Journal of the Korea Academia-Industrial cooperation Society
Vol. 16, No. 9 pp. 6290-6296, 2015

http://dx.doi.org/10.5762/KAIS.2015.16.9.6290
ISSN 1975-4701 / eISSN 2288-4688

효율적인 계획 수립을 위한 동작-기반의 휴리스틱

김현식[1*]
[1]서일대학교 인터넷정보과

A Action-based Heuristics for Effective Planning

Hyun-Sik Kim[1*]
[1]**Dept of Internet Information, Seoil University**

요 약 정보력이 높은 휴리스틱들은 해 계획을 찾기 위한 탐색을 보다 효율적으로 유도할 수 있다. 하지만 일반적으로, 계획 문제 명세로부터 이러한 정보력이 높은 휴리스틱을 추출하는 것은 매우 많은 계산 노력을 요구한다. 이러한 문제점들에 효과적으로 대처하기 위해서, 본 논문에서는 계획문제로부터 계획 수립을 보다 효율적으로 풀 수 있는 상태-동작 기반 계획 그래프와 동작-기반 휴리스틱을 제안한다. 상태-동작 기반 계획그래프는 계획문제 풀이를 위한 휴리스틱 계산에 이용되는 간략화된 계획그래프를 부속 목표와 목표조건들 간의 상호작용을 찾는데 적용할 수 있도록 확장한 자료구조로써, 상태-동작 기반 계획그래프를 이용하는 동작 기반 휴리스틱은 보다 효과적인 방법으로 부속 목표와 목표조건들 간의 상호작용을 찾아내고, 이들을 목표 도달 거리 계산에 이용한다. 따라서 동작-기반 휴리스틱은 종래의 최대 휴리스틱, 합산 휴리스틱 보다 더 높은 정보력을 가지며 겹침 휴리스틱보다 더 적은 계산 노력을 통해 동일한 결과를 얻을 수 있다. 본 논문에서는 동작-기반 휴리스틱을 계산하는 알고리즘을 제시하고, 동작-기반 휴리스틱의 정확성과 효율성을 알아보기 위한 실험적 분석에 대해 설명한다.

Abstract More informative ones of heuristics can help to conduct search more efficiently to obtain solution plan. However, in general, to derive highly informative heuristics from problem specifications requires lots of computational effort. To address this problem, we propose an State-Action based Planning Graph(SAPG) and Action-based heuristics for solving planning problems more efficiently. The SAPG is an extended one to be applied to can find interactions between subgoal & goal conditions from the relaxed planning graph which is a common means to get heuristics for solving the planning problems, Action-based heuristics utilizing SAPG graphs can find interactions between subgoal & goal conditions in an effective way, and then consider them to estimate the goal distance. Therefore Action-based heuristics have more information than the existing max and additive heuristics, also requires less computational effort than the existing overlap heuristics. In this pager. we present the algorithm to compute Action-based heuristics, and then explain empirical analysis to investigate the accuracy and the efficiency of the Action-based heuristics.

Keywords : Classical Planning, Delete Relaxation, Planning Graph, Search Heuristic, Supporting Actions

1. 서론

일반적으로 전통적인 인공지능 계획방식(Classical Planning)에서는 상태 공간상의 탐색을 통한 휴리스틱을 이용하여 계획생성을 수행한다. 이때, 특정 영역(domain)에 귀속되지 않고 계획문제 명세로부터 정보를 추출하여 상태 공간상의 탐색을 수행하여 계획생성을 하기 위해서는 영역-독립적인(domain- independent) 탐색 휴리스틱을 이용할 필요가 있다. 이러한 영역-독립적인 탐색 휴리스틱을 얻기 위해서 널리 이용되고 있는 효과적인 방법이 간략화된 계획그래프(Relaxed Planning Graph, RPG)[1,2,11-13]이다. 이러한 간략화된 계획그래프는 각

*Corresponding Author : Hyun-Sik Kim(Seoil University)
Tel: +82-2-490-7388 email: hskim@seoil.ac.kr
Received August 24, 2015 Revised September 10, 2015
Accepted September 11, 2015 Published September 30, 2015

동작의 부정 효과(negative effect)를 모두 삭제하여 동작들 사이의 부정적 상호작용을 배제하는 삭제 간략화(delete relaxation)[3,4]를 이용한다. 일반적으로 이러한 삭제 간략화를 이용하는 간략화된 계획그래프의 해 계획(solution plan)은 허용 가능한 휴리스틱(admissible heuristic)[5,6,14]을 제공할 수 있는 것으로 알려져 있다. 하지만, 휴리스틱을 이용한 탐색을 통해서 사용자 또는 시스템이 요구하는 양질의 해 계획을 구하기 위해서는 휴리스틱의 계산 효율성, 정확성 등이 요구된다. 여기서 말하는 휴리스틱의 정확성은 주어진 상태로부터 목표까지의 거리, 즉 실제 거리에 얼마나 가까운 휴리스틱을 구할 수 있는가에 따라 결정이 된다. 이러한 휴리스틱의 정확성이 높아질수록 정확한 탐색을 유도해 탐색의 효율성은 높아진다. 하지만, 이러한 실제 거리에 가까운 휴리스틱을 구하는 과정은 실제 거리에 가까우면 가까울수록 휴리스틱 계산 과정이 복잡하고 계산량이 매우 많아진다는 문제점을 가지고 있다. 이와 같은 문제들로 인하여 계산량을 줄이기 위해 정보력이 낮은 최대 휴리스틱(max heuristic)[1]과 같은 단순한 형태의 휴리스틱을 이용하거나 최적의 해 계획을 보장하지는 못하지만 휴리스틱의 정보력이 조금 더 높은 합산 휴리스틱(additive heuristic)[7], 또는 계산량이 많고 복잡하지만 정보력이 높은 겹침 휴리스틱(overlap heuristic)[4]과 같은 휴리스틱을 이용하고 있다. 따라서 효율적 탐색을 통해서 해 계획을 보다 효과적으로 구하기 위해서는 계산량을 최소화하면서도 정보력이 높은 휴리스틱에 대한 연구가 필요하다[9].

본 논문에서는 이러한 효율적인 휴리스틱 계산을 위해서, 계산 노력을 줄이면서도 정보력이 높은 휴리스틱 계산을 위한 새로운 자료구조인 상태-동작 기반 계획그래프(State-Action based Planning Graph, SAPG)와 이를 바탕으로 한 동작-기반 휴리스틱(Action-based Heuristic)을 소개한다. 상태-동작 기반 계획그래프는 기존의 전통적 계획문제 풀이에 사용되던 간략화된 계획그래프(Relaxed Planning Graph)의 기능과 함께 상태 정보를 구성하는 상태조건들을 만족시키는 동작 정보를 보다 쉽게 파악할 수 있는 정보를 포함한 자료구조이다. 본 논문에서는 상태-동작 기반 계획그래프와 이를 바탕으로 한 동작-기반 휴리스틱의 계산 과정에 대한 소개에 이어 동작-기반 휴리스틱의 정확성과 효율성을 분석하기 위한 비교 실험 결과를 설명한다.

2. 그래프 기반 탐색 휴리스틱

2.1 간략화된 계획그래프

주어진 계획문제로부터 영역-독립적인 휴리스틱(domain-independent heuristic)을 자동적으로 구하기 위한 간략화된 계획그래프(Relaxed Planning Graph)는 Graphplan[8]에서 소개된 원래의 계획그래프(Planning Graph)에서 부정적 효과를 무시하는 삭제 간략화(delete relaxation)를 통하여 휴리스틱을 손쉽게 계산할 수 있도록 도와주는 대표적인 자료구조이며, 전통적 계획문제를 해결하기 위한 상태 공간 계획 알고리즘과 휴리스틱 계산에 널리 이용되고 있다[10].

이러한 간략화된 계획그래프는 Fig. 1에서 보는 바와 같이 상태 층(literal layer)과 동작 층(action layer)으로 구성되어 있으며 이러한 상태 층과 동작 층이 교대로 펼쳐지는 하나의 계층화된 자료구조이다. 계획그래프는 이전 상태 조건과 동작을 유지하며 목표조건을 모두 만족할 때까지 그래프를 확장한다. 이때, 부정 효과가 제거된 동작들만이 사용되며 동작들 간의 관계는 검사하지 않는다. 따라서 임의의 한 상태에서 단위 목표까지의 최단 도달거리를 추정할 수 있는 정보를 제공한다.

간략화된 계획그래프로 얻어진 각 단위 목표조건까지의 최단 도달거리 추정치를 이용하여 전체 목표조건들을 모두를 만족하는 상태까지의 휴리스틱을 계산하는 대표적인 방법들로는 최대 휴리스틱(max heuristic)과 합산 휴리스틱(additive heuristic), 그리고 겹침 휴리스틱(overlap heuristic) 등이 있다.

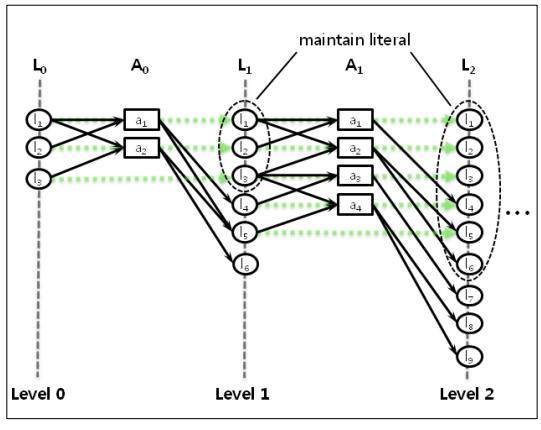

Fig. 1. Relaxed Planning Graph

2.2 탐색 휴리스틱

최대 휴리스틱(max heuristic)과 합산 휴리스틱(additive heuristic)은 식 (1), (2)에서와 같이 상태 s에서 각 단위 목표(g_k)까지의 최단 도달거리 중 최대값(max)과 합산값(additive)을 각각 휴리스틱 값으로 이용한다.

식 (1)에서 정의한 것과 같이 최대 휴리스틱은 목표를 구성하는 모든 목표조건들 간에 긍정적 상호작용이 존재한다고 가정하기 때문에 휴리스틱으로 최대값을 이용한다. 이러한 최대 휴리스틱은 휴리스틱 허용성을 만족하고 계산이 간편하다는 장점을 가지지만, 일반적으로 휴리스틱 정보력이 낮고 서로 다른 상태들 간의 휴리스틱 평가치의 차이가 적어 탐색 과정에서 변별력이 떨어지는 문제점을 가지고 있다. 때문에 휴리스틱 허용성을 반드시 만족해야 하는 계획문제이거나 탐색 공간이 적은 문제에 많이 사용되고 있다.

$$h_{\max}(s) = \max \{\mathbf{dist}_{\text{delete-relax}}(s, g_k)\} \quad (g_k \in G,\ k = 1, \ldots, n) \tag{1}$$

최대 휴리스틱과 달리 합산 휴리스틱은 목표조건이 독립된 관계로 존재한다고 보고 각 목표조건을 달성하기 위해서는 목표조건을 위한 개별 동작들을 수행해야 한다고 가정한다. 이러한 합산 휴리스틱은 휴리스틱 값의 선택 폭이 넓어지기 때문에 최대 휴리스틱에 비해 정보력이 높아 일반적으로 최대 휴리스틱 보다 더 효율적으로 주어진 목표에 도달할 수 있지만, 실제 목표 도달거리와 차이가 크며 여전히 사용자의 요구를 수용할 수 있을 만큼의 정보력을 갖추지는 못했다는 문제점을 가지고 있다.

$$h_{\text{additive}}(s) = \sum \mathbf{dist}_{\text{delete-relax}}(s, g_k) \quad (g_k \in G,\ k = 1, \ldots, n) \tag{2}$$

겹침 휴리스틱(overlap heuristic)은 계획기 FF[4]에서처럼 간략화된 계획그래프 위에서 목표와 관련한 지지 동작들(supporting actions)을 역방향으로 분석하며 목표조건들 간의 연관 관계와 사용된 동작들 간의 상호작용을 계산함으로써 주어진 목표 집합을 모두 달성하는데 필요한 간략화된 해 계획을 구하고, 그 해 계획의 길이를 휴리스틱 값으로 취한다. 연관관계와 상호작용을 고려하기 때문에 일반적으로 최대 휴리스틱과 합산 휴리스틱에 비해서 실제 목표 도달거리에 보다 가까운 휴리스틱을 구할 수 있지만, 상태 조건을 만족시키는 동작을 결정하기 위한 정보가 부족하여 중복 검사, 지지 동작 선택 문제 등으로 인해 그 과정이 복잡하고 계산량이 많다는 문제점을 가진다. 이러한 겹침 휴리스틱은 식(3)과 같이 정의할 수 있다.

$$h_{\text{overlap}}(s) = \text{Merge}\{\mathbf{dist}_{\text{delete-relax}}(s, g_k)\} \quad (g_k \in G,\ k = 1, \ldots, n) \tag{3}$$

계산 노력을 최소화하면서도 휴리스틱의 정보력을 높이기 위해서는 목표상태를 구성하는 목표조건들의 집합과 그 과정을 달성하기 위한 과정의 상태 조건들 간의 상호의존성을 보다 쉽게 파악하고 이를 휴리스틱 계산에 반영할 수 있어야 한다.

기존의 계획그래프는 Fig. 1에서 보는바와 같이 상태 정보를 나타내는 상태 조건들이 초기 상태 층(L_0)부터 계속 유지되고(maintain literal) 동작들 또한 계속 유지되게 된다. 겹침 휴리스틱과 같은 경우, 목표를 달성하기까지의 상태 조건들 간의 상호의존성을 파악하기 위해서 동작들을 일일이 검사하여 상태조건과 관련된 동작들을 추출해야 하고, 추출된 동작 정보를 이용하여 정보를 일일이 모두 추적하는 과정을 반복해야 한다. 이는 상태 층과 동작 층에서 동작들과 상태조건들을 제외한 다른 정보는 가지고 있지 않을 뿐 아니라 실제 상태 조건들이 어느 층에서 어느 동작으로 인해 만족했는지를 확인하기 어렵기 때문이다. 때문에 정보력이 높은 휴리스틱을 구하기 위해서는 상태 조건들 간의 상호의존성 파악이 필요하며, 이를 보다 쉽게 파악하고 이를 휴리스틱에 반영할 수 있도록 하는 연구가 필요한 상황이라고 할 수 있다.

3. 동작 기반 휴리스틱

상태 조건들 간의 상호의존성을 파악하기 위해서는 휴리스틱 계산 과정에서 상태 조건의 만족 과정과 그와 관련한 정보를 추적할 수 있도록 다양한 정보 제공이 필요하다. 하지만, 기존의 계획그래프는 동작들과 상태조건들을 제외한 다른 정보의 부족으로 인하여 중복 검사 문제와 동작의 선택 문제 등이 발생하고 이를 해결하기

위해서는 많은 계산 노력을 필요로 한다. 본 논문에서는 이러한 상태 조건들 간의 상호의존성을 보다 쉽고 효율적으로 파악할 수 있는 새로운 계획그래프인 상태-동작 기반 계획그래프와 이를 바탕으로 한 동작-기반 휴리스틱을 제안한다.

계획문제 P=(s_0, G, O)는 초기상태 s_0와 목표상태를 나타내는 목표조건들의 집합 G, 그리고 동작들의 집합 O로 주어진다. 이러한 계획문제 P=(s_0, G, O)를 풀기 위해서 상태-동작 기반 계획그래프는 초기상태 s_0에 대해서 Fig. 2와 같이 계획그래프 전개 과정을 수행한다. 초기상태 s_0에서부터 목표조건들의 집합 G에 포함된 모든 목표조건들이 등장할 때까지 기존의 간략화된 계획그래프와 같이 계획그래프를 전개한다. 이때, 계획그래프의 각 초기 상태 층을 포함한 계층별 전개과정에서 상태 조건을 만족시키는 동작들을 확인하여 다음과 같은 정보부여 과정을 수행한다.

정보부여 과정 :

- Fig. 2에서와 같이 초기 상태 층(L_0)을 구성하는 모든 상태조건들에 대해서 초기상태 조건임을 나타내는 init 정보를 부여한다.
- 계획그래프 전개 과정에서 Fig. 2의 l_4, l_5, l_6와 같이 확장된 계층에서 새롭게 만족되는 상태 조건들에 대해서는 (l_4, a1), (l_5, a1), (l_6, a2)와 같이 해당 조건을 만족시키는 동작 정보를 부여한다. 이때, 유지 동작을 포함하여 동일한 동작 층에서 상태 조건을 만족시키는 새로운 동작이 발생하더라도 정보부여 과정을 통해 부여된 정보는 수정되지 않는다.

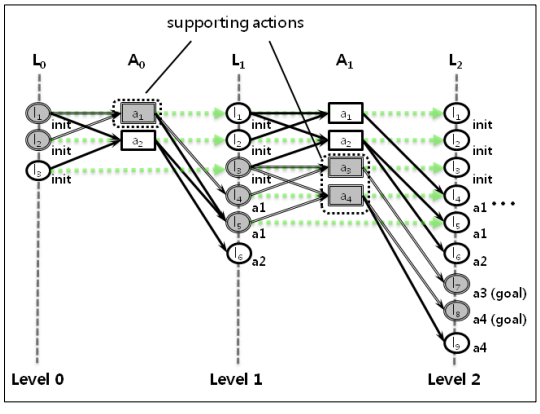

Fig. 2. State-Action based Planning Graph

이러한 상태-동작 기반 계획그래프를 통해서 계산되는 동작-기반 휴리스틱의 알고리즘은 Table 1과 같다. 먼저 휴리스틱 알고리즘 계산에 필요한 자료구조를 초기화한다(Table 1, line 3~8). 그리고 초기상태가 목표상태 조건들을 만족하지 않으면 초기상태 조건들에 정보부여 과정을 수행한다(Table 1, line 9~13). 초기상태 조건들에 대한 정보부여 과정이 완료되면 모든 목표조건들이 만족될 때까지 계획그래프를 한 단계씩 확장하면서 (Table 1, line 14~15) 새롭게 등장한 상태조건들을 만족시키는 동작들을 확인한다(Table 1, line 16). Table 2는 이러한 새롭게 등장한 상태조건들을 만족시키는 동작들을 확인하는 과정을 나타낸다. 새롭게 등장한 상태조건들에 대해서만 확인하기 때문에 이전 상태 층을 구성하는 상태조건들을 제외한 새롭게 등장한 상태조건들에 대해서만 정보부여 과정을 수행한다(Table 2, line 3~8). 이때, 새롭게 등장한 상태조건이라 하더라도 해당 상태조건을 만족시키는 동작이 다수 존재할 수 있기 때문에 정보부여 과정이 이미 수행된 상태조건인지 확인한다 (Table 2, line 6~7).

Table 1. Action-based Heuristic Algorithm

```
1.
2.    Action-based_Heuristic(s0, G, O)
3.    Begin
4.      L0 = s0; /* Initial Literal Layer */
5.      k = 1; /* Extend Level */
6.      action_count = 0; /* Action-based Heuristic */
7.      goal_Dependency_action = null
8.      actionTable = null /* Action-Literal Interaction Map */
9.      if (G ⊄ L0) {
10.       For each literal li ∈ L0 do
11.         actionTable.put(li, init);
12.       End
13.   }
14.     while (G ⊄ Lk-1) {
15.       (Ak-1, Lk) = APG_Expand_Level(Lk-1, O);
16.       Find_AL_Relation(Ak-1, Lk , Lk-1);
17.       k = k +1;
18.     }
19.     if ( ¬actionTable.EMPTY() ) {
20.       action_count = Find_Action_Dependency(G)
21.     }
22.     if (action_count == 0)
23.       /* s0 is goal state */
24.     else
25.       return action_count;
26.   End
```

Table 2. Find_AL_Relation Function

```
1.
2.    Find_AL_Relation(A_{k-1}, L_k , L_{k-1})
3.    Begin
4.      For each new literal li ∈ L_k − L_{k-1} do
5.        Select an action o from A_{k-1} s.t. li ∈ effect(o);
6.        If ¬actionTable.CONTAINS(li) then
7.          actionTable.put(li, o);
8.      End
9.      return null;
10.   End
```

정보부여 과정과 함께 모든 목표조건들이 등장할 때까지 계획그래프 전개 과정이 완료되면 정보부여 과정을 통해 기록된 정보를 바탕으로 목표조건들을 만족시키는 관련 동작들과 상태조건들에 대한 검사를 수행한다 (Table 1, line 19~21). Table 3은 이러한 목표조건들을 만족시키는 관련 동작들과 상태조건들에 대한 검사를 수행하는 과정을 나타낸다. 계획그래프 확장 과정에서 정보부여 과정을 통해 부여된 정보를 이용하여 모든 목표조건들에 대해서 목표조건을 만족시키는 동작을 추출하고 해당 동작을 계획생성에 이용된 목표 관련 지지 동작(supporting actions)에 포함한다. 이때, 동작 검사를 통해서 목표 관련 지지 동작 간의 중복 저장 및 추적 과정이 발생하지 않도록 한다(Table 3, line 5~14). 검출된 동작이 지지 동작에 포함되게 되면 해당 동작이 수행되기 위한 전제조건(precondition)을 경로 추적을 위한 검사 항목에 포함시킨다. 이때, 동작이 새롭게 지지 동작으로 선택되어 저장되는 경우에만 해당 동작의 전제조건이 검사항목에 포함되게 된다(Table 3, line 10~13). 검사 항목에 저장된 상태조건들을 바탕으로 동작 확인 과정과 저장 과정을 반복하며 경로를 추적하게 된다(Table 3, line 15~30). 하나의 검사 항목에 대한 추적 과정은 초기 상태 조건에 도달하거나, 또는 이미 경로 탐색을 진행한 경우 마치게 되며, 검사 항목이 없을 때까지 반복된다. 이러한 과정을 통해서 구해진 목표 관련 지지 동작들은 주어진 목표조건을 달성하기 위한 최소한의 관련 동작들이 되고 이를 바탕으로 휴리스틱을 계산하게 된다.

이와 같이 상태-동작 기반 계획그래프를 이용한 동작-기반 휴리스틱은 계획그래프 확장 과정에서 생성되는 정보를 상호의존성 파악에 이용하기 때문에 그래프 확장 과정 이외에 추가적인 노력을 필요로 하지 않으며 동작층에 포함된 동작들을 일일이 검사하지 않고도 관련 동작을 쉽게 추출할 수 있다. 또한 목표조건들과 관련된 지지 동작 검사 과정에서 중복 검사 문제와 동일 상태조건에 대한 복수의 지지 동작 선택 문제 등을 방지할 수 있다. 따라서 상태-동작 기반 계획그래프를 이용한 동작-기반 휴리스틱은 종래의 최대 휴리스틱과 합산 휴리스틱보다 더 높은 정보력을 가지며 겹침 휴리스틱보다 더 적은 계산 노력을 통해서 동일한 휴리스틱 평가치 계산이 가능하다.

Table 3. Find_Action_Dependency Function

```
1.    Find_Action_Dependency(G)
2.    Stack new_literal = null;
3.    boolean query_result = true;
4.    Begin
5.      For each g ∈ G_k do
6.        Select an action o from A_{k-1}
7.                                  s.t. g ∈ effect(o);
8.          If each g ∈ effect(o) s.t o is different then
9.            action a_k = actionTable.query(g);
10.           If          ¬goal_Dependency_action.CONTAINS(a_k)
11. then
12.             goal_Dependency_action.put(a_k);
13.             new_literal.PUSH(precondition(a_k));
14.     End
15.     while ( ¬new_literal = null) {
16.       li = new_literal.POP();
17.       action a_k = actionTable.query(li);
18.       if ( a_k == init) {
19.         /* a_k is initial state condition */
20.         } else {
21.           query_result                                    =
22. goal_Dependency_action.CONTAINS(a_k);
23.         }
24.         if (query_result == false) {
25.           goal_Dependency_action.put(a_k);
26.           new_literal.PUSH(precondition(a_k));
27.     }
28.   /* if query_result is true then a search is already action a_k
29. */
30.     }
31.     return goal_Dependency_action.count();
32.   End
```

4. 평가

본 논문에서 제안한 동작-기반 휴리스틱을 평가하기 위하여 제안하고 있는 상태-동작 기반 계획그래프의 기반이 되는 간략화된 계획그래프를 이용하는 대표적인 휴리스틱인 최대 휴리스틱과 합산휴리스틱, 그리고 겹침 휴리스틱과 비교 실험을 전개하였다. 이를 통해서 동작-기반 휴리스틱의 정확성과 계산 효율성을 확인하고자 하였다. 일반적으로 휴리스틱의 정확성이 높을수록 탐색을 보다 정확히 수행하기 때문에 효율적인 탐색을 유도한다는 것을 알 수 있다. 따라서 최소 도달거리와 휴리스틱

평가치의 비교를 휴리스틱의 정확성을 확인하기 위한 평가 척도로 이용하였으며, 이 과정에서 필요한 동작 검사 횟수를 이용하여 계산 효율성에 대한 평가 척도로 사용하였다.

실험에서 탐색을 위한 알고리즘으로는 전통적 계획문제에 가장 일반적으로 사용되는 A* 알고리즘을 이용하였으며, 비교 실험 도메인으로는 생활 서비스 도메인과 블록 도메인을 이용하였고 각 도메인마다 임의로 생성한 5개의 계획문제를 이용하여 비교 실험을 전개하였다. 비교 실험에 이용된 도메인을 간단히 살펴보면, 생활 서비스 도메인은 가정에서의 로봇 서비스와 사용자 일정 생성을 위한 서비스가 결합된 도메인으로서 사용자의 일정과 로봇에게 주어진 업무에 대한 계획 생성을 동시에 진행하게 된다. 이때, 로봇에게 주어지는 업무와 사용자 일정은 서로 별도의 계획이 아닌 물을 가져다주는 동작(give_the_water_to_user)과 같이 서로 연관된 업무 등으로 구성되어 있으며, 이를 위해서 로봇이 공간 사이를 이동하기 위한 move, 문을 여는 open_door, 물건 나르기와 같은 사용자 지시 사항을 수행과 관련한 carry, putdown 등의 동작과 user_move, visit, have_dinner, watching_TV, lesson과 같은 사용자 일정 생성을 위한 동작들을 포함하고 있다. 블록 도메인은 pickup, pickup-table, putdown, putdown-table과 같은 블록을 집거나 내려놓기 위한 동작들을 포함하고 있다.

Table 4. Heuristic Estimates and Minimum Cost to Goal State from the Initial State

Domain	Life Service Domain				
Heuristic	p1	p2	p3	p4	p5
Minimum Cost	5	8	9	13	15
Max	3	7	6	10	12
Additive	7	16	12	22	26
Overlap	5	9	8	12	14
Action-based	5	9	8	12	14
Domain	Block Domain				
Heuristic	p1	p2	p3	p4	p5
Minimum Cost	5	7	11	14	19
Max	3	4	4	4	6
Additive	8	17	22	34	47
Overlap	5	8	11	16	22
Action-based	5	8	11	16	22

Table 4는 동작-기반 휴리스틱의 정확성을 알아보기 위한 실험 결과로써, 주어진 계획문제의 초기 상태에서 목표까지의 실제 최소 도달비용(거리)와 이것을 추정한 휴리스틱 평가치를 비교하고 있다. 결과를 비교해보면, 두 도메인 모두에서 최대 휴리스틱과 합산 휴리스틱에 비해 제안하고 있는 동작-기반 휴리스틱의 평가치가 더 정확하다는 것을 알 수 있다. 또한 블록 도메인의 경우에는 동작-기반 휴리스틱의 정확성이 상대적으로 매우 높음을 알 수 있다. 이러한 결과는 실험에 사용된 도메인의 특성을 감안했을 때, 목표를 구성하는 목표조건의 수와 적용 가능한 동작이 많아 상호작용이 보다 복잡할수록 상호의존성 분석이 필요하기 때문으로 판단된다.

Table 5. Number of Action Check for Supporting Actions

Domain	Life Service Domain				
Heuristic	p1	p2	p3	p4	p5
Overlap	5	9	9	14	16
Action-based	5	8	9	13	15
Domain	Block Domain				
Heuristic	p1	p2	p3	p4	p5
Overlap	11	14	24	27	32
Action-based	6	10	14	18	22

Table 5는 동작-기반 휴리스틱의 계산 효율성을 알아보기 위한 실험 결과로써, 휴리스틱 계산에 이용되는 목표 관련 지지 동작을 구하기 위해 검사한 동작들의 수를 비교한 결과이다. 비교 대상 휴리스틱 가운데 최대 휴리스틱과 합산 휴리스틱의 경우, 상호의존성을 계산하지 않기 때문에 겹침 휴리스틱만을 비교 대상으로 하였다. 겹침 휴리스틱과 동작-기반 휴리스틱 모두, 휴리스틱 계산을 위해서는 확장이 완료된 계획그래프 상에서 목표와 관련한 지지 동작을 구하여 이를 휴리스틱으로 이용하기 때문에 목표지지 동작을 구하기 위해 검사한 동작의 수를 통해서 계산 효율성을 분석할 수 있다. 결과를 비교해보면 동작-기반 휴리스틱이 겹침 휴리스틱에 비해 지지 동작을 구하기 위해 더 적은 수의 동작 검사를 수행했음을 알 수 있다. 이를 통해서 동작-기반 휴리스틱이 실험 도메인에서 최대 휴리스틱과 합산 휴리스틱보다 더 높은 정확성을 가지며 겹침 휴리스틱보다 휴리스틱 계산 효율성이 더 높음을 알 수 있었다.

5. 결론

본 논문에서는 보다 효율적으로 양질의 탐색 휴리스틱을 자동으로 도출하기 위한 새로운 자료구조인 상태-동작 기반 계획그래프와 이를 이용한 새로운 휴리스틱인 동작-기반 휴리스틱을 제안하였다. 그리고 비교 실험을

통해 동작-기반 휴리스틱이 목표조건들 간의 상호의존성을 분석하지 않는 기존의 최대 휴리스틱이나 합산 휴리스틱보다 더 높은 정확성을 보인다는 것을 확인하였다. 또한 같은 비교 실험을 통해서 같은 정확성을 보인 겹침 휴리스틱에 비해 본 논문에서 제안하고 있는 동작-기반 휴리스틱이 더 높은 계산 효율성을 나타냄을 확인하였다. 추후에는 제안하는 휴리스틱의 활용 가능성을 보다 명확하게 하기 위하여 다양한 도메인과 보다 많은 계획 문제를 이용한 실험 보완과 지속적으로 연구되고 있는 다른 휴리스틱과의 비교도 필요할 것으로 판단된다. 또한 실세계 계획문제에 적용하기 위하여 조건부 계획문제(contingent planning)로의 확대 적용을 위한 연구가 필요할 것으로 판단된다.

References

[1] D. Bryce and S. kambhampati, “A Tutorial on Planning Graph-Based Reachability Heuristics", AI Magazine, vol.28, no.1, pp.47-83, 2006.
DOI: http://dx.doi.org/10.1609/aimag.v28i1.2028

[2] D. Cai, M. Yin and J. Wang, “Improving Relaxed Plan-Based Heuristics via Simulated Execution of Relaxed-Plans”, ICAPS, 2009.

[3] B. Bonet and H. Geffner, “Planing as Heuristic Search”, Artificial Intelligence, pp.5-33, 2001.
DOI: http://dx.doi.org/10.1016/S0004-3702(01)00108-4

[4] J. Hoffmann and B. Nebel, “The FF Planning System: Fast Plan Generation through Heuristic Search”, Journal of AI Research, vol.14, pp.253-302, 2001.
DOI: http://dx.doi.org/10.1613/jair.855

[5] P. Haslum and H. Geffner, “Admissible Heuristics for Optimal Planning”, AIPS2000, pp140-149, 2000.

[6] S. Russell and P. Norving, Artificial Intelligence: A Modern Approach, 3nd Edition, PearsonEducation, 2010.

[7] E. Keyder and H. Geffner, “Set-Additive ad TSP heuristics for planning with action costs and soft goals”, ICAPS, 2007.

[8] A. Blum and M.Furst, “Fast Planning through Planning through Planning Graph Analysis", IJCAI, 1995.
DOI: http://dx.doi.org/10.1016/S0004-3702(96)00047-1

[9] Rosslin John Robles, Fast Nearest-Neighbor Search Algorithms Based on High-Multidimensional Data, Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.3 No.1, June (2013), pp.17-24,
DOI: http://dx.doi.org/10.14257/AJMAHS.2013.06.01

[10] Yvette E. Gelogo, Haeng-Kon Kim, Enterprise Resource Planning System Deployment on Mobile Cloud Computing, *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, Vol.3 No.1, June (2013), pp.1-8,
DOI: http://dx.doi.org/10.14257/AJMAHS.2013.06.02

[11] S.-U. Lee, “Travelling Salesman Problem Based on Area Division and Connection Method,” *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, Vol. 15, No. 3, pp. 211-218, 2015.
DOI: http://dx.doi.org/10.7236/JIIBC.2015.15.3.211

[12] J.-Y. Chang, “Efficient Retrieval of Short Opinion Documents Using Learning to Rank,” *The Journal of The Institute of Internet, Broadcasting and Communication,* VOL. 13 No. 4, pp. 117-126, 2013.
DOI: http://dx.doi.org/10.7236/JIIBC.2013.13.4.117

[13] K.-Y. Lee, I.-H. Seo, M.-J. Lim, K.-H. Kim, J.-L. Kim, “Design and Implementation of a Efficient Search Engine Using Collaborative Filtering,” *The Journal of The Institute of Webcasting, Internet and Telecommunication,* VOL. 12 No. 3, pp. 23-28, 2012.
DOI: http://dx.doi.org/10.7236/JIWIT.2012.12.3.23

[14] Y.-M. Kwon, I.-R. Lee, M.-G. Kim, “A Study on Clustering of SNS SPAM using Heuristic Method,” T*he Journal of The Institute of Internet, Broadcasting and Communication (IIBC),* Vol. 14, No. 6, pp.7-12, Dec. 31, 2014.
DOI: http://dx.doi.org/10.7236/JIIBC.2014.14.6.7

[15] S.-U. Lee, M.-B. Choi, “Simple Solution for Multi-commodity Transportation Problem,” *The Journal of The Institute of Internet, Broadcasting and Communication*, VOl. 13, No. 5, Oct. 2013.
DOI: http://dx.doi.org/10.7236/JIIBC.2013.13.5.173

김 현 식(Hyun-Sik Kim) **[종신회원]**

- 2004년 2월 : 경기대학교 일반대학원 전자계산학과 (이학석사)
- 2010년 8월 : 경기대학교 일반대학원 전자계산학과 (이학박사)
- 2011년 3월 ~ 현재 : 서일대학교 인터넷정보과 조교수

<관심분야>
자동계획, 시멘틱웹, 지능로봇, 에이전트, 데이터마이닝