

# 시간논리와 목표트리를 이용한 화학공장의 운전절차 합성 알고리즘

허보경, 황규석  
부산대학교 화학공학과

## Algorithm for Operating Procedure Synthesis of chemical Plants using Temporal logic and Goal Tree

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

### 1. 서론

화학공장의 자동화와 안전성 향상을 위해서는 공장의 개시·정지·비상사태·보수유지 조업에 관한 운전절차를 합성하고 분석하는 작업이 필요하다. 공정의 초기상태(initial state)로부터 최종 목표상태(final or goal state)로 안전하게 운전 목표를 달성시키는데 필요한 장치의 조작(operation) 절차를 생성하는 운전절차 합성(Operating Procedure Synthesis, OPS) 시스템은, 오류가 없는 운전절차를 합성하는데 도움을 주고 특정 공정에 대한 전문가들의 지식을 구조·문서화하여 보관하고 공유할 수 있다는 장점과 함께 조업자 교육용 시스템으로의 활용 뿐 아니라 공정의 이상상황(abnormal condition)에 유연하게 대처할 수 있는 시스템으로 발전할 수 있을 것으로 예상된다.

그러나 기존의 연구자들이 개발한 OPS 시스템은 지식표현이나 합성 알고리즘은 많은 문제점을 가지고 있어 대규모 화학공장의 운전절차를 합성할 수 없다. 따라서 본 연구에서 시간논리와 목표트리를 이용하여 대규모 화학공장의 운전절차를 합성할 수 있는 알고리즘을 제시하고자 한다.

### 2. 기존 OPS 알고리즘의 문제점

기존의 OPS 시스템은 선형조건을 전제로 개발되었으므로 병렬조작을 표현하지 못하며, 많은 후퇴작용으로 인한 시간소요를 유발시키므로 해답을 찾지 못하는 경우가 많다. 이러한 단점을 극복하기 위해 다른 연구자들이 비선형 조건을 다룰 수 있는 시스템을 개발하였으나 목표들 간의 상호작용을 효과적으로 다루지는 못하였다. 또한 조합폭발 문제를 효과적으로 다루지 못하여 매우 단순한 경우의 문제들에만 적용되었다.

기본적인 비선형 OPS 알고리즘의 개선된 형태로는 비선형 제약부가 알고리즘이 존재한다. 이것은 최소결정 전략(least commitment strategy)을 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.

Fig. 1과 같은 공정에 대해 각각 선형과 비선형 방식의 알고리즘을 적용하여 그 장·단점을 논의하기로 하자. 초기에 산소가 inlet 1에서 outlet 1로 흐르고 있는 상태를 산소와 탄화수소의 혼합에 의한 폭발이 발생되지 않도록 하면서 메탄을 inlet 2에서 outlet 2로 흐르게 하여 최종 목표상태를 달성시키는 운전절차를 합성하는 문제이다. 각 조작연산자의 예비조건과 효과, 상황(frame axioms) 판단 규칙은 Table 1과 같이 정의한다.

선형 알고리즘인 경우, 우선 목표상태를 구성하고 있는 목표 중에서 하나를 선택한다. 예를 들어, (flowing methane)을 선택하여 이 목표를 달성하기 위해 필요한 조작연산자 (establish x)의 x에 methane을 바인딩한 후, 적용하지만 변화된 공정상태가 상황 판단 규칙의 첫 번째 규칙을 위배하여 (explosion)을 유도하므로 planner는 후퇴를 하여 다른 목표인 not(explosion)을 선택한다. 이 때에 planner는 (purge x)의 x에 oxygen을 바인딩 할지 아니면 methane을 바인딩 할지를 결정해야 한다. 그러나 선형 알고리즘은 전에 탐색한 경로를 기억하고 있지 않으므로 methane이 선택되어 (flowing methane) 목표는 달성될 수 없게 된다. 이러한 방식은 매우 비효율적이며 단순한 것으로 화학공장의 운전절차 합성에 적합하지 않다.

비선형 알고리즘인 경우, 선형 알고리즘과 마찬가지로 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(Chapman, 1985). 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 바인딩 하지 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 다음과 같다.

(stop-flow oxygen) > (purge oxygen) > (establish-flow methane)

위 결과를 요약하면, 선형 알고리즘은 엄청난 탐색 경비를 소요하지만 비선형 알고리즘은 후퇴를 하지 않고 실행 가능한 운전절차를 합성해 줄 수 있다. 또한 복잡한 화학공장의 운전절차를 합성하기 위해서는 조작과 상황 사이의 의

미론적인 상관관계의 인식이 필요하다. 이것은 조작에 의한 상황의 변화를 인식하기 위해 반드시 필요하며, 공정변수를 구성요소로 하는 상황리스트로써 나타내는 것이 일반적이다. 즉, 지식베이스 내의 지식 중에서 공장모델(plant model)과 운전상황에 관한 경험규칙이 적절하게 묘사되어야 한다.

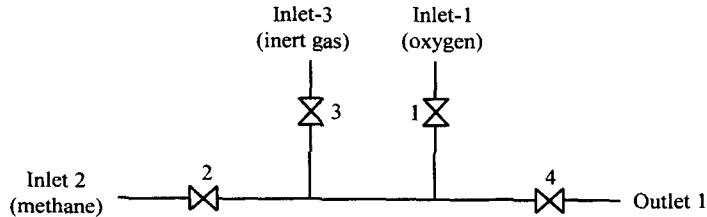


Fig. 1. Example of planning problem.

Table 1. 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)

### 3. 시간논리를 이용한 제약조건의 표현

최초의 OPS 시스템(Rivas and Rudd, 1974)은 다음과 같은 다섯 가지 형태의 위험상황을 체크할 수 있었으나 그 이후에 개발된 시스템들은 모두 다 체크해 주지 못하였다. 즉 (1) 두가지 이상 물질의 혼합, (2) 두 장치사이의 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. 또한 사용되어진 기존의 운전절차 합성 방법들도 많은 단점들을 가지고 있다. 일반적으로 복잡한 공장의 운전절차 합성은 조합 폭발의 문제를 해결하기 위해 경험 지식과 운전상의 정해진 절차들을 필요로 한다. 어떤 운전절차 합성 문제는 운전절차 합성 시스템으로 하여금 시간을 추론할 수 있는 능력을 요구한다.

어떤 문제에서 임의의 사건은 고정된 시간 안에 일어나야 한다. 예를 들어, “특정한 화학물질은 매우 불안정하기 때문에 합성된 후에 가능한 빨리 사용되어야 한다”. 또한 어떤 문제에서 임의의 사건은 다른 사건이 일어난 다음, 어느 정도 시간이 경과된 후에 일어나야 한다. 예를 들어, “배치가 유입되고 3시간 후에 용기의 내용물을 추가한다”. 운전절차 합성 시스템은 이러한 시간적인 제한조건들을 나타내기 위해서 시간에 대한 추론 능력을 가지고 있어야 한다.

본 연구에서는 연산자의 선택을 위한 효과적인 탐색을 사용하기 위해서 공정 제한조건, 경험적인 규칙, 운전상황의 인과관계와 시간적인 정보들을 이용한다. 제한조건을 이용한 탐색제어는 운전절차 합성시스템이 후진(backtracking)하는 것을 방지한다.

따라서 우리는 시간적인 제한조건을 나타내기 위해 일차 선형 시간논리(LTL)를 제안한다. LTL은  $\cup$ (Until),  $\square$ (Always),  $\diamond$ (Eventually),  $\circ$ (Next)와 같은 시간표현 양식과  $\forall$ (All),  $\exists$ (Some)과 같은 한정사(quantifier), 그리고  $\wedge$ (And),  $\vee$ (Or),  $\Rightarrow$ (Implies),  $\neg$ (Not)와 같은 논리 기호(logical symbol)를 이용한다. LTL의 기본적인 작성 규칙은 Table 2의 규칙에 의해 확대된다.

**Table 2. Meaning of temporal modality**

Temporal Modality	Meaning
$\circ f$	$f$ holds in the next world
$\square f$	$f$ holds in current world and in all future worlds
$\diamond f$	$f$ either holds now or in some future world
$f_1 \cup f_2$	Either now or in some future world $f_2$ holds and until that world $f_1$ holds

조작자 선택과 조작 순서화를 위한 지침으로 시간적인 제한조건을 사용하기 위해 과거, 현재, 그리고 미래의 운전상황에 대한 정보를 알아야 한다. 그래서 우리는 운전상황의 변화를 파악하기 위해 운전상황 진행 알고리즘을 제시한다(Fig 2.). 여기서,  $\forall[x:Y(x)]f \equiv \forall x.Y(x) \Rightarrow f$  는  $Y(x)$ 을 만족하는 모든  $x$ 가 참이면,  $f$ 가 유지됨을 의미하고,  $\exists[x:Y(x)]f \equiv \exists x.Y(x) \wedge f$  는  $Y(x)$ 을 만족하는 어떤  $x$ 가 참이면,  $f$ 가 유지됨을 의미한다. 또한 LTL은 Fig. 3과 같은 단축형(abbreviation)을 사용한다.

<b>Inputs:</b> An LTL formula $f$ and a world $w$
<b>Output:</b> A new LTL formula $f'$ representing the progression of $f$ through the world $w$
<b>Algorithm</b> $Progress(f, w)$
<b>Case</b>
1. $f = \emptyset$ ( $\emptyset$ contains no temporal modalities): $f' := \text{TRUE}$ if $w \models f$ , <b>FALSE</b> otherwise
2. $f = f_1 \wedge f_2$ : $f' := Progress(f_1, w) \wedge Progress(f_2, w)$
3. $f = f_1 \vee f_2$ : $f' := Progress(f_1, w) \vee Progress(f_2, w)$
4. $f = \neg f_1$ : $f' := \neg Progress(f_1, w)$
5. $f = \bigcirc f_1$ : $f' := f_1$
6. $f = f_1 \cup f_2$ : $f' := Progress(f_2, w) \vee (Progress(f_1, w) \wedge f)$
7. $f = \diamond f_1$ : $f' := Progress(f_1, w) \vee f$
8. $f = \square f_1$ : $f' := Progress(f_1, w) \wedge f$
9. $f = \forall [x:Y(x)]f_1$ : $f' := \bigwedge_{\{c:w \models Y(x/c)\}} Progress(f_1(x/c), w)$
10. $f = \exists [x:Y(x)]f_1$ : $f' := \bigvee_{\{c:w \models Y(x/c)\}} Progress(f_1(x/c), w)$

**Fig. 2. The progression algorithm of operating conditions.**

$f_1 \vee f_2 \equiv \neg(\neg f_1 \wedge \neg f_2)$
$f_1 \Rightarrow f_2 \equiv \neg f_1 \vee f_2$
$\diamond f \equiv true \cup f$
$false \wedge f$ or $f \wedge false \equiv false$
$true \wedge f$ or $f \wedge true \equiv f$
$\neg true \equiv false, \neg false \equiv true$
$\square f \equiv \neg \diamond \neg f$

**Fig. 3. The standard abbreviations.**

#### 4. 목표트리와 시간논리를 이용한 OPS 알고리즘

최종 운전목표의 달성을 위해 필요한 중간목표들을 목표트리 상에 나타내고 중간목표사이의 시간적 인과관계와 제약조건을 규정하기 위해 시간논리를 사용하는 새로운 OPS 알고리즘을 다음과 같다.

1) 사용자로부터 장치의 연결관계(topology), 초기상태와 목표상태, 목표트리 그리고 제약조건들을 입력받는다. 입력된 데이터의 논리적 타당성은 제약조건 전파로 검증한 다음에 각 장치의 공정변수 값을 초기화한다. 또한 흐름상황 인식 연산자를 이용하여 각 라인의 흐름상태를 파악한 후, 시간논리로 표현된 특정 영역의 운전상황과 공정 전체의 운전상황을 파악하여 각각 초기화한다.

2) 입력받은 초기 공정상태와 달성하기 원하는 최종 목표 상태가 서로 일치되

는 경우, 운전절차를 제시하고 종료된다.

3) 운전상황 진행 알고리즘으로 운전상황의 인과적·시간적 변화를 인식하여 한 후, 목표트리가 있는 경우, 중간목표 달성에 필요한 장치와 선택 가능한 조작 세트를 생성한다. 만일 목표트리가 없는 경우, 초기상태와 목표상태를 비교하여 그 차이를 줄이는데 필요한 조작세트를 생성한다.

4) 조작의 세트 중에서 작동 준비조건과 운전상황 진행 조건(progression condition)을 만족하는 조작을 하나 선택한다

5) 전역적 제한조건에 의해 더 이상 선택할 조작이 없는 경우, 적절한 퍼지 조작을 조작 세트에 추가한 후, 4)로 간다. 퍼지 조작을 행할 수 없는 경우나 운전 목표를 달성하는데 필요한 조작이 없는 경우, 사용자에게 운전목표를 달성할 수 없음을 알리고 종료된다.

6) 제약조건 전파를 이용하여 선택된 조작의 실행으로 변화된 공정상태를 모사한다.

7) 변화된 공정상태가 전역적·지역적·시간적 제약조건을 침범하지 않고 안전한지를 체크한다. 만일 위험한 경우에는 공정을 조작 실행 이전의 상태로 되돌린 다음, 4)번으로 가서 조작 세트에서 다른 조작을 하나 선택한 후, 4)~7)의 과정을 반복 실행한다. 안전한 경우에는 운전절차에 그 조작을 추가한 다음, 2번으로 가서 2)~7)의 과정을 반복 실행한다.

## 5. 결론

본 연구에서 제시한 시간논리와 목표트리를 이용한 OPS 알고리즘을 대규모 화학공장에 적용한 결과, 그 유효성을 입증할 수 있었다. 즉, 기존의 선형·비선형 OPS 알고리즘을 대규모 화학공장에 적용할 경우에 발생하는 조합폭발 (combinatorial explosion)의 문제를 해결하기 위해 목표트리와 시간논리를 이용하여 운전목표의 달성에 불필요한 탐색경로를 제거하도록 하였다.

## 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)