

## 운전절차 합성을 위한 제약조건 전파 기반의 통합 프레임워크

허보경 · 황규석<sup>†</sup>

부산대학교 화학공학과, 생산기술연구소  
(2000년 7월 18일 접수, 2001년 3월 20일 채택)

### Integrated Framework for Operating Procedure Synthesis Based on Constraint Propagation

Bo Kyeng Hou and Kyu Suk Hwang<sup>†</sup>

Dept. of Chem. Eng., Research Institute of Industrial Technology, Pusan National University, Busan 609-735, Korea  
(Received 18 July 2000; accepted 20 March 2001)

#### 요 약

화학공장의 운전절차 자동 합성은 공장 안전을 향상시킬 뿐 아니라 설계 비용을 줄여 주지만 기존 방법들은 조작에 의한 공정상태의 변화를 모사하는 방법, 운전목표 달성에 필요한 조작과 중간목표의 선택 방법에 많은 한계를 가지고 있다. 따라서 본 연구에서는 목표사이의 복잡한 상호작용으로 인한 조작선택의 어려움을 해결하고 공정지식을 보다 효과적으로 표현하기 위해 필요한 제약조건 전파 기반의 통합 프레임워크를 제안하였다. 제안된 방법론은 중간목표사이의 상호작용이 매우 심한 공정에 적용하여 그 유효성을 입증하였다.

**Abstract** – The automatic synthesis of operating procedures for chemical plants improves plant safety and reduces design costs. However, the existing operating procedure synthesis techniques have limitations in simulating a change of process states by the operation and selecting operations and subgoals needed to achieve the operational goal. So in this study, we have proposed the integrated framework based on constraint propagation to resolve problems about operation selection due to subgoal interaction among subgoals and to effectively use the process knowledge. The proposed methodology has been successfully applied to the process with serious subgoal interaction.

**Key words:** Operating Procedure, Constraint Propagation, Integrated Framework

#### 1. 서 론

경제성장과 더불어 화학공장들이 점차 대형화되고 업종이 다양화 되어감에 따라 사고발생 빈도가 점차 증가하고 있으며, 사고에 따른 인명 및 재산피해도 점점 대형화되고 있는 추세에 있다. 국내 · 외의 석유화학공장에서 발생한 104건의 사고를 근거로 하여 사고 원인을 분석해 보면 운전 미숙(운전절차의 오류 및 불이행, 오조작 등)으로 인한 사고가 21%를 차지하고 있다[19].

따라서 화학공장의 자동화와 안전성 향상을 위해서는 공장의 개시, 정지, 비상사태, 보수유지 조업에 관한 운전절차를 합성하고 분석하는 작업이 필요하다[17, 18]. 공정의 초기 상태로부터 최종 목표상태를 만족시키기 위한 조작(operation)의 시퀀스(sequence)를 만들어 내는 운전절차 합성(Operating Procedure Synthesis, OPS) 시스템은, 오류가 없는 운전절차를 합성하는데 도움을 주며, 특정 공정에 대한 전문가들의 지식을 구조화, 문서화하여 보관하고 공유할 수 있다는 장점과 함께 운전자 교육용 시스템으로의 활용 뿐 아니라 공정의 이상상황에 유연하게 대처할 수 있는 시스템으로 발전할 수 있을 것으로 예상된다[1, 5, 8, 10, 13].

그러나 기존 시스템은 공정 구조의 수정이나 조업 조건의 변화에 적용할 수 있는 운전절차를 합성하는데 부적합하고 공정 장치와 시스템의 구성요소에 대한 완벽한 지식베이스를 요구한다. 또한 표층지식(shallow knowledge)인 공정 데이터를 심층지식(deep knowledge)인 운전상황으로 인식하지 못하여 운전상황의 시간적 변화와 인과 관계를 추론하지 못할 뿐 아니라 조작 선택과 순서 결정에 대상 공정의 지식을 제대로 활용하지 못한다. 일반적으로 기존의 OPS 시스템이 내포하고 있는 지식 표현의 문제점은 다음과 같이 요약할 수 있다[2-4, 6].

(1) 조작의 효과는 조작되는 시점에서 바로 발생한다고 가정되므로 시간 경과에 의한 조작의 영향은 표현하지 못한다. 예를 들어, 조작이 실행되고 몇 분 후에 반응기의 넘침(overflow) 현상이 발생하는 경우를 말한다.

(2) 흐름의 방향은 시스템의 설계자가 의도한 방향으로만 흐르며 역류(back flow)는 존재하지 않는다고 가정한다. 또한 공정의 정성적 변화만을 고려할 뿐 물질의 혼합비나 농도는 무시한다.

(3) 공정의 전체영역에 걸쳐 물질의 혼합, 접촉으로 발생할 수 있는 위험을 규정한 전역적 제약조건(global constraint)과 장치의 운전 준비조건 미비로 발생하는 위험을 규정한 지역적 제약조건(local constraint)은

<sup>†</sup>E-mail: kshwang@hyowon.pusan.ac.kr

고려하지만 임의의 운전상황에서 한시적으로 적용되는 위험을 규정 한 시간적 제약조건(temporal constraint)은 고려하지 않는다.

또한 기존 운전절차 합성 알고리즘의 문제점을 간단하게 요약하면 다음과 같다[7, 9, 11, 12].

- (1) 조합폭발(combinatorial explosion)의 문제를 해결하기 위해 조작 선택과 순서 결정을 위한 불완전한 경험규칙을 사용하므로 가능한 모든 해를 얻을 수 없다.
- (2) 중간목표(subgoal) 사이의 상호작용을 인식하지 못하여 엄청난 후진(backtracking) 비용을 요구한다.
- (3) 안전한 운전절차는 제시하지만 조업자의 실수나 장치고장에 의한 위험상황을 안전하게 되돌려 주지 못한다.

따라서 본 연구에서는 기존 연구의 지식 표현과 합성 알고리즘의 문제점을 자세히 파악하고 조작과 연관되어 있는 상황 의존적 정보를 인식하기 위해 필요한 공정 지식의 분류와 공정의 기능적·구조적인 세분화 및 조작의 표현 방법을 소개하고자 한다. 또한 공정에 존재하는 제약조건을 이용하여 조작 선택과 순서 결정의 지침(guideline)으로 삼는 방법을 제시하고자 한다.

본 연구에서는 조작에 의한 공정상태의 변화를 인식하기 위해 제약조건 전파기법을 사용한다. Lakshmanan[10]이 개발한 공정 제약조건 언어인 PCON(Process CONstraints Language)은 인공지능 분야의 TCON(Tiny CONstraint Language) 모델에 객체지향 모델링 프레임워크를 도입한 것으로 공정을 쉽게 모델링할 수 있을 뿐 아니라 전향·후향추론이 가능하다[9-11]. 그러나 PCON은 공정의 운전상황과 실제 공정 상에 존재하는 여러 가지의 흐름특성을 고려하지 않을 뿐 아니라 각 변수 사이의 기능적인 연관성을 인식하지 못하여 잘못된 결과를 나타내므로 운전절차 합성에는 적합하지 않다. 따라서 본 연구에서는 기존의 공정 제약조건 기법을 다양한 공정흐름 특성과 조작형태에 적용할 수 있도록 수정하였다.

## 2. 공정 모델링 및 상태모사 방법

### 2-1. 조작연산자 모델

STRIPS(STanford Research Institute Problem Solver) 연산자(operator) 기반의 OPS 시스템은 현재의 공정상황과 장치의 연결관계를 고려하여 장치의 조작을 표현해야 하므로 엄청난 개수의 연산자를 필요로 한다[14-16]. 또한 조작과 연관되어 있는 상황과 무관한 상황을 정의해야 하는 프레임 문제(frame problem)는 “조작 실행에 의한 효과를 나타내는 리스트(list)에 기재된 요소에만 영향을 주고 기재되지 않은 요소에는 영향을 주지 않는다”라고 가정(STRIPS assumption)하여 해결한다. 예를 들어, Fig. 1과 같은 공정에서 밸브 x가 잠겨져 있는 경우, 밸브 y를 열어도 그것이 공정에 끼치는 영향은 그렇게 크지 않다. 그러나 밸브 x가 열려져 있는 경우, 밸브 y를 열면 outlet-1로 수소 흐름이 발생한다. STRIPS assumption을 만족시키기 위해서는 밸브 y의 조작에 필요한 연산자를

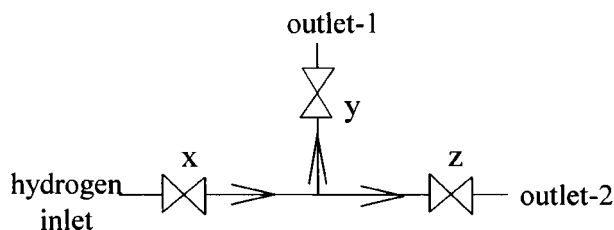


Fig. 1. Simple process.

```
(define Unit
  (:superclass (Graph))
  (:name ?name)
  (:energy ((sigma (mass-flow-{pi}) over pi in PORTS)
    (for-each j in *chemical-species*
      (sigma (x-{j})-{pi} * mass-flow-{pi})
        over pi in PORTS))
    (for-each pi in PORTS
      (1 - (sigma (x-{j})-{pi})
        over j in *chemical-species*))))))
  (:material ((sigma (mass-flow-cp-{pi}) * T-{pi})
    over pi in PORTS) + q-in)
    (sigma (mass-flow-cp-{pi}) over pi in PORTS)
    (for-each pi in PORTS
      (mass-flow-cp-{pi} - (mass-flow-{pi} * cp-{pi}))))))
  (:precondition ?precondition)
  (:constraints ?constraints)
  (:ports ?ports)
  (:purge ?purge)
  (:var ?var)
  (:state-equation ?equation)
  (:sink ?sink)
)
```

Fig. 2. Operator of process unit.

적어도 두 개 이상 필요로 한다. 즉, 밸브 x의 on/off 상태에 따라 각각 다른 밸브 y의 조작 연산자를 설계해야 한다. 따라서 하나의 장치가 가지고 있는 조작 연산자가 서로 유사하고 많은 경우, 잘못된 조작 연산자의 선택은 엄청난 후진을 발생시키는 원인이 된다.

이러한 STRIPS 연산자를 기반으로 하는 OPS 시스템의 단점을 극복하기 위해 복잡한 조작모델을 사용하는 전향추론 OPS 시스템은 어떤 효과를 달성하는데 필요한 조작을 제외할 수 없거나 운전절차를 생성하는 동안에 발생하는 목표들 사이의 상호작용을 인식하지 못할 수 있다.

따라서 본 연구에서는 조작연산자의 내부에 다른 장치나 흐름경로(flow path)의 상태에 대한 정보가 필요한 STRIPS 기반의 조작연산자의 한계를 극복하기 위해 장치가 가지고 있는 기능들을 장치중심의 상태함수로 표현한 객체지향 조작 연산자모델을 Fig. 2와 같이 제시한다. 객체지향 조작연산자의 각 슬롯은 상위클래스(superclass), 장치 이름(:name), 물질 수지(:material), 에너지 수지(:energy), 장치의 사전 운전 준비조건(:precondition), 운전중 제약조건(:constraint), 입력/출력포트(:port), 퍼지(purge)의 실행 여부와 퍼지 가능한 물질 쌍(:purge), 상태 변수들(:var; 온도, 압력, 농도, 작동상태, 상 등), 상태 전이식(:state-transition), 물질 혼합의 가능 여부 확인(:sink)을 나타낸다.

위와 같은 장치의 조작연산자는 미리 작성해 둔 객체지향 데이터베이스에서 유도되며, 데이터베이스는 지식의 일관된 표현 방법으로 지식의 제사용 및 확장성을 보장할 뿐 아니라 여러 가지 형태(정성, 정량)의 지식을 제공하므로 다양한 수준의 추론을 지원한다. 운전목표 달성에 필요한 조작연산자의 선택과 조작에 의한 공정상태의 변화를 모사하기 위해 흐름경로의 운전상황을 판단한 후, 각 장치가 가지고 있는 제약조건 전파(constraint propagation) 모델을 이용한다. 여기서, 제약조건 전파 모델은 조작연산자의 슬롯 중에서 :energy, :material, :state-equation 슬롯에 저장된다.

### 2-2. 제약조건 전파를 이용한 상태모사

조작에 의한 공정상태의 변화 모사와 사용자가 입력한 공정상태 정보의 논리적 타당성을 입증하기 위해서는 정성·정량적인 정보를 모두 이용할 수 있어야 하며 다양한 수준(level)의 운전목표에 대응할 수 있는 모사가 필요하다. 다시 말해, 단순한 형태의 정성적 모사에서 정량적 모사까지의 모든 영역을 포함하는 유연성을 가지고 있어야 할 뿐 아니라 사용자의 상위 레벨(high-level)의 상태 입력정보를 하위 레벨(low-

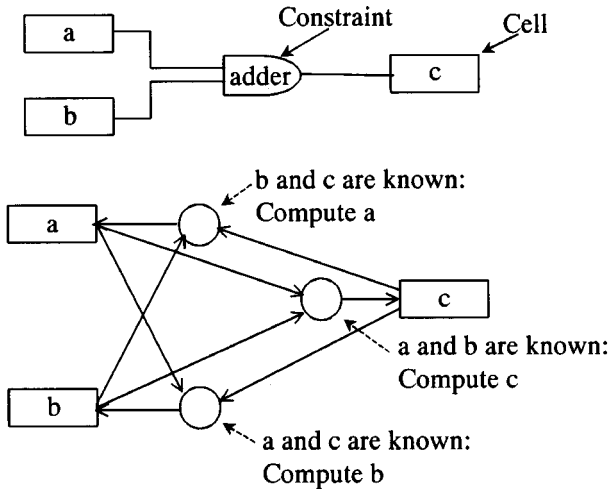


Fig. 3. Constraint propagation.

level)의 상태 입력정보로 인식할 수 있는 구조를 가지고 있어야 함을 의미한다.

본 연구에서는 조작에 의한 상태의 변화를 모사하기 위해 인공지능 분야의 연구자들이 개발한 TCON(Tiny CONstraint Language)에 객체 지향 모델링을 도입한 공정 제약조건 언어인 PCON(Process CONstraints Language)을 운전절차 합성에 맞게 수정하였다[10].

제약조건 전파란 변수를 나타내는 셀(cell)과 셀 사이의 제약조건(constraint)을 이용하여 셀의 값을 결정하는 방법을 말한다(Fig. 3). 예를 들어, Fig. 3에서 셀 A와 셀 B의 값을 알고 있는 경우는 constraint adder의 룰(rule)인  $A+B=C$ 를 이용하여 셀 C의 값을 구하고, 셀 A와 셀 C의 값으로 셀 B의 값을 구하는 경우는  $B=C-A$ 를 이용한다. 나머지의 경우도 위와 같은 방법으로 구한다. 즉, 임의의 공정 변수 값이 바뀌는 경우, 제약조건으로 연결되어 있는 다른 공정 변수의 값을 자동으로 변화시켜 준다. 이러한 계산능력을 이용하여 조작에 의한 공정상태의 변화를 모사할 뿐 아니라 사용자가 입력한 초기상태와 목표상태의 논리적 타당성을 검증하기 위해 의존 네트워크(dependency network) 기능을 이용할 수 있다. 즉, 사용자가 논리에 맞지 않는 서로 상반된 상태를 입력하는 경우, 제약조건 전파는 의존 네트워크 기능을 사용하여 이러한 모순을 유발시키는 원인을 찾아내어 논리적으로 타당한 초기상태와 목표상태의 입력이 가능하도록 하는 것이다.

기존의 PCON 모델은 다음과 같은 문제점들을 가지고 있어 운전절차 합성에 적합하지 않다.

(1) 사용자가 서로 다른 수준의 공정상태 정보를 입력하는 경우에 발생하는 ramification 문제를 해결할 수 있는 방법이 모색되어야 한다. 예를 들어, 밸브를 통해 유체가 흐르고 있다면 그 밸브는 열려 있어야만 된다는 사실을 유추할 수 있어야 한다. 이러한 문제는 지식의 레벨을 달리하여 적용하는 경우에 반드시 해결해야 되는 문제 중의 하나이다.

(2) 공정의 특정 영역과 연관되어 있는 운전상황을 인식하기 위해서는 그 영역과 상호작용하고 있는 공정의 다른 영역의 운전상황의 이해가 반드시 필요하다. 그러나 기존의 제약조건 전파기법은 이러한 기능을 수행할 수 없는 단점을 가지고 있어 완벽한 공정상태의 변화를 모사하지 못한다. 예를 들어, 반응기 밖으로 생산물이 나오고 있는 상태라면 반응기 안에서 반응이 진행되고 있어야 할 뿐 아니라 반응기 유입구로 반응물이 유입되고 있음을 인식해야 한다.

(3) 물질 수지식이식 (1)-(4)와 같은 파이프 PCON 모델에 Lakshmanan [10]이 사용한 일반적인 제약조건 전파 알고리즘을 적용할 경우, 입력포

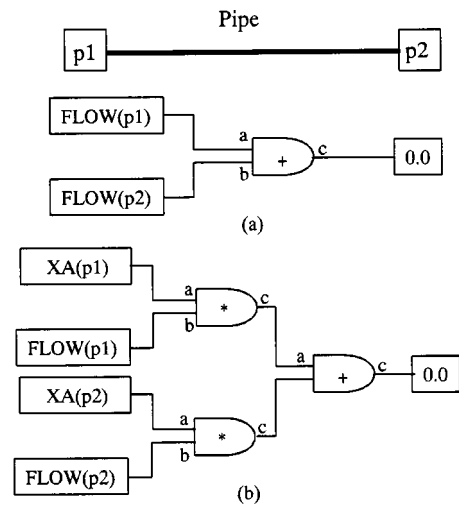


Fig. 4. Simple model for a pipe.

(a) Overall mass balance, (b) Component mass balance on A.

트 p1과 출력포트 p2의 총괄 물질 수지(total material balance)는 p1의 흐름 FLOW(p1)과 p2의 흐름 FLOW(p2)를 항상 동일하게 유지시킨다(Fig. 4). 또한 성분 A에 관한 물질 수지(component balance on A)는 p1의 조성 XA(p1)과 p2의 조성 XA(p2)를 항상 일치시키므로 서로 다른 성분의 물질들이 섞이지 않고 분리되어 동일한 파이프 내를 흐르는 경우는 표현할 수 없다.

$$\begin{aligned} \text{FLOW}(p1) + \text{FLOW}(p2) &= 0 & (1) \\ \text{XA}(p1) * \text{FLOW}(p1) + \text{XA}(p2) * \text{FLOW}(p2) &= 0 & (2) \\ \text{XA}(p1) + \text{XB}(p1) &= 1 & (3) \\ \text{XA}(p2) + \text{XB}(p2) &= 1 & (4) \end{aligned}$$

따라서 본 연구에서는 기존 PCON 알고리즘의 단점을 극복하기 위해 시작장치(start-unit)와 목표장치(target-unit) 사이에 존재하는 흐름경로의 형태를 고려하는 새로운 제약조건 전파 알고리즘(constraint propagation for OPS, CONOPS)을 다음과 같이 제안한다. 여기서, 흐름경로의 형태는 막혀 있는 구간(Block), 폐쇄된 구간(Trap), 흐르고 있는 구간(Flow), 분기된 구간(Branch)으로 분류한다(Fig. 5).

- (1) IF {“목표장치의 상부흐름(upstream) 형태가 Flow나 Branch이다” AND “그 경로상의 수송장치가 작동하고 있거나 상부흐름의 압력이 목표장치 압력보다 크다”}
- THEN {“상부흐름의 물질들이 분리되어 흐르는 경우, 목표장치의 입력포트에서 가까운 곳에 존재하는 물질로부터 상부흐름에 존재하는 모든 물질에 대하여 순차적으로 장치들의 :energy와

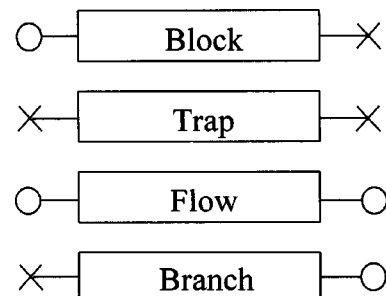


Fig. 5. Types of flow path.

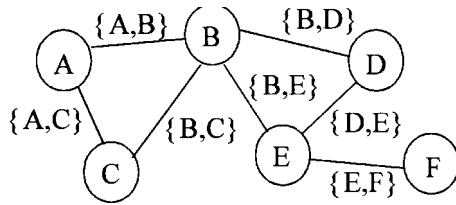


Fig. 6. Digraph.

:material 슬롯에 저장되어 있는 PCON모델을 이용하여 상태변수의 값을 전파시킨다”}

- (2) IF {“상부흐름의 흐름형태가 Block, Trap, Branch이다”} OR {“상부흐름의 흐름형태가 Branch이다” AND “수송장치가 작동하지 않거나 상부흐름의 압력이 목표장치보다 크지 않다”}

THEN {“상태변수의 값을 전파시키지 않는다”}

- (3) IF {“장치내에서 물질의 생성과 소멸이 존재한다”}

THEN {“상태변수식(:state-equation)을 이용하여 상태변수의 값을 전파시킨다”}

- (4) 목표 장치의 입력 공정변수 값이 결정되면, 그 장치의 제약조건전파 모델을 이용하여 출력변수 값을 결정한다.

- (5) 위와 같은 과정을 하부흐름(downstream)의 모든 장치에 반복 적용하여 실행한다.

## 2-3. 공정의 연결구조 표현과 흐름경로 탐색 방법

장치의 연결 구조(topology)는 새로운 장치의 정의 및 추가, 공정 흐름의 방향성 표현이 용이한 유향그래프(digraph)를 사용하여 나타낸다. 유향그래프는 장치를 나타내는 노드(node, 원으로 표시), 노드와 노드사이의 연결 관계를 나타내는 아크(arc, 선으로 표시)로 구성된다(Fig. 6). 여기서, 아크에 표시된 노드쌍(node pair)은 흐름의 방향성을 의미한다. 예를 들어, 노드 A와 노드 B 사이에 존재하는 아크의 노드쌍이 {A, B}이면 노드 A에서 노드 B로 흐름을 의미하며, 만일 {A, B}와 {B, A}가 동시에 존재하면 노드 A와 노드 B사이에는 양방향의 흐름이 가능함을 의미한다. 또한 역류는 아크에 존재하는 노드쌍의 순서를 바꿔주어 나타낸다.

객체지향 데이터베이스로부터 각 장치의 인스턴스를 생성한 후, 유향 그래프를 사용하여 연결하면, 각 장치 인스턴스의 상태변수들은 Fig. 7

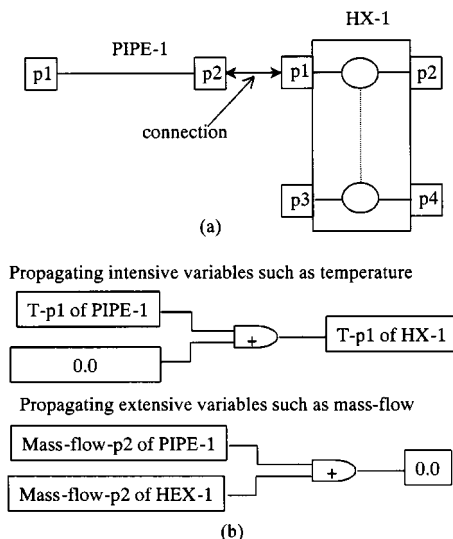


Fig. 7. The connection between units.

(a) Connecting two units, (b) Transferring variable values.

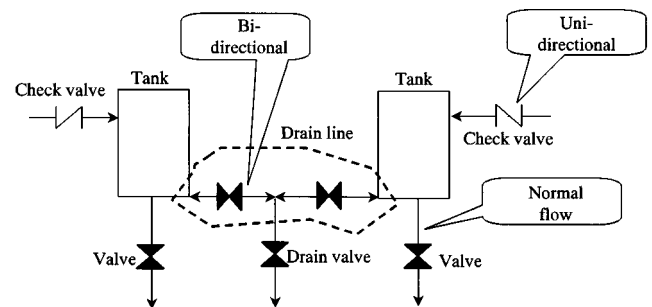


Fig. 8. Direction of flow path.

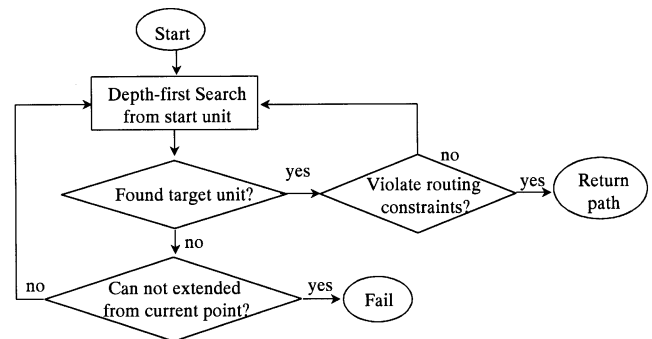


Fig. 9. Basic routing algorithm.

과 같이 서로 연결된다.

흐름경로 탐색의 용이함을 위해 본 연구에서는 흐름 방향성의 형태를 정상 흐름(설계자가 의도한 흐름인 동시에 역류의 가능성이 있는 흐름, normal flow), 단일 흐름(한 방향의 흐름이 존재하며 역류의 가능성이 없는 흐름, uni-direction flow), 양방향 흐름(양방향의 흐름이 가능한 흐름, bi-direction flow)으로 나눈다. Fig. 8에서 공유 드레인 라인(drain line) 위에 존재하는 모든 밸브와 파이프는 양방향 흐름이며, check valve는 단일 흐름이고 나머지는 모두 정상 흐름이라 할 수 있다.

물질이 흐를 수 있는 흐름경로는 전역적 제약조건(물질의 혼합, 접촉, 물질의 존재 유무 등), 사용금지 제약조건(사용해서는 안될 장치, Don't use constraint), 흐름경로의 방향성, 그리고 운전 목표(operating target)와 같은 경로 제약조건(routing constraints)을 참조하여 발견한다(Fig. 9).

일반적으로 흐름경로를 통한 물질흐름의 생성방법은 흐름경로 상에 존재하지 않는 주위의 밸브들을 잠근 후에 경로상의 모든 밸브를 열어 물질 흐름을 발생시킨다. 그러나 이러한 방법은 다른 물질의 흐름경로를 막아 조작 불능 상태에 빠뜨릴 수 있을 뿐 아니라 운전절차상에 필요 없는 조작이 중복되어 나타난다는 단점을 가진다. 따라서 각 중간목표의 달성에 필요한 흐름경로를 파악하고 서로 중복되지 않게 조정하여 조작 불능 상태에 빠지지 않도록 해야 한다.

만일 물질 수송을 위한 경로가 위험물질에 오염되어 있고 다른 우회 경로가 존재하지 않는 경우에는 경로의 오염물질을 깨끗이 청소하기 위한 퍼지 작업이 실행된다. 즉 퍼지 구간을 판별하는 연산자(Purge)와 물질로 오염되어 있는 구간을 판별하는 연산자(Contamination)를 사용하여 퍼지가 필요한 구간을 선택한 다음, 전역적 제약조건을 참조하여 퍼지 물질과 방법을 찾아 오염된 구간을 퍼지한다.

## 3. 운전상황의 인식과 조작연산자 선택 방법

화학공장의 개시·정지 조업을 위한 운전절차의 합성은, 초기상태로부터 목적상태로 상태가 전이되는 동안에 반드시 만족되어야 하는 중

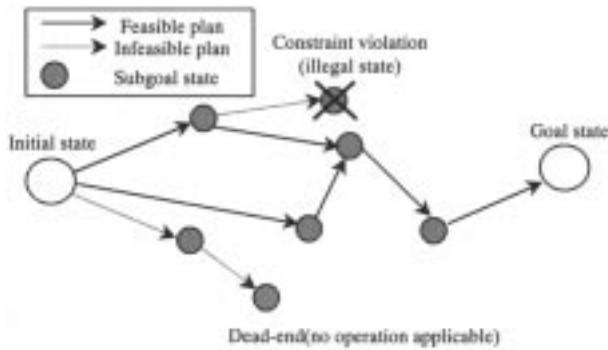


Fig. 10. Schematic of a typical OPS problem.

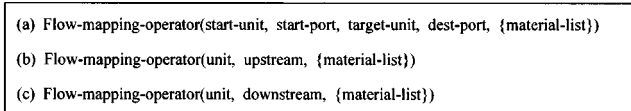


Fig. 11. Types of flow-mapping-operator.

간목표 상태(subgoal state)를 필요로 한다(Fig. 10). 이러한 중간목표 상태는 일반적으로 사용자가 입력해 주거나 미리 작성되어 있는 경험지식(장치의 표준 운전절차)에 의해 작성된다.

본 연구에서는 흐름경로의 운전상황을 고려한 조작연산자의 선택을 위해 흐름상황 인식 연산자(Flow-mapping-operator)를 정의하고(Fig. 11), 이 연산자를 이용하여 운전목표의 달성에 필요한 흐름을 생성시키는 조작연산자를 선택하도록 한다.

이러한 흐름상황 인식 연산자는 시작장치(start-unit)의 시작포트(start-port)와 목적장치(target-unit)의 종결포트(dest-port) 사이에 존재하는 구간[Fig. 11(a)]이나 장치(unit)의 상부흐름[upstream, Fig. 11(b)]·하부흐름[downstream, Fig. 11(c)]에 물질(material-list)이 어떠한 형태로 흐르는지를 판단한다. 여기서, 흐름상황 인식 연산자의 종류는 흐름 형태를 파악하는 Flow, Block, Trap, Branch가 있고, 오염여부를 확인하는 Contamination과 오염물질의 퍼지 구간을 확인하는 Purge가 있다. 또한 장치의 공정상태 변수의 값을 참조하여 장치의 운전상태를 파악한다.

조작연산자의 선택 방법은 공정의 현재상태와 목표상태와의 차이를 비교한 후, 그 차이를 줄이는데 필요한 공정흐름 및 장치의 상태변수의 값을 변화시키는 조작을 생성하여 이루어진다. 즉 조작에 의해 변화된 공정상태의 인식 뿐 아니라 목표상태의 도달에 필요한 조작 선택에 있어 흐름상황 인식 연산자를 사용한다.

#### 4. 공정 제약조건인 표현 방법

최초의 OPS 시스템[15]은 다음과 같은 다섯 가지 형태의 위험상황을 체크할 수 있다. 즉, (1) 두 가지 이상 물질의 혼합, (2) 두 장치 사이의 흐름, (3) 특정흐름의 차단(blocking), (4) 특정한 유출구에 닿는 특정 화합물의 유·무, (5) 특정 장치에 도착한 화학 물질 유·무 등에 의한 위험상황을 말한다.

본 연구에서는 흐름경로에 관한 제약조건이 존재하는 경우, 흐름상황 인식 연산자와 상태변수를 이용하여 장치와 흐름경로의 운전상황을 인식한 후, 미리 정의되어 있는 위험상황과 비교하여 일치하는 경우에는 현재의 공정상태가 위험하다고 판단한다. 예를 들어, tank1에서부터 tank2로 물질 A의 흐름이 중단되면, 위험상황이 발생하는 경우에는 다음과 같이 나타낸다. 여기서, ¬는 부정(negation)을 나타내는 기호이다.

(IF [¬flow(tank1, output-port, tank2, input-port, {A})

THEN [hazard-situation])

또한 물질의 혼합이나 접촉으로 인해 발생하는 위험상황은 전역적 제약조건으로 규정하고, contains(?unit, ?port, material)와 같은 연산자를 사용하여 나타낸다. 예를 들어, 수소와 산소가 혼합되어서는 안된다는 제약조건은 다음과 같이 나타낸다. 여기서, ?unit와 ?port는 모든 장치와 입력출력포트에 대해 적용된다는 의미이다.

(IF [contains(?unit, ?port, {hydrogen}) AND contains(?unit, ?port, {oxygen})]  
THEN [explosion])

#### 5. 합성 알고리즘

전향 추론(forward chaining)에 의한 전순서(total order) 합성 알고리즘은, 목적 지향성(goal direction)의 부족과 연속적인(sequential) 조작 시퀀스를 행하므로 병렬조작의 표현이 어렵지만, 중간목표(subgoal) 상태의 정보와 제약조건을 이용한 탐색이 가능하다.

부분순서(partial order) 알고리즘은 합성하는 과정 중에 조작의 순서를 미리 결정하지 않고 순서결정 제약조건(order constraint)만을 추가하므로 후진 감소와 병렬 조작 시퀀스가 가능하지만, 탐색의 제어가 힘들고 전순서 조작 시퀀스로의 전환이 어렵다.

중간목표의 상호작용이 존재하는 경우, 잘못된 조작의 선택은 엄청난 후진을 요구한다. 예를 들어 Fig. 12와 같은 간단한 공정에서 “반응기내의 물질을 드레인(drain)하라[Drain(Reactor)=True, Aperture(Valve 3)=Close, Aperture(Valve 1)=Close, Aperture(Valve 2)=Close]”는 목표상태를 달성하기 위해 먼저 Valve 3을 여는 조작인 Open(Valve 3)을 행한 후에 2분을 기다리는 조작인 Waitfor(2 min)를 행한다. 그러나 앞선 조작들 [Open(Valve 3), Waitfor(2 min)]이 Valve 3의 상태를 Aperture(Valve 3)=Open으로 변화시키므로 중간목표인 Aperture(Valve 3)=Close는 유지될 수 없다. 즉 중간목표 Drain(Reactor)=True와 Aperture(Valve 3)=Close 사이에 상호작용이 존재함을 알 수 있다. 만일 Waitfor(2 min)보다 Close(Valve 3)을 먼저 행하면, Drain(Reactor)=True는 달성될 수 없으므로 후진이 발생한다.

그러나 조작 연산자의 개수가 많아지는 경우에 순서결정 제약조건만으로는 충분히 중간목표 상호작용을 제어할 수 없으며, 부분순서들 사이에 존재하는 순서결정과 연관되어 있는 제약조건을 유추하지 못한다.

따라서 본 연구에서는 공정의 제약조건을 최대한 이용할 수 있는 제한조건전파 기법을 이용하는 새로운 OPS 알고리즘을 다음과 같이 제안한다.

- (1) 사용자로부터 장치의 연결관계, 초기상태와 목표상태, 공정 제약조건들을 입력받아 초기화한 후, 흐름상황 인식 연산자를 이용하여 각 흐름경로의 상태를 파악한다.
- (2) 공정의 현재상태와 목표상태가 서로 일치하는 경우, 그때까지 합성된 운전절차를 제시하고 프로그램은 종료된다.
- (3) 현재상태와 목표상태를 비교하여 그 차이를 줄일 수 있는 조작세

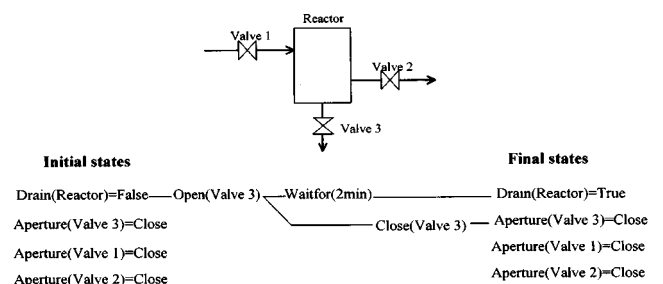


Fig. 12. Subgoal interaction.

트를 생성한다. 만일 전역적 제약조건에 의해 더 이상 선택할 조작이 없는 경우, 전역적 제약조건을 피할 수 있는 퍼지 조작세트의 생성한다. 단, 퍼지 조작인 경우에 퍼지 작업이 완료되면, 반드시 그 조작을 닫는 조작을 자동적으로 추가한다.

(4) 퍼지 조작을 행할 수 없는 경우나 운전목표를 달성하는데 필요한 조작이 없는 경우, 사용자에게 운전목표를 달성할 수 없음을 알리고 종료된다.

(5) 조작세트 중에서 사전 운전준비조건을 만족하는 조작을 하나 선택한다. 만일 조작의 사전 운전준비조건이 만족되지 못한 경우에는 사전 운전준비조건을 만족시키는데 필요한 조작을 먼저 선택한다.

(6) 각 장치의 제약조건 전파 모델을 이용하여 조작 실행으로 인한 공정상태의 변화를 모사한다.

(7) 변화된 공정상태가 전역적·지역적 제약조건을 침범하지 않고 안전한지를 체크한다. 만일 위험한 경우에는 공정의 상태를 조작 실행 이전으로 되돌리고 (4) 단계로 이동한 다음, 조작세트에 있는 나머지 조작들에 대해 (4)-(7) 단계를 반복 실행한다. 만일 조작에 의해 변화된 공정상태가 안전한 경우에는 운전절차에 그 조작을 추가한 다음, (2) 단계로 이동한 후에 (2)-(7) 단계를 반복 실행한다.

## 6. 통합 프레임워크의 구성

기존 비선형 부분순서 OPS 시스템의 구조는 수치계산을 할 수 없을 뿐 아니라 외부 agent들을 이용할 수 없지만 본 연구에서 제안한 시스템 구조는 모두 가능하다. 제안된 OPS 시스템은 Allegro common lisp for windows 5.01로 작성되었으며 Fig. 13과 같은 구성요소로 이루어져 있다. 각 구성요소의 기능은 아래와 같다.

**GUI(Graphic User Interface):** 사용자가 대상공정의 연결관계, 공정의 초기상태와 목표상태, 제한조건 등과 같은 정보를 입력할 뿐 아니라 최종적으로 합성된 운전절차를 사용자에게 제시해 주는 모듈이다.

**Knowledge-Base:** 각 장치의 대체지향 조작연산자, 화학물질의 물리적·화학적 특성, 작업자의 경험지식 등을 라이브러리로 작성하여 저장해 둔 지식베이스이다.

**CONOPS-Classifier:** 사용자로부터 입력받은 대상공정의 지식과 데이터를 계층화할 뿐 아니라 논리적 질의에 응답하기 쉬운 형태의 네트워크로 변형시키는 모듈이다. 즉, 장치 사이의 물리적·기능적인 인과관계를 유지하고 운전절차의 합성에 필요한 정보들을 추출하는 역할을 수행한다.

**Mapping-Module:** 운전상황의 인식을 위해 필요한 모듈로 조작에 의해 변화된 공정상태를 인식하여 운전상황의 존재 유·무를 판단한다.

**Safety-Checker:** 전역적·지역적 제약조건을 침범하는 위험상황이 발생하는지의 여부를 판단하는 모듈이다.

**CONOPS-Planner:** 중간목표 순서화, 조작연산자의 선택, 그리고 운전목표의 달성 여부를 체크하여 위험상황을 발생시키지 않고 목표상태에

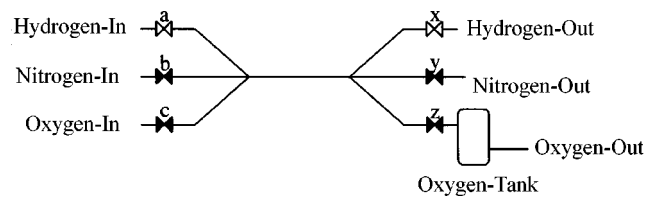


Fig. 14. The bottleneck problem.

Table 1. Initial and final states

Units	Initial state	Final state
a	open	open
b	close	close
c	close	close
x	open	open
y	close	close
z	close	close
Oxygen-tank		oxygen

도달하는데 필요한 운전절차를 합성하는 모듈이다.

**Simulator: CONOPS-Planner**가 제시한 후보조작의 실행으로 인한 공정상태 변화를 확인하는데 사용되는 모사기 모듈이다.

**Flow-Path-Searcher:** 필요한 공정흐름 경로를 발견하기 위해 사용되는 탐색 모듈이다.

## 7. 적용 사례

본 연구에서 제시한 공정 모델링과 제약조건 전파 기반의 합성 알고리즘의 유효성을 검증하기 위해 Fig. 14와 같이 중간목표 사이의 상호작용이 매우 심한 공정에 적용하여 보았다[20]. 기존의 방법은 각 장치의 초기상태에서 목표상태로의 달성을 위한 조작이 단지 하나만 요구되는 문제에만 적용될 수 있을 뿐이다. 즉, 각 장치의 목표상태에 도달하기 위해 두 개 이상의 조작을 필요로 하는 경우에는 적용할 수 없다는 단점을 가진다. 또한 초기상태와 목표상태의 차이가 거의 없기 때문에 목표달성에 필요한 조작연산자를 선택하지 못하고 각 중간목표사이의 상호작용을 인식하지 못해 잘못된 조작연산자를 제시하여 조합폭발의 문제가 발생한다.

공정의 초기상태로는 밸브 a, x가 열려져 있어 hydrogen-in에서 hydrogen-out으로 수소가 흐르고 있으며, 산소탱크(oxygen-tank)안에는 산소가 채워져 있지 않고 밸브 b, c, y, z는 닫혀져 있는 상태이다(Table 1). 운전목표는 산소탱크에 산소를 채우고 난 다음, hydrogen-in에서 hydrogen-out으로 수소를 다시 흐르게 하는 것이다. 이 공정의 전역적 제약조건은 “산소와 수소가 접촉되어서는 안된다[Fig. 15(a)]”와 “수소와 질소, 산소와 질소는 접촉 가능하므로 산소와 수소를 질소로 퍼지할 수 있지만 퍼지할 때 유출되는 가스의 배출은 nitrogen-out으로만 가능하다[Fig. 15(b)]”이다. 이 공정은 물질흐름을 생성하는데 필요한 파이프를 공유하고 있기 때문에 병목(bottleneck)현상이 발생한다. 또한 매우 복잡한 중간목표 상호작용이 발생하며 운전목표를 달성하기 위해 한 밸브의 상태가 두 번 이상 바뀌어야 한다.

제안한 OPS 알고리즘은 필요한 모든 데이터를 초기화한 다음, 초기상태와 목표상태를 비교하여 산소탱크에 산소를 채우는 Flow(oxygen-in, out-port, oxygen-tank, in-port, {oxygen}) 흐름상황을 먼저 생성한다. 흐름경로의 주위로 물질이 새어나가는 것을 방지하기 위해(close a)와(close x)를 선택·실행한 후, 공정상태의 변화를 제약조건 전파로 모사하여 전역적 제약조건의 침해 여부를 체크한다. 위험상황이 발생하지 않으면

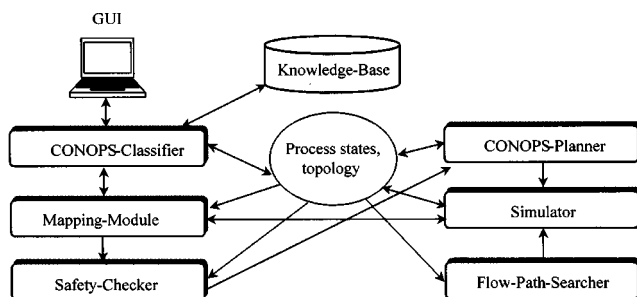


Fig. 13. Integrated framework.

```

(a) (IF [contains(?unit, ?port, {oxygen}) AND
      contains(?unit, ?port, {hydrogen})]
    THEN [hazardous-situation])

(b) (IF [contains(hydrogen-out, in-port, {nitrogen, hydrogen}) OR
      contains(oxygen-tank, in-port, {nitrogen, oxygen})]
    THEN [hazardous-situation])

```

Fig. 15. Global constraints of case study.

로 (close a)와 (close x)를 운전절차에 추가한다.

흐름경로와 공정상태의 변화를 인식한 다음, Flow(oxygen-in, out-port, oxygen-tank, out-port, {oxygen})에 해당하는 흐름경로의 생성이 전역적 제한조건을 침범하고 우회경로가 존재하지 않으므로 Purge(nitrogen-in, in-port, nitrogen-tank, out-port, {oxygen})가 생성된다. 산소의 퍼지를 위해 필요한 흐름경로를 여는 (open b)와 (open y) 조작을 먼저 선택하여 실행한 후, “퍼지 실행에 기여한 조작은 다시 닫아야 한다”는 기능적 특성에 의해 (close b)와 (close y)를 실행한다.

Flow(oxygen-in, in-port, oxygen-tank, out-port, {oxygen})에 해당하는 흐름경로가 폐지되었으므로 흐름경로를 여는 (open c)와 (open z) 조작을 선택하여 산소탱크에 산소를 채운다. 그 다음, hydrogen-in에서 hydrogen-out으로 수소를 다시 흐르게 하기 위해 Flow(hydrogen-in, in-port, hydrogen-tank, out-port, {hydrogen})의 생성을 시도하지만 해당 흐름경로를 여는 (open a)와 (open x)의 실행이 전역적 제약조건을 침범하므로 조작실행 이전의 상태로 후퇴한다. 그러나 시스템이 우회경로를 발견하지 못하므로 오염된 경로를 폐지한다. 폐지 작업에 앞서서 먼저, “폐지할 때 유출되는 가스의 배출은 nitrogen-out으로만 가능하다 [Fig. 15(b)]”라는 제약조건을 만족시키기 위해 (close c)와 (close z) 조작을 실행한 후, (open b)와 (open y) 조작을 선택하여 오염경로를 폐지한 후, 다시 (close b)와 (close y) 조작을 실행하여 폐지 작업을 완료한다. 이제 Flow(hydrogen-in, in-port, hydrogen-tank, out-port, {hydrogen})를 위한 흐름경로가 깨끗이 폐지되었으므로 (open a)와 (open x) 조작 실행으로 운전목표가 달성된다. 따라서 합성된 최종 운전절차 결과는 다음과 같다. {(close a), (close x)}, (open b), (open y), (close b), (close y), (open c), (open z), (close c), (close z), (open b), (open y), (close b), (close y), (open a), (open x)}.

## 8. 결 론

선형이나 비선형 계획 방식을 사용하는 기존 운전절차 합성 방법들은 공정 제약조건을 쉽게 이용할 수 없고 중간목표간의 복잡한 인과관계로 인한 조합폭발의 문제를 적절하게 해결하지 못한다. 따라서 본 연구에서는 이러한 문제를 해결하기 위해 운전상황의 변화를 인식하여 중간목표사이에 존재하는 상호 인과관계를 파악한 후에 조작연산자를 선택하고 제약조건 전파 기법을 이용하여 조작 실행에 의한 공정상태의 변화를 모사하는 통합 프레임워크를 개발하였다. 또한 제안된 방법론을

중간목표간의 상호작용이 매우 복잡한 공정에 적용하여 그 유효성을 입증하였다. 즉, 운전상황을 고려한 조작연산자의 선택이 어려운 기존 방법과는 달리 본 연구에서 제시한 제약조건 전파 기법은 흐름경로의 특성을 파악하여 중간목표 달성에 필요한 조작연산자를 제시해 줄 뿐 아니라 순서 결정을 위한 정보를 제공한다. 따라서 중간목표사이에 복잡한 상호작용을 가지고 있는 공정에 적용할 경우, 잘못된 조작연산자의 선택으로 인한 후퇴를 감소시켜 준다.

향후, 공정의 시간적 제약조건을 다룰 수 있는 방법과 함께 대규모 화학공장의 운전절차 합성을 위해 필요한 지식베이스의 작성, 그리고 다른 시스템과의 효율적 연계를 위해 필요한 시스템 아키텍처(architecture)의 개발이 요구된다.

## 감 사

본 연구는 부산대학교 생산기술연구소의 일부 연구비 지원으로 수행되었으며 이에 감사드립니다.

## 참고문헌

1. Aelion, V. and Powers, G. J.: *Comp. Chem. Eng.*, **15**, 349(1991).
2. Crooks, C. A. and Macchietto, S.: *Chem. Eng. Commun.*, **114**, 117(1992).
3. Foulkes, N. R., Walton, M. J., Andow, P. K. and Galluzzo, M.: *Chem. Eng. Commun.*, **12**, 1035(1988).
4. Fusillo, R. H. and Powers, G. J.: *Comp. Chem. Eng.*, **4**, 369(1987).
5. Fusillo, R. H. and Powers, G. J.: *Comp. Chem. Eng.*, **12**, 1023(1988).
6. Huasheng, L., Lu, M. L. and Naka, Y.: *Comp. Chem. Eng.*, **23**, 657(1999).
7. Hwang, K. S., Tomita, S. and O'shima, E.: *Kagaku Kogaku Ronbunshu*, **14**, 728(1988).
8. Tomita, S., Hwang, K. S., O'shima, E. and McGreavy, C.: *J. Chem. Eng. Jpn.*, **22**, 364(1989).
9. Lakshmanan, R. and Stephanopoulos, G.: *Comp. Chem. Eng.*, **12**, 985(1988).
10. Lakshmanan, R. and Stephanopoulos, G.: *Comp. Chem. Eng.*, **12**, 1003(1988).
11. Lakshmanan, R. and Stephanopoulos, G.: *Comp. Chem. Eng.*, **14**, 301(1990).
12. Li, H. S., Lu, M. L. and Naka, Y.: *Comp. Chem. Eng.*, **21**, 899(1997).
13. Naka, Y., Lu, M. L. and Takiyama, H.: *Comp. Chem. Eng.*, **9**, 997(1997).
14. Rivas, J. R. and Rudd, D. F.: *AIChE J.*, **20**, 311(1974).
15. Rivas, J. R. and Rudd, D. F.: *AIChE J.*, **20**, 320(1974).
16. Rostein, G. E., Lavie, R. and Lewin, D. R.: *AIChE J.*, **40**, 1650(1994).
17. Hou, B. K., An, D. M., Hwang, K. S.: *HWAHAK KONGHAK*, **34**, 750(1996).
18. An, D. M. and Hwang, K. S.: *HWAHAK KONGHAK*, **34**, 727(1996).
19. Rafael, B. P.: Ph.D. Dissertation, Tokyo Institute of Technology, Yokohama, Japan(1997).
20. Soutter, J. K.: Ph.D. Dissertation, Loughborough Univ., Loughborough, UK(1996).