

# 상태 분할 기법을 이용한 집사 로봇의 작업 계획 시스템

## A Task Planning System of a Steward Robot with a State Partitioning Technique

김 용 휘<sup>1</sup>, 이 형 옥<sup>2</sup>, 김 현 희<sup>3</sup>, 박 광 현<sup>4</sup>, 변 중 남<sup>†</sup>

Yong-Hwi Kim<sup>1</sup>, Hyong-Euk Lee<sup>2</sup>, Heon-Hui Kim<sup>3</sup>,  
Kwang-Hyun Park<sup>4</sup>, Z. Zenn Bien<sup>†</sup>

**Abstract** This paper presents a task planning system for a steward robot, which has been developed as an interactive intermediate agent between an end-user and a complex smart home environment called the ISH (Intelligent Sweet Home) at KAIST (Korea Advanced Institute of Science and Technology). The ISH is a large-scale robotic environment with various assistive robots and home appliances for independent living of the elderly and the people with disabilities. In particular, as an approach for achieving human-friendly human-robot interaction, we aim at 'simplification of task commands' by the user. In this sense, a task planning system has been proposed to generate a sequence of actions effectively for coordinating subtasks of the target subsystems from the given high-level task command. Basically, the task planning is performed under the framework of STRIPS (Stanford Research Institute Problem Solver) representation and the split planning method. In addition, we applied a state-partitioning technique to the backward split planning method to reduce computational time. By analyzing the obtained graph, the planning system decomposes an original planning problem into several independent sub-problems, and then, the planning system generates a proper sequence of actions. To show the effectiveness of the proposed system, we deal with a scenario of a planning problem in the ISH.

**Keywords** : Task planning system, Backward split plan, State partitioning technique, Steward robot

### 1. 서 론

첨단 기술의 발전과 더불어 인간의 삶의 질을 높이기 위한 스마트 홈(smart home)에 대한 연구가 활발히 이루어지고 있다. 콜로라도 대학의 Adaptive house<sup>[1]</sup>, 마이크로소프트의 Easy-living<sup>[2]</sup>, MIT의 House of the future<sup>[3]</sup> 등이 대표적인 예이다. 특히 노약자나 장애인의 경우에는 신체적인 제약 때문에 주거공간 내에서의 자율적인 이동과 행동에 불편을 느낀다. 이에 대하여 최근의 스마트 홈 연구

는 노약자 및 장애인을 대상으로 하여 독립적인 생활을 보장하기 위한 시도가 많아지고 있다. 지능형 휠체어 및 이동기의 개발은 이러한 노력의 하나라 할 수 있다.

국내에서도 장애인을 위한 스마트 홈에 대한 연구가 활발히 진행되고 있는데, 한국과학기술원 인간친화 복지 로봇시스템 연구센터의 지능형 주거공간 ISH가 대표적이다<sup>[4]</sup>. ISH는 노약자와 장애인의 독립적인 생활을 돕기 위해 지능형 침대, 지능형 휠체어, 이동기(robotic hoist), 손 동작 기반 인터페이스, 건강 모니터링 장치를 포함하고 있다.

ISH는 사용자에게 다양한 서비스를 제공하기 위하여 많은 장치와 인간-기계 인터페이스(human-machine interface)를 가지고 있다. 그러나 장치와 인터페이스의 수가 증가함에 따라서, 사용자가 주거 공간 내의 장치들의 기능과 역할을 모두 습득하여야 하기 때문에, 불편함을 느낄 가능성이 높다. 본 논문에서는 인간친화적 인간-로봇 상호작용 관점에서 해결 방법 중 하나로 '작업 명령의

※ 본 연구는 과학기술부/한국과학재단 우수연구센터육성사업의 지원으로 수행되었음(R11-1999-008).

† 교신저자: 한국과학기술원 전자전산학과 교수  
(E-mail : zbien@ee.kaist.ac.kr)

<sup>1</sup> 한국과학기술원 전자전산학부 박사과정  
(E-mail : pigfrogbrother@kaist.ac.kr)

<sup>2</sup> 한국과학기술원 인간친화 복지 로봇시스템 연구센터 선임연구원  
(E-mail : helee@ctrsys.kaist.ac.kr)

<sup>3</sup> 한국과학기술원 인간친화 복지 로봇시스템 연구센터 연구원  
(E-mail : kimhh@ctrsys.kaist.ac.kr)

<sup>4</sup> 광운대학교 정보제어공학과 조교수 (E-mail : akaii@robotian.net)

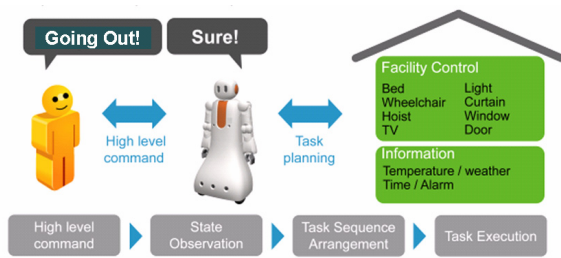


그림 1. 집사 로봇 조이의 작업 계획 시스템

단순화'를 대상으로 한다.

집사 로봇 조이(JOY)<sup>6)</sup>는 주거 공간 내의 장치들과 인간 사이의 매개체 역할을 한다. 작업 명령의 단순화 관점에서 집사 로봇 조이는 작업 계획 시스템(task planning system)을 보유하고 있다(그림 1). 만약 사용자가 외출을 원한다면, 작업 계획 시스템은 현재의 환경 정보를 수집하고, 외출 준비 서비스를 제공하기 위하여 필요한 행동들을 선택하고, 그 순서를 조합한다. 그리고 얻은 순서에 따라 장치를 제어한다.

ISH는 로봇 들과 장치들을 이용하여 사용자에게 서비스를 제공한다. 때문에, 사용자 만족의 관점에서 서비스 준비 시간을 고려하여야 한다. 즉, 작업 계획 시스템에서 서비스 준비 시간에 영향을 주는 계산 시간을 분석하여야 한다.

집사 로봇의 작업 계획 시스템은 STRIPS 표현법<sup>7)</sup>을 이용하여 상태(state)와 행동(action)을 표현하고, 이를 바탕으로 순서 탐색을 수행한다. 인공지능 분야에서는 STRIPS 표현법을 기반으로 다양한 계획법을 연구하고 있다. 그래프 계획<sup>8)</sup>은 계획 그래프(planning graph)를 이용한 방법 중 하나이다. 계획 그래프는 상태 내에 존재하는 각 성분들이 어떻게 천이를 일으키는지를 가지적으로 나타내는 방법이다. 그래프 계획은 계획 그래프를 탐색하여 원하는 행동의 순서를 찾는 방법이다. 그래프를 이용한 계획 시스템은 계획 그래프를 구성하는 방법에 따라 성능에 차이가 있다. 분리 계획 그래프(split planning graph)<sup>9)</sup>는 그래프 계획의 계획 그래프와 달리 초기 상태 내의 모든 성분이 목표 상태의 모든 성분으로 대응될 때 거쳐가는 행동을 분석하는 그래프이다. 분리 계획 그래프는 그래프 계획에서 사용한 계획 그래프보다 계산 시간을 줄일 수 있다. 본 논문에서는 이러한 분리 계획 그래프를 사용한 방법을 그래프 계획과 구별하기 위하여 분리 계획(split plan)이라 명명하였다.

분리 계획의 성능에 영향을 주는 요인은 크게 3가지로 분석할 수 있다. 먼저 분리 계획은 휴리스틱 탐색

(heuristic search)법을 기반으로 하고 있다. 따라서 탐색에 사용하는 휴리스틱 함수에 따라서 성능에 차이가 생길 수 있다. 둘째, 계획 그래프 구성 방향에 따라서 성능에 차이가 있을 수 있다. 즉, 그래프를 역방향으로 구성(backward construction)하느냐 전방향으로 구성(forward construction)하느냐에 따라서 성과 성능이 변한다. 마지막으로 상태의 그룹화 방법에 따라 성능이 달라진다. 경우에 따라서 주어진 계획 문제를 여러 개의 독립적인 부분 문제(sub-problem)로 나눌 수 있다. 이 경우 하나의 계획 문제를 한번에 해결하는 것이 아니라 여러 개의 부분 문제로 나누어 해결하면, 계획 그래프의 노드의 수와 행동의 수를 줄일 수 있어 계획 시간을 단축시킬 수 있다.

본 논문에서는 위의 3가지 관점에서 작업 계획의 시간을 단축하는 방법에 대하여 논한다. 먼저 2절에서는 배경지식으로 STRIPS 표현법에 대하여 서술하고, 위의 3가지 관점을 분석한다. 3절에서는 기본 계획 모듈로 분리 계획을 변형한 역방향 분리 계획(backward split plan)을 제안한다. 4절에서는 계획 문제를 분할하기 위한 기법으로 상태 분할 기법(state partitioning technique)을 제안한다. 5절에서는 기존에 알려진 계획문제를 바탕으로 제안한 기법의 일반성을 확보하고, ISH에 정의된 시나리오를 바탕으로 제안된 시스템의 효율성을 검증한다. 6절에서는 본 논문을 정리하고 결론에 대하여 서술한다.

## 2. 배경 지식

### 2.1 STRIPS 표현법

STRIPS 표현법은 1971년 Fikes에 의해 제안된 표현법으로, 현재 상태와 행동을 표현하는 방법을 나타낸다. STRIPS 표현법에서 상태는 사실(fact)들의 집합으로 정의된다. 여기서 사실은 현재 상황을 묘사한 것으로, 인스턴스(instance)와 술어(predicate) 두 가지로 구성되어 있다. 인스턴스는 정의된 환경 모델에 존재하는 구별되는 물체를 나타낸다. 술어는 하나 이상의 인스턴스를 가지는 이진 함수(binary-valued function)를 나타낸다. 만약  $on(A,B)$ 을 B위에 A가 있다라는 것을 나타내는 술어, user와 bed는 인스턴스라 가정하면, "사용자가 침대 위에 있다"라는 상황을 사실로 표현하면 "on(user,bed)"와 같이 표현할 수 있다.

인공지능 분야에서는 행동을 다음과 같은 개념으로 사용한다. Valle는 행동의 역할을 현재 환경을 변화시키는 것이라 하였다<sup>9)</sup>. Fikes는 행동이 현재 상태에 적용될

때, 새로운 상태가 계산된다고 하였다<sup>7)</sup>. Rintanen은 상태를 임의의 공식에 의하여 변화시키는 것을 행동이라 하였다<sup>10)</sup>. 이러한 내용들로 미루어 볼 때 행동은 현재 상태에서 새로운 상태로 천이하는 매핑(mapping)이라 정의할 수 있다. 행동은 선행 조건(precondition), 제거 효과(delete effect), 추가 효과(add effect)를 요소로 가진다. 선행 조건은 행동 수행 직전에 충족하여야 하는 사실들의 집합이다. 제거 효과는 새로운 상태가 계산될 때 제거되는 사실들의 집합이고, 추가 효과는 새로운 상태가 계산될 때 추가되는 사실들의 집합이다.

작업 계획 시스템에 주어진 조건을 계획 문제(planning problem)라 한다. 계획 문제의 표시법은  $P = \langle A, I, G \rangle$  와 같으며,  $A$ 는 정의된 행동들의 집합,  $I$ 는 현재 인지된 초기상태(initial state),  $G$ 는 원하는 목표 상태(goal state)를 나타낸다. 따라서 작업 계획이란 초기 상태에서 목표 상태로 성공적으로 천이시키는 행동의 순서(Q)를 찾는 것이라 정의할 수 있다<sup>11)</sup>.

### 2.2 분리 계획의 성능 분석

분리 계획의 계산 속도에 영향을 주는 요인은 3가지로 살펴볼 수 있다. 먼저 분리 계획은 휴리스틱 탐색<sup>13)</sup>,<sup>14)</sup>을 기반한 방법이다. 탐색에 사용하는 휴리스틱 함수는 현재 상태와 원하는 목표 상태 사이의 거리를 나타내는 함수로, 정의하는 방법이 매우 다양하고, 그에 따라서 속도, 탐색의 완전성(completeness)에 영향을 준다. 이 중 셋-레벨 (set-level)<sup>11,20)</sup> 휴리스틱 함수는 ISH 환경에서 실험한 결과, 해를 구할 수 있었다. 따라서 이 함수를 적용하여 탐색을 수행하였다.

둘째, 그래프 구성 방향에 따라 성능이 영향을 받을 수 있다. Bsr-그래프 계획<sup>12)</sup>은 그래프 계획의 계획 그래프를 역방향으로 변형하여 구성하는 방법이다. 그러므로 Bsr-그래프 계획은 그래프 계획과 비교하여 목표 상태를 얻는데 필요 없는 초기 상태 내의 사실의 수가 증가하더라도 그래프 구성의 속도에는 영향을 미치지 않기 때문에 해당 시스템에서 계산 시간을 단축시킬 수 있다. 만약 계획 문제의 목표 상태가 초기 상태의 일부 분만을 필요로 할 경우, 역방향 그래프 구성이 더 쉽게 속도를 증가시킬 수 있다<sup>15)</sup>. 즉 초기 상태에서 불필요한 사실들(redundant facts)이 존재하면 그래프 구성 시간에서 이득을 얻을 수 있다. 이와 같은 방법으로 분리 계획 그래프를 역방향으로 구성한 경우 유사한 결과를 얻을 수 있다.

마지막으로 입력된 계획 문제를 하나의 문제로 보지 않고, 여러 개의 독립적인 부분 문제로 그룹화한 후 계

획하면 전체 계산 속도를 증가시킬 수 있다. 하지만 상태를 잘못 그룹화하면 원하는 결과를 얻을 수 없다. 즉, 서로 의존하는 두 부분 문제들을 독립적인 부분 문제로 간주하여 작업 계획을 수행할 경우, 원하는 목표를 얻을 수 없고, 잘못된 작업 수행을 일으킬 수 있으므로, 부분 문제에 대한 독립성 판단법 및 필요한 정보 획득 방법을 제안하여야 한다.

## 3. 역방향 분리 계획

분리 계획 그래프를 역방향으로 구성할 경우, 목표 상태로 천이하는데 초기 상태 내에 필요 없는 여분의 사실들이 있으면, 계산 시간을 감소시킬 수 있다. 즉 역방향 그래프 구성을 수행할 경우 원하는 목표를 얻는데, 입력된 모든 센서 정보 중 필요한 정보만을 선택하는 효과를 얻을 수 있다. 이를 역방향 분리 계획이라 명명한다.

역방향 분리 계획은 3가지 과정으로 나눌 수 있다. 먼저 역방향 분리 계획 그래프(backward split planning graph)를 구성한 다음, 계획 그래프를 휴리스틱 함수 값, 클래스, 천이 사실 집합에 대하여 분석한다. 마지막으로 분석한 휴리스틱 함수 값을 바탕으로 행동의 순서를 탐색한다.

### 3.1 역방향 분리 계획 그래프

작업 계획에서 현재 상태에서 다음 상태를 추론하는 방식에는 전방향 추론(forward reasoning)과 역방향 추론(backward reasoning)이 있다<sup>12,14)</sup>. 전방향 추론은 임의의 행동에 의하여 현재 상태에서 어떠한 상태로 변화하는지를 나타낸 방법이다. 이에 비하여 역방향 추론은 임의의 행동에 의해서 현재 상태가 발생하였을 때, 행동을 취하기 이전의 상태가 무엇인지를 나타내는 방식이다. 즉 전방향 추론은 초기 상태에서부터 목표 상태로 천이할 때 상태 변화가 어떠한지 추론하는 반면, 역방향 추론은 목표 상태에서 초기 상태로 추론할 때 상태 변화가 어떠한지 추론한다. 역방향 분리 계획 그래프는 이러한 역방향 추론을 기반으로 한다.

현재 상태를  $S$ , 임의의 행동을  $a$ , 그리고 이전 상태를  $S'$ 라 할 때, 역방향 추론에 의한 상태 변화식은  $S' = \text{Backward}(S, a) = S \cup \text{pre}_a - \text{add}_a$ <sup>[12]</sup>와 같다. 이때  $\text{add}_a$ 는  $a$ 의 추가 효과,  $\text{pre}_a$ 는  $a$ 의 선행 조건이다.

역방향 분리 계획 그래프는 레벨화된 트리 구조를 가진다. 그리고 그 기호법은 기존의 분리 계획 그래프와 유사하다. 역방향 분리 계획 그래프는 노드(node)로 구성이 되어있고, 노드는 상태 노드(state node)와 행동 노드(action node)로 나뉜다.

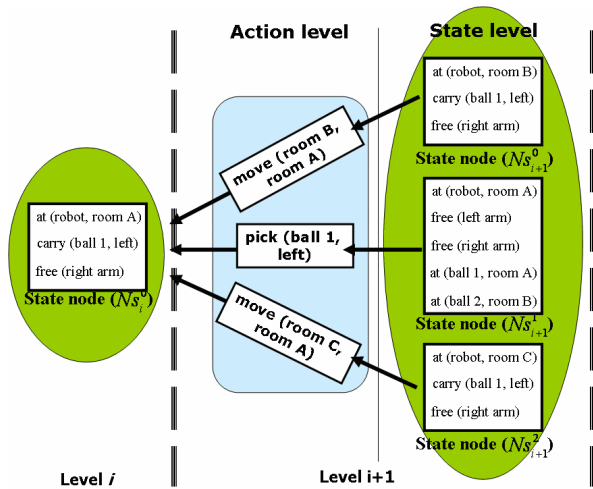


그림 2. 역방향 분리 계획 그래프의 구조

그림 2는 역방향 분리 계획 그래프의 구조를 나타낸다. 먼저, 현재 레벨의 상태 노드에 적용할 수 있는 (applicable) 행동들을 찾은 후 각각을 다음 레벨의 행동 노드로 설정한다. 여기서 현재 상태  $S$ 에 행동  $a$ 를 적용할 수 있다는 것은  $add_a \cap S \neq \emptyset$ ,  $del_a \cap S = \emptyset$ 가 성립한다는 것을 나타낸다<sup>20)</sup>. 여기서  $del_a$ 은 행동  $a$ 의 제거 효과를 나타낸다. 역방향 분리 계획 그래프의 하나의 레벨은 상태 레벨(state level)과 행동 레벨(action level)로 구성되어 있다. 여기서 상태 레벨은 현재 레벨에 포함된 모든 상태 노드의 집합이고, 행동 레벨은 현재 레벨에 포함된 모든 행동 노드의 집합이다.

그래프의 첫 번째 레벨의 상태 노드로 기존의 분리 계획 그래프에서는 초기 상태가 설정되는 반면, 역방향 분리 계획 그래프에서는 역방향 추론을 기반으로 하기 때문에 목표 상태가 설정된다. 다음으로 현재 상태 노드를 지지하는 행동을 먼저 선택하고, 각각을 하나의 행동 노드로 만든다. 그리고 각 행동 노드에 대하여 역방향으로 추론하여 다음 레벨의 상태 노드들을 생성한다. 이러한 과정을 상태 노드가 초기 상태의 부분 집합이 될 때까지 반복한다.

역방향 분리 계획 그래프는 상호 간섭 관계(mutex: mutual exclusive relation)를 고려하여야 한다. 상호 간섭 관계란 한 행동의 제거 효과가 동시에 수행하는 행동들의 선행 조건과 추가 효과에 영향을 주는 경우를 말한다.

분리 계획 그래프를 구성할 경우, 레벨이 증가할수록 상태 노드의 수와 행동 노드의 수가 크게 증가하고, 이는 계산 시간을 지수적으로 증가시킬 가능성이 있다. 따라서 계산량을 줄이기 위하여 암기 과정(memorization mechanism)이 필요하다. 이는 계획 그래프의 이전 레벨

**PROCEDURE** MakeBackwardSplittedGraph ( $\mathcal{A}, \mathcal{I}, \mathcal{G}$ )

```

 $Ns_0^i := \mathcal{G}$ 
 $i := 0$ 
SET  $\mathcal{H}$  to  $\langle \rangle$ 
WHILE  $Ns_j^i \notin \mathcal{I}$  for all  $Ns_j^i \in Ns_i$  DO
  FOR  $j=1$  to  $ns_i$ 
    SET  $A_{applicable}$  to the actions that is applicable all  $f \in Ns_j^i$  in  $\mathcal{A}$ 
    WHILE  $a$  for all  $a \in A_{applicable}$  DO
       $S := \text{Backward}(S_j^i, a)$ 
      IF facts in  $S$  are created by actions that are not mutex THEN
        IF  $S \notin \mathcal{H}$  THEN
          IF  $S \notin Ns_{i+1}$  THEN
            INSERT  $S$  in the  $Ns_{i+1}$ 
          ENDIF
          INSERT  $a$  in the  $Na_{i+1}$ 
        ENDIF
      ENDIF
    ENDIF
  ENDIF
  ENDFOR
   $i \leftarrow i+1$ 
  ADD  $Ns_i$  at the end of the HASHTABLE
ENDWHILE
 $g_k := \bigcup_{i=0..k} (Na_i \cup Ns_i)$ 
RETURN  $g_k$ 

```

**\* Notation**  
 $\mathcal{I}$ : Initial state,  $\mathcal{G}$ : Goal state,  $\mathcal{A}$ : A set of actions in planning problem  
 $Ns_j^i$ : The  $j$ -th state node of the level  $i$   
 $S_j^i$ : The state in state node  $Ns_j^i$   
 $Na_j^i$ : The  $j$ -th action node of the level  $i$   
 $Ns_i = \{Ns_1^i, \dots, Ns_{ns_i}^i\}$ : The  $i$ -th state level ( $ns_i$ : The number of state nodes at level  $i$ )  
 $Na_i = \{Na_1^i, \dots, Na_{na_i}^i\}$ : The  $i$ -th action level ( $na_i$ : The number of actions nodes at level  $i$ )  
 $\mathcal{H}$ : Storage for memorization mechanism  
 $g_k$ :  $k$ -order backward splitted planning graph  
 $\text{Backward}(S, a) = S - add_a \cup pre_a$

그림 3. 역방향 분리 계획 그래프 구성의 pseudo 코드

에 만들어진 상태 노드를 모두 저장하고, 새로운 상태 노드가 추가될 때, 이미 저장된 상태 노드와 같은 노드는 그래프에 추가하지 않는 과정을 말한다. 이러한 방법을 사용할 경우 그래프 내의 불필요한 노드를 추가하지 않아 그래프 계산 시간을 줄일 수 있다. 그림 3은 역방향 분리 계획 그래프의 생성을 pseudo 코드로 나타낸 그림이다.

**3.2 계획 그래프 분석**

역방향 분리 계획 그래프를 분석하면 4가지 정보(상태 클래스, 행동 클래스, 가능한 사실, 휴리스틱 함수의 값)를 얻을 수 있다.

먼저 그래프 분석에서 가장 중요한 정보는 클래스(class)이다. 그래프 내에 존재하는 마지막 레벨의  $j$ 번째 상태 노드와 시작 노드 사이의 경로에 연결된 상태 노드들의 집합을 상태 클래스(state class)라 하고, 행동 노드들의 집합을 행동 클래스(action class)라 한다. 만약에 초기 상태와 관련이 있는 상태 노드가 포함된 상태 클래스인 경우 대응하는 행동 클래스는 계획 문제에 대한 예측된 해라고 할 수 있다. 만약 마지막 레벨의  $j$ 번째

사실 노드를 얻는 데, 여러 가지 경로가 존재할 경우 서로 다른 클래스로 구분한다.

다음으로 휴리스틱 함수는 2.3절에서 살펴본 것과 같이 셋-레벨 휴리스틱 함수를 사용하였고, 표현식은 다음과 같다.

$$h(S) = lev(S) = \min_{S \subseteq N_{s_i^j}, j \in [1, n_{s_i}]} i$$

- $G \subseteq N_{s_i^j}$  인 상태 노드는 0으로 설정한다.
- 그래프에 임의의 상태  $S$  내의 모든 사실을 포함한 노드가 없을 경우,  $h(S) = \infty$ .

( $G \subseteq N_{s_i^j}$ : i-레벨 j번째 상태 노드,  $n_{s_i}$ : i번째 상태 레벨에 포함된 상태 노드의 수)

마지막으로 역방향 분리 계획 그래프에서 천이 사실 집합(a set of transition facts)을 얻을 수 있다. 천이 사실 집합은 초기 상태와 목표 상태 사이에 존재할 수 있는 사실들에 대한 정보를 포함하고 있으며, 상태 클래스로부터 쉽게 구할 수 있다. 즉 초기 상태가 속해있는 상태 클래스를 얻을 수 있고, 그 클래스에 존재하는 모든 사실들이 천이 사실 집합이 된다. 천이 사실 집합은 다음과 같이 정의된다.

**정의 (천이 사실 집합:  $\mathcal{F}$ )** 초기 상태와 목표 상태 사이의 경로에 연결된 중간 상태 노드들에 속한 사실들의 집합을 천이 사실 집합( $\mathcal{F}$ )이라 한다.

### 3.3 탐색 과정

행동들의 순서( $\mathcal{Q}$ )를 탐색하기 위하여 사용한 탐색 알고리즘은 휴리스틱 탐색에서 사용하는 enforced hill climbing 알고리즘<sup>[16, 17]</sup>을 사용하였다. 휴리스틱 탐색은 시작 상태 노드  $G \subseteq N_{s_i^0}$ 에서부터 원하는 최종 상태 노드  $N_{s_i^j} \subseteq \mathcal{I}$ 를 얻을 때까지 탐색을 시도한다. 이 때, 탐색 수행 시, 항상 휴리스틱 함수의 값이 작아지는 방향으로 행동을 선택하여야 한다. 휴리스틱 함수의 값이 커진다는 것은 현재 상태와 최종 상태 노드 사이의 거리가 멀어진다는 의미가 되기 때문이다. 따라서 최종 상태 노드에서 휴리스틱 함수 값은 0이 되어야 하고, 시작 상태 노드에서 휴리스틱 함수의 값은 최대값이 되어야 한다. 하지만, 셋-레벨 휴리스틱 함수는 시작 상태 노드에서 휴리스틱 함수 값이 0이고, 원하는 최종 상태 노드에서 휴리스틱 함수의 값이 최대값이 된다. 그러므로, 행동들의 순서를 탐색하기에 앞서, 정의한 셋-레벨 휴리스틱 함수를 수정하여야 한다. 이

를 위하여 휴리스틱 함수를  $h(S) = k - lev(S)$ 로 수정하여 사용한다<sup>[9]</sup>. 이때 k는 계획 그래프의 레벨 수를 나타낸다.

## 4. 상태 분할 기법

상태 분할의 목적은 목표 상태와 초기 상태를 분할하여 그림 4(a)와 같은 하나의 계획 문제를 그림 4(b)와 같이 독립적인 부분 문제로 나누어 계획 속도를 증가시키는 것이다. 부분 문제는 부분 초기 상태(sub-initial state)와 부분 목표 상태(sub-goal state)로 구성된 계획 문제로 정의한다. 이 때, 부분 목표 상태는 목표 상태의 일부분이고, 부분 초기 상태는 부분 목표 상태를 얻는데 필수적으로 필요한 초기 상태 내의 사실들의 집합이다.

상태 분할 기법은 그림 5와 같이 2가지 과정을 거친다. 먼저 목표 상태 내의 각 사실들과 초기 상태 내의 사실들과의 관계를 분석하기 위하여 초기 분할 과정을 수행한다. 그 다음, 초기 분할 과정에서 얻은 초기 분할된 문제 사이의 의존성(dependency), 독립성(independ-

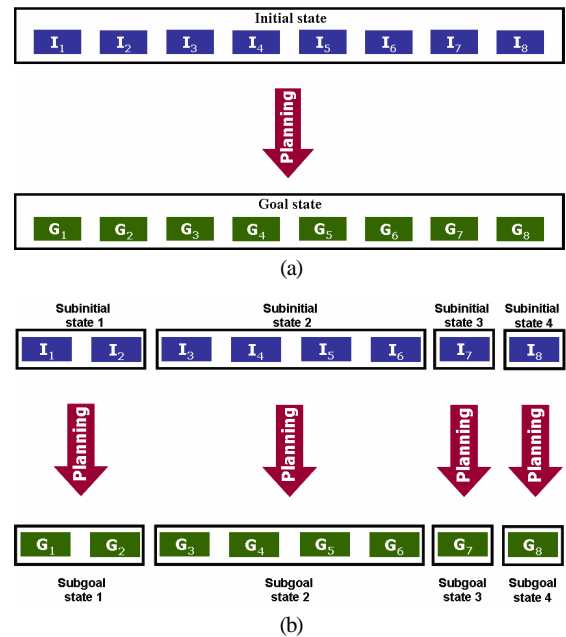


그림 4. 상태 분할 기법의 개념도

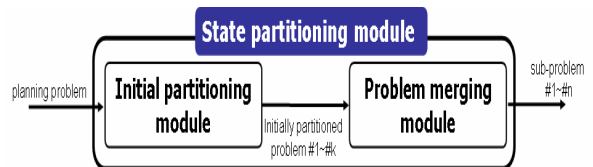


그림 5. 상태 분할 기법의 과정

ency)을 분석한 후 의존하는 문제에 대해서는 하나로 합치는 통합 과정을 거친다.

#### 4.1 초기 분할 과정

상태 분할 기법에서 부분 목표 상태와 그것을 얻는데 필수적으로 필요한 부분 초기 상태를 어떻게 그룹화하여 하나의 부분 문제로 분할하느냐가 매우 중요한 문제이다. 이를 위하여 상태 분할 기법을 적용하는데 필요한 기초 정보를 얻어야 하며, 3장에서 살펴본 역방향 분리 계획 그래프를 이용한다. 역방향 분리 계획 그래프의 마지막 상태 레벨의 상태 노드는 원하는 목표 상태를 얻는데 필수적인 초기 상태 내의 사실들로 구성되어 있다. 즉, 주어진 부분 목표 상태에 대응하는 부분 초기 상태를 얻을 수 있음을 의미한다. 뿐만 아니라 목표 상태와 초기 상태 사이의 경로에 존재하는 천이 사실들에 관한 정보 역시 얻을 수 있다. 이러한 천이 사실들은 초기 분할된 문제들 사이의 의존성, 독립성을 판단하는데 중요한 역할을 한다.

초기 분할 과정은 2단계로 나누어 진행된다. 먼저 목표 상태에 속한 각 사실들로 구성된 초기 분할된 부분 목표 상태  $G_{ini}^n$  (n-번째 초기 분할 된 부분 목표 상태)들로 분할한다. 다음으로 역방향 분리 계획 그래프를 구성하고, 이를 바탕으로  $G_{ini}^n$ 에 대응하는 초기 분할된 부분 초기 상태  $I_{ini}^n$  (n-번째 초기 분할 된 부분 초기 상태)를 얻을 수 있다. 이러한 과정을 거치면 목표 상태 내의 각 사실들을 얻는데 필요한 초기 상태 내의 사실들을 얻을 수 있고, n-번째 초기 분할된 각 문제의 천이 사실 집합  $F_{ini}^n$ 를 얻는 것이 가능하다.

#### 4.2 통합 과정: 의존하는 문제 구분 및 통합 과정

먼저 의존하는 부분 문제를 구분하기에 앞서 독립성과 의존성을 정의하여야 한다. 의존성과 독립성은 계획 분야에서 다양한 의미로 해석된다. 그 중 Knoblock<sup>[18]</sup>은 각 목표 상태를 얻는데 필요한 각 순차적인 행동들이 개별적으로 수행하여 얻은 목표 상태와 동시에 수행을 하여 얻은 목표 상태가 같은 경우 두 행동의 순서는 독립적이라 하였다. 즉, 두 순차적인 행동들이 동시에 실행 가능하면 둘 사이는 독립적이라는 의미이다. 여기서 행동의 순서는 각 계획 문제를 해결하는 해이므로, 행동의 순서들 사이의 관계가 독립적이라는 말은 두 문제가 독립적이라는 말과 같다. 이를 정리하면, 두 문제를 해결하는데 필요한 각 순차적인 행동들이 서로 동시에 실행 가능하면 두 문제는 독립적이라고 할 수 있다.

역방향 분리 계획 그래프에서 천이 사실 집합을 얻을 수 있다. 이는 각 문제를 해결하는데 꼭 거쳐가는 천이 사실들이 무엇인지를 나타내기 때문에, 이들의 관계를 파악하면 문제 사이의 의존성을 판단할 수 있다.  $F^1$ 과  $F^2$ 를 각각 문제 1, 2를 해결할 수 있는 해의 천이 사실의 집합이라 할 때,  $F^1$ 과  $F^2$ 에 속한 사실들 중 서로 같은 것이 존재하면 두 문제는 의존적이라고 할 수 있으며,  $F^1$ 과  $F^2$ 에 속한 사실들에 상호 배제 관계가 존재하는 경우에도 두 문제는 의존적이라 할 수 있다. 이는 각 문제를 해결하는 두 순서들이 동시에 수행되지 못하고, 서로 영향을 받는다는 것을 의미한다.

초기 분할된 문제 사이의 독립성을 판단한 후, 의존하는 초기 분할된 문제에 대해서는 통합 과정을 거쳐야 한다. 두 문제가 상호 의존 관계이면,  $I_{sub} = I_{ini}^1 \cup I_{ini}^2$ ,  $G_{sub} = G_{ini}^1 \cup G_{ini}^2$ 에 의하여 새로운 부분 초기 상태와 부분 목표 상태를 얻는다. 만약 한 초기 분할된 문제가 다른 문제들과 항상 독립적이라고 하면, 통합 과정 없이 하나의 부분 문제가 된다.

#### 4.3 상태 분할의 적용

상태 분할은 기존의 계획 모듈에 그림 6과 같이 추가하여 적용할 수 있다. 상태 분할에서 얻은 각 부분 문제에 대하여 작업 계획을 수행하면, 각 부분 문제마다 행동의 순서를 얻을 수 있다. 하지만 이러한 순서를 직접 제어에 활용할 수는 없다. 따라서 각 순서를 하나의 순서로 합치는 순서 통합 과정(sequence merging process)을 거쳐 통합된 행동의 순서를 얻는다.

ISH는 여러 로봇이 통합된 공간이기 때문에 로봇 간의 충돌이 발생할 가능성이 있다. 그러므로 충돌 문제를 고려하여 순서를 통합하여야 한다. 예를 들어, 그림 7은 두 로봇의 이동 경로를 나타낸 그래프이다. 그림에서

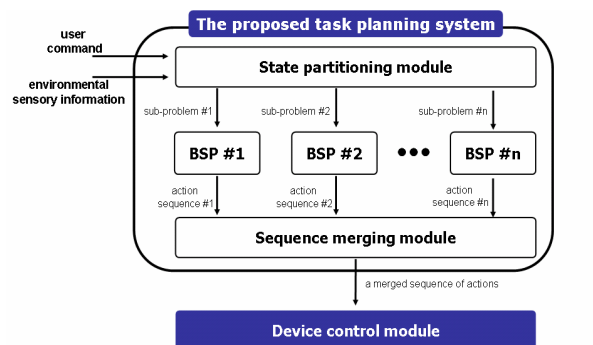


그림 6. 제안된 작업 계획 시스템의 전체 구조

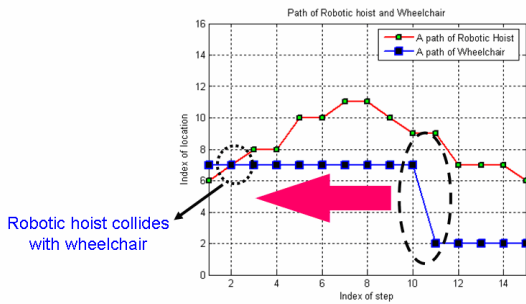


그림 7. 두 로봇의 이동 경로 및 충돌

보면 행동 순서의 스텝 2에서 두 로봇이 7번 위치에 같이 존재함을 알 수 있다. 이는 이동기와 휠체어가 충돌을 일으키는 것을 의미한다.

행동의 순서 내에서 충돌을 없애기 위하여 충돌을 일으키는 로봇에 관련된 행동들을 먼저 수행하면, 로봇 간의 충돌을 해결할 수 있다. 이를 간단히 나타내면, 그림 7의 그래프 상에 원으로 표시된 부분을 왼쪽으로 이동시키면 된다. 즉, 충돌을 일으키는 로봇들 중 하나를 선택하고, 그 로봇과 관련된 행동들의 순서를 얻은 후, 충돌하는 다른 로봇보다 먼저 수행한다.

### 5. 실험 결과

#### 5.1 실험 1

ISH에 대한 실험에 앞서 기존의 알려진 계획 문제를 바탕으로 실험을 하였다. 사용한 계획 문제는 상자 쌓기 문제(block world problem)에 대하여 실험 하였다. 상태 분할 기법의 효용성을 보이기 위하여 상자 쌓기 문제는 그림 8의 왼쪽과 같은 초기 상태에서 오른쪽과 같은 목표 상태를 만드는 문제를 다루었다. 실험은 각 문제에 대하여 독립적인 부분 문제의 수를 증가 시켜 보았다. CPU P4-2.8GHz 컴퓨터를 사용하여 다음과 같이 2가지 경우에 대하여 실험을 하였다.

(a) 분리 계획

(b) 역방향 분리 계획 + 상태 분할

그 결과 그림 9의 결과를 얻을 수 있다. 그림에서 보는 것과 같이 분리 계획은 부분 문제의 수가 증가하면 할수록 그 시간이 크게 증가하는 것을 볼 수 있다.

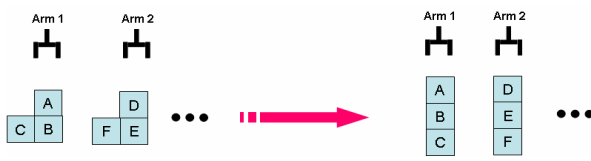


그림 8. 상자 쌓기 문제(block world problem)

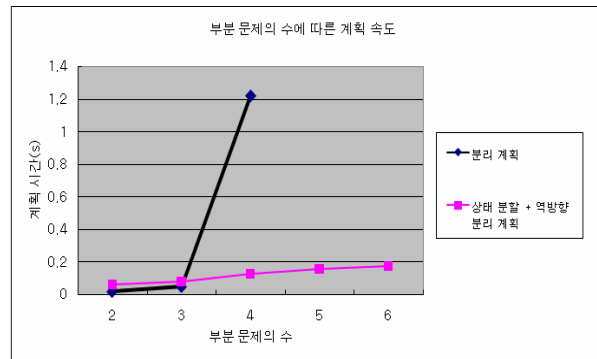


그림 9. 상자 쌓기 문제에 대한 실험 결과

하지만 상태 분할 기법을 사용하여 부분 문제로 나누어 계획하면 그 시간이 크게 단축되는 결과를 알 수 있다. 이는 일반적인 계획 문제에 대하여 독립적인 부분 문제로 충분히 나누는 것이 가능한 경우, 상태 분할 기법을 적용 시키면 계산 시간을 크게 단축시킬 수 있음을 의미한다.

#### 5.2 실험 2

ISH에서 외출 준비 작업을 계획하기 위하여, 행동을 STRIPS 표현에 의하여 정의하였다(표 1). ISH에서 로봇이 위치 가능한 장소는 그림 10과 같이 14개의 위치로 가정하였다. 이 때 공간 추상화(world abstraction)<sup>[9]</sup>를 적용하여 ISH의 위치를 그림과 같이 계층 구조로 모델링 하였다. 공간 추상화는 장소에 관련된 인스턴스 및 행동을 줄이는 기법 중 하나로 전체 계획 시간을 단축시키는 효과가 있다.

외출 준비 작업은 ISH에서 대표적인 상위 단계 작업이다. 이는 하반신이 불편한 장애인을 침대에서 휠체어

표 1. ISH에서 정의된 행동

한글 명칭	영문 명칭	선행 조건	추가 효과	제거 효과
전진 이동	Forward (robot) (x location) (y location)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)	left(robot, y)/ right(robot, y)/ up(robot, y)/ down(robot, y)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)
후진 이동	Backward (robot) (x location) (y location)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)	left(robot, y)/ right(robot, y)/ up(robot, y)/ down(robot, y)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)
좌회전	Turn left (robot) (x location)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)	down(robot, x)/ up(robot, x)/ left(robot, x)/ right(robot, x)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)
우회전	Turn right (robot) (x location)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)	up(robot, x)/ down(robot, x)/ left(robot, x)/ right(robot, x)	left(robot, x)/ right(robot, x)/ up(robot, x)/ down(robot, x)
사용자올림	Pick up the user	on(user, Bed) right(RH, P11)	on(user, Bed)	on(user, Bed)
사용자 내림	Put down the user	on(user, RH) down(RH, P7) up(wheelchair, P2)	on(user, wheelchair)	on(user, RH)
전원 켜기	Turn on(TV/light)	off(TV/light)	on(TV/light)	off(TV/light)
전원 끄기	Turn off (TV/light)	on(TV/light)	off(TV/light)	on(TV/light)
커튼 닫음	Open the curtain	close(curtain)	open(curtain)	close(curtain)
커튼 열음	Close the curtain	open(curtain)	close(curtain)	open(curtain)

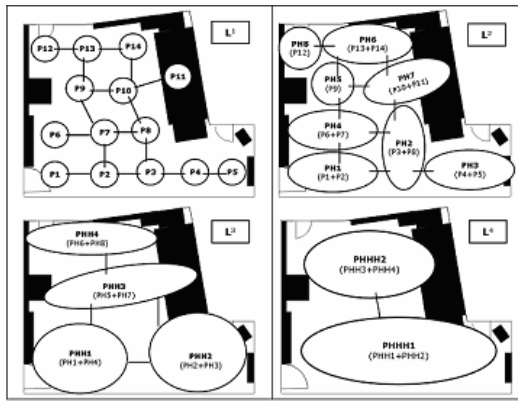


그림 10. ISH의 추상화된 공간 모델

표 2. 외출 준비 작업

상위 단계 작업: 외출 준비 작업	
초기 상태	on (user, bed), right (RH, P6), left (Wheelchair, P5), on (TV), on (lights), open (curtain), right (JOY, P1)
목표 상태	on (user, Wheelchair), right (RH, P6), off (TV), off (lights), close (curtain)
작업 수행 순서 요약	<ol style="list-style-type: none"> <li>1. 이동기가 지능형 침대로 이동한다.</li> <li>2. 사용자는 침대에서 이동기로 이동한다.</li> <li>3. 휠체어와 이동기가 도킹 장소로 이동한다.</li> <li>4. 이동기와 휠체어가 도킹한다.</li> <li>5. 이동기는 도킹 후, 사용자를 휠체어로 이동시킨다.</li> <li>6. 가진 제품의 전원을 차단한다.</li> <li>7. 이동기는 원래 위치로 돌아간다</li> </ol>

로 옮기는 과정, 절전을 위하여 가전기기의 전원을 차단하는 과정 2가지로 구분할 수 있다. 표 2는 외출 준비 작업의 계약적인 순서를 나타내고, STRIPS 표현법을 이용하여 모델링한 것이다.

외출 준비 작업에 대한 상태 분할 기법 및 역방향 분리 계획을 실험하였다. 앞서 살펴 보았던 외출 준비 작업에 대하여 계획을 수행하였으며, CPU P4.2.8GHz 컴퓨터를 사용하여 다음의 6가지 경우에 대하여 실험을 하였다.

- (a) 분리 계획
- (b) 분리 계획 + 공간 추상화
- (c) 역방향 분리 계획
- (d) 역방향 분리 계획 + 공간추상화
- (e) 역방향 분리 계획 + 상태 분할
- (f) 역방향 분리 계획 + 상태 분할 + 공간 추상화

위의 6가지 경우에 대하여 실험한 결과를 표 3에 정리하였다. 표 3(a)~(d)의 결과에서 보는 것과 같이 역방

표 3. (a)~(f)에 대한 실험 결과

Method	Graph level	Graph node	A sequence of actions		Partition time (s)	Graph construction time (s)	Serach time (s)	Total time (s)
			Step	The number of actions				
(a)	22	62597	14	22	-	545.97	0.02	545.99
(b)	22	5289	14	22	-	38.81	0.03	38.84
(c)	22	30184	14	22	-	152.15	0.02	152.17
(d)	22	4916	14	22	-	8.12	0.03	8.13
(e)	22	3727	14	22	1.59	1.56	0.02	3.17
(f)	22	611	14	22	0.45	0.38	0.01	0.84

향 분리 계획법을 사용할 경우 ISH 환경에서 좀더 좋은 성능을 보임을 알 수 있다. 그래프 노드의 수도 줄어들음을 알 수 있다. 표 3의 (e)와 (f)는 상태 분할 기법을 적용한 결과로, 전체 계획 시간이 1/50가량 줄어들음을 알 수 있다. 뿐만 아니라 공간 추상화 기법을 같이 적용한 경우 1초 이하로 계산 시간을 단축시킬 수 있는 것을 알 수 있다.

## 6. 결론

본 논문에서는 기존의 작업 계획 시스템의 성능 개선에 대하여 다루었다. 성능 개선을 위하여 먼저 기존의 분리 계획법과 달리 그래프 구성 방향이 변화된 역방향 분리 계획을 제안하였다. 그리고 상태 분할 기법을 이용하여 계산 속도를 크게 증가시켰다. 상태 분할 기법은 상호 의존하는 부분 문제를 잘못 세분화하여 계획하면, 원하는 결과를 얻을 수 없다. 따라서, 본 논문에서는 이를 해결하기 위하여 역방향 분리 계획 그래프를 이용하여 부분 문제 사이의 독립성과 의존성을 판단할 수 있는 방법을 제안하였다. 상태 분할 기법과 역방향 분리 계획을 ISH 환경에 적용시킬 경우, 계산 시간이 크게 단축됨을 실험을 통하여 알 수 있었다.

그러나 로봇의 수와 협동 작업이 늘어나는 경우 대상 시스템의 복잡도가 증가할 것이다. 이는 로봇의 행동이 서로 충돌할 경우가 증가할 것이고, 앞서 정리한 충돌 회피 방법으로는 한계가 있다. 그러므로 행동 들의 충돌 문제를 계획 단계와 구별하여 고려하는 것이 아니라, 작업 계획 단계에서 고려하여야 한다. 또한 로봇의 충돌 문제뿐 아니라, 어느 상황에서 어느 로봇을 사용하는 것이 비용(cost)을 최적화 할 수 있는지에 대한 판단도 작업 계획 시스템이 계획하여야 한다. 따라서 추후에는 동적으로 변화하는 로봇 환경에 대하여 성능을 개선할 필요가 있다.



참 고 문 헌

- [1] M.C.Mozer, "The neural network house: An environment that adapts to its inhabitants," The American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110-114, 1998
- [2] <http://research.microsoft.com/easyliving/>
- [3] [http://architecture.mit.edu/house\\_n/](http://architecture.mit.edu/house_n/)
- [4] J.W.Jung, J.H.Do, Y.M.Kim, K.S.Suh, D.J.Kim and Z.Bien, "Advanced robotic residence for the elderly/the handicapped : Realization and user evaluation", Proceedings of IEEE 9th International Conference on Rehabilitation Robotics, pp.492-495, 2005
- [5] Z.Bien, K.H.Park, D.J.Kim and J.W.Jung, "Welfare-oriented service robotic systems: Intelligent Sweet Home & KARES IP", Lecture Notes in Control and Information Sciences, vol.306, pp.57-94, 2004
- [6] Z. Zenn Bien, Hyong-Euk Lee, Young-Min Kim, Yong-Hwi Kim, Jin-Woo Jung, and Kwang-Hyun Park, "Steward Robot for Human-friendly Assistive Home Environment", Proceedings of the 2006 International Conference on Aging, Disability and Independence (ICADI 2006), 2006
- [7] R.E.Fikes and N.J.Nilsson. "STRPIS: A new approach to the application of theorem proving", Journal of Artificial Intelligence, vol.2, no.3, pp.189-208, 1971
- [8] A.L.Blum and M.L.Furst, "Fast planning through planning graph analysis", Journal of Artificial Intelligence, vol.90, no.1, pp.281-300, 1997
- [9] S.M.L.Valle, "Planning algorithms", Cambridge University Press, 2006
- [10] J.Rintanen, "Introduction to automated planning", Lecture notes of the AI planning course, Albert-Ludwigs-University Freiburg, 2006
- [11] Y.Zemali, "Controlled reachability analysis in AI planning: Theory and practice", Lecture Notes in Artificial Intelligence, vol.3698, pp.264-278, 2005
- [12] E.Parker, "Making graphplan goal-directed", Proceedings of the 4th European Conference on Planning, pp.333-346, 1999
- [13] B.Bonet and H.Geffner, "HSP: Heuristic search planner", Proceedings of the Artificial Intelligence Systems 98 (AIPS-98) Planning Competition, 1998
- [14] B.Bonet and H.Geffner, "Planning as heuristic search: New result", Proceedings of the 5th European Conference on Planning (ECP'99), pp.60-72, 1999
- [15] D.Weld, "An introduction to least commitment planning", AI Magazine, vol.15, no.4, pp.27-61, 1994
- [16] J.Hoffmann and B.Nebel, "The FF planning system: Fast Plan Generation through heuristic search", Journal of Artificial Intelligence Research, vol.14, pp.253-302, 2001
- [17] J.Hoffmann, "The Metric-FF planning system: Translating "Ignoring delete lists" to numeric state variables", Journal of Artificial Intelligence Research, vol.20, pp.291-341, 2003
- [18] C.A.Knoblock, "Generating Parallel Execution Plans with a Partial Order Planner", Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems, pp.98-103, 1994
- [19] C.Galindo, J.Fernández and J.González, "Improving Efficiency in Mobile Robot Task Planning Through World Abstraction", IEEE Transactions on Robotics, vol.20, no.4, pp.677-690, 2004
- [20] X.Nguyen, S.Kambhampati and R.Nigenda, "Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search", Artificial Intelligence Published by Elsevier, vol.135, pp.73-123, 2002



김 옹 휘

2005 세종대학교 전자공학과 (공학사)  
 2007 한국과학기술원 전자전 산학과(공학석사)  
 현재 한국과학기술원 전자전 산학과 박사과정

관심분야: 작업 계획, 학습 이론, 지능 로봇



이 형 욱

2000 한국과학기술원 전자전 산학과(공학사)  
 2002 한국과학기술원 전자전 산학과(공학석사)  
 2007 한국과학기술원 전자전 산학과(공학박사)

현재 한국과학기술원 인간친화 복지 로봇시스템 연구센터 선임 연구원

관심분야: 학습 이론, 지식 발견, 패턴인식, 서비스 로봇, 지능 로봇



**김 현 희**

- 1997 한국해양대학교 기관공학과 (공학사)
- 2002 한국해양대학교 제어계측공학과(공학석사)
- 현재 한국과학기술원 인간친화 복지 로봇시스템 연구센터 연구원

관심분야: SLAM, 추정이론, 인간-로봇 상호작용, 보행 보조 로봇



**박 광 현**

- 1994 한국과학기술원 전자전산학과(공학사)
- 1997 한국과학기술원 전자전산학과(공학석사)
- 2001 한국과학기술원 전자전산학과(공학박사)

현재 광운대학교 정보제어공학과 조교수  
관심분야: 학습 이론, 지능 로봇, 재활공학, 인간-로봇 상호작용



**변 증 남**

- 1969 서울대학교 전자공학과 (공학사)
- 1972 미국 IOWA 대학교 전기공학과 (공학석사)
- 1975 미국 IOWA 대학교 전기공학과 (공학박사)

- 1976~1977 미국 IOWA대학교 전기공학과 조교수
- 1990~1995 한국퍼지시스템학회 회장
- 1999 제어자동화시스템공학회 부회장
- 2001 대한전자공학회 회장
- 2003~2005 국제퍼지시스템학회(IFSA) 회장
- 2003~2006 한국로봇공학회(KRS) 회장
- 1977~현재 한국과학기술원 전자전산학과 교수
- 1999~현재 한국과학기술원 인간친화 복지 로봇시스템 연구센터 소장
- 2005~현재 한국전력공사 석좌교수
- 2007~현재 IEEE Fellow

관심분야: 지능제어, 학습제어, 퍼지이론, 소프트 컴퓨팅, 서비스 로봇, 재활 공학 시스템