

모순 문제 해결을 위한 의사결정트리 기반 나비 알고리즘의 개발과 적용

현정석[†] · 고예준^{††} · 김용결^{†††} · 전승재^{††††} · 박찬정^{†††††}

요 약

모순에 대한 일반적인 생각은 모순을 해결 가능성이 전혀 없는 공집합이나 논리적으로 틀린 것이다. 두 가지 대안 중에서 어느 쪽도 바람직하지 못한 결과를 초래하는 딜레마는 그 안에 숨어 있는 모순을 해결해야 하므로 해결이 어렵다. 하지만 이런 특성으로 인해 역설적으로 모순 해결은 혁신적이고 창의적인 문제 해결로 간주 되어왔다. 문제의 해법을 모순 해결의 관점에서 분석하는 트리즈(TRIZ)는 그동안 컴퓨터보다는 인간의 관점에서 문제 해결 방법으로 사용되었다. 트리즈처럼 모순 해결 중심으로 문제를 분석하는 나비 모형은 문제 해결의 자동화 관점에서 기호 논리학을 이용하여 모순 문제의 유형을 분석하였다. 모순 문제유형별 구체적 해결전략을 적용하기 위해 본 연구에서는 의사결정트리 기반의 나비 알고리즘을 설계하였다. 본 연구는 파이선 tkInter를 바탕으로 주어진 모순 문제의 구체적 해결전략을 찾아 사용자들에게 제시하는 시각화 도구를 개발하였다. 개발한 도구를 검증하기 위하여 중학교 3학년 학생들이 나비 알고리즘을 학습한 후, 나무지지대의 모순 문제를 분석하도록 하였다. 학생들이 새로운 해결책을 찾아 발명대회에 참가하여 대상을 받았다. 본 연구에서 개발한 의사결정트리 기반 나비 알고리즘은 문제 해결 초기에 문제의 해결공간을 체계적으로 줄여주어 시행착오 없이 모순 문제를 해결하는데 도움을 줄 수 있다.

주제어 : 모순 해결방법, 알고리즘, 트리즈, 창의성, 시각화

Development and Application of the Butterfly Algorithm Based on Decision Making Tree for Contradiction Problem Solving

Jung Suk Hyun[†] · Ye June Ko^{††} · Yung Gyeol Kim^{†††} ·
Seungjae Jean^{††††} · Chan Jung Park^{†††††}

ABSTRACT

It is easy to assume that contradictions are logically incorrect or empty sets that have no solvability. This dilemma, which can not be done, is difficult to solve because it has to solve the contradiction hidden in it. Paradoxically, therefore, contradiction resolution has been viewed as an innovative and creative problem-solving. TRIZ, which analyzes the solution of the problem from the perspective of resolving contradictions, has been used for people rather than computers. The Butterfly model, which analyzes the problem from the perspective of solving the contradiction like TRIZ, analyzed the type of contradiction problem using symbolic logic. In order to apply an appropriate concrete solution strategy for a given contradiction problems, we designed the Butterfly algorithm based on decision making tree. We also developed a visualization tool based on Python tkInter to find concrete solution strategies for given contradiction problems. In order to verify the developed tool, the third grade students of middle school learned the Butterfly algorithm, analyzed the contradiction of the wooden support, and won the grand prize at an invention contest in search of a new solution. The Butterfly algorithm developed in this paper systematically reduces the solution space of contradictory problems in the beginning of problem solving and can help solve contradiction problems without trial and errors.

Keywords : Contradiction Solving, Algorithm, TRIZ, Creativity, Visualization

[†]중심회원: 제주대학교 경영정보학과

^{††}정 회원: 탐라중학교

^{†††}중심회원: 제주대학교 컴퓨터교육과(교신저자)

논문접수: 2018년 11월 22일, 심사완료: 2019년 1월 22일, 게재확정: 2019년 1월 28일

* 본 논문은 2018년 제주대학교 과학영재교육원의 지원으로 수행되었음. 2018 한국컴퓨터교육학회 하계 학술대회 발표 논문인 '분할과 결합 전략의 설계를 중심으로 한 나비 모형에 기반을 둔 모순 문제 해결 알고리즘'을 확장하였음.

^{††††}정 회원: 노형중학교

^{†††††}정 회원: 제주제일중학교

1. 서론

최근 IT 영역에서는 4차 산업혁명 시대와 맞물려 인공지능에 대한 관심이 고조되고 있다. 컴퓨터가 사람 대신에 처리할 수 있는 일이 계속 증가하고 있다. 한편, 컴퓨터가 문제를 해결하고 처리하는 방법인 알고리즘 영역에서는 전통적으로 분할정복법(divide and conquer), 되추적기법(back-tracking) 등과 같이 특정 유형의 문제들을 해결하는 방법이 개발되어왔다[1]. 본 연구에서는 여러 가지의 문제 유형 중에서 모순 문제에 초점을 두고 있다. 모순 문제는 원하는 결과를 얻고자 도입한 것이 원하지 않는 다른 결과를 초래하는 상황에서 발생한다. 예를 들어, 악성종양을 앓고 있는 환자를 치료하기 위해 쎬 방사선을 악성종양에 쏘면 악성종양을 파괴할 수 있지만, 악성종양을 둘러싸고 있는 일반 세포도 함께 파괴되는 문제가 있다. 결국, 악성종양을 파괴하려면 방사선의 강도는 세야 하지만, 일반 세포를 보호하려면 방사선의 강도는 약해야 하는 모순 문제가 있다. 방사선 강도의 문제는 360도 각도에서 약한 방사선을 쏘아 환자의 악성종양에 모이게 하면 가운데로 모여드는 방사선의 강도가 세져 해결된다. 모순 해결은 혁신적이고 창의적인 문제 해결 과정으로 간주되어 왔다[2][3][4].

러시아의 학자 알트슐러(Altshuller)는 혁신적이고 창의적인 특허 사례들을 분석하던 중에 문제 해결 원리의 공통점을 찾고 이들을 분류하여 트리즈(TRIZ) 이론으로 정립하였다[5][6]. 트리즈에는 발명 아이디어를 내는 40가지 방법, 기술시스템 진화 법칙, 멀티스크린 분석과 함께 모순 문제를 체계적으로 해결하기 위해 아리즈(ARIZ) 85-c 알고리즘[7]을 포함하고 있다. 트리즈를 이해하고 적용하려면 80시간 이상의 많은 시간을 학습해야 할 뿐만 아니라 시행착오 요소를 갖고 있다. 반면, 모순 문제를 해결하기 위한 나비 모형(Butterfly Model)은 기호 논리학을 바탕으로 모순 문제의 구성요소를 정의하고 모순 문제의 유형을 찾아 대응되는 문제 해결 전략을 제시한다. 이를 통해 문제 해결자들은 자신들이 해결해야 할 모순 문제의 유형에 맞는 해결전략을 구체적으로 정할 수 있다. 올바른 문제 해결 전략을 정함으로써 문제 해결 공간(solution

space)을 줄여 문제 해결을 위한 시행착오를 줄일 수 있다.

본 연구에서는 나비 모형 이론[8][9][10][11][13][14]에서 정의한 모순 문제 유형과 모순 문제유형별로 추상적인 문제 해결 전략을 소개한다. 이를 바탕으로 모순 문제의 유형에 맞게 기존 이론보다 구체적인 해결전략을 찾기 위해 의사결정트리 기반의 나비 알고리즘을 개발하였다. 본 연구는 파이선의 기본 GUI 모듈인 tkinter를 바탕으로 문제 해결자들이 해결해야 할 모순 문제의 구성요소를 입력하면 모순을 정의하고 구체적 해결전략을 찾아서 시각적인 다이어그램을 제시하는 도구를 설계하고 구현하였다.

또한, 본 연구에서 개발한 시각화 도구의 유용성을 검증하기 위해서 중학교 3학년 학생들에게 나비 알고리즘을 학습하게 한 후, 나무지지대의 모순 문제를 분석하도록 하였다. 학생들이 나무지지대 문제에 대한 구체적 해결전략을 탐색함으로써 나무지지대의 모순을 일으키는 요소를 정확하게 찾아내어 새로운 발명 아이디어를 끌어냈다. 학생들은 나무지지대의 모순 문제를 분석하고 발명 아이디어를 도출한 과정을 정리하여 광역지방자치단체의 발명대회에 참가하여 발표하였다. 나비 알고리즘을 이용해 문제를 해결한 학생들의 아이디어는 발명대회에서 가장 좋은 평가를 받아 학생 부문에서 대상을 받았다.

본 연구의 구성은 다음과 같다. 2장의 관련 연구에서는 기존 나비 모형을 기술한다. 내용으로 나비 모형의 구성요소, 나비 다이어그램의 역할, 모순 정의, 모순별 추상적 문제 해결 전략이 포함된다. 3장에서는 시간 및 구성요소 특성에 의해 모순 해결방법을 기술하고 알고리즘을 제시한다. 4장에서는 나비 다이어그램을 만드는 도구를 개발하고 적용 사례를 기술한다. 5장에서 결론을 맺는다.

2. 관련 연구

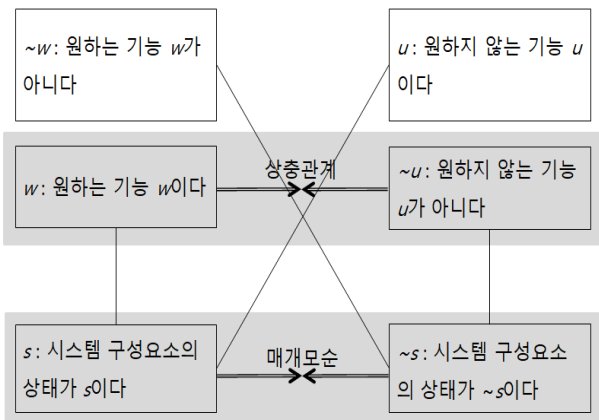
이 장에서는 기존의 나비 모형에서 정의한 모순 유형 및 유형별 추상적 문제 해결 전략을 기술한다.

나비 모형은 문제 해결을 위하여 시스템 구성요소를 시스템이 원하는 기능인 w (wanted function, w)와 w 로부터 발생하는 시스템 구성요소의 상태

s (state, s), s 로부터 발생하는 원하지 않는 기능인 u (unwanted function, u) 그리고 w, s, u 의 각각 부정(not)인 $\sim w, \sim s, \sim u$ 로 정의한다. 여기서 시스템이란 문제 해결에서 요구하는 기능과 구성요소들의 집합체이다.

나비 모형에서 원하는 기능(w)을 수행하면 s 상태에 도달하고, s 는 다시 원하지 않는 기능(u) 수행을 초래하기 때문에, 문제의 원하는 기능과 원하지 않는 기능인 w 와 u 는 s 가 초래한다고 가정한다[10][11][12][13][14]. 예를 들어, 명제 논리가 ($w \rightarrow s$)이고 ($s \rightarrow u$)이면, 대우 법칙에 따라 ($\sim u \rightarrow \sim s$)이고 ($\sim s \rightarrow \sim w$)이다. 즉 원하지 않는 기능인 u 의 수행을 차단하기 위해서 u 의 부정인 $\sim u$ 의 수행은 상태 $\sim s$ 를 초래하고, $\sim s$ 는 다시 원하는 기능 w 가 아닌 $\sim w$ 의 수행에 이르게 된다[13][14].

이를 그림으로 나타내면 다음 [그림 1]과 같다 [13][14]. 모순 문제에 있는 상충 관계(trade-off relation)와 매개모순 관계(mediating contradiction relation)를 다이어그램으로 나타낸 것을 나비 다이어그램(The Butterfly Diagram)이라 부른다. 나비 다이어그램은 모순 문제의 구조를 분석하고 모순을 정의하여 문제 해결에 도움을 준다 [13][14].



[그림 1] 나비 모형에서 문제의 구성요소 의미

원하는 기능 w 와 원하지 않는 기능 u 가 아닌 $\sim u$ 사이의 관계를 상충관계라 정의하며 w 와 $\sim u$ 는 문제 해결을 위해서 시스템이 모두 만족시켜야 할 문제 해결 목표(problem solving objective)를 구성한다. 상충관계를 발생시키는 것은 w 가 초래한 상태 s 와 $\sim u$ 가 초래한 상태 $\sim s$ 이다. 즉, w

와 $u, \sim u$ 와 $\sim w$ 관계를 매개시켜주는 역할을 수행한다. 따라서 s 와 $\sim s$ 간 모순 관계를 매개모순이라 하였다[13][14]. 또한, s 와 $\sim s$ 는 문제 해결을 위해서 시스템이 모두 만족시켜야 할 문제 해결 전략(problem solving strategy)을 구성한다.

나비 모형은 시스템 구성요소인 w, s, u 를 명제로 정의한 후, 이들 간의 관계를 조건 명제로 정의하였다. 임의의 두 조건 명제 p 와 q 의 관계에는 충분조건, 필요조건, 필요충분조건이 존재한다. 마찬가지로, 문제의 기능 w 와 상태 s 간에 충분조건 ($w \rightarrow s, w$ 이면 s 이므로 w 는 s 의 충분조건), 필요조건 ($w \leftarrow s, w$ 는 s 의 필요조건), 필요충분조건 ($w \leftrightarrow s, w$ 는 s 의 필요충분조건) 중 하나의 조건 명제 관계를 갖는다. s 와 u 간의 조건 명제 관계도 마찬가지로 충분조건 ($s \rightarrow u$), 필요조건 ($s \leftarrow u$), 필요충분조건 ($s \leftrightarrow u$) 중 하나의 조건 명제 관계를 갖는다.

w 와 s 그리고 s 와 u 사이의 관계가 각각 3가지씩 존재하므로 모순 유형은 <표 1>의 좌측 열과 같이 모두 9개로 나눌 수 있다. 나비 모형에서는 모순 문제의 유형을 정의한 후 각 모순 문제유형별로 문제 해결 목표와 문제 해결 전략을 정의하였다. 문제를 구성하는 기능 $w, u, \sim u$ 가 어떤 관계를 갖는지 논리 연산자로 정의하였다. 문제 해결 전략은 문제 해결 목표를 충족하기 위한 구성요소의 상태와 기능 간의 관계로 증명하였다[13][14]. 9개의 모순 유형에 대해 유형별 문제 해결 목표와 문제 해결 전략은 <표 1>과 같다[12][13][14]. 본 연구에서는 기존 연구에서 정의한 문제 해결 전략을 추상적 해결전략과 구체적 해결전략이라 구분하였다. 또한, 구체적 해결전략을 어떤 기준에서 적용하면 타당한지 의미 분석을 통해 주어진 문제에 적합한 구체적 해결전략을 선택할 수 있도록 의사결정트리를 구축하고자 한다.

3. 분할과 결합 전략 알고리즘 설계

이 장에서는 분할과 결합 전략이 필요한 논리 연산자 \oplus 를 설명하고, \oplus 연산자를 문제 해결의 전략으로 갖는 모순 유형들(①, ③, ⑦, ⑨)의 구체적 문제 해결 전략을 찾는 의사결정트리 기반의 나비 알고리즘을 소개한다.

<표 1> 나비 모형에서 모순 문제 유형과 유형별 문제 해결 목표 및 문제 해결 전략

모순 문제 유형	문제 해결 목표	추상적 문제 해결 전략
① $(w \rightarrow s) \wedge (s \rightarrow u)$	$w \oplus \sim u$	$s \oplus \sim s$
② $(w \leftarrow s) \wedge (s \rightarrow u)$	$w \wedge \sim u$	$w \wedge \sim s$
③ $(w \leftrightarrow s) \wedge (s \rightarrow u)$	$w \oplus \sim u$	$s \oplus \sim s$
④ $(w \rightarrow s) \wedge (s \leftarrow u)$	$w \wedge \sim u$	$s \wedge \sim u$
⑤ $(w \leftarrow s) \wedge (s \leftarrow u)$	$w \wedge u$	$w \wedge s$
⑥ $(w \leftrightarrow s) \wedge (s \leftarrow u)$	$w \wedge \sim u$	$s \wedge \sim u$
⑦ $(w \rightarrow s) \wedge (s \leftrightarrow u)$	$w \oplus \sim u$	$s \oplus \sim s$
⑧ $(w \leftarrow s) \wedge (s \leftrightarrow u)$	$w \wedge \sim u$	$w \wedge \sim s$
⑨ $(w \leftrightarrow s) \wedge (s \leftrightarrow u)$	$w \oplus \sim u$	$s \oplus \sim s$

일반적으로 식(expression)은 연산자와 피연산자들의 집합으로 구성된다. 식은 값을 갖는다. 식에 논리 연산자가 사용되면 결과값은 참, 거짓 중 하나를 갖는다. <표 1>의 \oplus 는 논리 연산자 중에서 배타적 논리합(exclusive-or) 연산자의 개념을 바탕으로 정의하였다. 배타적 논리합 연산자는 두 피연산자 중에 하나만이 참이 되어야 연산식의 결과값이 참이 되는 연산자이다. 예를 들면, $A \oplus B$ 라는 식에서 A가 참이고 B가 거짓이거나 A가 거짓이고 B가 참인 경우와 같이 A와 B가 서로 배타적인 값을 갖는 경우에만 결과값이 참이 된다. $s \oplus \sim s$ 는 식의 정의에 따라 s 가 참이어야 하거나 $\sim s$ 가 참이어야 한다(<표 2> 참고). 따라서 상충관계인 w 와 $\sim u$ 를 해결하기 위한 매개모순 관계 $s \oplus \sim s$ 를 나타내기 위해서는 배타적 논리합 연산자인 \oplus 가 적합하다.

<표 2> $s \oplus \sim s$ 연산자

s	$\sim s$	$s \oplus \sim s$
참	거짓	참
거짓	참	참

하지만, 문제를 해결하기 위한 특정 상태에 부여되는 상반된 요구상황인 매개모순을 해결하기 위해서는 상태 s 가 참이어서 $\sim s$ 가 거짓인 상황과 상태 s 가 거짓이어서 $\sim s$ 가 참인 상황을 모두 만족

시켜야 한다. 이는 논리적으로 불가능하다. 따라서, 본 연구에서는 이 문제를 해결하기 위하여 분할-결합 원리를 적용하였다[14]. 분할-결합 원리는 트리즈의 분리 원리[5][6]를 근간으로 하였으나 트리즈에서는 정의하지 못한 모순을 해결하기 위해 추출과 양자택일이라는 모순 해결방법을 제시하였다[14]. <표 3>에서는 분할-결합의 기준으로 시간과 구성요소를 설정하여 매개모순을 해결하기 위해 시간만 분할-결합(①), 구성요소만 분할-결합(②), 시간과 구성요소를 모두 분할-결합(③)하는 방법을 제시하였다[14].

<표 3> 나비 모형에서 연산자 \oplus 를 충족하기 위한 분할-결합 조건

구성요소	①	
시간	C_1	C_2
T_1	s	$\sim s$
T_2	$\sim s$	$\sim s$

<표 3>에서 ①은 시간 분할-결합 방법을 의미한다. T_1 에서는 s 가 참이고 T_2 에서는 $\sim s$ 가 참이다. ②는 같은 시간대에 구성요소 C_1 에 대해서는 s 가 참이고 C_2 에 대해서는 $\sim s$ 가 참이 된다. ③은 시간과 구성요소를 모두 분할하여 T_1 과 C_1 에 대해서는 s 가 참이고 T_2 와 C_2 에 대해서는 $\sim s$ 가 참이 된다.

<표 4>에는 추상적 문제 해결 전략, $s \oplus \sim s$ 에 대한 구체적 문제 해결 전략이 기술되어 있다. 트리즈의 이론과 다르게 본 연구에서는 공간 분리(separation principle) 개념을 좀 더 구체적으로 세분화하여 시스템 구성요소의 특징을 기반으로 ① 단일 시스템을 분해하는 구성요소 분할과 결합, ② 다중시스템을 이루는 구성요소 분할과 결합 원리를 제시하였다. 또한, 모순을 일으키는 유해한 구성요소에 대해 선추출(pre-extraction)과 후추출(post-extraction)의 원리를 추가하였다[12][14]. 또한, s 와 $\sim s$ 중에 어느 하나만을 만족시켜야 하는 경우에는 양자택일의 원리를 추가하였다.

<표 4> 나비 모형에서 추상적 문제 해결 전략인 $s \oplus \sim s$ 에 대한 구체적 문제 해결 전략

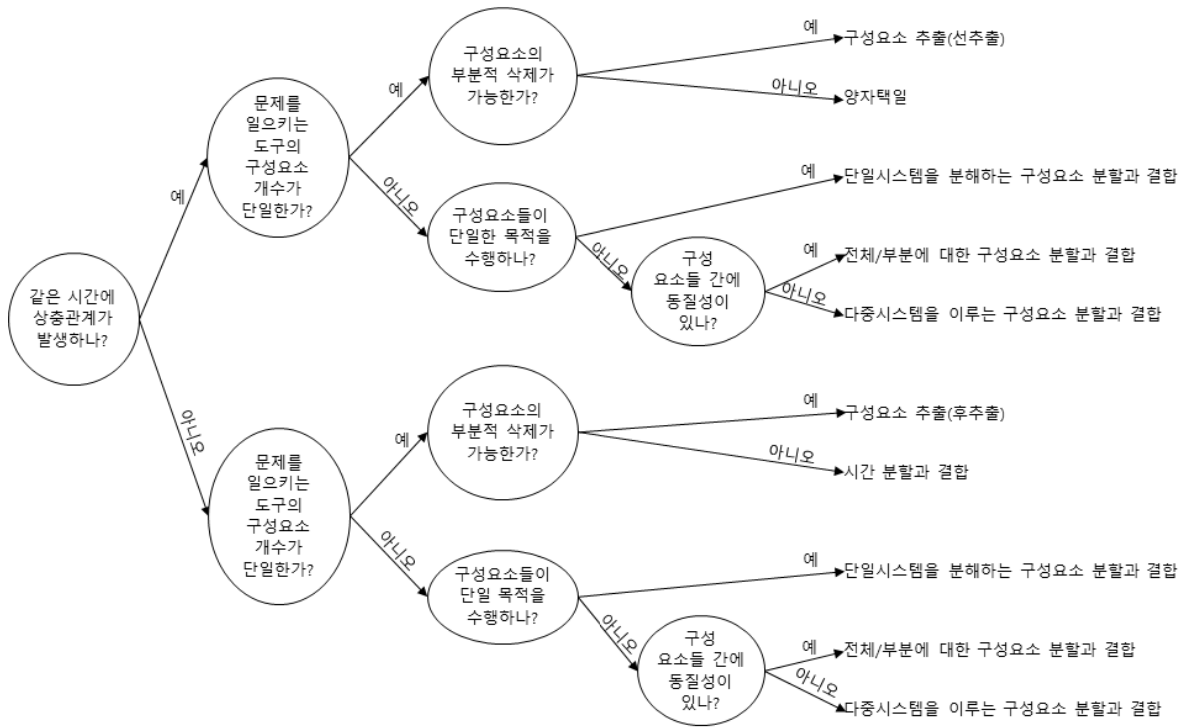
추상적 문제 해결 전략	$s \oplus \sim s$	
구체적 문제 해결 전략	분할-결합 원리를 적용한 배타적 논리합	① 시간 분할과 결합 ② 구성요소 추출 ③ 단일 시스템을 분해하는 구성요소 분할과 결합 ④ 전체/부분에 대한 분할과 결합 ⑤ 다중시스템을 이루는 구성요소 분할과 결합
	배타적 논리합	⑥ 양자택일

본 연구에서는 기존 나비 모형 이론에서는 제안하지 못한 구체적 문제 해결 전략을 적용하는 방법을 개발하였다. 이를 위하여 [그림 2]와 같이 의사결정트리 기반 나비 알고리즘을 설계하였다.

- ① 문제의 상충 관계를 해결하기 위하여 매개모순을 이루는 두 상태인 s 와 $\sim s$ 를 서로 다른 시간 값으로 나누어 해결할 수 있는지 제일 먼저 확인한다. 예를 들면, 비행기 바퀴의 모순과 같이 이/착륙 시에는 비행기 바퀴가 있어야 한다. 비행 시에는 비행기 바퀴가 없어야 한다. 동시에 비행기 바퀴가 있으면서 (s) 없을 수($\sim s$) 없지만, 시간을 분할하여 결합하면 두 상태를 모두 만족시킬 수 있다.
- ② 문제의 상충 관계를 해결하기 위하여 문제를 일으키는 유해한 부분을 제거(추출)할 수 있는지 확인한다. 예를 들면, 보안 나사못과 같이 열지 못하도록 못을 고정하기 위해서 나사못의 머리에 나사못의 홈이 있어야 한다. 하지만, 홈이 있다면 다시 나사못이 풀려 열릴 수가 있다. 나사못을 열지 못하도록 잠금 방향으로의 나사못 홈은 그대로 두고(s) 열림 방향으로의 나사못 홈($\sim s$)을 선추출하면 나사못 홈의 모순 문제를 해결할 수 있다 [15].
- ③ 문제의 상충 관계를 해결하기 위하여 문제를

해결하기 위한 한 개의 도구를 여러 개로 분할할 수 있는지 확인한다. 예를 들면, 정보보안을 위한 키(key)는 본래 남에게 노출이 되면 안 되도록 암호화 되어야 한다(s). 하지만, 최근의 전자상거래와 같이 익명의 사람들과 안전한 거래를 하려면 키는 알려져야 한다($\sim s$). 이와 같은 키 배분의 모순 문제를 해결하고자 Diffie and Hellman은 키를 비밀키(private key)와 공개키(public key)의 두 개념으로 분할하였다. 송신자가 전송할 대상인 수신자의 공개키로 메시지를 암호화하여 전송하면 수신자는 자신의 비밀키로 메시지를 복호화한다[16]. 키 배분의 문제는 단일 시스템이었던 한 개의 키를 비밀키와 공개키로 나누어 두 가지 기능을 하도록 분할-결합하였다[13].

- ④ 문제의 상충 관계를 해결하기 위하여 구성요소의 분할이 가능한 경우, 분할되는 구성요소의 성질이 동질적인지 이질적인지를 확인한다. 예를 들면, 방사선의 문제와 같이 악성종양에는 강한 방사선 빛(s)을 쏘여야 하고 건강한 일반 세포에는 약한 방사선 빛($\sim s$)을 쏘여야 한다. 이를 위해 빛의 성질을 활용하여 360도 각도에서 각각의 일반 세포에는 약한 빛을 쏘이게 하고, 빛들이 전부 악성종양으로 모이게 한다면, 악성종양에는 강한 방사선 빛이 쏘여지게 된다.
- ⑤ 문제의 상충 관계를 해결하기 위하여 다중 목적을 수행하는 경우, 문제를 해결하기 위하여 여러 개인 구성요소가 동질적 성질을 갖는지 아닌지를 확인한다. 예를 들면, 이중초점 렌즈와 같이 가까운 곳을 보기 위해서는 원시용 렌즈, 먼 곳을 잘 보기 위해서는 근시용 렌즈가 필요하다. 이때, 각각의 렌즈는 반대 기능을 수행할 수 없다. 기능 간의 모순을 해결하기 위해 두 시스템을 결합한다[17].
- ⑥ 문제의 상충 관계를 해결하기 위하여 s 가 참이고 $\sim s$ 가 거짓이 되는 상태를 선택하거나 s 가 거짓이고 $\sim s$ 가 참이 되는 상태를 선택한다.



[그림 2] 의사결정트리 기반 나비 알고리즘

본 연구와 기존 연구의 차이를 요약한다면, 기존 연구에서는 추상적 해결전략 $s \oplus \sim s$ 에 대한 구체적 해결전략으로 6가지를 존재함을 밝힌 것이라면, 본 연구에서는 주어진 어떤 모순 문제에 대해 문제의 특성과 조건에 적합한 구체적 해결전략이 정확히 무엇인지 제시함으로써 시행착오를 거치지 않고 해결공간을 줄여 줄 방법을 제시하고 있다.

4. 구현 및 적용

4.1 시각화 도구 구현

본 연구에서는 파이썬 언어의 기본 GUI를 지원하는 tkinter 모듈을 이용하여 모순을 해결하는 모순 트리 알고리즘을 구현하였다. 알고리즘에 대한 단계별 과정을 [그림 3]에서 '연필+지우개'의 발명 과정을 이용하여 설명하였다. 첫째, [그림 3]의 (가)에서는 사용자들에게 w, s, u 를 정의하고, [그림 3]의 (나)에서 사용자들이 $\sim w, \sim s, \sim u$ 가 올바른지 확인하게 한다. [그림 3]의 (다)에서 모순 유형을 찾게 한 후, [그림 3]의 (라)에서 문제에 맞는 문제 해결 전략을 찾는다.

[그림 3]의 (라)에는 [그림 2]의 단계별 사용자 선택이 결정되면 해당하는 다음 단계의 질문 문항을 제시하는 과정이 설명되어 있다. 즉 의사결정트리의 루트 노드에서부터 질의하면서 사용자들에게 의사결정을 할 수 있도록 문제 해결의 단서를 지속적으로 제시한다. 결과적으로 사용자들의 선택을 통해 최종 구체적 문제 해결 전략에 도달하도록 돕는다.

연필과 지우개의 문제에서는 두 도구가 한 시간대에 함께 사용되는 것이 아니며, 두 개의 기능을 수행하기 때문에 두 개의 구성요소가 존재한다고 할 수 있다. 또한, 두 개는 각각의 목적을 가지고 기능을 수행하며 이질적이다. 따라서 본 알고리즘에 의해 '다중시스템 결합을 통해 시간에 따라서 다른 기능을 수행하는 문제 해결 방법'이 구체적 해결전략이 된다.

사용자는 나비 모형의 원하는 기능(w), 시스템 구성요소의 상태(s), 원하지 않는 기능(u)과 $\sim u, \sim s$ 를 차례로 입력한다. 사용자가 같은 시간에 시스템 구성요소인 s 와 $\sim s$ 간에 매개모순 관계가 발생하는지 확인한다.

[그림 4]에는 사용자 입력을 바탕으로 한 나비 다이어그램이 제시되어 있다. 또한, 시간과 구성요

소별 필요한 기능과 상태를 정의하여 문제 해결자들에게 구체적 문제 해결 전략을 제시하고 있다.

나비모형

w, s, u의 입력

원하는 기능 또는 목적 (w):

원하는 기능으로 인한 상태 또는 원하는 기능을 위한 수단 (s):

원하지 않는 기능 또는 결과 (u):

사용자 입력

(가) w, s, u의 입력

~u와 ~s의 확인

w와 상충관계를 갖는 기능 또는 목적 (~w)로 수정:

s와 매개모순 관계를 갖는 상태 또는 수단 (~s)으로 수정:

사용자 추가 입력

(나) s와 u에 대한 입력을 바탕으로 한 ~u와 ~s의 제시 및 사용자 수정

w, s, u 간의 필요충분조건 입력

글자를 쓴다. (w)와 연필로 쓴다. (s)의 관계는? 충분조건 필요조건 필요충분조건

연필로 쓴다. (s)와 쓰면 지울 수 없다. (u)의 관계는? 충분조건 필요조건 필요충분조건

(다) w, s, u간의 조건 규명

글자를 쓴다. (w)와 글자를 지운다. (~w)가 같은 시간대에 상충관계가 발생하는가? 예(같은 시간대) 아니오(다른 시간대)

문제를 일으키는 도구의 구성요소의 개수가 단일한가? 예(1개) 아니오(2개 이상)

구성요소들이 단일한 목적을 수행하는가? 예(단일) 아니오(다중)

글자를 쓴다. (w)와 글자를 지운다. (~w)가 구성요소들 간에 동질성이 있나? 동질성 이질성

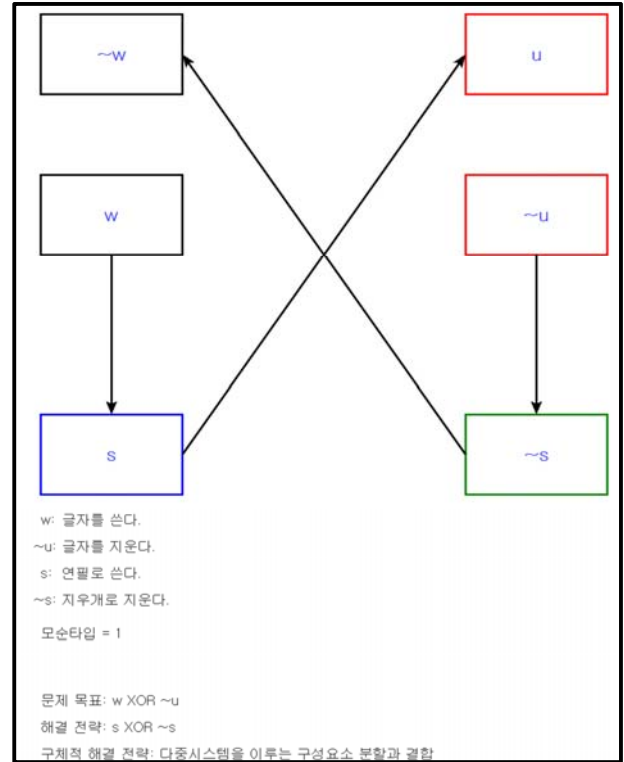
(라) 모순 트리 노드에 대한 질의와 응답

[그림 3] 모순 문제 해결을 위한 질의와 의사결정

4.2 제안한 알고리즘의 실생활 문제 적용

본 연구에서는 개발한 알고리즘의 타당성을 검증해보고자 본 연구에 참여한 학생들에게 새로운 모순 문제를 제시하여 개발한 프로그램에서 구체적

해결안을 찾은 후, 스스로 해결안을 찾아보도록 하였다.



(가) 나비 다이어그램

	시간1	시간2
기능	글자를 쓴다.	글자를 지운다.
상태	연필로 쓴다.	지우개로 지운다.

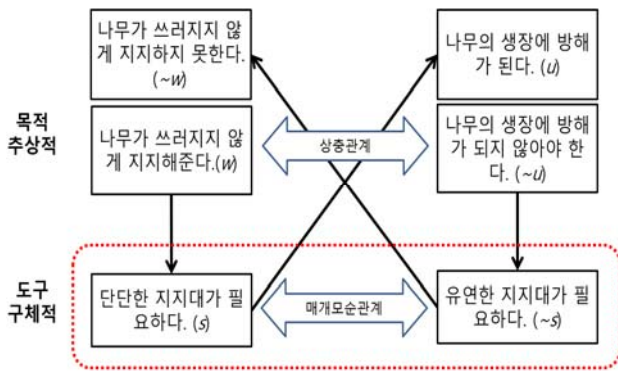
	구성요소1	구성요소2
기능	글자를 쓴다.	글자를 지운다.
상태	연필로 쓴다.	지우개로 지운다.

(나) 시간과 구성요소에 따른 기능과 상태

[그림 4] 나비 다이어그램과 구체적 문제 해결 전략의 자동 생성

기존 연구인 [18][19][20]에서 제시한 나비 다이어그램 교육 효과와는 다르게 본 연구에서 개발한 프로그램이 자동 생성한 해결전략의 방향성에 대한 적합성 여부와 학생들이 제안한 해결안이 질적으로 우수한지 검증해보았다.

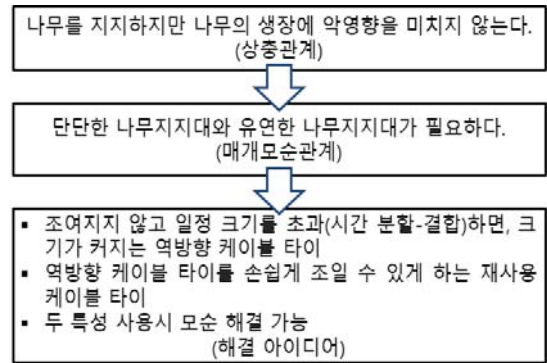
[그림 5]에서와 같이 학생들에게 어린나무가 바람에 쓰러지지 않게 하기 위한 나무지지대를 해결하도록 하였다. 나무지지대 문제는 나무가 쓰러지지 않게 지지하려면 단단한 지지대가 필요하지만, 단단한 지지대는 나무의 생장에 방해가 된다. 그러므로 나무의 생장에 방해가 되지 않는 지지대가 되려면 유연해야 한다. 결국, 나무지지대는 단단하면서도 유연해야 하는 모순 문제를 갖는다. 나무지지대 문제의 모순 정의는 [그림 5]와 같고 나무지지대 문제는 시간에 따라서 나무지지대가 단단하고 유연해야 하는 1번 유형의 모순 문제임을 알 수 있다.



[그림 5] 나무지지대의 모순

중학생들은 모순 해결을 위한 의사결정 나비 알고리즘을 활용하여 나무지지대와 철사라는 두 문제 구성요소가 '지지하고(w) 생장에 방해가 안 되는(~w)' 두 기능을 수행하기 때문에 (1) 같은 시간대가 아닌 문제이고, (2) 철사라는 단일한 구성요소가 문제를 일으키고, (3) 지지하기 위해 철사의 기능을 없앨 수 없으므로 '시간 분할'이라는 구체적 해결전략을 도출하였다. 즉, 나무 생장에 방해가 될 정도로 조여지는 문제를 해결하기 위하여 나무가 성장하게 되면 지지대에서 철사의 역할을 수행하는 부분을 안으로는 조여지지 않고 밖으로 넓어지게 만드는 역 케이블 타이의 방법을 도출하였다(그림 6) 참조).

본 연구팀에서는 제안한 아이디어와 해결 과정을 정리하여서 한 광역지방자치단체에서 실시한 발명대회에 출품하여 대상을 받아 아이디어의 질을 입증할 수 있었다.



[그림 6] 문제의 상충관계와 매개모순 관계 정의 및 실제 해결아이디어 생성

4.3 기존 연구와의 차별성 및 학생들이 기여한 부분

이 절에서는 기존의 연구와 본 연구와의 차별성을 기술하고 본 연구에 참여한 학생들이 기여점을 진술한다. 연구 [12]와는 다르게 본 연구에서는 주어진 모순문제 해결을 위하여 기존 연구에서 제시한 구체적인 해결전략 중에서 정확히 어떤 전략을 선택해야 하는지 탐색 과정을 알고리즘으로 설계하였다. 또한, 알고리즘을 구현함으로써 모순 해결 방법론을 모르는 일반 문제 해결자들도 시각화 도구로 3~4회의 의사결정으로 정확한 해결전략을 얻게 되어 시행착오를 줄일 수 있게 되었다. <표 5>에서 기존 연구와 본 연구의 차별성을 기술하였다.

<표 5> 연구의 연도 순서에 의한 나비 기존 연구와 본 연구의 차별성

기존 문헌의 제목(계제연도)	연구의 차별성
The Butterfly Model for Supporting Creative Problem Solving (2012년) [8]	창의성 교육 방법에서 채택하고 있는 '발산 후 아이디어의 수렴'이라는 접근방법 대신, 초기 나비 다이어그램을 제안하여 '수렴 후 아이디어의 발산'이라는 역의 접근방법을 처음 제안한 논문임. 본 연구와 다르게 명제논리학에 근간을 두고 있지 않음.
모순 해결 나비 모형의 알고리즘과	본 연구와 다르게 무방향성 그래프인 나비 다이어그램을 작성하는 방법에 대한 교육 효과 검증을 수행한 연구. 풀기 어려운 Dunker의 방사선

<p>교육 효과 (2012년) [18]</p>	<p>문제에 대한 나비 모형 교육생들의 문제해결력을 기존 연구와 비교하여 나비 모형 교육의 긍정적 효과성을 입증한 논문임. 본 연구와 다르게 명제논리학에 근간을 두고 있지 않음.</p>
<p>분할-결합 원리와 상태모형에 대한 학습이 모순 문제 해결과 성장 마인드셋에 미치는 영향 (2013년) [19]</p>	<p>복잡하고 어려운 트리츠 이론을 초중등학생이 쉽게 학습할 수 있도록 나비 다이어그램과 함께 트리츠의 분리 원리에 초점을 두고 원리를 설명하여 학생들의 문제해결력 증진과 긍정적 마인드셋 함양에 영향을 미쳤음을 밝힌 논문임. 본 연구와 다르게 명제논리학에 근간을 두고 있지 않음.</p>
<p>Logical Interpretation about Problem Types and Solution Strategies of the Butterfly Model for the Automation of Contradiction-based Problem Solving (2014년) [9]</p>	<p>명제논리학을 근간으로 모순의 유형을 모두 나열하여 이론이 정확함을 증명한 논문임. 본 논문의 근간이 되는 논문이기도 함. 이전 논문과의 차별점은 트리츠의 개선 관점이라기보다 나비 모형의 이론은 시행착오를 없애고 논리적으로 문제 정의를 하여 수렴 후 발산이라는 아이디어를 구체화하기 위한 연구의 시작점이었음. 또한, 이전까지의 연구에서는 기술적 모순 관계를 이루는 기능, w와 u, 상태 s로만 나비 모형을 소개한 것과 다르게 부정 연산자인 ~을 포함한 6개의 구성요소로 새롭게 나비 모형과 나비 다이어그램을 재정의함.</p>
<p>트리츠의 물리적 모순에 대한 모순 해결 나비 모형의 모순 관계와 해결차원 분류 (2014년) [10]</p>	<p>트리츠의 물리적 모순을 나비 다이어그램으로 정의하는 과정을 설명함. 또한, 트리츠가 물리적 모순 해결방법을 단순히 나열한 한계를 극복하고자 모순 문제유형별로 물리적 모순 문제 해결방법을 분류하였음.</p>
<p>창의 혁신을 위한 모순 해결 방법론과 사례 (2018년) [13]</p>	<p>나비 다이어그램에서 명제논리학의 대우 개념을 처음으로 소개하고 지금까지는 무방향 그래프였던 나비 다이어그램을 방향 그래프로 전환한 논문임. 트리츠에서는 다루지 못하는 모순 유형인 혼합, 전화위복, 양자택일,</p>

	<p>모순 심화, 제3자원의 도입과 같은 새로운 모순 유형과 모순 해결방법을 사례 중심으로 기술한 논문임.</p>
<p>모순 해결을 위한 나비 모형의 분리-결합 기반 구체적 해결전략 도구 (2018년) [12]</p>	<p>9개의 모순 유형 중에서 분리-결합 원리에 기반을 두어 문제 해결 전략을 갖는 모순 문제들에 대하여 연구 [10]에서 찾지 못한 새로운 구체적인 해결전략을 제시한 논문임. 본 연구와 다르게 언제, 어떻게 구체적인 해결전략을 적용하면 좋은지는 제시하지 못함.</p>

초중등학생들의 나비 모형에 대한 교육적 효과를 검증하려 했던 연구 [18]과 연구 [19]와는 다르게 본 연구에서는 본 연구에 참여한 중학생 3명이 나비 모형의 이론을 32시간 동안 학습한 후, 나비 모형을 모르는 문제 해결자도 순차적인 질문을 통해 모순 문제를 해결할 수 있도록 알고리즘을 교수자와 함께 설계 및 구현해 보았다. 또한, 개발한 알고리즘의 정확성을 검증해보기 위해 학생들에게 나무지대 문제를 제시하여 정해진 순서에 따라 구체적 해결전략을 찾아내도록 하였다. 그 결과, 학생들은 쉽게 역 케이블 타이라는 발명 아이디어를 제시하게 되었고 발명대회에서 대상이라는 평가 결과를 얻었다.

본 연구에서는 9가지 모순 유형 중에서 4가지 유형으로 연구 범위를 한정하였다. 중학교 영재 학생들과 함께 사사 과정 중에 연구한 논문이어서 모든 모순 유형에 대한 구체적 해결전략을 구분하고 구현하는데 시간적 제약이 있었다. 추후 나머지 문제 유형(<표 1>의 ②, ④, ⑤, ⑥, ⑧)에 대한 의미적 고찰과 해결책 고안이 진행될 예정이다.

5. 결론

진퇴양난의 딜레마는 문제 안에 모순을 갖는 경우가 많다. 즉, 어느 한쪽을 만족시키려 하면 다른 한쪽이 나빠지는 상충관계는 무엇(something)이면서 무엇이 아닌(not something) 모순 관계를 해결해야만 풀리는 경우가 있다. 모순에 대한 연구자들의 일반적인 생각은 모순을 전혀 가능성이 없는

공집합이나 논리적으로 틀린 것이다. 하지만 자연과학의 창의적 연구자들과 혁신적인 발명가들은 문제에 숨어 있는 모순을 해결한 경우가 많다. 예를 들어, 제임스 와트(James Watt)는 증기기관의 실린더가 뜨거워야 하고 차가워야 하는 문제를 해결하였다. 루이 파스퇴르(Louis Pasteur)는 공기는 통과하지만 공기 중의 미생물은 통과하지 못하도록 백조 목 플라스크를 만들었다. 프레드릭 바우어(Fredric Baur)는 감자칩이 맛있도록 얇은 두께의 감자칩을 포장지에 넣으면 감자칩이 쉽게 부스러지는 문제를 해결하였다[21]. 로렌스 룰렌(Lawrence Luellen)은 깨지지 않는 컵을 만들기 위해 물에 젖지 않는 종이컵을 발명하였다[22].

본 연구는 풀기 어려운 모순 문제에 초점을 두고 모순 문제를 나비 다이어그램으로 정의하였다. 모순 유형을 파악하고 유형별 해결전략을 체계적으로 제시하여 문제 해결자들이 시행착오를 겪지 않고 문제를 해결하도록 시각화하였다. 또한, 모순 유형 중에서 분할-결합에 근간을 둔 모순 문제에 대한 구체적 해결전략을 제시하였고, 이를 컴퓨터에서 동작할 수 있는 의사결정트리 기반의 알고리즘을 설계 및 구현하였다. 본 연구가 제시하는 알고리즘의 효과를 검증하기 위하여 의사결정트리 기반의 알고리즘을 학습한 중학생들이 발명 대회에 참가하였다. 학생들이 제시한 문제 해결과정과 문제 해결 아이디어는 가장 높은 평가를 받아 대상을 받았다.

지금까지 창의적 문제 해결에 관한 거의 모든 연구는 문제 해결 과정을 발산적 사고를 먼저 하고 수렴적 사고를 나중 단계에 하는 과정을 밟는다. 즉, 브레인스토밍처럼 많은 아이디어를 낸 다음에 논리적 평가를 통하여 좋은 대안을 추리는 방식이다[23][24]. 본 연구의 의사결정트리 기반 나비 알고리즘은 문제 해결자에게 먼저 논리적인 수렴적 사고를 통하여 올바른 문제 해결 전략을 분석하도록 돕는다. 문제유형별로 올바른 해결전략이 정해지고 나면 발산적 사고를 통하여 창의적 문제 해결 과정을 밟아 구체적인 해결안을 다양하게 내놓게 된다. 의사결정트리 기반 나비 알고리즘의 적용으로 문제 해결 공간이 줄어들어 따라 효율적인 문제 해결이 가능해진다.

컴퍼스 침 문제를 예로 들어 설명하면 다음과

같다. 종이에 원을 그리려면 컴퍼스 침이 있어야 하지만 날카로운 컴퍼스 침 때문에 안전사고 위험이 큰 경우가 있다. 만약에 문제 해결자가 컴퍼스 침의 문제를 컴퍼스를 사용할 때는 컴퍼스 침이 있다가 컴퍼스를 사용하지 않을 때는 컴퍼스 침을 숨기는 것으로 정의할 수 있다. 이를 이행하는 방법은 (1) 컴퍼스 침을 볼펜 심처럼 눌러 컴퍼스 침을 나왔다가 들어가게 하는 방법(한국실용신안, 20-0300111), (2) 컴퍼스의 발을 접으면서 발 안으로 컴퍼스 침을 숨기는 방법(한국특허, 10-2007-0053192), (3) 스프링이 달린 레버를 밀면 컴퍼스 발 안으로 침이 들어가는 방법(한국특허, 10-0952819), (4) 손으로 덮개를 밀어 컴퍼스 침을 덮는 방법(한국실용신안, 20-0443527) 등으로 문제가 해결된다. 수렴적 사고를 먼저 하고 나중에 발산적 사고를 하는 것은 시행착오를 줄이면서도 문제 해결을 할 수 있는 다양한 아이디어를 내도록 돕는다.

본 연구에서 제시한 모형과 알고리즘은 중학생들이 이해하고 문제 해결에 적용하기에 용이하다. 많은 시간을 할애하여 학습해야 활용할 수 있는 트리 이론과는 달리 나비 모형과 나비 알고리즘은 나비 다이어그램이라는 시각화 도구를 사용함으로써 어린 학생들도 쉽게 모순을 정의하고 문제 해결 전략에 도달하는데 많은 시간을 요구하지 않는다.

알트슐러(Altshuller)는 아리즈 85-c 알고리즘 [7] 서문에서 아리즈는 복잡하기 때문에 아리즈를 적용하려면 먼저 트리즈를 80시간 이상 학습할 것을 당부했다. 아리즈 알고리즘의 문제 해결 과정은 논리적 기호 대신 말과 그림으로써 기술하는 과정을 밟도록 한다[14]. 이와 달리, 본 연구의 의사결정트리 기반 나비 알고리즘은 기호 논리학을 이용한 증명을 활용해 모순 문제유형별로 적용해야 할 해결전략을 올바르게 정하도록 돕는다.

참 고 문 헌

- [1] Horowitz, E., Sahni, S., and Rajasekaran, S. (1998). *Computer Algorithms*. New York, Computer Science Press.
- [2] Simonton, Dean K. (2004). *Creativity in Science*. Cambridge University Press.

- [3] Simon, Herbert A. (1987). *The Sciences of the Artificial*. The MIT Press.
- [4] Polya, G. (1981). *Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving*. John Wiley & Sons, Inc.
- [5] Altshuller, G. S. (1984). *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems*. CRC Press.
- [6] Hipple, J. (1999). The Use of TRIZ Separation Principles to Resolve the Contradictions of Innovation Practices in Organizations. *TRIZ Journal*. 4. (URL: <http://www.triz-journal.com>).
- [7] Salamatov, Y. (1999). *TRIZ: The Right Solution at the Right Time: a Guide to Innovative Problem Solving*. Hattem: Insytec.
- [8] Hyun, J. S. and Park, C. J. (2012). The Butterfly Model for Supporting Creative Problem Solving. *Knowledge, Information and Creativity Support Systems(KICSS), 2012 Seventh International Conference on IEEE*. 28-34.
- [9] Hyun, J. S. and Park, C. J. (2014). Logical Interpretation about Problem Types and Solution Strategies of the Butterfly Model for the Automation of Contradiction-based Problem Solving. *IEEE International Conference on Teaching Assessment, and Learning for Engineering 2014*. 1, 1-6.
- [10] 현정석·박찬정 (2014). 트리즈의 물리적 모순에 대한 모순 해결 나비 모형의 모순관계와 해결차원 분류. *지식경영연구*. 15(4), 15-34.
- [11] Hyun, J. S. and Park, C. J. (2016). The Butterfly Algorithm: A Contradiction Solving Algorithm based on Propositional Logic for TRIZ. *International Journal of Software Engineering and Its Applications*. 10(1), 27-34.
- [12] 현정석·박찬정·이경은 (2018). 모순 해결을 위한 나비 모형의 분리-결합 기반 구체적 해결전략 도구. *Convergence Research Letter*. 4(1), 141-144.
- [13] 현정석·박찬정 (2018). 창의혁신을 위한 모순 해결 방법론과 사례. *Korea Business Review*. 22(2), 99-126.
- [14] 현정석 (2018). **창의적 모순 해결의 원리와 실제**. 도서출판 청람.
- [15] Simone, J. V. and Olmstead, C. H. (1977). *Security Screw. US Patent 4171662*.
- [16] Diffie, W. and Hellman, M. E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*. 22(6), 644-654.
- [17] Suh, N. P. (2001). *Axiomatic Design: Advances and Applications*. Oxford University Press.
- [18] 현정석 (2012). 모순 해결 나비 모형의 알고리즘과 교육효과. *Korea Business Review*. 16(3), 101-132.
- [19] 현정석·박찬정 (2013). 분할-결합 원리와 상태모형에 대한 학습이 모순 문제 해결과 성장 마인드셋에 미치는 영향. *지식경영연구*. 14(4), 19-46.
- [20] Hyun, J. S. and Park, C. J. (2015). Effects of the Butterfly Diagram Education in Primary and Secondary Schools on Contradiction Problem Solving Ability. *IEEE International Conference on Frontiers in Education Conference(FIE)*. 1-8.
- [21] Baur, F. J. 1966). *Packaging of Chip-Type Snack Food Products*. US Patent 3498798.
- [22] Luellen, L. W. (1911). *Cup*. US Patent 1284728.
- [23] 김영채 (2013). **사고력 교육: 이론과 실제**. 유원북스.
- [24] Runco, Mark A. and Acar, S. (2012). Divergent Thinking as an Indicator of Creative Potential. *Creativity Research Journal*. 24(1), 66-75.



현 정 석

1991 서강대학교
경영학과(경영학사)
1993 서강대학교 대학원
경영학과 (경영학석사)

1998 서강대학교 대학원 경영학과(경영학박사)
2002 ~ 현재 제주대학교 경영정보학과 교수
2019 제주대학교 연구업적 우수교수상 수상
2008 제주대학교 대학을 빛낸 교수상 수상
2012 특허청장상 수상
관심분야: 마케팅, 행동의사결정론, 트리즈, 창의성
교육, 영재교육
E-Mail: jshyun@jejunu.ac.kr



박 찬 정

1988 서강대학교
전자계산학과(공학사)
1990 한국과학기술원
전산학과(공학석사)

1998 서강대학교 대학원 전자계산학과(공학박사)
1990~1994 한국통신 소프트웨어연구소 전임연구원
1998~1999 한국통신 멀티미디어연구소 전임연구원
1999~현재 제주대학교 컴퓨터교육과 교수, 제주
대학교 교육과학연구소 연구원
2010 ~ 2011 미국 UC Berkeley 방문학자
2017 미국 Stockton University 방문학자
관심분야: 추상적 사고력, 문제해결 알고리즘, 창의
인성 교육, 영재교육
E-Mail: cjpark@jejunu.ac.kr



고 예 준

2016 ~ 현재 노형중학교
2018 제주대학교 과학영재교육원
관심분야: 기계학습, 컴퓨터, 정보,
IT

E-Mail: koyejune0302@naver.com



김 용 결

2016 ~ 현재 탐라중학교
2018 제주대학교 과학영재교육원
관심분야: 컴퓨터, IT융합

E-Mail: yung7999@naver.com



전 승 재

2016 ~ 현재 제주제일중학교
2018 제주대학교 과학영재교육원
관심분야: 머신러닝, IT, 컴퓨터

E-Mail: wo110112@naver.com