

부분순서 알고리즘을 이용한 화학공장의 운전절차 합성

허보경 · 황규석[†]

부산대학교 화학공학과
(2000년 7월 24일 접수, 2001년 4월 19일 채택)

Operating Procedure Synthesis of Chemical Plants using Partial Order Algorithm

Bo Kyeng Hou and Kyu Suk Hwang[†]

Dept. of Chem. Eng., Pusan National University, Pusan 609-735, Korea
(Received 24 July 2000; accepted 19 April 2001)

요 약

운전절차 자동합성에 관한 기존 연구의 한계점을 극복하기 위해 본 연구에서는 운전절차 합성에 관한 새로운 접근법을 개발하였다. 이것은 부분순서 계획 알고리즘을 기반으로 하여 안전과 밸브의 시퀀스 문제를 해결하는 새로운 접근 방법이다. 제안된 시스템은 주어진 제약조건을 침해하지 않고 주어진 목표 세트를 만족시키기 위해 조작연산자로부터 조작을 선택하여 그 순서를 결정한다. 제안된 시스템으로 기존의 시스템이 풀 수 없었던 문제를 해결할 수 있음을 증명하였다.

Abstract – In this study we have proposed a new operating procedure synthesis (OPS) methodology to overcome the limitations of the earlier work. The proposed approach is based on partial order planning algorithm to solve the safety and valve sequencing problems, and choose and order actions from the operators so as to achieve a given set of operational goals with satisfying a given set of safety constraints. We have demonstrated that the proposed OPS system is capable of solving problems that no previous system could solve.

Key words: Operating Procedure Synthesis, Partial Order Algorithm, Safety

1. 서 론

화학공장은 다른 어떤 산업보다도 공정에서의 위험성을 예방차원에 서 관리하여야 하며 그 관리는 체계적이어야 함을 강조할 수밖에 없다. 또한 경제성장과 더불어 화학공장들이 점차 대형화되고 업종이 다양화 되어감에 따라 사고발생 빈도가 점차 증가하고 있으며 사고에 따른 인명 및 재산피해도 점점 대형화되고 있는 추세에 있다. 또한 개시 조업의 지연 사유를 살펴보면, 장치 부족으로 인한 지연이 40-61%, 디자인의 부적합으로 인한 지연이 10%, 조작 실수로 인한 지연이 13-30%를 차지하고 있다.

따라서 화학공장의 개시 조업의 자동화와 안전성을 향상시키기 위해서는 개시 조업에 필요한 운전절차를 합성하고 분석하는 작업이 필요하다. 즉 공정의 초기상태(initial state)로부터 최종 목표상태(final or goal state)로 제약조건(constraint)을 침범(violation)하지 않고 안전하게 운전 목표를 달성시키는데 필요한 운전절차를 합성하는 것이 요구된다[1-7]. 기술적으로 완벽한 표준 운전절차(Standard Operating Procedure, SOP)는 OSHA의 공정안전관리 기준을 만족해야 하며 공장의 개시·정지·비상사태·보수유지 조업 시에 행해야 되는 일, 기록해야 될 데이터, 유지해야 되는 운전상황, 수집해야 될 샘플(sample), 이상상황 발생과 연관되어 있는 경고(alarms), 그리고 대처 조작 등이 명시되어 있어야 한다[8-11].

대부분의 운전절차 합성(Operating Procedure Synthesis, OPS) 시스템은 공정에 적용될 수 있는 조작들(예, 밸브를 열거나)과 피해야 될 위험 상황(hazardous condition; 예를 들어, 수소와 산소는 혼합되어서는 안 된다는 조건), 그리고 공정의 각 구성요소 사이의 관계(예, pipe-1은 valve-1과 연결되어 있다) 등이 정의되어 있는 지식베이스(knowledge base)를 이용한다. 이러한 지식베이스 기반 OPS 시스템은 지식을 재사용할 수 있을 뿐 아니라 플랜트의 구조가 바뀌는 경우, 시스템의 수정에 소요되는 시간이 적게 걸린다는 장점을 가지고 있다[12, 13]. 그러나 기존 OPS 시스템은 지식 표현과 알고리즘에 많은 문제점을 가지고 있어 운전목표 사이에 매우 심한 상호작용(interaction)을 보이는 대규모 화학공장에 적용할 경우, 조합폭발 문제(combinatorial explosion)를 수반하여 해를 발견하지 못하는 단점을 가지고 있다[14-16].

따라서 본 연구에서는 기존의 전순서(total-order) 알고리즘의 단점을 극복하기 위해 부분순서(partial-order) 알고리즘을 기반으로 한 화학공장의 운전절차 합성기법을 제시하고 화학공장에 적용하여 그 유효성을 검증하였다.

2. 기본 개념

인공지능 분야의 계획법은 Fikes와 Nilsson이 STRIPS(Stanford Research Institute Problem Solver)를 개발한 이후 크게 발전하여 다양한 문제 영역에 응용되었다. 일반적으로 계획법에 필요한 구성요소는 domain,

[†]E-mail: kshwang@hyowon.pusan.ac.kr

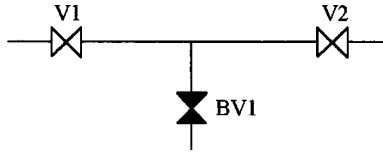


Fig. 1. A double block and bleed valve arrangement.

Table 1. Example statements in the first order predicate calculus

Predicate	Meaning
open(valve-1)	valve-1 is open
\neg activate(heater-104)	heater-104 is not active
contents(pipe-5, methane)	pipe-5 contains methane
pressure(vessel-1, medium)	vessel-1 is at medium pressure

Table 2. Example statements in the function literal representation

Predicate	Meaning
aperture(valve-1) is open	valve-1 is open
state(heater-104) is off	heater-104 is not active
contents(pipe-5, methane) is true	pipe-5 contains methane
pressure(vessel-1) is medium	vessel-1 is at medium pressure

state, action이 있다. 예를 들어, Fig. 1과 같이 두 개의 block 밸브(V1, V2)와 한 개의 bleed 밸브(BV1)로 구성되어 있는 공정(domain)에서 운전목표의 달성에 필요한 밸브의 on/off 조작(action)은 각 밸브와 파이프의 상태(state)에 따라 어떤 밸브를 열고 닫아야 하는지를 명확하게 이해해야 가능하다. 어떤 시점에서의 state는 Table 1과 같은 서술 논리(predicate logic)나 명제 논리(propositional logic)로 구성하지만, 보다 상세한 정보는 Table 2와 같은 function literals를 이용한다.

기존의 OPS 문제에서도 많이 응용된 STRIPS의 조작 표현법은 각 조작을 예비조건 리스트(precondition-list), 삽입 리스트(add-list), 삭제 리스트(delete-list)의 세 부분으로 나누어 모델링(modeling)한다. 여기서, 예비조건 리스트는 그 조작이 실행되기 위해 미리 만족되어야 하는 사실(fact)을 나타내고 삽입 리스트와 삭제 리스트는 그 조작의 실행 후에 나타나는 효과를 기술한다. 즉, 예비조건 리스트의 사실들을 모두 만족하는 조작의 실행은 상태공간에 삽입 리스트의 사실들을 추가하거나 삭제 리스트에 있는 사실들을 삭제하여 새로운 상태를 생성시킨다. 예를 들어, 압축기 조작을 나타내는 STRIPS 연산자에서 예비조건(preconditions)은 압축기가 정지(off)되어 있어야 함을 나타내며, 효과(effects)는 압축기의 가동(on)에 의해 유량이 커지고 압력이 높아짐을 의미한다(Fig. 2). 또한 Fig. 4는 forced reboiler 공정(Fig. 3)에서 pump-201을 가동시킨 후, domain의 상태가 어떻게 변화하는지를 잘 보여 주고 있다. 그러나 화학공장은 장치 사이에 매우 복잡한 형태의 상호작용과 동적 거동이 발생하므로 STRIPS 기반의 조작 표현은 적합하지 않다.

2-1. 선형 계획시스템

STRIPS 시스템으로 대표되는 선형 계획시스템들(linear planning systems)은 주어진 목표들을 임의 순서대로 한 번에 하나씩 차례대로 달성해가

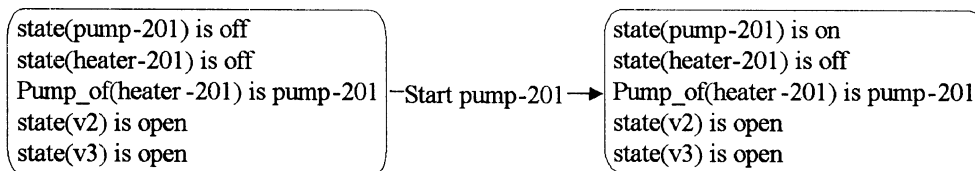


Fig. 4. State transition caused by starting pump-201.

```

operator Start-compressor
{
  compressor ?c;
  effects
  bulk-flow of ?s is high;
  pressure of ?s is high;
  state of c? is on;
  preconditions
  state of ?c is off;
  end
}
    
```

Fig. 2. Example of a STRIPS-like operator.

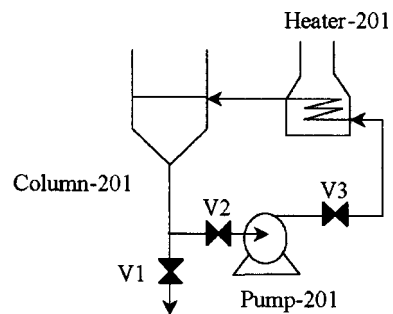


Fig. 3. A forced reboiler shown with valves.

면서 그때마다 필요한 조작들을 계획(plan)에 선형적으로 추가해간다. 여기서, 계획이란 조작이 추가되는 시퀀스(sequence) 리스트를 의미하며 이것은 OPS 분야의 운전절차에 해당한다.

이러한 선형 계획수립 방식은 제어구조가 단순하여 구현하기 쉬운 장점이 있으나 목표들 간의 상호작용으로 인해 경비가 많이 소요되는 후퇴(backtracking)를 자주 유발하게 되고 계획수립 과정 초기부터 중요도가 낮은 세부사항들까지 결정을 내려야 함으로써 탐색공간이 넓어져 복잡도가 크다는 단점을 가지고 있다. 그러므로 조작들 사이에 복잡한 상호작용을 내포하고 있는 화학공장에 적용하기에는 많은 한계점을 가진다. 다시 말해 화학공장은 복잡한 장치로 서로 연결되어 있는 네트워크 구조를 가지므로 어떤 조작의 효과는 연결되어 있는 다른 장치로 전파되어 다른 장치의 공정 상태와 상호 간섭되기 때문이다.

2-2. 비선형 계획시스템

선형 계획시스템들의 단점을 극복하고자 하는 노력으로 비선형 계획수립 방식인 계층적 계획법(hierarchical planning) 방식과 부분순서 계획법(partial-order planning)이 등장하게 되었다. ABSTRIPS에서 처음 시도된 계층적 계획법은 각 조작들의 예비조건들 중에서 중요도가 낮은 것들을 차례대로 삭제함으로써 목표들을 계층적으로 추상화하고 최상위 레벨의 추상목표들을 만족시키는 한 추상계획(abstract plan)을 먼

저 구해낸 다음, 이것을 미리 정의해둔 추상계층(abstraction hierarchy)에 따라 세분화함으로써 계획을 수립하는 방식이다. 이 방식에 따르면 별로 중요하지 않은 세부사항들에 대한 고려는 뒤로 미루어 쉽게 추상 계획을 구할 수 있어 탐색공간을 대폭 축소시켜 준다.

한편 NOAH 시스템으로부터 시작된 부분순서 계획수립 방식은 최소 결정 전략(least commitment strategy)을 사용하는 방식으로서 한 계획을 이루는 조각들의 부분순서와 조각들에 포함된 변수들의 바인딩(binding)에 대한 부분제약을 허용함으로써 계획수립 과정 동안 꼭 필요한 조각들과 그들에 대한 제약들만을 점진적으로 부가해 나간다. 부분순서 계획시스템들은 선형 계획시스템들이나 전순서(total order)계획시스템들에 비해 불필요한 임의 결정들에 의한 후퇴 발생을 감소시킬 수 있으며, 계획을 반드시 선형적으로 확장하지 않고 임의 방향으로 확장할 수 있다는 장점이 있다.

부분순서 계획시스템에서는 하나의 독립적인 부분순서 계획수립 공간으로 볼 수 있는 각 추상 레벨 내에서의 탐색 효율에 큰 영향을 미치는 제어요소로는 다음과 같은 것이 있다. 즉, (1) 부분계획들 중 어느 것을 선택하느냐는 계획 선택 제어와 (2) 선택된 계획이 포함하고 있을 미달성 목표들과 위협(threat)들 중 어느 것을 먼저 해결할 것인지를 결정하는 목표/위협 선택 제어, (3) 미달성 목표들 중 어느 것을 우선적으로 성취할 것인지를 결정하는 목표 선택 제어, (4) 어떤 유형의 위협부터 먼저 제거해야할지를 결정하는 위협 선택 제어 등이다.

3. OPS 시스템의 구성요소

3-1. Planner

OPS 시스템에서의 planner는 각 에이전트(agent)를 이용하여 제약조건들을 침해하지 않고 공정의 초기상태에서 원하는 목표상태로 안전하게 도달할 수 있는지의 여부를 판단하는 역할을 수행한다(Fig. 5).

상태그래프(state graph)를 기반으로 한 OPS는 현재 상태에서 원하는 목표상태에 도달하기 위해서 필요한 경로를 그래프상에서 탐색하여 발견한다. 그러나 이 OPS 시스템에서는 벨브의 모델링과 안전운전에 관계된 문제를 풀기 위한 모델링이 그래프 내에 포함되어야 한다.

Means-ends 방법(초기상태와 목표상태의 차이를 비교하여 그 차이를 줄이는데 필요한 조작을 행하는 방법)은 경험 규칙을 사용하기 위해 후향추론(backward chaining or regression)을 이용해야 한다. 또한 운전절차가 안전한지를 검증하기 위해 운전절차의 영향을 한 단계씩 조사해야 한다. 후향추론과 least commitment 접근법을 사용하는 planner의 경우, 위험성 체크 알고리즘이 대단히 복잡하며 action synergy(조작에 의한 공정상태 변화의 모사)와 같은 문제를 해결하기 어렵다.

대부분의 전향추론(forward chaining or progression)을 사용하는 OPS 시스템은 계획수립(운전절차 합성)하는데 많은 시간을 요한다. Fusillo와 Powers[4], Rivas와 Rudd[15] 시스템은 시스템의 효율을 향상시키기

위해 용납할 수 없는 경험규칙(positive operator)을 사용하였다. Hwang [7, 8] 등은 전향추론과 후향추론을 사용한 하이브리드(hybrid) 시스템을 채용하고 있으나, 전향추론엔진은 후향추론에 의해 발견된 중간목표와 통합하여 조각의 순서를 바로잡기 위해 다시 계획수립이 이루어져야 된다는 단점을 가지고 있다. 또한 조각들을 먼저 생성하고 나서 순서를 결정함으로써 subplan 사이의 충돌과 상호작용을 해결하기 위해 재계획(replanning) 수립이 필요하다.

3-2. Safety evaluator

전역 제약조건(global constraint)은 단순히 위험상태/상황들을 표현한 것으로 Fusillo와 Powers에 의해서 처음으로 사용되었다. 예를 들어, 'Do not mix chlorine and methane unless the system temperature is high'와 같은 것이다. 이러한 제약조건은 위험상황을 나타내는 리스트로 표현되며 어떤 공정상태에서의 위험상황을 나타내는 리스트의 모든 요소들의 값이 참이면 해당 위험상황이 존재한다고 간주된다.

전역 제약조건의 침범 여부를 확인하기 위해 각 계획수립 과정 중에 있는 운전절차를 체크한다. 만일 운전절차가 안전한 상태에서 시작하여 각 계획수립 과정 중에 전역 제약조건을 침해하지 않는다면, 완성된 최종 운전절차도 안전한 것으로 간주한다. 위험상황을 일으키는 것이 명백한 운전절차들을 계속 진행시킨다는 것은 시간 낭비이므로 완성된 운전절차에 대하여 안전체크를 하는 것보다 진행중인 운전절차에 대하여 안전성을 체크하는 것이 더 효율적이다. 그러나 Rivas와 Rudd가 개발한 OPS 시스템을 제외한 모든 시스템들은 특정한 공정의 구간 내에만 존재하는 전역적 제약조건을 적절하게 다루지 못하고 있다.

3-3. Process simulator

STRIPS 기반의 planner는 비교적 간단한 언어를 사용하여 묘사되며 각 조작은 예비조건과 효과를 나타내는 요소로 구성되어 있다. 어떤 상태에서 어떤 조작을 행하기 전에 조작 예비조건 요소들이 모두 참이면 조작이 실행되어 효과가 나타난다.

그러나 STRIPS 기반의 planner를 실제계에 적용하기엔 많은 한계를 가지고 있다. 왜냐하면 조작의 그룹은 단지 개개의 조작 실행으로는 보일 수 없는 효과를 나타낼 수 있기 때문이다. 이것이 바로 action synergy이다. 특히, 이러한 action synergy는 조작에 의한 공정상태의 변화 모사와 벨브의 조작 시퀀스의 결정에 매우 중요하게 사용되는 요소이다.

4. 부분순서 운전절차 합성알고리즘

4-1. 계산시간을 줄이는 방법

최소결정 전략은 여러 가지 대안이 존재할 경우에 선택 가능한 조작 세트를 한정하여 상태공간의 범위를 축소화할 수 있어 매우 유용하다. 예를 들어, 파이프로부터 응축되어 있는 물질을 제거하고자 할 경우, 필요한 조작이 'purge with chemical ?c'라고 한다면, 이때에 변수 ?c를 끄거나 온 상태의 질소나 산소로 한정시켜 이 변수에 따른 물질이 바인딩되는 것을 막아주므로 후퇴를 최소화할 수 있다는 장점을 가지고 있다.

부분순서 계획법은 최소결정 전략으로 각 중간목표(subgoal)를 달성하면서 발생하는 충돌(conflict)을 해결하기 위해 순서 제약조건(ordering constraint)을 부가한다. 여기서, 순서 제약조건은 $x < y$ 와 같은 형태로 표현되고 그 의미는 'action x comes before action y'를 말하며 충돌은 임의의 목표가 한번 달성된 후에 어떤 조작이 그 목표를 부정(negation)하는 경우를 말한다. 모든 중간목표들이 달성되면 각 중간목표들 사이의 상호 인과관계를 나타내는 순서 제약조건을 이용하여 조각의 순서를 재조정한다.

펌프(pump1)와 히터(heater1)를 조작하여 컬럼(column1)으로 물질을 예열, 순환시키는 공정(Fig. 3)에 부분순서 계획법을 적용해 보자. 공정의 운전 제약조건은 "펌프를 가동하기 전에 히터를 작동시켜서는 안된

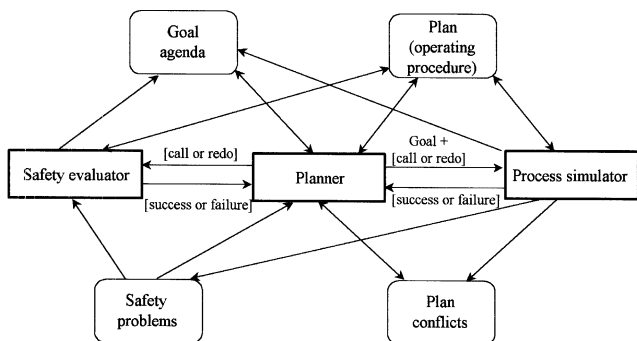


Fig. 5. The structure of the proposed OPS system.

다"이다. 왜냐하면 물질의 흐름이 존재하지 않는 경우, 히터의 파손이 우려되기 때문이다. 위와 같은 방법을 사용하여 운전절차를 합성한 결과, Plan ordering constraints(Action2: START-PUMP1)<(Action3: START-HEATER1)]- '펌프를 먼저 가동하고 난 다음, 히터를 작동하라'-고 제시하였다.

또한 계산시간을 감소시키는 매우 유용한 기법으로 intelligent variable 을 이용하는 방법이 있다. 예를 들어, Fig. 3과 같은 공정에서 조작 'start heater-201'은 heater-201과 연관되어 있는 펌프를 나타내기 위해 변수 ?p를 가지고 있다. 만일 공정에 n개의 펌프가 있다면, ?p는 n번의 서로 다른 선택이 가능하므로 잘못된 조작선택으로 인한 엄청난 후퇴를 수반하게 된다. 이러한 것을 방지하기 위해 codesignation과 noncodesignation 관계를 이용한다. 여기서, codesignation 관계는 'symbol=symbol'의 형태로 표현되어 두 symbol이 같은 값을 가져야 된다는 것을 의미한다. 예를 들어, '?x=pump-1'은 변수 ?x의 값이 pump-1이어야 한다는 것을 나타낸다. 또한 noncodesignation 관계는 'symbol≠symbol'로 표현되어 두 symbol이 같은 값을 가져서는 안됨을 나타낸다. 예를 들어, '?x ≠?y'는 변수 ?x의 값과 변수 ?y의 값이 다르다는 것을 의미한다.

4-2. 중간목표 사이에 존재하는 상호작용을 해결하는 방법

화학공장의 운전목표 달성은 각 세부 목표들을 차례로 수행해야 할 뿐만 아니라 중간목표 사이의 시간적이고 인과적인 관계를 모두 만족시켜야만 가능하다. 일반적으로 특정 장치의 운전절차는 정형화되어 (formalized) 있는 경우가 많으며 특정 공정의 운전절차도 어느 정도의 규칙을 가지고 실행되는 경우가 많으므로 서로 연관성이 있는 부분을 기능 단위별로 계층화할 수 있다. 예를 들어, 'the compressor is isolated' 라는 상위 레벨(high-level)의 운전목표는 'the suction, discharge and bleed valves of the compressor are all closed'로 세 가지 하위 레벨(low-level)의 운전목표로 계층화가 가능하다. 이와 같은 방법은 각 목표들 사이에 존재하는 상호작용을 미리 계층적으로 표현하여 두는 것이다.

그러나 이와 같은 방법은 상위 레벨의 운전목표와 하위 레벨의 세부 운전목표들 사이의 일치성을 유지해야 되는 ramification problem을 일으킨다. 예를 들어, Fig. 6과 같은 압축기의 정지조업을 위한 운전목표는 Fig. 7과 같이 나눌 수 있으므로 상위 레벨의 운전상황인 'isolated (compressor) is true'는 하위 레벨의 운전상황인 'aperture(suction) is closed', 'aperture(bleed) is closed', 그리고 'aperture(discharge) is closed'로 세분화된다(Fig. 8). 그런데 Fig. 9와 같은 합성과정에서 조작 depressure의 예비조건과 운전상황인 'isolated(compressor) is closed'와의 충돌을 인식하지 못하므로 bleed 밸브를 닫는 조작을 수행하지 않는다. 즉, 시스

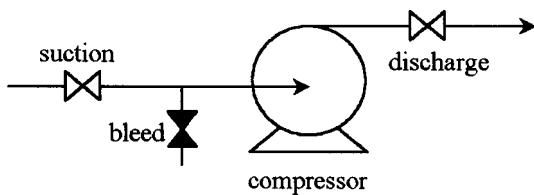


Fig. 6. The compressor problem.

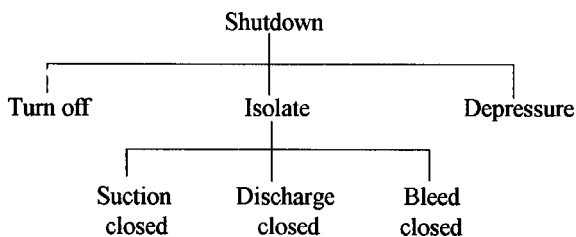


Fig. 7. The breakdown of the goals in the compressor problem.

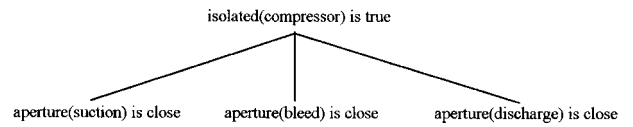


Fig. 8. A hierarchy of literals.

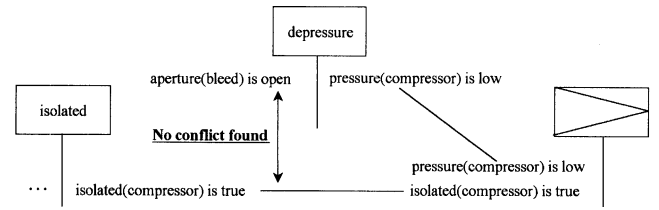


Fig. 9. An example of the ramification problem.

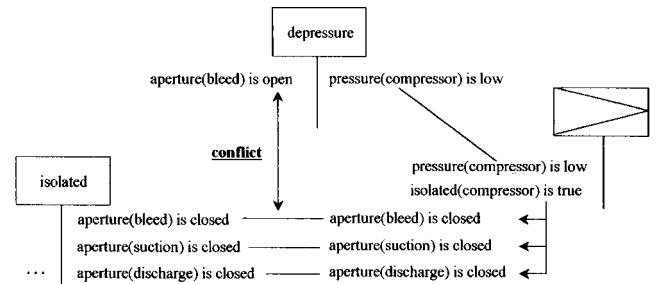


Fig. 10. Conflict in a plan.

템은 'isolated(compressor) is true'와 'aperture(bleed) is closed' 사이의 관계를 이해하지 못한다. 이러한 ramification의 문제는 모든 운전상황을 더 이상 확장이 불가능한 기본상황(primitive conditions)으로 확장하여 해결할 수 있다. 이와 같은 문제점을 해결하기 위해 상위 레벨의 'isolated (compressor) is true' 상황을 기본상황인 'aperture(suction) is closed', 'aperture (bleed) is closed', 'aperture(discharge) is closed'로 확장하면, 조작 depressure를 행함으로 발생하는 충돌을 발견할 수 있다(Fig. 10). 이와 같은 것을 목표확장(goal expansion)이라고 한다.

그러나 계층적인 구조의 형태로 운전목표를 세분화하여 표현한다고 할지라도 역시 어떤 목표를 먼저 선택하여 실행하느냐의 문제를 해결해야 한다. 이 과정에서 적용될 수 있는 방법으로는 각 중간목표에 우선 순위를 미리 정해두고 그 순서대로 적용하는 방법이 있다. 예를 들어, 밸브의 on/off 조작보다 물질 흐름을 위해 필요한 경로를 먼저 발견하도록 우선 순위를 정해 놓는 것을 말한다. 즉, 상위 레벨의 운전목표를 먼저 달성한 다음, 하위 레벨의 운전목표를 달성하도록 하여 잘못된 상위 레벨의 운전목표 선택으로 인한 후퇴를 방지할 수 있다. 이 방법은 중간목표와 조작 선택의 실수를 운전절차 생성의 초기 단계에서 미리 막아 후퇴를 줄일 수 있는 반면, 가능성이 있는 다른 운전절차들을 제거하는 역효과를 가진다.

중간목표 사이의 상호작용을 해결하기 위해 사용될 수 있는 다른 방법으로 조작의 세트인 매크로 조작(macro action)이 있다. 예를 들어, 압축기의 특별한 타입(type)은 압축기를 정지(off)시킨 후, 모터의 스핀(spin)이 정지할 때까지 기다려야 되는 경우가 있다. 즉, 모터의 스핀이 정지할 때까지는 압축기가 활동중인(active) 것으로 간주되게 모델링(modeling) 되었다는 의미이다. 이 경우에 해당하는 매크로 조작은 Fig. 11과 같이 표현된다. 압축기의 정지를 위해 물론 앞에서 언급한 목표확장을 이용하여 표현하는 방법도 가능하다(Fig. 12).

압축기를 수리하기 위해 작동을 중지시키고 난 후에 다시 압축기를 가동시키는 경우, 수리하는 동안에는 압축기가 항상 중지 상태에 놓여 있어야 한다. 그러나 매크로 조작을 사용하지 않고 목표확장을 이용한다면, OPS 시스템은 수리하는 동안에 압축기를 가동시키는 조작이 행

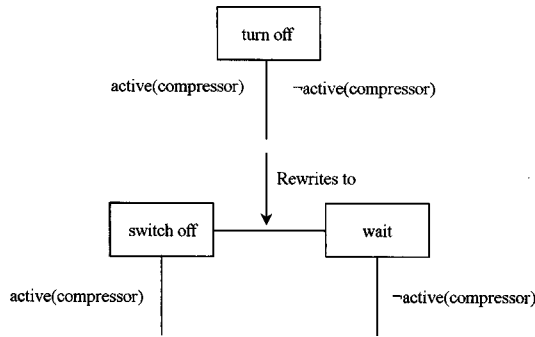


Fig. 11. The macro action representation of turning off a compressor.

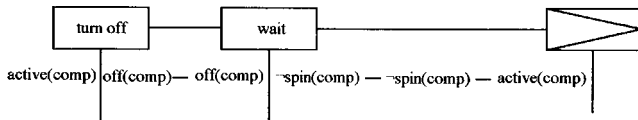


Fig. 12. The goal expansion representation of turning off a compressor.

```

Operator StartReboiler {
  heater ?boiler;      ;;Variable ?m boiler is defined as boiler
  pump ?pump;
  column ?c;

  expand [reboiler-state of ?c is on;] ;; The effect of StartReboiler
  using [heater of ?c is ?boiler;      ;; The precondition of StartHeater
  pump of ?c is ?pump;
  state of ?boiler is on;
  state of ?pump is on;]
}
    
```

Fig. 13. The representation of the expansion operator.

해지는 것을 막을 수 없다. 따라서 매크로 조작은 중간목표들 사이에 존재하는 복잡한 인과관계와 상호작용을 해결하는데 매우 유용한 수단으로 이용할 수 있다.

본 연구에서는 위와 같은 다양한 기법을 사용하기 위해 목표확장이 가능한 조작연산자를 Fig. 13과 같이 제시한다.

4.3. 부분순서 운전절차 합성 알고리즘의 개요

기본적인 비선형 부분순서(nonlinear partial-ordered) OPS 알고리즘은 Fig. 14와 같다. 비선형 부분순서 알고리즘이 어떻게 작동되는지를 알아

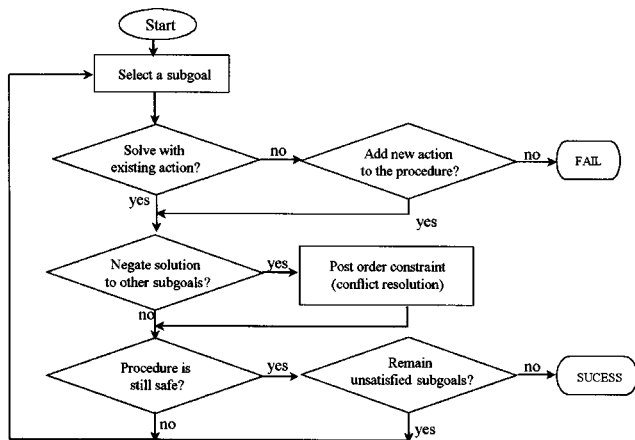


Fig. 14. Basic algorithm for partial ordered OPS.

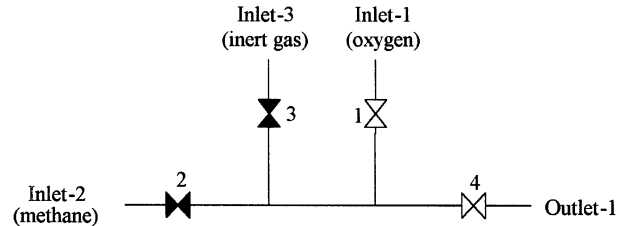


Fig. 15. Example of planning problem.

Table 3. Data of planning problem

Item	Contents
Initial state	(flowing oxygen) not(explosion)
Goal state	(flowing methane) not(explosion)
Operators	(stop-flow x) pre-conditions: (flowing x) post-conditions: not(flowing x) (establish-flow x) pre-conditions: () post-conditions: (flowing x) (purge x) pre-conditions: not(equal inert-gas x) post-conditions: not(present x) (flowing inert-gas)
Frame axioms	(present methane) & (present oxygen) => (explosion) (flowing x) => (present x)

보기 위해 Fig. 15와 같은 공정에 적용해 보기로 하자. 즉, 공정의 초기에 산소가 inlet-1에서 outlet-1로 흐르고 있다. 산소와 탄화수소의 혼합에 의한 폭발이 발생되지 않도록 하면서 메탄을 inlet-2에서 outlet-1로 흐르게 하여 최종 목표상태를 달성시키는 운전절차를 합성하는 문제이다. 각 조작연산자의 예비조건과 효과, 상황(frame axioms) 판단 규칙은 Table 3과 같이 정의한다.

비선형 부분순서 알고리즘인 경우, 목표상태의 목표 중에서 하나인 (flowing methane)을 선택한 후, 그 목표를 달성하는데 필요한 조작연산자 (establish x)의 x에 methane을 바인딩하여 실행하면 (explosion)이 발생함을 planner는 인식한다. 그러나 비선형 알고리즘은 후퇴를 시도하지 않고 위험을 유발시키는 조작연산자의 효과를 완화시킬 수 있는 다른 조작연산자들을 찾아서 미리 실행하므로 위험이 발생하지 않도록 한다. 이러한 방법을 'promotion'이라고 한다. 즉, 이 예제의 경우, (explosion)을 막기 위해 (establish hydrocarbon)을 행하기 전에 미리 (purge x)를 행한다. 선형 알고리즘인 경우, 모든 변수는 즉시 바인딩 되지만 비선형적 알고리즘은 변수 x를 당분간 바인딩하지 않은 상태로 둔다. 현재 생성되어 있는 중간목표는 {not(flowing x) and (equal inert-gas x)}이다. 여기서, (flowing methane)은 목표상태에서 요구하고 있기 때문에 x의 유일한 바인딩 대상은 oxygen뿐이다. 따라서 비선형 알고리즘은 not (flowing oxygen)을 달성하기 위해 stop-flow 조작연산자를 사용하여 성공시킨다. 이렇게 하여 생성된 운전절차는 (stop-flow oxygen) > (purge oxygen) > (establish-flow methane)과 같다. 위 결과를 요약하면, 선형 전순서 알고리즘은 엄청난 탐색 경비를 소요하지만 비선형 알고리즘은 후퇴를 하지 않고 실행 가능한 운전절차를 합성해 줄 수 있다.

5. 적용 사례

Fig. 16과 같은 공정에서 히터(heater)와 압축기(compressor)의 가동,

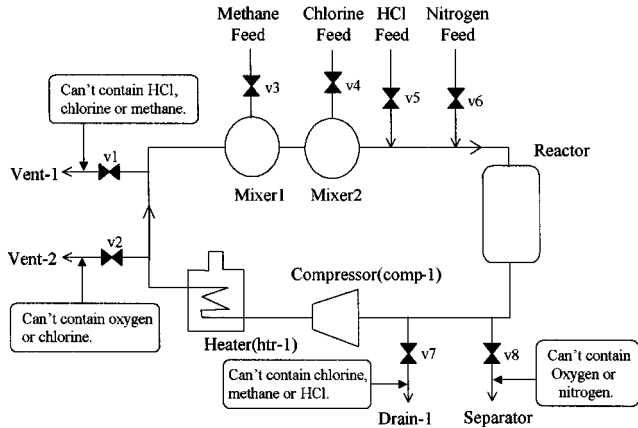


Fig. 16. Simplified flowsheet of chlorination process.

Table 4. Process constraints

1	not{contains(chlorine) and contains(methane) and temperature != high}
2	not{contains(water) and contains(HCl)}
3	not{contains(oxygen) and temperature != low}
4	not{contains(methane) and contains(oxygen)}
5	not{vent1 = open and contains(?any_toxic)}
6	not{drain1 = open and contains(?any_toxic)}
7	not{vent2 = open and contains(oxygen)}
8	not{vent2 = open and contains(chlorine)}
9	not{separator_line = open and contains(?any_non_condensable)}
10	?any_non_condensable = {oxygen, nitrogen}
11	?any_toxic = {chlorine, methane, HCl, chloronated_hydrogen}

Table 5. Initial and final state

Initial state	Final state
temperature=low	temperature=high
pressure=low	pressure=high
contains(system, nitrogen)	contains(system, HCl)
contains(system, oxygen)	contains(system, methane)
contains(system, water)	contains(system, chlorine)
closed(all valves)	open(v3)
	open(v4)
	open(v5)
	open(v8)

메탄(methane)과 염소(chlorine)의 유입구에서의 흐름생성과 같은 네 가지의 운전목표를 만족시키는 운전절차를 합성하고자 하였다. 만족되어야 하는 제약조건과 초기·최종 공정상태는 각각 Table 4, 5와 같다.

이 문제의 해답은 먼저 압축기를 가동하고 나서 히터를 작동시킨 후에 메탄을 유입한 다음, 마지막으로 염소를 유입시키는 것이다. 합성된 최종 운전절차는 다음과 같다.

- (1) Open valve v1.
- (2) Start compressor comp-1.
- (3) Start heater htr-1.
- (4) Close valve v1.
- (5) Open valve v7.
- (6) Close valve v7.
- (7) Open valve v2.
- (8) Open valve v8.
- (9) Open valve v3.

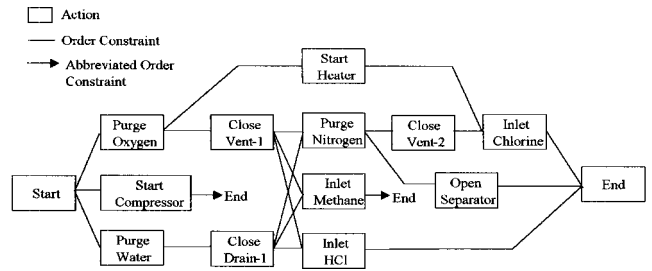


Fig. 17. The partially ordered operating procedure.

- (10) Open valve v5.
- (11) Close valve v2.
- (12) Open valve v4.

전향후론에 의한 선형 전순서 합성 알고리즘은 목적 지향성의 부족과 연속적인 조작 시퀀스를 행하므로 병렬조작의 표현이 어렵고 잘못된 조작의 선택으로 엄청난 후퇴를 수반하지만, 비선형 부분순서 알고리즘은 Fig. 17에서 볼 수 있는 것과 같이 합성하는 과정 중에 조작의 순서를 미리 결정하지 않고 순서결정 제약조건만을 부가하므로 후퇴 감소와 병렬 조작 시퀀스가 가능하다.

6. 결 론

기존의 선형 전순서(linear total-ordered) OPS 시스템은 병렬조작을 표현하지 못할 뿐 아니라 많은 후퇴를 수반하므로 엄청난 계산시간을 요구하거나 해를 발견하지 못하는 경우가 많다. 이러한 단점을 해결하기 위해 제안된 비선형 부분순서 OPS 알고리즘을 개선한 비선형 제약부가 알고리즘은, 최소결정 전략을 사용하여 한 계획을 이루는 조작들간의 순서와 이 조작들에 포함된 변수들의 바인딩에 대한 부분제약(partial constraints)을 허용함으로써 계획수립 과정 동안 꼭 필요한 조작연산자들과 그들에 대한 제약들만을 점진적으로 부가해 나간다. 또한 탐색의 효율성과 목표들의 효율적인 추상화를 위해서 계층적인 구조를 가지며 특정 공정의 제약조건과 운전 특성을 표현할 수 있는 방법을 사용한다. 제안된 시스템을 제약조건이 복잡한 화학공정의 운전절차 합성에 적용해 본 결과 그 유효성을 검증할 수 있었다. 향후에는 수정된 플랜트 구조에 적용할 수 있는 운전절차의 합성과 함께 공정의 디자인 초기단계에서 운전성(operability)을 검증할 수 있는 OPS 시스템의 개발이 요구된다고 생각된다.

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(1991).

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).