

## 유연한 중간저장탱크 사용이 가능한 공유된 중간 저장탱크 방안에서의 다품종 회분식 공정 생산계획

정재학 · 이인범 · 양대륙 · 장근수

공정산업의 지능자동화연구센터/포항공과대학 화학공학과  
(1993년 5월 3일 접수, 1993년 12월 21일 채택)

### Scheduling for Multi-product Batch Processes with Common Intermediate Storage Policy as a Flexible Intermediate Storage Policy

Jae Hak Jung, In-Beum Lee, Dae Ryook Yang and Kun Soo Chang

Automation Research Center, Pohang University of Science and Technology, P.O. Box 125, Pohang 790-600  
(Received 3 May 1993; accepted 21 December 1993)

#### 요 약

회분식 다품종용 생산공정에서 mixed intermediate storage(MIS) 방안보다 더 효율적인 중간저장탱크 방안으로써 common intermediate storage(CIS) 방안을 제안한다. CIS 방안은 MIS 방안의 한 블록으로써 연구되어진 shared storage 시스템의 전 공정으로의 확장으로 볼 수 있으며, 장치 사이의 변화될 수 있는 저장탱크 방안을 표현해 줄 수도 있어 각 장치사이의 변화되는 저장탱크 방안에 유연하게 대처할 수도 있다. 또한 CIS 방안의 새롭고 간결한 완료시간 결정 알고리즘을 개발하였고, 이를 이용하여 CIS 방안에 대한 2가지 생산계획법을 적용해 보았다. 첫째 10개 이하의 제품의 문제만을 다룰 수 있지만 최적값을 항상 제공하는 combinatorial search 알고리즘이며, 둘째 항상 최적치를 얻지는 못하나 제품 11개 이상의 큰 문제에서 매우 효율적인 simulated annealing 방법이다. 다품종 공정에서 각 생산 사이클에 따라 생산해야 할 제품의 종류가 바뀌는 경우, 그에 따라 각 장치 사이에서 저장탱크사용 방안이 변화되는 경우가 있다. 이 때 CIS 방안과 적당한 on/off 밸브와 펌프 등을 이용해 변화하는 저장탱크 방안을 표현해 줄 수 있다. 또한 CIS 방안의 단점인 조업의 복잡성과 그에 따른 안전성문제를 극복하기 위한 safety interlock을 구성하였으며, 기초적인 회분식 다품종 생산공정의 자동화 모델의 한 부분을 제시하였다.

**Abstract**—In this study, we propose the common intermediate storage(CIS) policy which is more efficient than mixed intermediate storage(MIS) policy. The CIS policy is a form of shared storage system, which is extended to overall processes and can be applied to flexible inter-stage intermediate storage policies. Using this strategy, we developed a new completion time algorithm and applied it to two kinds of optimal or near optimal scheduling method; the combinatorial search and the simulated annealing method. We also extended the CIS policy for flexible operation of intermediate storage policy with proper on/off electronic valves and pumps. It thus can be accepted as a basic form of FMS(Flexible Manufacturing System) for operating various storage policies. Finally we suggested the PLC logic of safety interlock to complement the difficult application of CIS policy, i.e., complication of operation and safety, resulting in a part of basic batch process automation mode.

## 1. 서 론

최근 산업은 양적 생산체제에서 질적 생산체제로의 전환이 두드러지고 있다. 화학공업 역시 과거의 연속식 대량 생산체제 중심에서 서서히 다품종, 소량 생산 및 고부가가치 고품위 생산의 반연속, 회분식 생산체제가 각광을 받기 시작하였다. 단일 제품의 수명이 짧고 소비자의 취향도 다양하여 고품질 우선의 소비 패턴이 나타나게 되었고 여기에 대응하기 위한 비교적 규모가 작은 회분식 공정의 도입은 계속 늘어날 것으로 보여진다. 이러한 회분식 공정에서 발생하는 여러 가지 문제가 연구되기 시작하였는데, 그 중 경제성에 큰 영향을 끼치는 분야가 다품종 생산공정의 제품 생산 계획이다.

화학공학에서의 이러한 연구는 1980년대 초에 시작되어 1980년대 중반부터 매우 활기를 띄어 왔다. 이러한 문제는 주로 다수의 주어진 생산품들을 제조하는데 드는 총 조업시간을 최소화할 수 있는 제품의 조업순서를 결정하는데 있으며 생산 설비의 유형에 따라 다목적(multi-purpose) 혹은 다제품(multi-product) 조업으로 나누어 생각할 수 있다. 그 중 본 연구에서는 제품을 만드는 설비의 순서가 일정하게 주어진 다제품 생산 계획에 관한 연구에 국한한다.

이들 다제품 생산 계획에 관한 연구는 매우 활발하여 Reklaitis[1, 2]와 Ku 등[3]이 여러 차례 review 하기도 하였다. 이들 review paper에 따르면 화학공학의 회분식 조업에서 그 특성에 따라 중간저장탱크를 운용하는 경우가 많은데, 공정 특성상 그 사용 유형을 UIS(unlimited intermediate storage), NIS(no intermediate storage), FIS(finite intermediate storage), ZW(zero wait) 등으로 나눈다.

그외에 기본적인 4가지 유형의 중간저장탱크 운용방법이 적용되기 어려운 현장 문제가 발생하여, 이들을 표현해 주는 몇 가지 다른 사용 방안이 제안되었다. 최근 Wiede JR.와 Reklaitis[4]는 한 시스템에서 이들 기본적인 4가지 중간저장탱크 운용 방안을 조합적으로 사용하여야 하는 경우를 제안하여 이를 MIS(mixed intermediate storage)라 하였고 Ku와 Karimi[5]는 FIS의 중간저장탱크 운용 방안에서 중간저장탱크가 사용중일 경우 unit가 중간제품을 갖고 있을 것인가(holding), unit가 중간제품을 갖고 있지 못할 것인가에 따라 두 가지 FIS의 세부적 운용 방안으로, NIS/FIS와 ZW/FIS를 제안하였다. 또한 FIS의 운용 방안을 응용하여 중간저장탱크를 공유하는 shared storage block을 제안하였다.

중간저장탱크를 여러 개의 unit가 공유하는 경우 또 하나의 복잡한 문제가 생길 수 있다. 즉 여러 가지 중간

제품들이 어떤 시간대에 동시 다발적으로 중간저장탱크의 사용을 요구하게 될 것이다. 이 경우 어떤 제품에 중간저장탱크의 사용권을 줄 것인가를 결정해야 한다. 조금이라도 먼저 중간저장탱크의 사용을 요구하는 제품에 중간저장탱크의 사용권을 주는 경우를 사건 중심(event priority)의 생산 계획이라 하며, 다중 제품들의 조업 순서가 가장 빠른 제품에 우선권을 주는 경우를 제품 중심(product priority)의 생산 계획이라 부른다. Ku와 Karimi[5]는 대다수의 경우 조업 순서가 빠른 제품에 우선권을 주어야 그 뒤에 따라오는 제품들에 조업 시간 지연의 영향을 덜 주게 되므로 전자보다 후자의 결과가 더 우수함을 밝힌 바 있다.

본 연구에서는 우선 Ku와 Karimi의 중간저장탱크의 공유화(shared storage)를 전공정에 확장 적용시켜 CIS(common intermediate storage) 방안으로 명명하였고, CIS 방안이 MIS 방안보다 더 우수함을 보였다. 또한 CIS 방안에서 사건 중심(event priority) 저장탱크 운용 방안이 제품 중심(product priority) 저장 탱크 운용 방안 보다 더 우수한 경우를 보여 Ku와 Karimi가 제시한 바와 달리 반드시 제품 중심의 저장탱크 운용 방안이 더 우수한 경우가 대다수가 아님을 보였다.

CIS 방안의 경우 다품종 회분식 조업에 대한 각 unit에서의 새롭고 보다 간략한 조업 완료 시간(completion time)을 계산하는 알고리즘을 제안하였으며, 이를 이용한 최적, 준 최적의 생산 계획법으로 combinatorial search법 및 simulated annealing법을 제시하였다. 또한 CIS 방안과 PLC(programmable logic controller)와 on/off 밸브 등을 이용하여 다른 여러 종류의 저장탱크 방안(예, MIS 방안)의 표현이 가능함을 보였는데, 이것은 생산될 제품이 바뀔 때 따라 같이 바뀌어야 하는 저장탱크 운용 방안을 유연성있게 대처할 수 있으므로 여러 가지 저장탱크의 운용 방안에 대한 FMS(flexible manufacturing system)의 기초적인 모델로 제시될 수 있다. 마지막으로 CIS의 단점인 조업의 복잡성과 안전성을 보완하기 위해 interlock블럭을 따로 두었으며, 이들을 모두 조합하여 기초적인 단계의 회분식 공정 자동화(batch process automation)의 한 모델로 제시하였다.

## 2. CIS(Common Intermediate Storage) 시스템

한정된 수의 저장탱크를 다수의 unit가 공유하는 경우를 shared storage 시스템이라 할 때, 시스템의 한 특수 예로서 Fig. 1에서와 같이 모든 unit가 유한 개의 저장탱크를 공유하는 시스템을 생각할 수 있다. 이것을

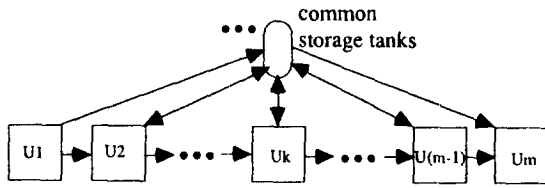


Fig. 1. CIS intermediate storage system.

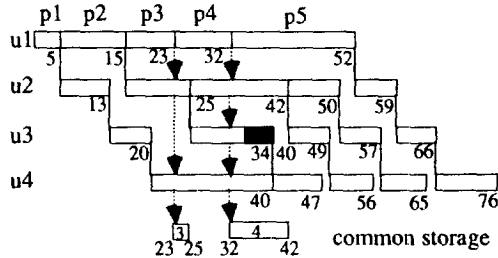


Fig. 2. Event priority scheduling for CIS policy.

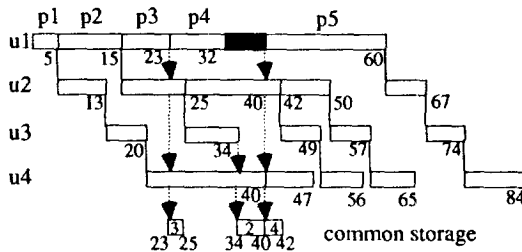


Fig. 3. Product priority scheduling for CIS policy.

CIS라 명명하였으며, CIS 방안의 복잡성은 앞서 언급했듯이 중간저장탱크 운영의 세부적인 경우가 생기는 것으로, 이를 설명하기 위해 사건 중심의 저장탱크 운영 방안과 제품중심의 운영 방안의 예를 5개의 제품이 4개의 unit를 거치는 공정에서 주어진 조업순서가 1-2-3-4-5일 때의 경우를 다음 Fig. 2, 3에 나타내었다. Fig. 2는 사건중심의 저장탱크 운영 방안에 대한 최소완료시간 Gantt chart이고 Fig. 3은 제품중심의 저장탱크 운영 방안에 대한 최소완료시간 Gantt chart이다. Fig. 2와 3에서는 제품중심의 저장탱크 운영 방안이 더 우수한 결과를 보이고 있으며, 결국 제품중심의 저장탱크 운영 방안이든, 사건중심의 저장탱크 운영 방안이든, 주어진 문제의 특성상 어떤 것이 뒷공정에 영향을 더 많이 끼칠 것인가에 따라 그 성능이 좌우된다.

CIS 방안과 최근 Wiede JR.과 Reklaitis[4]가 제시한 MIS 방안과의 성능을 살펴보기 위하여 4개의 제품이 5개의 unit를 거치는 문제에서 주어진 제품의 순서가

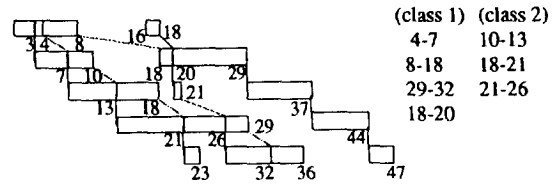


Fig. 4. MIS policy with class 1(unit 1, 4) and class 2(unit 2, 3) storages.

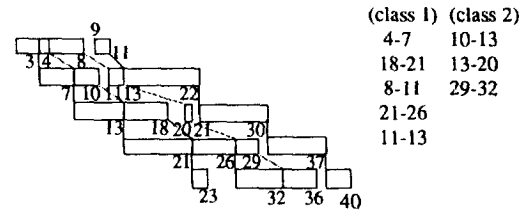


Fig. 5. MIS policy with class 1(unit 1, 3) and class 2(unit 2, 4) storages.

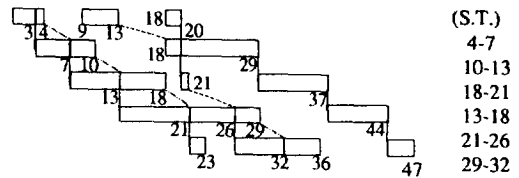


Fig. 6. CIS policy with one common storage.

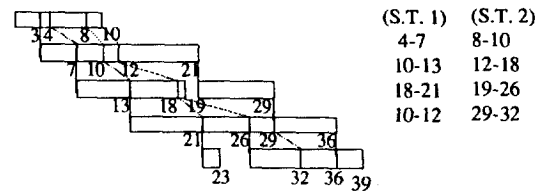


Fig. 7. CIS policy with two common storages.

1-2-3-4일 때에 대해 살펴보았는데, Fig. 4는 unit 1과 4가 class 1의 저장탱크를, unit 2와 3이 class 2의 저장탱크를 사용하는 경우의 최소완료시간의 Gantt chart이고, Fig. 5는 unit 1과 3이 class 1의 저장탱크를, unit 2와 4가 class 2의 저장탱크를 사용하는 경우의 MIS 방안의 한 예이다. 또한 Fig. 6은 같은 문제에서 한 개의 중간저장탱크 공유 시스템이며, Fig. 7은 두 개의 중간저장탱크 공유 시스템이다.

Wiede JR.과 Reklaitis[4]가 제시한 방법과의 비교를 위해 여기서는 그들이 사용한, unit에서 holding을 허용하지 않는 ZW방식으로 생산 계획 공정시간을 비교

하였다. CIS 방안의 경우 저장탱크 한 개만 사용하여도 MIS 시스템의 Fig. 4와 같은 결과가 도출되며, 두 개의 중간저장탱크의 공유시스템인 Fig. 7의 경우 가장 우수한 결과가 도출됨을 알 수 있다. 이 문제의 경우 더 이상의 저장탱크 공유는 무의미하여 2개이상의 저장탱크가 필요치 않게 된다. 실제산업의 설비에서 중간저장탱크를 그다지 많이 두지 않으며, 생산 계획을 위해 무리한 중간저장탱크의 설치가 어려운 실정이므로, 중간저장탱크의 수를 줄일 수 있는 생산계획법의 개발이 필요하게 된다. 이런 측면에서 볼 때 CIS 방안은 최소한의 투자비로 최대한의 효과를 얻을 수 있는 시스템이라 할 수 있다.

### 3. CIS 방안을 위한 조업 완료시간계산 알고리즘 및 생산계획

#### 3-1. CIS 방안을 위한 조업완료시간

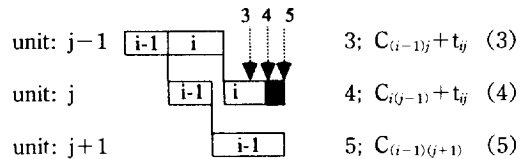
CIS 방안을 사용할 경우 복잡해지는 것은 공유된 중간저장탱크의 운용 방안이 다양해지는 것이다. 즉 제품우선의 방법인가 사건우선의 방법인가에 따라 그 생산계획이 달라짐을 앞서 보았다. 기존의 FIS 방안의 경우 Ku와 Karimi[11]가 매우 간략한 방법으로 조업완료시간을 한 수학적 회귀공식(recursive formula)로써 표현하였고, 이를 MILP로 풀이하여 최적생산계획의 해답을 제시하였다. 그러나 CIS 방안은 특정 unit가 주어진 중간저장탱크를 독점하지 못하고, 많은 unit가 공유함으로써 간단한 수학적 회귀공식을 이끌어 내기 어려울 뿐 아니라, 그렇기때문에 MILP형의 단순한 수식표현이 매우 어려워진다. 결국 Ku와 Karimi[5]는 5가지 경우에 대한 조사와 추가적으로 storage efficiency check를 통해 조업시간을 얻어내려 하였다. 본 연구에서는 한 스텝으로 완료시간을 결정하는 완전한 수학적 회귀공식은 아니지만 storage efficiency check가 필요 없는, 사용중인 중간저장탱크의 사용시간대 비교에 뒤따르는 보다 간략한 수학적 회귀공식표현을 얻어내었다. 또한 Ku와 Karimi[5]의 방법으로는 제품우선(product priority)의 운영방안에 대한 조업완료시간밖에 얻을 수 없지만 본 연구에서 구한 방법은 계산의 순서만 달리 해줌으로써 제품우선(product priority)의 운영 방안 및 사건우선(event priority)의 운영방안에 대한 조업완료시간을 모두 얻을 수 있는 장점이 있다.

중간저장탱크의 설치투자비를 고려하지 않는다면, UIS 방안이 가장 우수한 결과를 얻을 것이며 NIS나 ZW처럼 중간저장탱크를 사용하지 않은 방법이 가장 좋지 않은 결과를 얻게 된다. FIS 방안의 경우는 그 중간쯤의 결과가 도출되는데, CIS 방안에서는 FIS 방안의 중간저

장탱크 투자비와 같거나 적은 투자로써 FIS 방안보다는 좋은 결과를 얻자는 것이다. 결국 다음 식 (1)(UIS 방안)의 경우와 식 (2)(NIS 방안)의 경우 사이의 결과가 얻어질 것이다. UIS 방안은 식 (3)과 (4)의 비교에 의한 최대값이  $C_{ij}$ 가 되며 NIS 방안은 식 (3), (4) 그리고 (5)의 값중 최대값이  $C_{ij}$ 가 된다.

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

$$C_{ij}(\text{NIS}) = \max[C_{(i-1)j}, C_{i(j-1)}, C_{(i-1)(j+1)} - t_{ij}] + t_{ij} \quad (2)$$



위 그림의 예는 식 (5)가 식 (3), (4)에 비해 큰 경우를 나타내었다. CIS 방안의 경우, 식 (4)와 (5) 사이에 저장탱크가 비어있다면, 식 (5)보다 더 좋은 값을 얻을 수 있다. 그러므로 우선 여기서 저장탱크의 사용시간에 대한 정보가 필요하다. 여기서 새로운 변수  $SS(l, k)$ ,  $SE(l, k)$ 를 각각  $l$  제품이  $k$  장치후에 common storage 사용을 시작하는 시간과 끝 마치는 시간으로 정의하고, 변수  $SE^*(l, k)$  ( $l=1, 2, \dots, N$ ,  $k=1, 2, \dots, M$ )는  $SS(l, k) < C_{(i-1)(j-1)}$  및  $C_{ij}(\text{UIS}) < SE(l, k)$ 를 만족하는 모든  $[SS(l, k), SE(l, k)]$  ( $l=1, 2, \dots, N$ ,  $k=1, 2, \dots, M$ )들에서의  $SE(l, k)$ 들을 말한다. 만약 위 조건을 만족하는  $SE(l, k)$ 가 하나도 없으면  $SE^*(l, k)$ 는 0으로 둔다. 저장탱크 사용에 관한 초기치는  $SS(l, k) = SE(l, k) = 0$  (for  $l=1, \dots, N$ ,  $k=1, \dots, M$ )라고 둔다.

Case 1) 만약  $C_{ij}(\text{UIS})$  값이  $E_{(i-1)(j+1)}$  보다 크다면 CIS 시스템의  $j$  unit에서의  $i$  제품 완료시간  $[C_{ij}(\text{CIS})]$ 은  $C_{ij}(\text{UIS})$ 와 같다.

Case 2) 그렇지 않으면 저장탱크 사용가능여부를 검토하여야 한다. 이 때 발생하는 상태는 다음 두 가지로 요약된다:

Case 2-1) 만약 공유된 저장탱크중 하나이상  $j$  unit에서 조업을 마친  $i$  제품을 위해  $C_{ij}(\text{UIS})$ 로부터  $C_{ij}(\text{NIS})$  시간동안 사용 가능하다면  $C_{ij}(\text{CIS})$ 는  $C_{ij}(\text{UIS})$ 와 같다.

Case 2-2) 만약 모든 공유된 저장탱크가  $j$  unit에서 조업을 마친  $i$  제품을 위해  $C_{ij}(\text{UIS})$ 로부터  $C_{ij}(\text{NIS})$  시간동안 유용하지 않다면  $C_{ij}(\text{CIS})$ 는  $C_{ij}(\text{NIS})$ 와 같다.

Case 2-3) 만약 공유된 저장탱크중 하나이상  $j$  unit에서 조업을 마친  $i$  제품을 위해  $C_{ij}(\text{UIS})$ 로부터  $C_{ij}$

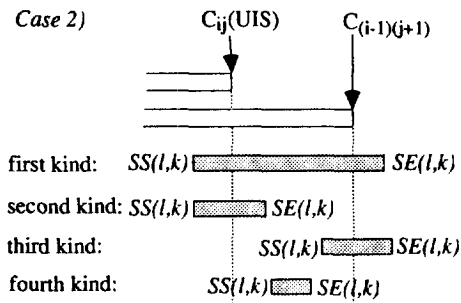


Fig. 8. Four kinds of storage using pattern of one common storage without the transfer and set-up times.

(NIS) 시간동안 부분적으로 사용 가능하다면 다음 두 가지 조건을 검토해야 한다. 1) 그 저장탱크의 사용 가능해지기 시작한 시각[i.e.,  $SE(i,k)$ ] 이  $C_{ij}(NIS)$ 보다 빠른가? 2) 그 저장탱크의 유용시간이  $C_{ij}(NIS)$ 까지 지속되는가? 만약 위 두 조건을 모두 만족한다면  $C_{ij}(CIS)$ 는  $SE(i,k)$ 이 되고, 그렇지 못하면  $C_{ij}(CIS)$ 은  $C_{ij}(NIS)$ 과 같다.

저장탱크의 사용시간들 즉  $SS(i,k)$ 와  $SE(i,k)$ 는 완료시간  $C_{ij}(CIS)$ 를 계산하는 과정에서 정해진다. Case 2의 경우 Fig. 8에 나타난 바와 같이 한 공유된 저장탱크의 사용 유형이 4가지로 나타날 수 있다. 즉 완료시간  $C_{ij}$ 를 계산하는중 Case 2가 될 때 저장탱크가 사용중인지 사용 가능한지를 계산을 통해  $C_{ij}(UIS)$ 와  $C_{ij}(NIS)$  사이에서의 그 저장탱크 상태를 알아봄으로써 정할 수 있다. 저장탱크가 그 시간간격동안 사용되지 않는다면 항상 j unit을 나온 i 제품은 저장탱크를 사용할 수 있다는 것이나, 위에서 말한 4가지중 어느 한 가지 유형으로 저장탱크가 사용중이면  $C_{ij}(CIS)$ 를 결정하는데 또다른 절차가 필요하다. 위에서 나타난 저장탱크의 사용유형 4가지는 모두 다음 두 가지의 조건(condition 1과 condition 2)을 만족한다는 공통점이 있고, 이 두 가지 조건으로 저장탱크가 사용중인지 사용가능함을 간단히 알 수 있다.

$$SS(i,k) < C_{(i-1)(j-1)} \quad (\text{condition 1})$$

$$SE(i,k) > C_{ij}(UIS) \quad (\text{condition 2})$$

만약 조건 1 및 2를 모두 만족한다면 완료시간 결정과정의 i 제품 j unit 스텝은 Case 2의 경우이며, 그렇지 않다면 즉 두 조건중 하나만이라도 위배된다면 그 과정은 Case 1의 경우이다. Case 2에서 저장탱크가  $C_{ij}(NIS)$  이전에 사용 가능한 상태로 되는 2번째(second kind)와 4번째(fourth kind)의 경우는  $C_{ij}(CIS) = SE(i,k)$

가 되고, 그외의 경우는  $C_{ij}(NIS) = C_{ij}(NIS)$ 가 된다. 즉 다음 식 (6)과 같이 표시될 수 있다.

$$C_{ij}(CIS) = \min[C_{ij}(NIS), SE(i,k)] \quad (6)$$

어떤 특별한 경우 Fig. 8의 2번째, 3번째, 4번째 경우의 저장탱크 사용유형은 한 저장탱크내에서  $C_{ij}(NIS)$ 와  $C_{ij}(NIS)$  시간간격내에 여러번 나타날 수도 있다. 결국 식 (6)은 다음 식 (7)과 같이 개선되어야 한다.

$$C_{ij}(CIS) = \min[C_{ij}(NIS), \max\{SE^*(i,k)\}] \quad (7)$$

definition 1)  $SE^*(i,k)$  in eq. (7) is a set of common storage ending(emptying) times[ $SE(i,k)$ s] which satisfy both the condition 1 and condition 2 for i product after j unit.

또한 만약 공유된 중간저장탱크의 갯수 P가 1개 이상 이라면( $P > 1$ ) 모든 저장탱크에 대해 사용여부를 고려해야만 할 것이다. 결국 식 (8)과 같이 개선되어진다.

$$C_{ij}(CIS) = \min[C_{(i-1)(j+1)}, \max\{SE_1^*(i,k)\}, \max\{SE_2^*(i,k)\}, \dots, \max\{SE_P^*(i,k)\}] \quad (8)$$

definition 2)  $SE_p^*(i,k)$  in eq. (8) and (9) is a set of common storage ending(emptying) times[ $SE(i,k)$ s] of the p th( $p=1, 2, \dots, P$ ) common storage which satisfy both the condition 1 and condition 2 for i product after j unit.

위의 식 (8)을 사용하기 위해 저장탱크의 사용시작 및 끝시간에 대한 정보를 미리 가지고 있어야 할 것이다. 결국 i 제품의 j unit에서의 조업완료시간을 결정하는 모든 i, j 스텝에서  $SS_p(i,j)$ 와  $SE_p(i,j)$  ( $i=1, 2, \dots, N, J=1, 2, \dots, M$ )를 같이 계산하여야 한다.  $SS_p(i,j)$ 와  $SE_p(i,j)$ 를 계산하는 알고리즘은 다음과 같다.

For given i and j:

If( $C_{ij}(UIS) < C_{(i-1)(j+1)}$ ) then

Do(for every introduced intermediate storages  $p(p=1, 2, \dots, P)$

If(there is no [ $SS_p(i,k), SE_p(i,k)$ ] to satisfy the condition

$SS_p(i,k) < C_{(i-1)(j+1)}$  and  $SE_p(i,k) > C_{ij}(UIS)$ ) then

$SS_p(i,j) = C_{ij}(UIS)$   
 $SE_p(i,j) = C_{(i-1)(j+1)}$

Else  
 If(there are one or more  $SE_p(i,k)$ s to sat-

isfy  $SE_p(l, k) < C_{(i-1)(j+1)}$  then  
 $SS_p(i, j) = \max(SE_p(l, k) \text{ for } l=1, \dots, N,$   
 $k=1, \dots, M) = \max(SE_p^*(l, k))$   
 $SE_p(i, j) = C_{(i-1)(j+1)}$   
 Else  
 $SS_p(i, j) = 0$   
 $SE_p(i, j) = 0$  and unit  $j$  should be holding  
 the product  $i$  until next unit  
 $j+1$  become ready to process  
 the product  $i$   
 End if  
 End if  
 Continue  
 Else  
 for every introduced intermediate storages  $p(p=$   
 $1, 2, \dots, P)$   
 $SS_p(i, j) = 0$   
 $SE_p(i, j) = 0$  this block can be eliminated by ini-  
 tial definition of every  $SS_p(i, j)$ , and  
 $SE_p(i, j)$  are zero.  
 End if

앞서 설명한 모든 계산과정을 토대로 본 연구에서는  
 간략화한 CIS 방안의 조업완료시간 결정을 보다 간결한  
 수학적 회귀공식으로 다음 식 (9)와 같이 개발하였다.  
 이 식은 앞서 보인  $SS_p(i, j)$ 와  $SE_p(i, j)$  결정 알고리즘과  
 모든  $i, j$ 스텝에서 병행하여 사용하여야 한다.

여기서 공유된 저장탱크가  $P$ 개( $P \geq 1$ ) 존재한다고  
 가정한다.

$$C_y(CIS) = \max[C_{(i-1)j}, C_{i(j-1)}, \min\{\max(SE_1^*(l, k)),$$
 $\max(SE_2^*(l, k)), \dots, \max(SE_P^*(l, k)),$ 
 $C_{(i-1)(j+1)}\} - t_{ij}] + t_{ij} \quad (9)$

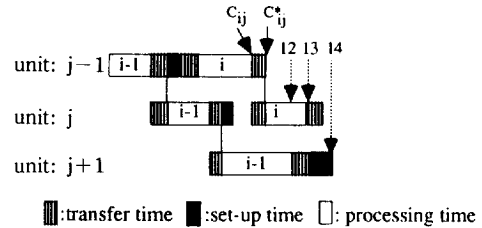
본 연구에서는 중간제품의 수송시간(transfer time)  
 과 unit의 다음 작업을 위한 준비시간(set-up time) 그  
 리고 중간저장탱크의 다음제품의 수송을 위한 정화시간  
 (clean-up time)을 고려하는 경우 간략화된 CIS 방안에  
 대한 조업완료시간 결정을 위한 수학적 회귀공식을 개  
 발하였다. 수송시간과 준비시간이 고려될 경우  $C_y(UIS)$ ,  
 $C_y(NIS)$ 도 다음 식 (10)과 (11)로 정의가 바뀌게 된다.

$$C_y(UIS) = \max[C_{(i-1)j} + a_{(i-1)j} + S_{(i-1)j}, C_{i(j-1)}]$$

$$+ t_{ij} + a_{i(j-1)} \quad (10)$$

$$C_y(NIS) = \max[C_{(i-1)j} + a_{(i-1)j} + S_{(i-1)j}, C_{i(j-1)},$$
 $C_{(i-1)(j+1)} + a_{(i-1)(j+1)} + S_{(i-1)(j+1)}$ 

$$- t_{ij} - a_{i(j-1)}] + t_{ij} + a_{i(j-1)} \quad (11)$$



$$7; C_{(i-1)j} + a_{(i-1)j} + S_{(i-1)j} + a_{i(j-1)} + t_{ij} \quad (12)$$

$$8; C_{i(j-1)} + a_{i(j-1)} + t_{ij} \quad (13)$$

$$9; C_{(i-1)(j+1)} + a_{(i-1)(j+1)} + S_{(i-1)(j+1)} \quad (14)$$

위 그림은 식 (14)가 식 (12), (13)에 비해 큰 경우의  
 예를 나타낸 것이다.  $a_{ij}$ 를  $i$  제품이  $j$  unit을 모두 빠져  
 나가는데 걸리는 수송시간,  $S_{(i-1)j}$ 를 unit  $j$ 에서  $i$  제품이  
 끝나고  $(i+1)$  제품이 시작될 수 있게 하기 위한 준비  
 시간,  $a_{i(j-1)}$ 는  $k$  unit 후에 저장탱크를 사용한  $i$  제품이  
 저장 탱크를 빠져 나간 후의 저장탱크 정화시간이라  
 한다면,  $C_{i(j-1)}$ 다음의  $C_{ij}$ 를 생각할 때는 제품의 변화가  
 없고 장치의 변화만 있으므로 식 (13)과 같이 단지 수  
 송시간만 고려한다( $j$  unit의 준비시간은 제품이 변화될  
 때 고려되므로). 그러나  $C_{(i-1)j}$ 다음의  $C_{ij}$ 를 생각한다면  
 제품의 변화가 생기고 장치의 변화가 없어서 식 (12)와  
 같이 수송시간과 준비시간을 모두 고려해 주어야 한다  
 (제품은 결국  $C_{i(j-1)}$ 에서  $C_{ij}$ 로 흐르므로  $a_{i(j-1)}$ 까지 고  
 려함). 또한  $C_{(i-1)(j+1)}$ 의 경우  $C_y$ 가 끝난후  $C_{i(j+1)}$ 을 시  
 작할 수 있는 시각을 알기 위한 것이므로 식 (14)와  
 같이 나타난다. 여기서 수식을 보다 간결하게 표현하기  
 위해 새로운 변수  $E_y = C_y + a_y + S_{(i-1)y}$ 을 도입하였다.

$SE^*(l, k)$  ( $l=1, 2, \dots, N, k=1, 2, \dots, M$ )는  $SS(l, k) <$   
 $E_{(i-1)(j+1)}$  및  $C_y(UIS) < SE(l, k)$ 를 만족하는 모든  $[SS$   
 $(l, k), SE(l, k)$  ( $l=1, 2, \dots, N, k=1, 2, \dots, M$ )]들에서의  
 $SE(l, k)$ 들을 말한다. 또한  $SS(l, k), SE(l, k)$ 의 결정에  
 서도 다소 차이가 있다. 수송시간과 준비시간을 고려  
 하는 경우 Fig. 9에서와 같이  $SS(l, k)$ 와  $SE(l, k)$ 에는  
 장치로부터 저장탱크로 또는 저장탱크로부터 장치로의  
 수송시간이 포함되어져 있으며, 제품  $i$ 이  $k$  장치로부터  
 다음장치로 수송되는데 필요한 시간  $a_{ik}$ 를 그대로 제품  
 $i$ 이  $k$  장치로부터 저장탱크로 또는 저장탱크로부터 다음  
 장치로 수송되는데 필요한 시간으로 사용된다.  $C_y(CIS)$   
 를 계산하는동안  $SS(i, j)$ 와  $SE(i, j)$ 는  $\max(SE^*(l, k)$   
 $+ a_{i(l, k)}) < E_{(i-1)(j+1)}$  및  $C_y(UIS) + a_y < E_{(i-1)(j+1)}$ 를 모  
 두 만족하는 경우에만  $SS(i, j) = C_y(UIS)$ ,  $SE(i, j) =$   
 $E_{(i-1)(j+1)} + a_{ij}$ 가 되고 그렇지 않은 경우는  $SS(i, j)$ 와  $SE$   
 $(i, j)$  모두 0으로 둔다. 조업완료시간 결정의 절차는  
 앞서 제시한 수송시간과 준비시간이 고려되지 않은 경

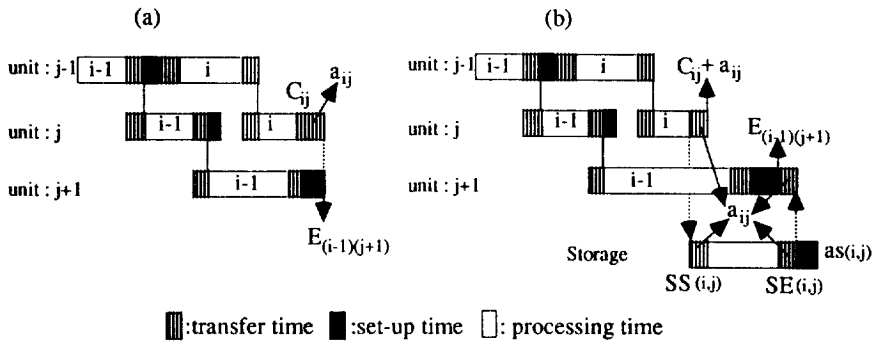


Fig. 9. Storage starting and finishing times with transfer and storage set-up times.

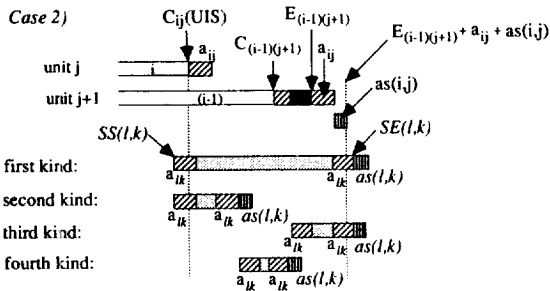


Fig. 10. Four kinds of storage using pattern of one common storage under consideration of non-zero transfer and set-up times.

우에서  $C_{(i-1)(j+1)}$ 을  $E_{(i-1)(j+1)}$ 로 또한  $SE(l, k)$ 를  $SE(l, k) + as(l, k)$ 로 대체시킨 것 이외에는 모두 같다.

수송시간과 준비시간을 고려하는 경우, case 2의 경우 저장탱크가 사용되는 4가지 상황을 Fig. 10에 나타내었다. 이 경우는 보다 복잡한 형태로써  $C_{ij}(\text{UIS})$ 와  $E_{(i-1)(j+1)} + a_{ij} + as(i, j)$  사이에 저장탱크 시작시간  $SS(l, k)$ 와 끝시간  $SE(l, k)$ 가 어떤 조건으로 놓이는가에 달려있다. 그 이유는 이 시간 간격동안 저장탱크를  $i$  제품이 사용하기 위하여  $E_{(i-1)(j+1)}$  시간 후  $(j+1)$  unit이 준비가 된다면 저장탱크로부터 수송시간  $a_{ij}$ 와 저장탱크 정화시간  $as(ij)$ 을 가능하게 하는 시간이 더 필요로 하기 때문이다. 결국 아래의 조건 3과 조건 4를 모두 만족하는  $SS(l, k)$ ,  $SE(l, k)$ 가 존재할 때가 case 2의 상태라고 할 수 있다.

$$SS(l, k) < E_{(i-1)(j+1)} + a_{ij} + as(i, j) \quad (\text{condition 3})$$

$$SE(l, k) + as(l, k) > C_{ij}(\text{UIS}) \quad (\text{condition 4})$$

수송시간과 준비시간을 고려하지 않은 경우와 같은 절차로 case 2 경우의  $C_{ij}(\text{CIS})$ 는 다음 식 (15)와 같이 얻어진다.

$$C_{ij}(\text{CIS}) = \min[E_{(i-1)(j+1)}, \max\{SE_1^*(l, k) + as(l, k)\}, \max\{SE_2^*(l, k) + as(l, k)\}, \dots, \max\{SE_p^*(l, k) + as(l, k)\}] \quad (15)$$

definition 3)  $SE_p^*(l, k)$  in eq. (15) and (16) is a set of common storage ending (emptying) times  $[SE(l, k)s]$  of the  $p$ th ( $p=1, 2, \dots, P$ ) common storage which satisfy both the condition 3 and condition 4 for  $i$  product after  $j$  unit.

이것은 제시된 전략이 같은 유형의 것들로  $l$ 과  $k$ 가 주어질 경우 각 저장탱크에서의  $a(l, k)$ 는 동일함을 전제로 한다. 또한 식 (15)를 사용하기 위해  $SS_p(i, j)$ 와  $SE_p(i, j)$ 의 값을 계산하는 알고리즘을 제시하였다.

For given  $i$  and  $j$ :

If( $C_{ij}(\text{UIS}) < E_{(i-1)(j+1)}$  (for operation(a)),

$C_{ij}(\text{UIS}) + a_{ij} < E_{(i-1)(j+1)}$  (for operation(b)) then

Do(for every introduced intermediate storages  $p(p=1, 2, \dots, P)$ )

If(there is no  $[SS_p(l, k), SE_p(l, k)]$  to satisfy both the condition 3 and condition 4) then

$$SS_p(i, j) = C_{ij}(\text{UIS})$$

$$SE_p(i, j) = E_{(i-1)(j+1)} + a_{ij}$$

Else

If(there are one or more  $(SE_p(l, k) + as(l, k))s$  to satisfy

$$SE_p(l, k) + as(l, k) < E_{(i-1)(j+1)} - a_{ij} \text{ then}$$

$$SS_p(i, j) = \max(SE_p(l, k) + as(l, k) \text{ for } l=1, \dots, N, k=1, \dots, M)$$

$$= \max(SE_p^*(l, k) + as(l, k))$$

$$SE_p(i, j) = E_{(i-1)(j+1)} + a_{ij}$$

Else

```

SSp(i, j) = 0
SEp(i, j) = 0 and unit j should be holding
the product i until next unit
j+1 become ready to process
the product i

End if
End if
Continue
Else
for every introduced intermediate storages p(p=
1, 2, ..., P)
SSp(i, j) = 0
SEp(i, j) = 0 this block can be eliminated by ini-
tial definition of every SSp(i, j), and
SEp(i, j) are zero.

End if

```

본 연구에서는 조업 완료시간을 각 제품  $i$ 가  $j$  장치에서 조업이 끝난 시간으로 보고있으나 조업이 끝난후 다음장치로 수송이 완료된시간( $C^*_{ij}$ )으로 보는 경우도 있다. 이런 경우  $C^*_{ij}(CIS) = C_{ij}(CIS) + a_{ij}$ 의 계산을 통해 조업이 끝난후 다음장치로 수송이 완료되는 조업 완료시간을 구할 수 있다.

공유될 중간저장탱크의 개수가 1개 이상( $p > 1$ )이 될 경우, CIS 방안의 조업 완료시간을 중간저장탱크 사용시간인  $[SS(ij), SE(ij)]$ 와 또다른  $[SS2(ij), SE2(ij)]$  등이 중복되는 것을 허용하여, 이들중 사용 가능한 것은 모두 사용할 수 있게 하여 계산하는 수학적 회귀공식을 다음과 같이 개발하였다.

$$C_{ij}(CIS) = \max[E_{(i-1)j}, C_{i(j-1)}, \min\{\max(SE_1^*(l, k) + as(l, k)), \max(SE_2^*(l, k) + as(l, k)), \dots, \max(SE_p^*(l, k) + as(l, k)), E_{(i-1)(j+1)}\} - t_{ij} - a_{i(j-1)}] + t_{ij} + a_{i(j-1)} \quad (16)$$

식 (16)의 경우  $C^*_{ij}(CIS) = C_{ij}(CIS) + a_{ij}$ 의 계산을 통해 조업이 끝난후 다음장치로 수송이 완료된 조업 완료시간을 구할 수 있다. 본 연구에서 개발한 조업 완료시간을 계산하는 회귀적 공식(recursive formula)의 장점은 storage efficiency check가 필요없이, 한 수식으로 제품중심의 저장탱크 운용 방안 및 사전중심의 저장탱크 운용 방안 모두를 표현해 줄 수 있으며 기존의 방법에 비해 컴퓨터 계산시간을 줄였다.

### 3-2. CIS 방안에 적합한 생산계획법

MILP의 최적화 생산계획법이나 경험적(heuristics) 방법은 CIS 방안의 중간저장탱크 운용 방안의 특수성

으로 인해 좋은 결과를 기대하기 어렵다고 보여지므로, 본 연구에서는 combinatorial search법이나 Ku와 Karimi[7] 및 DAS 등[8]이 제안한 simulated annealing 법에 의해 생산계획 문제를 풀이하였다. 본 연구에서는 제품의 수 10 이하의 비교적 작은 크기의 문제는 combinatorial search법을 이용하여 최적의 생산계획 해답을 얻어내고, 제품의 수가 11 이상인 비교적 큰 문제에서는 simulated annealing법을 이용하여 준 최적의 생산계획 해답을 얻어내는 방법을 제안하였다.

Combinatorial search의 경우 가능한 모든 조업의 순서를 얻어내는 알고리즘(permutation matrix search)과 본 연구에서 개발된 조업 완료시간 알고리즘의 조합으로 해답을 얻었다. 최적 생산순서를 얻어내기 위하여 우선 가능한 모든 생산순서의 가지수를 찾아내는 알고리즘과 이렇게 구해진 각 생산순서들에 대한 제품 생산완료시간을 새로이 제안한 회귀공식에 의해 계산하고 그 결과를 서로 비교하여 최소의 최종제품 생산 완료시간을 갖는 조업 순서를 찾아낸다. 컴퓨터 계산량을 줄이기 위한 방안으로, 현재까지 구하여진 최소의 최종제품 생산완료시간과 각 중간계산 단계에서 구하여진 제품 생산 시간 및 남아있는 생산에 필요한 최소한의 시간의 합을 비교하여 현재의 최소치보다 클 경우 계산에서 제외(fathoming)한다. 또한 공정간의 장치의 휴지시간(idle time)의 합이 현재까지 구하여진 최소 생산시간을 갖는 조업에서의 총 공정간 휴지시간의 합보다 더 커질 경우 계산에서 제외하여 준다. 이를 수식으로 표현하면 다음 식 (17)과 같다.

$$\begin{aligned} &\text{Fathoming if } C_{NM \min} < C_{ij} + \\ &\sum_{i=1+1}^N \sum_{j=j+1}^M P_{ij} + \sum_{i=i+1}^N \sum_{j=j+1}^M t_{ij} + \sum_{i=i+1}^N \sum_{j=j+1}^M S_{ijk} \\ &\text{Fathoming if } \Sigma(\text{Idle-Time of } C_{NM \min}) < \\ &\Sigma(\text{Idle-Time } C_{ij}) \end{aligned} \quad (17)$$

Simulated annealing법의 경우 uniform random number generation 알고리즘에 의해 조업 순서를 임의로 바꾼후 본 연구에서 제안된 조업 완료시간 알고리즘을 적용하였으며, uphill move의 허용을 random하게 얻어진 확률치와 비교하여 수행하는 에너지 함수의 control parameter  $kT$ 의 경우 Ku와 Karimi[7]가 제시한 평균 총 생산시간에 의한 방법을 사용하였으며, 이들을 이용한 metropolis 알고리즘을 사용하였다. Metropolis 알고리즘은 시행착오적 방법에 의해 얻어지는 각각의 해답을 물리적 계의 여러 다른 분자 배열들로 보고, 목적함수값을 물리계의 다양한 에너지 상태식으로 관련지어 보다 나은 분자배열 즉 시행착오적 해답을 얻

Table 1. Process dat for example 1

	u1	u2	u3
p1	61	2	56
p2	42	32	42
p3	1	31	71
p4	46	98	42
p5	11	14	20
p6	35	89	75
p7	66	8	25
p8	25	46	44

자는데 그 기본 개념이 있다. 이런 물리계의 현상을 이용한 컴퓨터 알고리즘을 최적화, 특히 생산계획에 접목시키기 위해 목적 함수인 에너지 함수를 Makespan으로 보아야 한다. 에너지 함수값을 초기에 무작위 변수 고정에 의해 얻고 이를  $E_1$ 이라 하며, 그 다음 번 무작위 변수고정에 의한 에너지 함수값을  $E_2$ 라 한다.

이 때  $E_1 > E_2$ 인 경우 새로운 해답이 보다 더 작은 값이 얻어진 것이므로 최소화를 목적으로 할 경우 새로운 중간 해답으로 받아들여지고,  $E_1 < E_2$ 인 경우는 새로운 해답이 과거의 해답보다 우수한 값이 도출되어 지지 않았으므로 버려지는 것이 아니라, 확률적 판단에 의해 받아들여지거나, 버림을 다시 한번 결정하게 된다. 즉 최소화 문제에 있어서 목적 함수치가 증가하는 경우 받아 들이는 확률을 다음 식 (18)와 같이 두었다.

$$P(\Delta E) = \exp[-(E_2 - E_1)/kT] \quad (18)$$

즉 구간  $[0, 1]$ 의 무작위 수(random number)를 발생시켜  $P(\Delta E)$ 와 비교하여  $P(\Delta E)$ 가 발생된 무작위 수보다 큰 경우 새로운 해답을 받아들이고, 작은 경우

새로운 해답을 버리고, 현재의 해답을 계속 유지하면서 다음 단계의 계산을 수행한다. 이와 같이 전단계의 해답보다 다음 단계의 해답이 나빠지더라도 어느 정도의 확률에 의해 그해답이 받아 들여 질 수 있는 것을 알 수 있는데 이것을 uphill movement라고 한다. 이 uphill movement는 위 확률식중 변수  $kT$ 와  $(E_2 - E_1)$ 에 영향을 받아 정해지는데, 이 중  $(E_2 - E_1)$ 은 조정 불가능한 변수이나,  $kT$ 값은 조정 가능하다. 계산 과정이 진행되면서 점점 최적 해답에 가까이 가게 되므로, 과정이 진행되는 동안 이 확률은 점점 줄어들어야 할 것인데, 그 확률을 줄어들게 할 수 있는 조정인자는  $kT$ 가 될 수 있다. 최종적으로 이 알고리즘의 종료는 미리 정해진 종료 조건, 즉 주어진 횟수만큼의 시행착오적 해답 계산에 의해서도 더 이상 우수한 결과가 도출되지 않는 경우에 의해 이루어진다.

무작위 수를 발생시켜 주어진 제품들의 초기 순서를 결정하였고, 그렇게 얻은 순서에 입각한  $C_{NM}$ 을 조업완료 시간을 나타내는 수학적 회귀공식에 의해 계산하였다. 초기의  $kT$ 값은 무작위 수 발생에 의해 제품 순서를 약 3000회 만들어보고 인접하여 얻어진 조업 순서의  $C_{NM}$  값들 간의 편차의 절대치 평균값을 계산하고 이 값의 약 1.5배를 함수의 반올림 정수를 초기  $kT$ 값으로 설정했다. 새로운 배열의 총 발생횟수를  $3N^3$ 으로 두고, 이를 20 step으로 나누어 매 step이 넘어 갈 때마다  $kT \times 0.95$ 를 하여 새로운 감소되어지는  $kT$ 값으로 설정하였다.

본 연구에서는 CIS 방안과 CIS 방안을 위한 새로운 completion time 알고리즘의 우수성을 보이기 위해 Ku와 Karimi[7]가 풀이하였던 다음 예제 1을( $8 \times 3$  문제) 각 저장탱크 방안에 대해 combinatorial search법과 simulated annealing법으로 풀이하여 비교해 보았다.

Table 2. Results of various intermediate storage policy of example 1

		Combinatorial search		Simulated annealing	
UIS	sequence	3-5-8-2-6-1-4-7		3-8-2-5-6-1-4-7	
		3-8-2-5-6-1-4-7			
		3-8-5-2-6-1-4-7			
	make span	411		411	
NIS	sequence	3-6-4-1-8-2-7-5		3-6-4-1-8-2-7-5	
		3-6-4-2-1-8-7-5			
		make span	452		452
FIS		ST : 1개	ST : 2개	ST : 1개	ST : 2개
		3-2-6-1-8-4-7-5	3-8-2-6-1-4-5-7	3-6-1-8-4-2-5-7	5-3-8-2-6-7-4-2
	외 4 경우	3-8-2-6-1-4-7-5			
	make span	429	417	429	427
CIS	sequence	3-8-2-6-1-4-5-7		3-8-2-6-1-4-7-5	
		3-8-2-6-1-4-7-5			
		make span	417		417

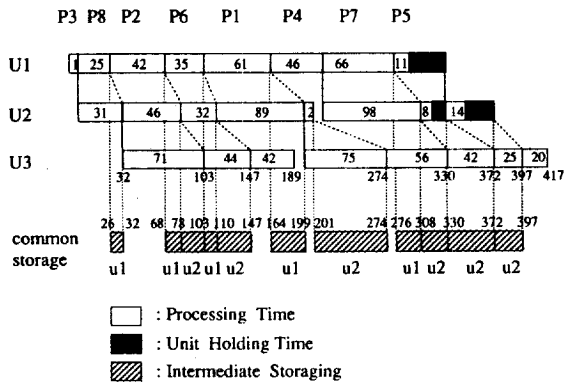


Fig. 11. Gantt chart of simulated annealing solution for example 1.

Table 1에 공정 데이터를 나타내었고, Table 2에 그 결과를 나타내었다. 또한 CIS 방안의 경우 얻어진 해답의 Gantt chart를 다음 Fig. 11에 나타내었다.

다음 예제 1의 경우 비교적 크기가 작은 문제이므로 combinatorial search법에 의한 결과는 모두 최적해임을 보장한다. 그러므로 simulated annealing법에 의한 해답이 매우 최적해에 근접한 준 최적 해임을 알 수 있다. 또한 CIS 방안이 투자비에 대비한 생산성 면에서 우수한 방안임을 다른 방안의 풀이 결과와 비교를 통해 알 수 있다.

#### 4. 유연한 중간저장탱크 사용 방안 및 회분식 공정 자동화의 기초

##### 4-1. CIS 방안의 중간저장탱크 운용 방안에 대한 FMS

CIS 방안의 우수한 점으로 또 하나 소개할 수 있는 것은 다른 종류의 저장탱크 사용 방안을 표현해 줄 수 있다는 점이다. 모든 중간저장탱크의 사용 방안은 밸브 및 펌프의 작동으로 유체를 unit에서 저장탱크로, 저장탱크에서 unit으로 이동시키게 한다. 중간저장탱크가 특정 unit사이에 고정되는 경우 그 저장탱크의 사용 방안은 고정되어지지만, CIS 방안의 경우 이들 밸브와 펌프의 적절한 작동으로 그 사용 방안이 유연성을 가질 수 있게 된다. 다음 Fig. 12의 경우 unit과 unit사이에 펌프 3개를 사용하는 경우이며, Fig. 13의 경우 보다 투자비를 줄이는 펌프 2개, 밸브 1개를 사용하는 경우이다.

Fig. 12의 경우 unit 2를 NIS 방안으로 사용하고자 한다면 펌프 5, 펌프 6을 잠그면 되고, Fig. 13의 경우는 단지 밸브 6만 잠그면 된다. 단, UIS와 같이 무제한적

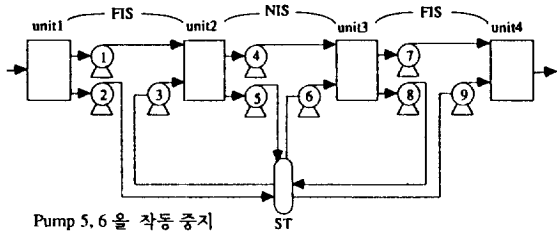


Fig. 12. CIS system with 3 pumps per each inter-unit.

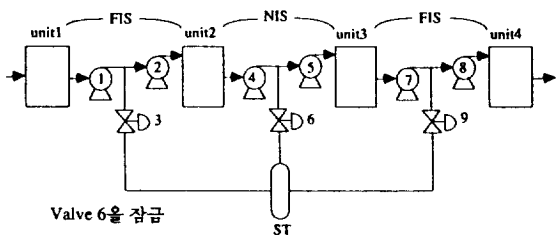


Fig. 13. CIS system with 2 pumps and 1 valve per each inter-unit.

중간저장탱크의 사용은 없는 것으로 가정해야 한다. 보다 많은 저장탱크 사용이 요구되어 진다면 CIS의 공유될 저장탱크 수를 늘려야 한다. 이와 같이 작동을 자유로이 변화시켜 여러 가지 유형, 특히 모든 MIS 방안의 시스템표현이 가능하다. 한 회분식 공정에서 제품의 수요가 항상 일정한 것은 아니고 일정기간이 지난후 제품의 수요가 바뀌어 생산해야 할 제품의 종류 및 양이 변하고, 이에 따라 생산계획이 변하며 또 바뀐 제품의 제조공정상의 특성이 각기 다양해서 지난번의 조업에서 요구되었던 중간저장탱크 방안이 모두 변해야 한다면 기존의 고정된 중간탱크 방안으로는 이를 대처하기 힘들게 된다. 이러한 측면에서 CIS는 다양하게 변화하는 중간저장탱크의 사용패턴을 때에 따라 맞추어 줄 수 있으므로 매우 유리하다. 여기서의 문제는 조업이 바뀔 때마다 밸브나 펌프의 조작이 복잡하게 달라져 조업 안전성에 문제를 줄 수 있을 것이다.

최근 그 성능이 매우 우수해져 자동화의 꽃으로 불리울 만큼 폭발적 수요를 나타내는 PLC는 이런 문제를 해결하기에 가장 적합한 장치이다. PLC logic을 이용하여 간단히 소프트웨어만 바꾸어 줌으로써 때에 따라 바뀌는 중간저장탱크의 사용 방안을 모두 해결해 줄 수 있다. 즉 CIS 방안과 PLC 및 전자식 펌프, 밸브를 사용함으로써 중간저장탱크의 운용 방안에 대한 FMS (flexible manufacturing system)를 실현할 수 있다. 조업의 안전성을 더하여 주기 위해서는 순차적 제어(sequence control)모듈과 별도로 interlock모듈을 두어야

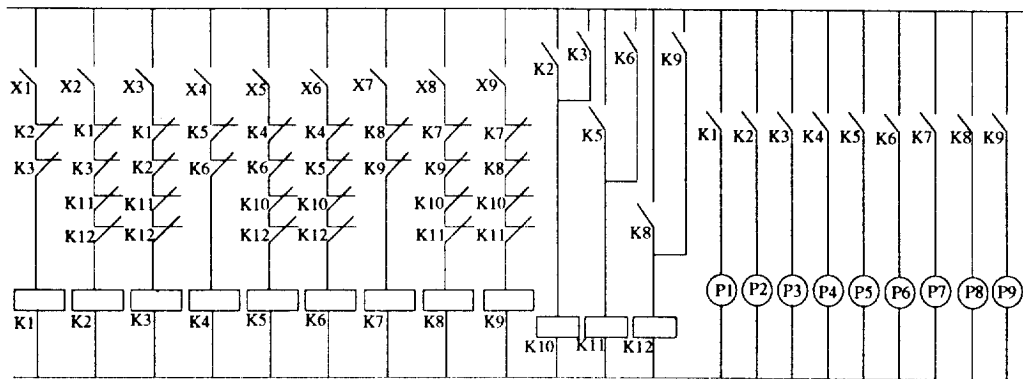


Fig. 14. Safety interlock ladder logic of PLC for the system of Fig. 10.

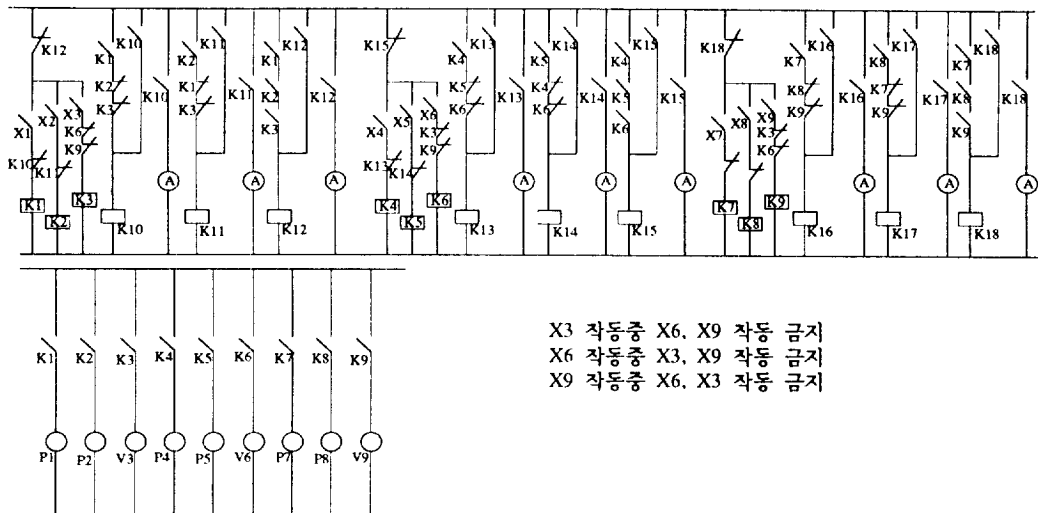


Fig. 15. Safety interlock ladder logic of PLC for the system of Fig. 11.

한다. 보통 조작이 복잡한 시스템에서 원치 않는 작동을 피하고, 오작동시 조업자에게 alarm 등을 통해 오작동임을 알려주며, 심각한 경우 전원차단을 하게 하는 것이 interlock모듈이다. 특정의 중간저장탱크 운용 방안을 원하는 경우 이 interlock모듈에서 불필요한 밸브나, 펌프를 작동치 않도록 하여 줄 수 있고, 이 밸브나 펌프가 작동을 할 때 alarm을 발생하도록 할 수 있다. 다음 Fig. 14와 15에 CIS 방안의 경우 Fig. 12 시스템과 Fig. 13 시스템에 대한 interlock모듈을 소개하였다.

이밖에 주어진 문제에 대해 최적 혹은 준 최적 생산 계획법에 의해 조업순서와 각 장치가 조업에 사용되는 시간이 결정 된다면 이 정보를 통해 PLC 알고리즘 (on/off switch, relay, timer 등의 작동과 조업자의 조업참조를 위한 PC와의 통신 등이 포함)을 구현할 수

있게 된다.

#### 4-2. 회분공정의 자동화 모델의 기초

회분식 공정의 자동화를 위해서는 여러 가지 하드웨어적 계층구조와 이들 복잡한 시스템들의 통합화, 나아가서는 CIM(computer integrated manufacturing) 시스템을 요구한다. 소프트웨어적으로는 합리적으로 데이터들을 관리하는 데이터 베이스 시스템, 상하위 레벨들로 나누어진 계층적 관리 시스템 등이 필수적으로 고려되어야 한다. 최근 Van Dooren[10]은 General Electric Plastic BV사에서 수행중인 실제산업의 회분식 자동화의 3가지 주요 소프트웨어적 기능(function)으로써 다음을 지적하였다. 첫째, 공정을 최적화하고 안전적 조업을 이끄는 주로 설정치(set point)의 추적(tracking)

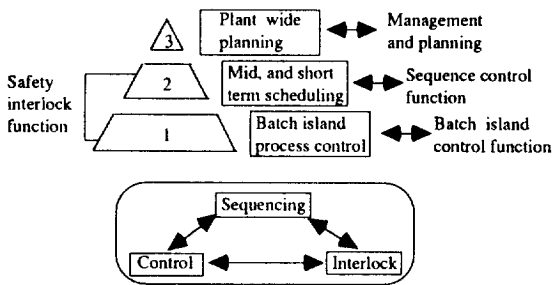


Fig. 16. Hierarchy of batch process automation mode.

cking)을 위한 회분식 공정들의 개별적 제어이며, 둘째, 밸브위치의 점검 등에 의해 공정중의 상황을 알려주는 safety interlock이며, 마지막으로 회분식 공정의 시작/종료, 수송시스템의 line-up 등을 담당하는 순차제어(sequence control)이다. Fig. 16에서와 같이 이 3가지 기능(function)들이 개별적으로 작동하면서 서로의 정보를 유기적으로 공유하여 적절한 작동의 지침을 세울 때 회분식조업의 자동화가 이루어질 수 있다.

회분식 공정의 전체공장 자동화(batch process plant wide automation)는 두 가지 유형의 계층적 구조를 갖는다. 첫째는 top-down 방식이며, 둘째는 bottom-up 방식이다. Bottom-up 방식 자동화의 예를 본 연구에 적용시켜 보면, 최하위의 각 회분식 공정의 제어를 최적으로 이행할 때 각 제품의 조업시간이 결정되어 진다. 이 정보를 바탕으로 CIS 방안의 중간저장탱크의 사용 방안을 FMS식 조업으로 생산계획을 할 수 있으며, 최적생산계획이 도출된 후 그 시스템에 맞는 safety interlock function을 개발하여 적용한다. 최종적으로 PC와 통신이 가능한 PLC와 PC로써 sequence control function을 구하여 주요 3가지 기능(function)을 한 데이터 베이스로 통합, 관리할 수 있게 하며 이것은 회분식 공정 자동화의 기초적 모델의 한 부분이 될 수 있다.

## 5. 결 론

본 연구를 통해 중간저장탱크 사용의 CIS 방안을 제시하였으며 이것은 효율적 측면에서 MIS 방안보다 우수함을 보인 동시에 중간저장탱크의 FMS를 실현하였고, 나아가서 PLC 및 PC와 함께 결합하여 회분식 자동화의 기초적 모델로써 사용될 수 있음을 제안하였다.

CIS 방안안에 있어서의 자동화를 위한 중간단계 레벨인 중기, 단기 생산계획을 위한 조업완료시간의 새롭고 간략한 수학적 회귀공식을 개발하였으며 이는 기존의 5 단계 if-else를 거치는 shared storage system의 완료

시간 결정에 비해 매우 간략한 수학적 회귀공식에 의한 완료시간 결정이므로 기존의 방법보다 효율적임을 보였다.

CIS 방안안에 적합한 생산계획법으로 combinatorial search와 simulated annealing을 이용하였으며, 마지막으로 회분식 공정의 자동화를 위한 가장 중요한 3가지 기능(function)중 하나인 safety interlock을 개발하였다.

## 감 사

본 연구는 한국과학재단지정 우수 연구센터인 공정 산업의 지능자동화연구센터의 연구비 지원에 의하여 이루어진 것으로 연구비를 지원해주신 연구센터에 감사드립니다.

## 사용기호

- $a_{ij}$  : transfer time of product  $i$  from unit  $j$  to unit  $(j+1)$
- $as(i, j)$  : clean-up time of storage tank after product  $i$  from unit  $j$
- $C_{ij}$  : completion time of product  $i$  from unit 1 to unit  $j$  without transfer time  $a_{ij}$
- $C^*_{ij}$  : completion time of product  $i$  from unit 1 to unit  $j$  including transfer time  $a_{ij}$
- $C_{NM}$  : completion time of final product  $N$  from unit 1 to final unit  $M$  without transfer time
- $M$  : number of units
- $N$  : number of products
- $S_{jrk}$  : set-up time for product  $j$  after product  $i$  on unit  $k$
- $SS(i, j)$  : common intermediate storage starting time of product  $i$  after processing on unit  $j$
- $SE(i, j)$  : common intermediate storage finishing time of product  $i$  after processing on unit  $j$
- $t_{ij}$  : processing time of product  $i$  on unit  $j$

## 참고문헌

1. Reklaitis, G. V.: *AIChE Symp. Ser.*, **78**, 119(1982).
2. Reklaitis, G. V.: *Proceedings of 4th International Symp. on PSE*, Montebello, Canada(1991).
3. Ku, H. M., Rajagopalan, D. and Karimi, I.: *Chem. Engng. Prog.*, August, 35(1987).
4. Wiede JR., W. and Reklaitis, G. V.: *Comput. Chem. Engng.*, **11**, 357(1987).
5. Ku, H. M. and Karimi, I.: *Comput. Chem. Engng.*, **14**, 49(1990).

6. Rajagopalan, D. and Karimi, I.: *Comput. Chem. Engng.* **13**, 175(1989).
7. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **30**, 163(1991).
8. DAS, H. et. al.: *Comput. Chem. Engng.* **14**, 1351 (1990).
9. Akamatsu, N. et. al.: IFAC DYCORT+'89, Maastricht, The Netherlands, 9323(1989).
10. van Dooren, J.: IFAC DYCORT+'89, Maastricht, The Netherlands, 9329(1989).
11. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **27**, 1840(1988).