

조작절차 합성을 위한 공정 모델링 및 알고리즘

허보경, 황규석
부산대학교 화학공학과

Process Modeling and Algorithm for Operating Procedure Synthesis

Bo Kyeng Hou, Kyu Suk Hwang
Dept. of Chemical Eng., Pusan National Univ.

1. 서론

가스나 화학물질을 다루는 공정의 자동화와 안전성 향상을 위해서는, 공정의 start-up 및 shutdown 조업의 조작절차를 합성, 분석하는 작업이 선행되어야 한다.

조작절차 합성(Operating Procedure Synthesis, OPS) 시스템은 예러가 없는 조작절차를 합성하는데 도움을 주며, 특정 공정에 대한 전문가들의 지식을 구조화, 문서화하여 보관하고 공유할 수 있다는 장점과 함께 운전자 교육용 시스템(OTS, Operator Training System)으로의 활용 뿐 아니라 공정의 이상상황에 유연하게 대처할 수 있는 시스템으로 발전할 수 있을 것으로 예상된다.

그러나 기존의 연구자들이 개발한 OPS 시스템은 지식표현이나 합성 알고리즘에 많은 문제점이 있기 때문에 복잡한 공정의 조작절차를 합성할 수 없다. 따라서 본 연구에서 복잡한 공정의 조작절차를 합성해 주는 알고리즘을 개발하고자 한다.

2. 기존 연구의 문제점

기존의 OPS 시스템은 공정 구조의 수정이나 조업조건 변화에 적용할 수 있는 조작절차를 합성하는데 부적합하며, 대상에 대한 완전한 지식을 필요로 한다. 즉 조작절차의 합성은 공정의 초기 운전조건으로부터 최종운전조건을 만족시키기 위한 조작의 시퀀스(sequence)를 만들어 내는 과정으로 공정장치, 시스템, 그리고 구성요소에 대한 완벽한 지식베이스를 요구한다. 따라서 공정의 디자인 초기에 공정의 조작성(operability)을 평가할 경우는 불완전한 공정의 지식을 사용해야 하므로 기존의 OPS 시스템으로는 조작절차를 합성할 수 없게 된다.

일반적으로 기존의 OPS 시스템이 내포하고 있는 지식표현의 문제점은 다음과 같이 요약할 수 있다.

1) 조작의 표현은 시간적인 개념을 도입하지 않음

조작의 효과는 조작되는 시점과 동시에 효력이 발생한다고 가정되기 때문에 시간경과에 의한 조작의 영향을 표현할 수 없다. 예를 들면 조작이 행하여지고 몇 분 후에 반응기가 overflow되는 조작의 효과는 나타내지 못한다. 또한 병렬조작을 고려하지 않으며, 절차에 있어서 병렬로 조작되어야 할 조작은 순차적으로 표현되며, 그 효과는 동일하다고 가정한다. OPS분야에서 병렬조작을 다룰 수 있는 시스템은 없으며, 단지 AI 계획시스템에서 시도된 바 있다.

2) 조작에 의한 공정상태의 변화를 완벽하게 인식 못함

역류(back flow)는 존재하지 않으며, 흐름의 방향은 시스템의 설계자가 의도한 대로 한 방향으로만 흐른다고 가정한다. 또한 정성적인 공정의 변화만을 고려하고, 물질의 혼합비나 농도는 무시한다.

3)공정의 제한조건을 적절하게 표현하지 못함

물질의 혼합, 접촉 등으로 발생하는 위험(global constraint)과 운전예비조건을 만족시키지 않고 장치를 운전하는 경우에 발생하는 위험(local constraint) 등은 표현하고 있으나, 임의의 운전 상황에만 적용되는 위험(temporal constraint)은 표현하지 못하고 있다.

또한 기존 OPS 알고리즘의 문제점은 다음과 같다.

1)조작의 선택과 순서를 결정하기 위해서 불완전한 경험규칙 사용

조합폭발(combinatorial explosion)의 문제를 해결하기 위해서 일반적으로 허용할 수 없는 규칙(예, Rivas와 Rudd의 positive operator)을 사용하는 경우가 많다. 이러한 규칙을 적용하는 경우에는 가능한 모든 해를 얻을 수 없으며, 이미 내정되어 있는 해답만을 얻을 뿐이다.

2)공정 지식에 내포되어 있는 제한조건들을 적절하게 활용 못하는 탐색전략

조작절차의 합성은 공정의 초기 운전상태로부터 실행 가능한 조작을 선택, 실행한 후에 변화된 공정상태의 위험성 여부를 판단하고, 최종 운전상태에 도달한지를 체크한다. 그러나 기존의 OPS 알고리즘은 탐색과정 중에 공정 지식에 내포되어 있는 제한조건들을 활용하지 못해서 잘못된 조작의 선택으로 인한 엄청난 후진(backtracking) 비용을 담당해야 한다.

3)공정의 위험상황을 정상적으로 회복시키지 못함

기존의 OPS 시스템은 안전한 조작절차를 제시하나 조작자의 실수나 고장에 의한 공정의 상태를 안전하게 되돌려 주는 시스템은 개발되지 않았다.

3. 연구내용

3.1 대상공정 모델링

화학공정의 안전한 개시/정지조업을 위해서는 장치의 적절한 개시/정지 운전절차와 밸브와 수송장치의 조작으로 물질의 흐름을 적절하게 조절해야 한다. 즉 공정의 위험상황은 물질의 흐름에 의해서 전파된다고 생각할 수 있다.

공정 장치의 연결구조(topology)와 공정흐름구조(PFD), 그리고 새로운 공정 장치의 정의와 추가를 쉽게 하기 위해서 양방향을 표시할 수 있는 유향그래프(digraph)로 공정을 나타낸다. 한 방향만을 나타낸 유향그래프의 공정흐름 표현은 역류에 대한 추론이 힘들고 설계된 흐름 경로 위의 장치 고장으로 인해 해당 경로를 사용할 수 없을 경우에 적당한 우회 경로를 찾을 수 없다.

본 연구에서는 아래와 같은 그래프(graph) 구조를 기본으로 하여 공정 구조를 표현한다.

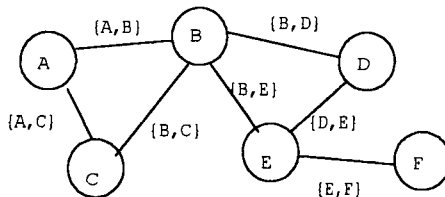


Fig. 1. Representation of digraph

Digraph(node, arc)는 그래프에서 원으로 표시되는 노드(node)와 노드사이에 연결된 선으로 표

시되는 아크(arc 또는 edge)로 구성되어 있다. 공정장치들은 모두 노드로 표시되며 장치사이의 연결관계는 노드사이의 아크로 표시한다(fig. 1). 여기서 노드사이의 방향성은 아크위에 표시하는 노드쌍의 순서에 의해서 결정된다. 만일 위의 그림에서 노드 A와 B 사이의 아크가 (A, B)이면 노드 A에서 B로 흐름이 존재함을 의미한다. 또한 (A, B)와 (B, A)가 둘 다 존재하면 노드 A와 B사이에는 양방향의 흐름이 존재함을 의미한다. 또한 임의의 아크위에 존재하는 노드쌍의 순서를 바꿔주면 역류에 대한 표현이 가능하게 된다.

매우 복잡한 공정을 다루는 경우에 가능한 모든 흐름을 다루는 것은 불가능하므로 모든 장치에 대하여 양방향(bi-direction) 흐름 표현은 사용하지 않는다. 단지 체크밸브는 한 방향(uni-direction)의 흐름만이 가능하다. 설계자 또는 운전자가 의도한 운전목표를 달성하기 위해 필요한 흐름방향을 지정 흐름(intended flow or normal flow)으로 정의한다(fig. 2). 지정 흐름과 한 방향 흐름의 차이는 한 방향흐름은 역류가 일어날 수 없으나 지정흐름은 가능하다.

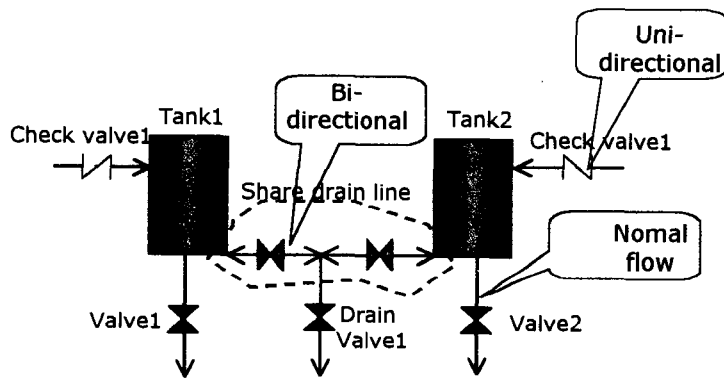


Fig. 2. The representation of process flow

물질 흐름을 위한 경로는 global constraint(물질의 혼합, 접촉, 물질의 존재 유무 등)와 Don't use constraint(사용해서는 안될 장치에 관한 제한 조건), 최종 운전 목표에 관한 내용을 참조하여 선택한다(Fig. 3).

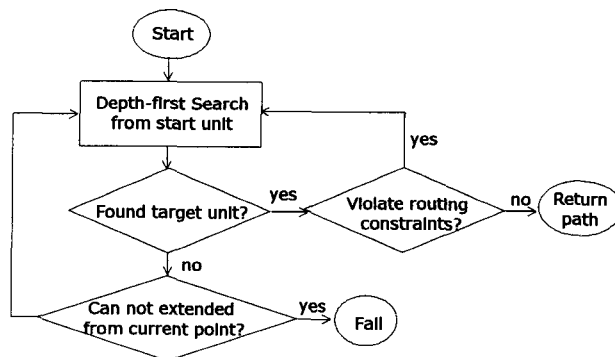


Fig. 3. Basic routing algorithm

공정에 존재하는 제한조건은 연결되어 있는 공정장치로 물질과 에너지가 흘러서 전파되므로, 각 공정장치는 정상상태를 기준으로 한 물질 수치, 에너지 수치, 그리고 공정장치의 고유 특성 지식(운전 준비조건, 일반적인 장치 개시/정지절차, 위험요소에 관한 지식)을 가진다. 물질수치와

에너지 수지는 장치의 일반적인 특성을 나타내는 함수로 공정의 구조와 상황에 따라 적용을 달리해야 한다. 본 연구에서는 제한조건의 전파는 algebraic equation과 qualitative equation을 모두 표현 가능한 forbus의 TCON(Tiny CONstraint Language)을 이용하여 모사한다(fig. 4, fig. 5).

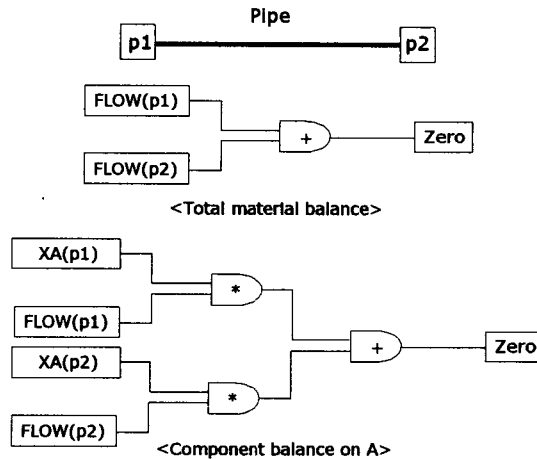


Fig. 4. TCON model of a pipe

```
(defclass process-unit (generic-process-unit)
  ((ports :initform '(p1 p2))
   (adiabatic? :initform t)
   (unit? :accessor unit? :initform nil)
   (mass-balance :accessor mass-balance
                 :initform
                 '((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*))))))
   (energy-balance :accessor energy-balance
                   :initform
                   '((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))))))
   (other-equations :accessor other-equations
                    :initform nil)
   (print-variables :accessor print-variables
                    :initform nil)))
```

Fig. 5. Modeling of process units for constraint propagation(LISP code)

3.2 합성 알고리즘

하부 목표(subgoal)들 사이에 상호작용(interaction)이 존재하는 경우, 잘못된 조작의 선택은 엄청난 후진(backtracking)을 요구한다. 예를 들어 아래와 같은 공정에서 “valve 3을 통해 반응기내의 물질을 드레인(drain)시킨 후, valve 3을 닫는다”라는 운전목표를 달성하는 조작 절차를 합성한다고 하자(fig. 6). 운전목표[Drain(Reactor)=True, Aperture(Valve 3)=Close, Aperture(Valve

1)=Close, Aperture(Valve 2)=Close]를 달성하기 위해서, 먼저 valve 3을 열고 2분을 기다린다 [Open(Valve3), Waitfor(2min)]. 그러나 앞선 조작들이 valve 3의 상태를 변화시켜 최종 운전상태에 있는 하부 목표인 Aperture(Valve 3)=Close는 유지될 수 없기 때문에 valve 3을 닫는 조작 [Close(Valve 3)]을 행하여야 운전목표를 달성할 수 있다. 즉, 하부목표 Drain(Reactor)=True와 Aperture(Valve 3)=Close 사이에 상호작용이 존재한다. 만일 Waitfor(2min)보다 Close(Valve 3)을 먼저 행한 경우에는 Drain(Reactor)=True를 달성할 수 없기 때문에 후진을 해야 한다.

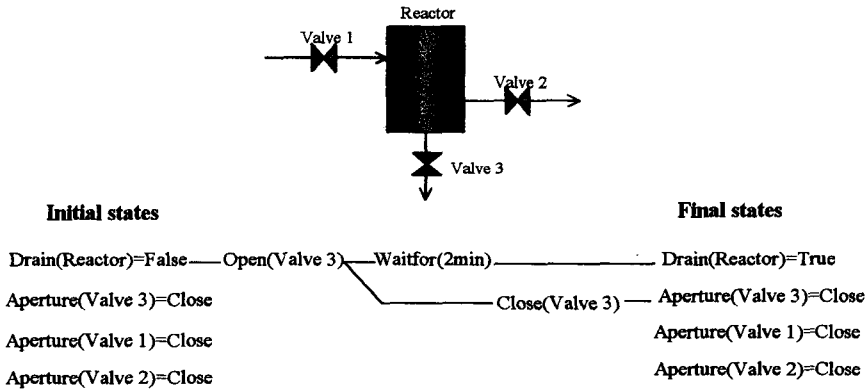


Fig. 6. Subgoal interaction

위와 같은 문제를 해결하기 위해서, 하부목표들 사이의 상호작용을 합성하는 과정 중에 조작의 순서를 미리 결정하지 않고 순서결정 제약조건(post order constraint)을 부가하므로 조작절차를 합성한다(Fig. 7).

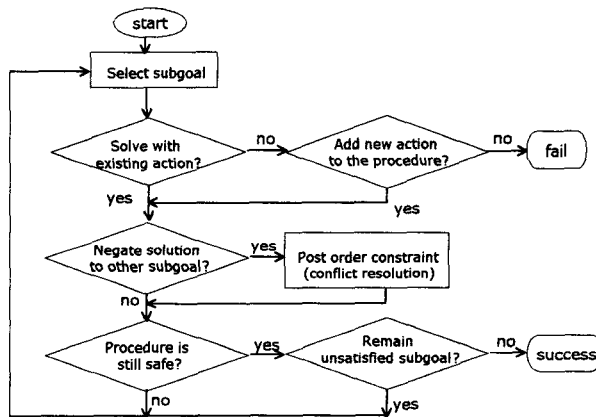


Fig. 7. Algorithm for operating procedure synthesis

4. 적용 및 결과

펌프(pump1)와 히터(heater1)를 조작하여 컬럼(column1)으로 물질을 예열, 순환시키는 공정(Fig. 8)에 적용하여 제시된 방법의 타당성을 입증하였다. 공정의 운전 제한조건은 "펌프를 가동하기 전에 히터를 작동시켜서는 안 된다"이다. 왜냐하면 물질의 흐름이 존재하지 않는 경우, 히

터의 파손이 우려되기 때문이다.

제시된 방법으로 조작절차를 합성한 결과는, Plan ordering constraints[(< (Action2: START-PUMP1) (Action3: START-HEATER1))] -“펌프를 먼저 가동하고 난 다음, 히터를 작동하라”- 로 제시하였다.

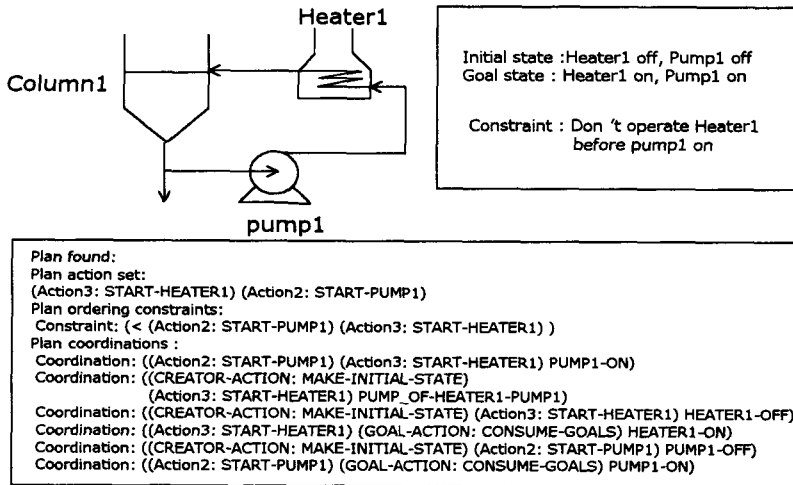


Fig. 8. Case study

향후, 조작절차 합성을 위한 공정모델의 라이브러리 작성과 제한조건을 이용한 보다 효율적인 합성전략의 개발로 복잡한 공정의 보다 구체적이고 실질적인 조작절차를 합성하고자 한다.

감사의 글

본 연구는 포항공과대학교 공정산업의 지능자동화 연구센터를 통한 한국과학재단 우수연구센터 지원금에 의한 것입니다.

참고문헌

1. Chapman, D.: "Planning for conjunction goals", *Artificial Intelligence*, 32(3), 287-301(1987)
2. Lakshmanan R. and Stephanopoulos G.: "Synthesis Of Operating Procedures for Complete Chemical Plants-I. Hierarchical Structural Modeling for Nonlinear Planning", *Comp. & Chem. Eng.*, 12, 985-1002(1988)