

# 사이버 문제은행에서 시뮬레이티드 어닐링을 이용한 2단계 문제세트 생성 휴리스틱\*

황인수\*\*

## Two Phase Heuristic for Test Set Generation Using Simulated Annealing in Cyber Testbank System\*

Insoo Hwang\*\*

### ■ Abstract ■

The widespread diffusion of Internet has enables every college and education institute to develop cyber education systems to meet the multiple needs of students, but it is not true that the effectiveness of cyber education is fruitful in terms of evaluation systems. Most of the early developed web-based evaluation systems for cyber education require that all the students should solve uniformed test set which are included in the predetermined static HTML pages. Therefore, it is impossible to dynamically provide a test set with consistency and reliability.

This paper purpose to describe the employment of simulated annealing in cyber testbank system for test set generation that satisfy all constraints. The constraints include number of items for each skill, method, domain, topic, and so on. This research developed two phase heuristic combining sequential test set generation algorithm with simulated annealing. As a result of computer simulations, it was found that the two phase heuristic outperforms the other algorithms.

## 1. 서론

인터넷 및 정보기술의 발달은 시간과 공간의 제

약을 극복하여 언제 어디서나 원격으로 학습할 수 있는 새로운 교육체제인 사이버 교육을 현실화시키고 있으며, 월드 와이드 웹(World Wide Web,

\* 본 연구는 전주대학교 학술연구비 지원으로 이루어졌음.

\*\* 전주대학교 정보기술학부 조교수

WWW)은 개발 및 접근의 용이성과 효율성 등을 바탕으로 교육의 정보화를 통한 열린교육을 가능케 하고 있다.

교육의 정보화의 일환으로 개발되어 왔던 대부분의 전통적인 교육용 시스템들은 텍스트나 간단한 그래픽 기반으로 설계·개발되었으며, 단방향 혹은 일방적인 학습과 평가를 수행하였다. 그러나 WWW는 양방향의 상호학습을 실시간으로 제공해주기 때문에 학습자의 수준에 맞는 다양한 교육 콘텐츠의 제공 및 학습능력의 평가가 가능하게 되었다(정성길 외, 1997).

이에 따라, 대학 및 각종 교육기관에서는 교육자가 강의실에서 교재를 중심으로 지식을 전달하는 전통적인 교육방법에서 탈피하여 CD-ROM 타이틀이나 인터넷 등 각종 멀티미디어 매체를 이용함으로써 교육효과를 극대화시키고 있다. 최근에는 면대면 학습을 수행하던 정규교육기관 외에 전산설비와 인터넷 시설을 이용한 사이버 대학, 사이버 학원 등 다양한 형태의 교육기관이 e-Business의 사이버 교육시장에 등장하고 있다. 일찍이 Drucker(1993)는 앞으로 전통적인 의미의 교육기관은 존재하지 않을 것이며, 원격교육(Distance Education)이 이를 대체할 것이라고 예측한 바 있다. 전통적인 의미의 대학 등 교육기관이 완전히 사라지리라고 생각하는 사람은 거의 없지만, 정보통신기술이 교수의 교육방법, 학습자의 학습방법, 그리고 교육장소 및 시간 등을 근본적으로 변화시킬 것이라는 데는 의심의 여지가 없다(전성빈, 1998).

최근에는 학습능력을 평가하여 자격을 부여하는 점성시험도 인터넷상에서 원격으로 실시되고 있는데, 평가방법의 대부분이 HTML로 구성된 정형화된 문제세트를 제공하거나 난수를 이용하여 작성한 임의의 평가문항들을 제공하여 평가하고 있다. 이와 같은 문제세트는 평가대상자 혹은 시험횟수마다 일관성이 없어서 문제유형이나 난이도가 달라질 수 있기 때문에 TOEFL이나 TOEIC 등과 같이 공인된 평가시험으로서의 신뢰성을 얻지 못하고 있다.

이에 따라, 본 연구에서는 문제은행시스템, 특히 인터넷 등 사이버 공간에서의 능력검정을 목적으로 하는 문제은행으로부터 문제영역, 문제유형, 난이도, 소재 등 다양한 세부영역별로 사전에 설정한 문제유형과 난이도의 문제세트를 생성함으로써 평가시험의 일관성을 유지하여 신뢰도를 향상시키는 방안에 대해 연구하였다. 본 논문은 문제은행으로부터 주어진 제약조건을 모두 만족하는 문제세트를 항상 추출할 수 있을 뿐만 아니라, 보다 효율적으로 생성할 수 있도록 시뮬레이티드 어닐링 기법을 적용하는 방안에 대해 기술하였다. 또한, 최적해의 탐색시간을 최소화하기 위한 방안의 하나로, 전통적으로 많이 사용되어 온 순차적인 문제추출 알고리즘과 시뮬레이티드 어닐링을 결합한 2단계 문제추출 휴리스틱의 성과도 함께 기술하였다.

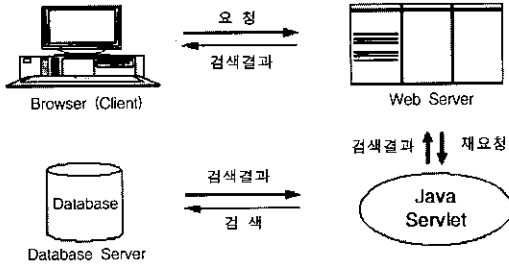
본 논문의 구성은 다음과 같다. 먼저, 2장에서는 시스템의 구성 및 동작원리를 기술하며, 3장에서는 시뮬레이티드 어닐링 기법을 간략히 소개한다. 4장에서는 전통적인 순차 추출알고리즘의 문제점을 기술하고, 5장에서는 시뮬레이티드 어닐링 기법의 적용과 2단계 문제추출 휴리스틱에 대해 기술한다. 끝으로, 6장에서는 본 연구의 요약과 함께 향후의 연구방향을 제시한다.

## 2. 시스템의 구성 및 동작원리

### 2.1 시스템 개발환경

본 연구의 대상인 T 능력평가시스템은 웹서버의 부하를 줄이기 위해 웹서버와 데이터베이스서버를 독립적으로 운영하는 3계층(tiers) 방식을 사용하고 있다. 웹서버는 펜티엄-III 1 GHz Dual Processor에서 Windows 2000 Server를 운영체제로 하며, 데이터베이스는 Sun의 Unix 운영체제에서 운영되는 오라클사의 Oracle 8i를 사용한다. 또한, 데이터베이스와 프로그램의 연동은 JDBC를 이용하였으며, 주요 프로그램 및 알고리즘은 자바 서블릿(servlet)으로 작성하였다. 이를 그림으로 나타내면

<그림 1>과 같다.

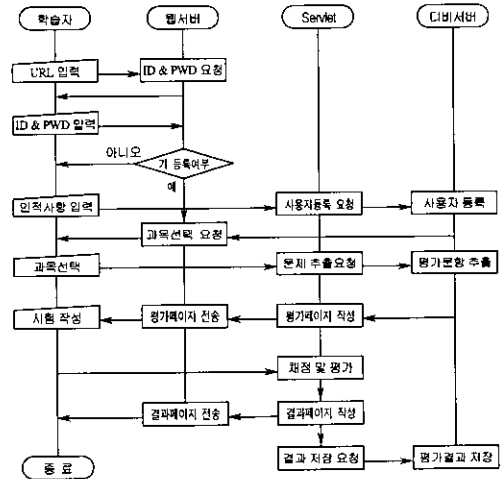


<그림 1> 문제은행시스템의 동작 흐름도

## 2.2 시스템 흐름도

본 연구의 근간을 이루는 웹기반 능력평가시스템의 구성 및 작동원리에 대한 이해를 돕기 위해서 시스템의 흐름을 간략히 기술하면 다음과 같다. 먼저, 신분이 인증된 출제자에 의해 입력된 평가문항과 정답은 데이터베이스에 저장되며, 학습자로부터 학습 성취도 평가요청이 있을 때, 학습자에 대한 인증과정을 거친 후에 문제은행 데이터베이스로부터 학습자가 선택한 과목의 문항을 동적으로 추출하여 제시한다. 문항의 추출은 웹서버에서 실행되는 서블릿 프로그램에 의해 이루어지기 때문에 서버에 많은 부하를 가하게 되므로, 가능한 한 효율적인 문제추출 알고리즘을 채택함으로써 웹서버의 부하를 경감시킬 필요가 있다.

학습자가 웹 브라우저에 나타난 문항을 풀어서 전송하면, 시스템은 데이터베이스에 있는 정답에 따라 채점한 평가점수와 함께 문항에 대한 해설을 제공한다. 이 때, 각 평가자에 대한 평가결과는 각 과목, 영역 및 문항별로 데이터베이스에 누적되어 개인 및 전체의 과목 및 영역별 학습성취도를 계산하여 제시함으로써 자신의 학습능력을 스스로 진단할 수 있도록 정보를 제공한다. 또한, 평가결과에 따라 각 문항의 난이도가 자동적으로 조정되기 때문에, 평가의 난이도를 항상 일정하게 유지할 수 있다. 이 시스템에서 평가가 이루어지는 과정을 흐름도로 나타내면 <그림 2>와 같다.



<그림 2> 웹기반 학습평가시스템의 평가흐름도

## 3. 시뮬레이티드 어닐링

금속의 담금질(annealing)이란 고체를 높은 온도에서 녹인 후에 다시 완전한 격자상태의 결정체가 될 때까지 식히는 물리적인 과정을 말하는 것으로, 이 과정에서 고체의 자유에너지(free energy)는 최소화된다. 오랜 경험에 따르면, 고체화되는 과정에서 지역 최소점에 빠지지 않도록 하기 위해서는 조심스럽게 서서히 식혀야 한다고 한다.

최적화(optimization) 문제도 이와 유사한 과정으로 설명할 수 있는데, 이는 여러 가지의 대안 중에서 최소비용이 소요되는 대안을 선택하는 문제로 전환이 가능하기 때문이다. 여기서 비용함수(cost function)와 자유에너지간의 관계, 그리고 해답과 물리적인 상태간의 관계를 정립함으로써 물리적인 담금질과정을 시뮬레이션을 통한 최적화문제의 해결과정으로 전환할 수 있는데, 이를 시뮬레이티드 어닐링(simulated annealing)이라 한다.

시뮬레이티드 어닐링은 Metropolis *et al.*(1953)에 의해 제안된 이후 Kirkpatrick *et al.* (1983)이 처음으로 최적화문제에 적용하여 성공적인 결과를 제시하였다. 시뮬레이티드 어닐링에서는 에너지를 최소화하는 것을 목표로 하기 때문에 새로운 해가

기존의 해보다 작은 에너지를 갖는 경우에는 무조건 개선된 해로 채택한다. 그러나 새로운 해가 기존의 해보다 나빠서 큰 에너지를 갖는 경우에도 사전에 설정된 확률함수에 따라 새로운 해를 채택(accept)하거나 혹은 거절(reject)한다는 특징을 갖고 있다. 따라서 만족해를 찾는 일반적인 휴리스틱 기법이 갖고 있는 가장 큰 문제인 지역 최소점(local optimum)에 빠지는 문제를 상당부분 해결할 수 있는 장점을 갖고 있다.

Lundy & Mees(1986)는 시뮬레이티드 어닐링은 탐색횟수가 증가될수록 전역해(global optimum)에 점진적으로 근접한다는 사실을 증명한 바 있다. 시뮬레이티드 어닐링에 관한 설명은 관련 논문이나 자료에서 쉽게 찾아볼 수 있기 때문에 알고리즘에 대한 상세한 설명은 생략한다.

### 4. 문제세트 생성 알고리즘

#### 4.1 문제에 대한 정의

본 논문에서 다루고자 하는 문제는 문항번호, 문제유형, 난이도 등 다양한 속성을 갖는  $n$ 개의 문항으로 구성된 문제은행으로부터 각종 제약사항을 모두 만족하는  $m$ 개의 문항을 추출하여 문제세트를 생성하는 것이다. 따라서,  $nCm$  개의 상태공간에서 최적해를 찾는 탐색문제로 평가될 수 있다.

본 연구에서는 전국적으로 시행되고 있는 T 능력평가시험을 대상으로 <표 1>에서 보는 바와 같은 문제유형, 소재, 난이도 등의 문항구성을 제약사항으로 설정하여 모든 제약사항을 만족하는 200개의 문항으로 이루어진 문제세트를 추출하기 위한 시뮬레이션을 실시하였다. 여기서, 영역은 어휘, 문법, 청해, 그리고 독해 등 각 평가영역을 의미하고, 유형은 빈칸 채우기, 오류 찾기 등의 평가유형을 의미하며, 세부영역은 각 영역을 보다 세분화한 것이다. 각 세부영역에 대한 상세한 내용은 해당 능력평가시험의 보안과 관련되어 편의상

A, B, C 등으로 기호화하여 표현하였으며, '각 2~3' 형식으로 표현된 항목들은 내부적으로는 세부영역별로 문항수가 확정되어 있으나 생략하여 기술한 것이다.

<표 1> 각각의 제약사항에 따른 영역별 출제 문항수

#### (a) 제약사항 1

영역	유형	세부영역	문항수	합계
A	AA	-	25	50
	AB	-	25	
B	BA	BAA, BAB	각 20	50
	BB	BBA, BBB	각 5	
C	CA	CAA	5	60
	CB	CAB	25	
		CBA~ABC	각 5	
CC	CCA~CCC	각 5		
D	DA	DAA	10	40
	DB	DAB	5	
		DBA	5	
		DBB	10	
DC	DCA	10		
총합				200

#### (b) 제약사항 2

영역	유형	세부영역	문항수	합계
A	AD	ADA~ADN	각 2~3	50
	AE	AEA~AEH	각 1~2	
B	BD	BDA~BDK	각 1~6	50
	BE	BEA~BEM	각 1~4	
C	CD	CDA~CDN	각 3~6	60
	CE	CEA~CEB	각 1~2	
D	DD	DDA~DDE	각 5	40
	DE	DEA~DEE	각 5	
총합				200

#### (c) 제약사항 3

영역	유형	세부영역	문항수	합계
A	AF~AJ	-	각 10	50
B	BF~BJ	-	각 10	50
C	CF~CJ	-	각 12	60
D	DF~DJ	-	각 8	40
총합				200

### 4.2 순차적인 문제추출 알고리즘

전통적으로 많이 사용되어 온 ‘순차적인 문제추출 알고리즘’은 문제은행으로부터 임의의 한 문항을 선택하여 문제세트에 포함시켰을 때 각 영역별 문항수 제약을 위반하지 않으면 채택하고, 어느 하나라도 위반하면 제외시킨다. 임의로 선택한 각 문항을 차례대로 평가하여 모든 제약을 만족하는  $m$ 개의 문항이 추출되면 문제세트의 구성을 성공적으로 종료하며, 문제은행 내에 있는 모든 문항을 평가했음에도 불구하고  $m$ 개의 문항이 추출되지 못하면 문제세트의 구성은 실패한다. 이를 자바 프로그래밍 언어의 문법에 따라서 메소드로 표현하면 <그림 3>과 같다.

```

public static boolean testSetGeneration
(int totalItems, int testItems) {
    int index, select, nCount;

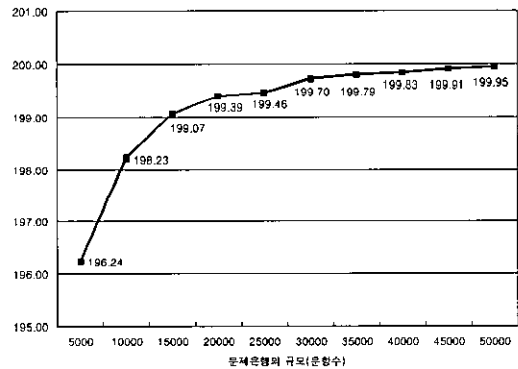
    Random rand = new Random();
    index=select=nCount=0;
    while (index < totalItems &&
           nCount < testItems) {
        select = rand.nextInt(totalItems-index) +
            index;
        swapItems(nCount, select); // 문항 교
        환 메소드
        if (verifyTestItems(nCount)) // 문제세
            트 검증 메소드
            nCount++;
        else
            swapItems(index, nCount);
            index++;
    }
    if (nCount == testItems) return true;
    else return false;
}
    
```

<그림 3> 순차적인 문제추출 알고리즘

여기서,  $m-1$ 개의 문항이 추출되고 마지막 한 문항을 추출해야 하는 경우를 가정하면, 문제은행에 남아 있는 문항 중에서 평가영역, 유형, 난이도 등 모든 조건을 정확히 일치하는 문항이 존재해야만 문제세트의 구성이 성공한다. 따라서, 문제은행

에 많은 문항이 있을수록 성공적으로 문제세트를 구성할 가능성은 높아질 것이지만, 모든 제약조건을 만족하는  $m$ 개의 문항을 항상 성공적으로 추출한다고는 보장하지 못한다.

<그림 4>는 문제은행내의 문항의 개수를 5,000 개로부터 50,000 개까지 차례대로 변화시키면서 각각 100회의 시뮬레이션을 실시하여 제약조건을 만족하도록 추출된 문항의 평균개수를 보여준다. 여기서, 문제은행의 규모가 커질수록 상대적으로 더 많은 문항이 추출됨을 알 수 있다.

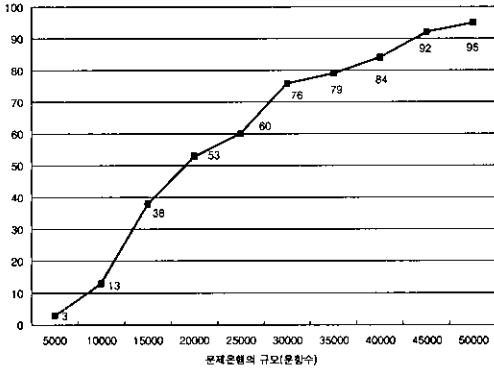


<그림 4> 문제은행의 규모에 따른 문항의 추출개수

<그림 5>는 문제은행의 규모에 따라서 각각 100회의 시뮬레이션을 실시하여 모든 제약조건을 만족하는 200개의 문항을 성공적으로 추출한 횟수를 그림으로 보여주고 있다. 여기서, 문제은행의 규모가 5,000 문항일 때에는 문제세트의 구성을 성공한 횟수가 100회의 시뮬레이션 중에서 3회에 불과하였으나, 문제은행의 규모가 50,000 문항으로 증가되면 95회로 향상되는 것으로 나타났다.

위의 결과로부터 문제은행의 규모가 작은 경우에 순차적인 문제추출 알고리즘을 적용하기 위해서는 모든 제약조건을 만족하는 문제세트가 구성될 때까지 문제추출과정을 여러 번 반복해야만 하는 불편함이 존재한다. 뿐만 아니라, 이 알고리즘은 문제은행의 규모가 커질 경우에도 모든 제약조건을 만족하는 문제세트를 100% 수준에서 생성하

는 항상 것을 보장하지는 못한다는 문제점을 갖고 있다.



〈그림 5〉 문제은행의 규모에 따른 문항추출 성공률

### 4.3 시뮬레이티드 어닐링

앞에서 기술한 순차적인 문제추출 알고리즘은 문제은행의 크기가 작을 경우 모든 제약조건을 만족하는 문제세트의 구성을 실패할 가능성이 상당히 높은 것으로 나타났다. 따라서, 본 연구에서는 모든 경우에 있어서 문제세트의 구성을 보장할 뿐만 아니라 보다 효율적으로 문제세트를 구성할 수 있도록 시뮬레이티드 어닐링을 도입하였으며, 그 과정을 간략히 기술하면 다음과 같다.

- ① 출제하고자 하는 문제영역, 문제유형, 난이도, 그리고 주제 및 소재 등의 각 영역별 출제문항수 비율을 기준으로 난수(Random) 함수를 이용하여  $n$ 개의 문항을 갖는 문제은행을 생성한다.
- ② 난수를 이용하여 임의의  $m$  문항을 추출하여 초기해로 채택한다.
- ③ 생성된 문제세트의 에너지(부적합도)를 계산한다. 여기서, 에너지는 영역별로 출제하고자 하는 문항의 수와 출제된 문항의 수의 함수로 표현된다.
- ④ 에너지가 0이면 모든 제약조건을 만족하는

문항이 추출되었음을 의미하므로, 문제추출을 종료하며, 0보다 크면 다음 단계로 넘어간다.

- ⑤ 시뮬레이티드 어닐링 기법에서 계산된 채택 확률( $P$ )이 난수로 만든 확률보다 크면 새로운 해로 채택하고, 그렇지 않으면 기각한다.
- ⑥ 온도함수에 따라서 온도를 감소시킨다.
- ⑦ 난수를 이용하여 문제세트 내에 있는 임의의 문항과 문제세트 외부에 있는 임의의 문항을 선택하여 서로 교환한 후, 위의 ③단계로부터 다시 실행한다.

시뮬레이티드 어닐링을 적용하기 위해서는 에너지함수와 채택확률을 결정해야 하는데, 이들 함수에 따라서 시뮬레이티드 어닐링의 적용성과는 현저히 달라지게 된다. 다음에서는 이들 함수의 설계와 테스트 결과에 대해 기술한다.

#### 4.3.1 에너지 함수

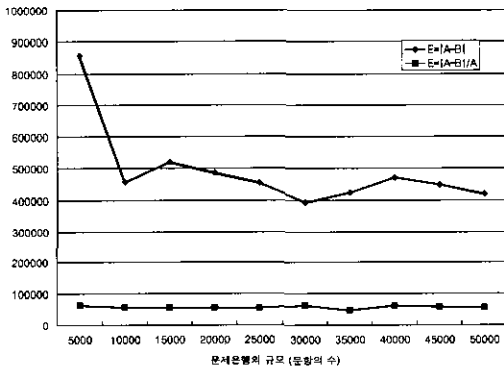
시뮬레이티드 어닐링을 적용하기 위해 몇 가지의 에너지 함수를 개발하여 시뮬레이션을 실시한 결과, 에너지함수가 시뮬레이티드 어닐링의 성과에 큰 영향을 미치는 것으로 나타났다. 에너지 함수는 에너지가 낮아질 수록 최적해에 근접해 가도록 설정해야 하는데, 본 문제에서는 세부 영역별로 출제하고자 하는 문항의 수( $A$ )와 문제세트에 포함된 문항의 수( $B$ )의 함수로 표현될 수 있다.

이에 따라, 본 연구에서는 다음과 같이 세부 영역별 출제문항수 차이의 절대값과 출제하고자 하는 문항수에 대한 비율 등을 에너지함수로 개발하여 시뮬레이션을 실시하였다. 이 외에도 지수 및 로그 함수를 이용하는 방안에 대해서도 시뮬레이션을 실시하였으나, 위의 두 번째 에너지 함수보다 나은 성과를 가져오지는 못하는 것으로 나타났다.

$$Energy_1 = |A - B|$$

$$Energy_2 = \frac{|A - B|}{A}$$

<그림 6>은 위에서 설정한 두 가지 에너지 함수를 이용하여 문제은행의 규모를 5,000 문항으로부터 50,000 문항으로 변화시키면서 각각 100회의 시뮬레이션을 실시하여 모든 제약조건을 만족하는 문항을 추출하기까지의 평균 탐색횟수를 그래프로 표현한 것이다. 여기서, 시뮬레이티드 어닐링의 성과에 영향을 미치는 또 다른 변수인 채택확률에 의한 영향을 제거하기 위해 채택확률(P)은 0으로 설정하였다.



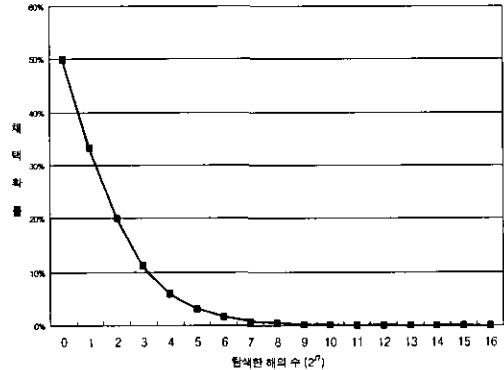
<그림 6> 에너지 함수에 따른 성과차이

시뮬레이션의 결과에서 볼 수 있는 바와 같이, 에너지 함수에 따라서 시뮬레이션의 성과에 많은 차이가 있음을 알 수 있으며, 두 번째 에너지 함수를 사용할 경우 탐색횟수가 약 11% 수준으로 감소되는 것으로 분석되었다. 따라서, 뒤에서는 비율을 이용하는 위의 두 번째 함수를 문제추출을 위한 에너지 함수로 적용한 결과에 대해서만 기술한다.

### 4.3.2 채택확률(P)

시뮬레이티드 어닐링의 성과는 온도함수 및 온도상수의 값에 따라서 달라질 수 있는데, 본 연구에서는 예비 시뮬레이션을 통하여 보다 나은 성과를 보였던 고속 시뮬레이티드 어닐링 온도함수를 사용하였다. 온도상수의 값은 에너지의 차이 ( $\Delta E$ )가 1일 때 시뮬레이션 초기에 약 50%로부터 말기에 약 0%의 채택확률(P)을 갖도록 설정하였

다. <그림 7>은 탐색횟수가 2"으로 증가할 때의 채택확률의 변화를 보여주고 있다.

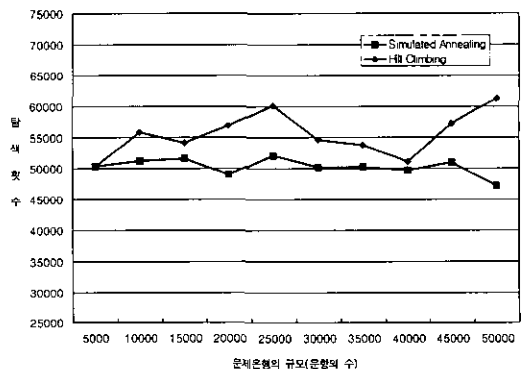


<그림 7> 탐색횟수 증가에 따른 채택확률의 변화

### 4.3.3 시뮬레이티드 어닐링의 적용

위에서 기술한 조건하에서 시뮬레이티드 어닐링의 적용성과를 평가하기 위해 보다 나은 해만을 채택하는 Hill Climbing Search의 성과와 비교하였다. 참고로, 시뮬레이티드 어닐링에서는 채택확률(P)에 따라 현재보다 나쁜 해도 채택할 수가 있기 때문에, 채택확률이 0이 되도록 설정하면 Hill Climbing Search가 된다.

<그림 8>은 문제은행의 규모를 5,000 문항으로부터 50,000 문항까지 변화시키면서, 각각에 대해 100회의 시뮬레이션을 실시한 후 모든 제약조건을 만족하는 문항을 추출하기까지 소요되는 평균탐색 횟수를 그래프로 나타낸 것이다.



<그림 8> 두 가지 알고리즘의 성과비교

시뮬레이션 결과 시뮬레이티드 어닐링을 적용할 경우 Hill Climbing Search 보다 탐색횟수가 약 10% 감소되는 것으로 나타났다. 그러나, 두 가지 알고리즘 모두 문제은행내의 문항의 수가 증가한다고 해서 최적해를 탐색하는 데 소요되는 시간이 크게 증가하거나 감소하지는 않았다. 따라서, 이들 기법을 사용하면 문제은행시스템을 개발하는 초기에 문제은행의 규모가 작더라도 모든 제약조건을 만족하는 문제세트를 성공적으로 구성할 수 있을 것이다.

4.4 2 단계 문제추출 휴리스틱

앞에서 기술한 순차적인 문제추출 알고리즘은 최적해를 찾는 데 실패할 수는 있으나, 짧은 시간내에 초기해를 구성할 수 있는 특징을 갖고 있다. 따라서, 본 연구에서는 위의 두 가지 방법을 결합한 2단계 문제추출 휴리스틱을 개발하여 시뮬레이션을 실시하였다.

즉, 1단계에서 순차적인 문제추출 알고리즘을 적용하여 초기해를 구성한 후에, 2단계로서 시뮬레이티드 어닐링을 적용하여 최적해를 찾으려 하는 것이다. 이 때, 1단계에서 이미 최적해를 발견하면 2단계는 수행되지 않으며, 1단계에서 최적해를 발견하지 못했을 경우에만 2단계가 수행된다. 이를 자바 프로그래밍 언어의 문법에 따라서 메소드로 표현하면 <그림 9>와 같다.

```
public static int twoPhaseHeuristic (int totalItems,
int testItems, double T0, double T1) {
    int inSet, outSet, nSearch;
    double oldEnergy, newEnergy, prob;

    Random rand = new Random();
    nSearch = 0;

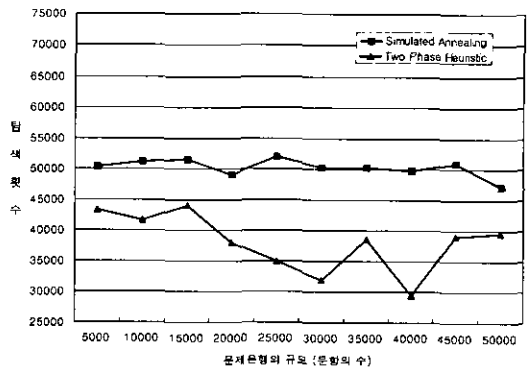
    firstPhase(totalItems, testItems);
        // 1단계 문제추출 메소드
    oldEnergy = energyForItems(testItems);
        // 에너지 계산 메소드
    while (oldEnergy > 0) {
        inSet = rand.nextInt(testItems);
        outSet = rand.nextInt(totalItems-
```

```
testItems) + testItems;
    swapItems(inSet, outSet);
        // 문항교환 메소드

    newEnergy = energyForItems
        (testItems);
    prob = Math.exp(-(newEnergy-
        oldEnergy)/(T1/(1+nSearch)+T0));
    if (prob > rand.nextDouble())
        oldEnergy = newEnergy;
    else
        swapItems(inSet, outSet);
        nSearch++;
    }
    return nSearch;
}
```

<그림 9> 2단계 문제추출 휴리스틱

2단계 문제추출 휴리스틱의 문제추출성과를 평가하기 위해 앞에서와 동일한 방법으로 문제은행의 규모를 5,000 문항으로부터 50,000 문항에 이르기까지 변화시키면서 각각 100회의 시뮬레이션을 실시하여, 1단계에서 해를 발견하지 못하여 2단계까지 수행한 경우 해를 찾기까지 추가적으로 요구되는 탐색횟수를 측정하였다. <그림 10>은 시뮬레이티드 어닐링과 2단계 문제추출 휴리스틱에서 문제구성에 요구되는 평균탐색횟수를 그래프로 표현한 것이다.



<그림 10> 2단계 문제추출 휴리스틱의 성과비교

그림에서 보는 바와 같이, 문제은행의 규모에 관계없이 2단계 문제추출 휴리스틱이 훨씬 더 나은



성과를 나타냈다. 특히, 이 그림은 2단계 휴리스틱을 적용할 때 1단계에서 최적해를 찾지 못해 추가적으로 2단계에 필요한 경우의 성과만을 나타낸 것이기 때문에, 1단계에서 최적해를 발견한 경우를 고려한다면, 2단계 휴리스틱은 훨씬 더 효율적으로 최적해를 발견하는 것으로 평가될 수 있을 것이다.

## 5. 결론 및 요약

인터넷 등 정보통신기술의 발달은 시간과 공간의 제약을 극복하여 언제 어디서나 원격으로 학습할 수 있는 새로운 교육체제인 사이버교육을 현실화시키고 있으며, 사이버공간에서 평가를 수행하는 각종 능력평가 및 자격검정시험이 등장하고 있다. 그러나, 대부분의 평가시험이 정해진 포맷에 따라 모든 평가자에게 동일한 문항을 제시하거나 혹은 문항의 유형과 난이도 등에서 일관성이 없는 문항을 출제하고 있는 실정이다.

이에 따라, 본 논문에서는 문제은행시스템, 특히 사이버공간에서의 능력평가를 목적으로 하는 문제은행으로부터 사전에 설정된 문제영역, 문제유형, 난이도, 주제 등 다양한 평가항목별 문항수를 정확히 일치하는 일관성 있는 문항을 추출하기 위해 시뮬레이티드 어닐링 기법을 도입하는 방안에 대해 기술하였다. 또한, 최적해의 탐색시간을 최소화하기 위해 전통적으로 많이 사용해 왔던 순차적인 문제추출 알고리즘과 시뮬레이티드 어닐링을 결합한 2단계 문제추출 휴리스틱의 적용에 대해서도 기술하였다. 시뮬레이션 결과, 에너지함수가 시뮬레이티드 어닐링에 매우 큰 영향을 미치는 것으로 확인되었고, 시뮬레이티드 어닐링 기법을 적용함으로써 모든 제약조건을 만족하는 문제세트를 100% 수준에서 생성하였으며, 최적해의 탐색시간도 현저히 감소하는 것으로 나타났다.

향후의 연구과제로서, 사이버 상에서 이루어지는 능력평가시험은 문제세트를 구성하기 위해 많은 탐색횟수가 요구될수록 서버에 많은 부하가 걸리기 때문에, 시뮬레이티드 어닐링 외에도 유전해

법 등 다양한 탐색기법을 적용함으로써 보다 효율적으로 문제세트를 구성하는 방안에 대해 연구하고자 한다.

## 참고 문헌

- [1] 백영균, "웹 기반 학습의 설계", 양서원, 1999.
- [2] 서대교 외, "인터넷을 이용한 시험 및 과제물 처리 방법에 관한 연구", 「한국정보처리학회」, '98 추계 학술발표 논문집, 제5권, 제2호, 1998.
- [3] 오성한 외, "개별학습을 위한 원격 교육 시스템", 「한국정보처리학회」, '98 추계 학술 발표 논문집, 제5권, 제2호, 1998.
- [4] 임경미, 김원영, 김치수, "Web 기반 평가도구의 개발과 교육적 활용", 「한국정보교육학회」, '99 하계 학술대회 논문집, 1999.
- [5] 임희숙, 김창근, 김수형, "CGI를 이용한 웹 기반 문제은행시스템 설계 및 구현", 「한국정보교육학회」, '99 하계 학술발표 논문집, pp.307-312, 공주교육대학교, 1999.
- [6] 전성민, "정보통신기술과 원격교육", 「경영교육연구」, 제2권, 제1호, 1998, pp.5-26.
- [7] 정성길외, "멀티미디어에 기반한 원격개별학습시스템 설계", 「한국정보처리학회」, '98 춘계 학술발표 논문집, 1998.
- [8] 황상연, 김두규, 임병민, 김정훈, 이재무, "웹을 기반으로 한 학습자 진단 및 조인시스템 구현", 「한국정보교육학회」, '99 동계 학술발표 논문집, 1999.
- [9] 황인수, 한재민, "유전해법에서 시뮬레이티드 어닐링을 이용한 개체선택의 효과에 관한 연구", 「경영과학」, 제14권, 제1호, 1997, pp.77-85.
- [10] Date, C., An Introduction to Database Systems, Addison Wesley, New York, 1983.
- [11] Drucker, Peter, Post Capitalist Society (자본주의 이후의 사회), 이재규 역, 한국경제신문사, 1993.

- [12] Holland, J.H. and J.S. Reitman, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [13] Kampke, T., "Simulated Annealing : Use of a New Tool in Bin Packing," *Annals of Operations Research*, Vol.16 (1988), pp.327-332.
- [14] Kirkpatrick, S., C.D Jr. Gelatt and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol.220 (1983), pp.671-680.
- [15] Lundy, M and A. Mees, "Convergence of an Annealing Algorithm," *Mathematical Programming*, Vol.34 (1986), pp.111-124.
- [16] McFadden, F.R. and J.A. Hoffer, *Database Management*, 3rd ed., Benjamin/Cummings, 1991.
- [17] Metropolis, W., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," *Journal of Chemistry Physics*, Vol.21 (1953), pp.1087-1092.
- [18] Sprague, R.H, "A Framework for the Development of Decision Support Systems," *MIS Quarterly*, Vol.4, No.4 (1980), pp.1-26.
- [19] Szu, H, "Fast Simulated Annealing," In J. Denker(Ed.), *Neural Network for Computing* : American Institute of Physics, New York (1986), pp.420-425.
- [20] Zhen-Ping Lo and B. Bavarian, "Optimization of Job Scheduling on Parallel Machine by Simulated Annealing Algorithms," *Expert Systems With Applications*, Vol. 4 (1992), pp.323-328.