정의 예이고, 제품수 4개와 장치수 4개로 이루어진 공정이다. 조업완료시간의 계산결과와 Gantt Chart를 Fig. 9에 나타내었다.

### 3-3. 예제 3

Fig. 10은 제품수 6개와 장치수 3개를 갖는 임의의 다목적 회분식

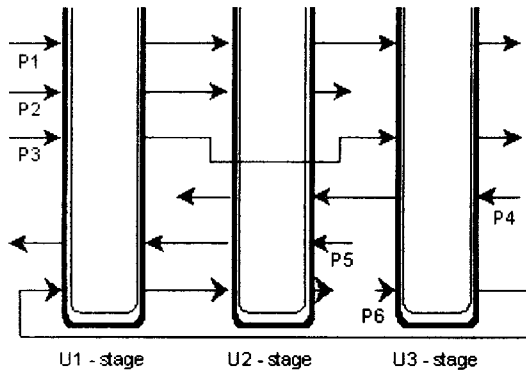


Fig. 10. Schematic diagram of example process 3.

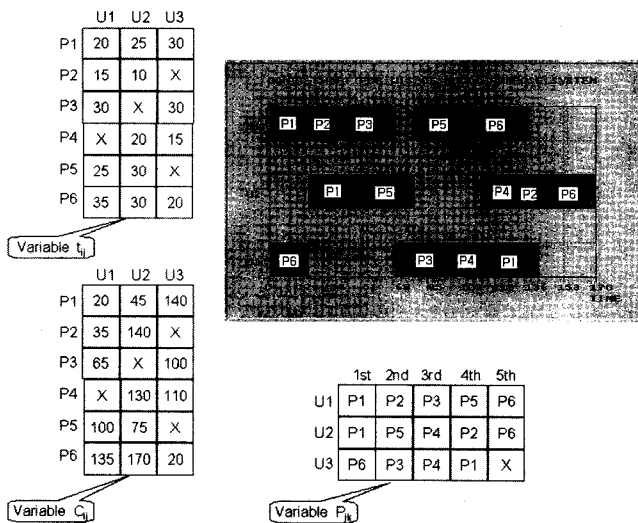


Fig. 11. Results of completion times calculation for example process 3.

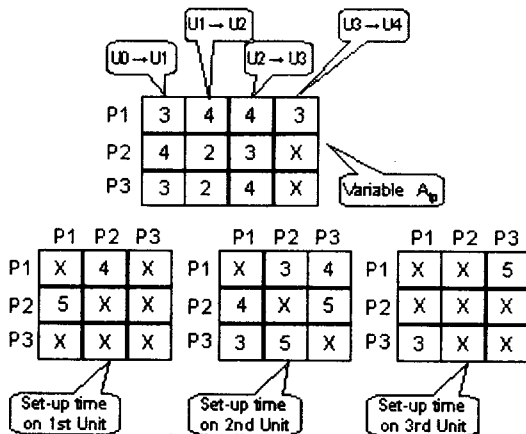


Fig. 12. Transportation times and set-up times of example process 1.

공정의 예이고 정방향, 역방향, 순환 제품들을 모두 갖는다. Fig. 11에는 그 결과값들을 나타내었다.

### 3-4. 예제 4

본 예제에서는 예제 1공정에 수송시간과 준비시간을 고려한 문제를 풀이해 보았다. Fig. 2의 공정에 대한 수송시간  $A_{ij}$  변수와 순서의존 준비시간  $S_{i,i+1,j}$  변수의 값을 Fig. 12에 나타내었다. Fig. 12에서  $U_0$ 은 원료 탱크,  $U_M$ 은 최종 제품 저장조를 의미한다. Fig. 13에서는 Fig. 2의 공정에 대한  $t_{ij}$ ,  $S_{i,i+1,j}$ ,  $A_{ij}$  변수의 값과 식 (4)를 적용하였을 때의 결

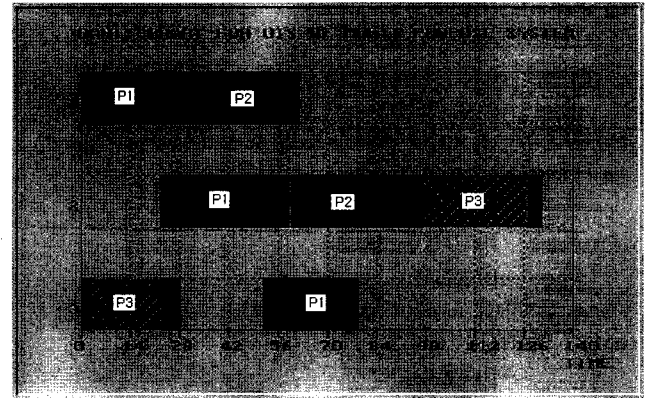


Fig. 13. Result of calculation for example process 1 with consideration of transfer times and set-up times.

과값과 Gantt Chart를 나타내었다.

Fig. 7, 9, 11에서 가로축은 시간, 세로축은 장치번호, 같은 무늬의 사각형들은 같은 제품을 의미한다. 그림에서 나타난 것처럼 다품종 회분식 공정들과는 달리 좌측 하단과 우측 상단에 생기던 Slack Time 이 줄어든 것을 알 수 있다.

## 4. 결 론

본 논문에서는 다품종 회분식 공정에 비해 상당히 복잡한 특성을 갖는 비순차 다목적 회분공정의 조업완료시간 알고리즘을 순환적 수식으로 표현하였고 컴퓨터 언어로 코딩하였다. 식 (3)에서는 수송시간과 준비시간을 고려하지 않았고, 식 (4)에서는 이들을 고려한 순환적 알고리즘을 유도하였다. 또, 제시된 알고리즘을 검증하기 위해 난수발생기를 사용하여 각 변수들-처리시간, 수송시간, 준비시간-의 값을 생성한 몇 가지 문제를 만들어 적용해 보았으며 그 결과 제시된 알고리즘은 완벽히 다목적 공정의 조업완료시간을 얻어냈다. 본 논문에서 제시한 예제 공정 외에 크기가 다른 여러 문제들에 대해서도 식 (3)과 (5)를 적용해 보았다. 그 문제들 중 크기가 가장 큰 제품수 10, 장치수 8인 문제의 경우에도 조업완료시간에 대한 계산 시간은 10초 이내에 결과를 얻을 수 있었다. 또, 주어진 종료조건 내에서 makespan을 계산하지 못할 경우, 본 논문에서는 그 제품순서를 불가능한 조업순서로 규정하여 계산을 종료시켰다. 그러나 본 알고리즘의 보완되어야 할 점은 종료조건에 해당하는 반복횟수의 최적값을 결정하는 방법을 제시하여 계산시간을 줄이는 것과 더 나아가 불가능한 제품순서를 종료조건없이 자동으로 제거할 방법을 찾는 것 등이 있다.

## 감 사

이 논문은 1997년 한국학술진흥재단의 공모과제(지방대 육성과제) 연구비에 의하여 연구되었습니다. 연구비를 지원해 주신 한국학술진흥재단에 감사드립니다.

### 부 록 I. 다목적 회분공정의 조업완료시간 알고리즘의 Source File

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define N 10
#define M 10
#define max(a,b) ((a)>=(b))?(a):(b)
/*array t[N][M] means processing times that is variable tij */
```

```

int t[N][M]={ {0,0,0,0,},
               {0,25, 25, 20,},
               {0,25, 25, 0,},
               {0, 0, 30, 20,},},};

/*array U[N][M] means product order on units that is U_im */
int U[N][M]={ {0,0,0,0,},
               {0,1,2,3,},
               {0,1,2,0,},
               {0,3,2,0,},},};

/*array O[N][M] means order of products on units that is O_ij */
int O[N][M]={ {0,0,0,0,},
               {0,1,2,3,},
               {0,1,2,0,},
               {0,0,2,1,},},};

/*array P[N][M] means processing path that is P_jk */
int P[N][M]={ {0,0,0,0,},
               {0,1,2,0,},
               {0,1,2,3,},
               {0,1,3,0,},},};

void main( )
{
    static int C[N][M],check[N][M];
    int i,j,k,Product=3,Unit=3,Smax=3,terminal_condition=7,done=0;
    int a,b,c,d;
    for(i=1;i<=Product;i++){
        for(j=1;j<=Unit;j++){
            C[i][j]=-1;
            check[i][j]=-2;
        }
    }

    for(i=1;i<terminal_condition;i++){
        for(k=1;k<=Smax;k++){
            for(j=1;j<=Unit;j++){
                a=P[j][k], b=P[j][k-1], c=O[a][j]-1, d=U[a][c];
                if(i==0){
                    if(C[b][j] !=-1 && C[a][d] !=-1 && P[j][k] != 0){
                        C[a][j]=max(C[b][j],C[a][d])+t[a][j];
                        check[a][j]=0;
                        done++;
                    }
                }
            }
        }
        if(i!=0){
            if(check[a][j] !=0 && C[b][j] !=-1 && C[a][d] !=-1 &&
                P[j][k] !=0){
                C[a][j]=max(C[b][j],C[a][d])+t[a][j];
                check[a][j]=0;
                done++;
            }
        }
    }
    if(done==terminal_condition){
        printf(" Number of iteration is %d\n\n",i);
        for(i=1;i<=Product;i++){
            for(j=1;j<=Unit;j++){
                printf("%5d",C[i][j]); }
            printf("\n"); }
        return;
    }
}

```

```

} } }

```

```

if(done!=terminal_condition){
    printf("\n\n\n\tThis case is impossible !!!");
    return; }
}

```

## 사용기호

- $A_{ip}$  : transportation time when  $i$  product is transferred from  $p$ th unit to  $p+1$ th unit  
 $C_{ij}$  : completion time of  $i$  product in  $j$  unit  
 $O_{ij}$  : order of  $i$  product in  $j$  unit  
 $P_{jk}$  : product which is processed  $k$ th in  $j$  unit  
 $S_{i,i+1,j}$  : set-up time which is required for preparing  $i+1$  product after  $i$  product in  $j$  unit  
 $t_{ij}$  : processing time of  $i$  product in  $j$  unit  
 $TC$  : terminal condition  
 $U_{im}$  : unit at which  $i$  product go through  $m$ th

## 참고문헌

1. Rajagopalan, D. and Karimi, I. A.: *Comput. Chem. Engng.*, **14**, 49(1990).
2. Kondili, E., Pentelides, C. C. and Sargent, R. W. H.: *Comput. Chem. Engng.*, **17**, 229(1993).
3. Ku, H. M. and Karimi, I.: *Comput. Chem. Engng.*, **14**, 49(1990).
4. Lee, I. B., Jung, J. H., Yang, D. R. and Chang, K. S.: FOCAP0 '93, U.S.A.(1993).
5. Jung, J. H., Lee, H. and Lee, I. B.: *Comput. Chem. Engng.*, **20**, 845 (1996).
6. Jung, J. H., Lee, H., Yang, D. R. and Lee, I. B.: *Comput. Chem. Engng.*, **18**, 537(1994).
7. Gaells, M., Espuna, A. and Puijaner, L.: *Comput. Chem. Engng.*, **20**, s1137(1996).
8. Kim, M., Jung, J. H. and Lee, I.: *Comput. Chem. Engng.*, **20**, s1149 (1996).
9. Raimon, G., Antonio, E. and Luis, P.: *Comput. Chem. Engng.*, **20**, s1143(1996).
10. Hasebe, S. and Hashimoto, I.: Proc. 4th Intl. Symp. on Proc. Systems Engng, Canada(1991).
11. Johnson, S. M.: *Naval Research Logistics Quarterly*, **1**, 1(1954).
12. Niwa, T.: Proc. Fourth Intl. Symp. on Proc. Systems Engng, Montebello, Canada(1991).
13. Voudouris, V. T. and Grossmann, I. E.: *Comput. Chem. Engng.*, **20**, 1335(1996).
14. Wiede Jr., W., Kuriyan, K. and Reklaitis, G. V.: *Comput. Chem. Engng.*, **11**, 337(1987).
15. Wiede Jr., W. and Reklaitis, G. V.: *Comput. Chem. Engng.*, **11**, 345(1987).
16. Wiede Jr., W. and Reklaitis, G. V.: *Comput. Chem. Engng.*, **11**, 357 (1987).
17. Kim, H. J., Jung, J. H., Kim, M. and Lee, I. B.: *Korean J. Chem. Engng.*, **14**, 225(1997).
18. Ha, J. K., Lee, B. S., Lee, I. B. and Lee, E. S.: *HWAHAK KONG-HAK*, **36**, 813(1998).
19. Yamalidou, E. C., Patsidou, E. P. and Kantor, J. C.: *Comput. Chem. Engng.*, **14**, 281(1990).