

이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색

(Reactive Tabu Search using Neighborhood Strategy
Switching Mechanism)

김재호[†] 이희상^{**} 한현구^{***}

(Jaeho Kim) (Heesang Lee) (Hyungoo Han)

요약 반응적 타부 탐색은 단순한 타부 탐색과 비교해서 중장기 메모리를 이용한 학습을 통하여 타부리스트의 크기를 반응적으로 변화시킴으로써 NP-hard 문제에 속하는 다양한 조합 최적화 문제에 대해서 좋은 해를 효율적으로 찾는다. 본 논문에서는 반응적 타부 탐색에 있어서 중장기 메모리를 이용한 탈출 메커니즘으로 이웃 해 전략 전환 메커니즘이라는 개념을 제시한다. 제시된 이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색을 특정 공과 대학의 강의 시간표 작성 문제와 외판원문제 (traveling salesman problem)에 적용하여 기존의 반응적 타부 탐색과 비교 분석을 하였다. 전산 실험 결과 제시된 알고리즘은 기존의 반응적 타부 탐색 알고리즘에 비교하여 더 좋은 해를 더 짧은 시간에 찾아주었다.

Abstract Reactive tabu search algorithms that exploit learning using long term memory to enhance the strength of searching good solutions of many NP-hard combinatorial optimization problems are compared to simple tabu search algorithms. In this paper we introduce a new concept so called neighborhood strategy switching mechanism (NSSM) that can be used as an escape mechanism for using long term memory in a reactive tabu search algorithm. We implement and experiment an NSSM tabu search algorithm with a university time tabling problem and traveling salesman problem. We compare the suggested algorithm with a known reactive tabu search algorithm. In our computational experiments, the suggested algorithm outperforms the other algorithms to find better solutions in shorter time.

1. 서론

최근 컴퓨터 성능의 급속한 발전에도 불구하고, 대형 수리 문제 특히 대형화 최적화문제 중에서 많은 문제가 NP-hard로써 문제크기가 커짐에 따라 컴퓨터는 용량과 계산시간의 한계를 갖게 된다. 이러한 NP-hard 문제는 일반적인 최적화 알고리즘으로는 주어진 시간 내에 최적 해를 얻는 것을 보장할 수 없다. 여러 가지 고전적 휴리스틱 또는 규칙기반 전문가 시스템에 근거한 탐색

방법을 통해 탐색시간을 줄일 수 있으나, 이러한 시도들은 보통 최적 해(optimal solution) 대신에 근사 해(near optimal solution)를 찾아주고, 최적 해에서 특정 범위 이내(performance guaranteed)의 좋은 근사 해를 보장하기도 어려울 뿐만 아니라 알고리즘이 해결하고자 하는 문제에 의존적이어서 문제마다의 특성에 맞추어 개발해야 하는 어려움을 가지고 있다. 이러한 문제로 인해 특정문제에 크게 국한됨 없이 다양한 문제에 적용이 가능하고 해공간의 탐색능력이 우수한 메타휴리스틱(meta-heuristic) 기법들에 대한 연구가 활발히 진행되고 있다.

이와 같은 메타 휴리스틱 방법들의 특징은 종래의 휴리스틱 기법들에 비하여 유연성이 뛰어나고 탐색 공간을 확대하는 등의 일반적 특징을 가지고 있다. 이 중 유전자 알고리즘은 구하려는 해를 염색체로 표현하여 최

[†] 정 회 원 : 전자부품연구원 고속네트워크연구센터 연구원
jhkim@kcti.re.kr

^{**} 비 회 원 : 한국외국어대학교 산업정보시스템공학부 교수
leehee@san.hufs.ac.kr

^{***} 비 회 원 : 한국외국어대학교 컴퓨터공학과 교수
hgghan@hufs.ac.kr

논문접수: 2000년 5월 19일

실사완료: 2001년 5월 18일

초 모집단인 해 집합을 다양한 유전연산을 통해 여러 세대 동안 진행시켜 결국 우수한 해를 만들어내는 방법으로 가능 해를 폭 넓게 탐색하는 장점을 가지고 있다. 시뮬레이티드 어닐링은 반복적인 개선 법을 기본으로 하되, 해의 비개선 움직임도 확률적으로 허용하는 방법이다. 이는 고체 물리학에서 물리적인 구조를 조합적 문제에서의 하나의 가능 해의 구조로 대응시키고 고체의 어닐링 과정에서 가장 낮은 에너지 수준을 갖는 상태가 최적 해에 대응된다는 점에 기반하고 있다. 그리고 본 연구에서 개선하고자 하는 타부 탐색은 유연한 메모리 구조를 사용하여 탐색의 방향성을 역사적인 정보를 기준으로 조정하는 탐색 방법으로 최근의 보고에 따르면 많은 문제의 해법연구에 좋은 결과를 보인 메타 휴리스틱 방법이다.

본 연구에서는 메타 휴리스틱 방법 중에서 타부 탐색 방법에 중장기적인 메모리 구조를 활용하여 타부의 크기를 적응적으로 유연하게 결정하는 반응적 타부 탐색 방법을 사용한다. 특히 많은 최적화 문제가 다양한 이웃 해 전략을 가질 수 있다는 특성에 주목하여, 이웃 해 전략을 전환하는 메커니즘을 통한 새로운 중장기 메모리 활용 기법을 개발, 기존의 반응적 타부 탐색 방법을 효과적으로 개선하였다.

2. 반응적 타부 탐색

타부 탐색은 1960년대 후반과 1970년대 초반에 처음 개념이 제시되었으며, Glover에 의해서 최적화 문제에 적용 가능하도록 개량되었다[1, 2, 3]. 타부 탐색은 인간의 기억 과정을 모방하여 탐색과정에 유연한 메모리를 사용하는 것이 그 기본 아이디어로 많은 문제에서 이제까지의 다양한 고전적 발견적 방법이나 다른 메타 휴리스틱 기법으로 찾아낸 해보다 더 좋은 해를 찾았다고 보고되었다[4]. 또한 정량화하기 어려운 제약 조건을 반영하거나 많은 제약사항을 동시에 고려할 수 있다는 등의 장점이 있다. 기존 언덕등산 휴리스틱(hill climbing heuristic)은 해를 개선하는 방향으로만 진행을 하므로 시작점에서 지역 최적해(local optimum)를 향하여 유일한 한쪽 방향으로만 진행을 하게 되며, 휴리스틱이 더 이상 개선을 하지 못할 때 최종적인 지역 최적해를 얻게 된다. 그러나 타부 탐색에서 움직임은 탐색 과정에서 과거의 타부 움직임을 기억하고, 타부 움직임에 들지 않는 움직임 중 최상을 선택함으로써(빠른 진행을 위해서 샘플링을 사용하기도 한다) 지역 최적해에서 이동을 가능하게 하여 새로운 방향으로 탐색을 진행할 수가 있다. 타부 움직임은 최근에 탐색한 역사적인 정보를 이용하

며 타부 리스트나 타부 조건 등으로 표현될 수 있다. 예를 들어 타부 리스트의 길이가 l 이라고 할 때 최근 l 번 동안의 움직임에 대한 반대 움직임들의 집합을 타부 리스트라고 정의 한다면, 해 s_0 에서 s_l 으로의 움직임이 최근 l 번만에 일어났다면 s_l 에서 s_0 으로의 움직임은 타부 리스트에 포함되어 타부 움직임이 된다.

우리가 해결하고자 하는 문제의 탐색공간을 S 라고 할 때, 가능해 $s(s \in S)$ 에 적용할 수 있는 움직임의 집합이 k 개가 존재 한다면, 적용가능 움직임 집합(M)을 $M = \{m_1, \dots, m_k\}$ 로 나타내자. 가능해 s 에 모든 움직임을 적용해서 얻어지는 해를 $M(s) = \{m_1(s), \dots, m_k(s)\}$ 라 하면 이중 가능 해들의 집합인 $N(s)(N(s) \subseteq M(s))$ 를 s 의 이웃 해라 한다. 선택된 초기 해를 $s_0(s_0 \in S)$ 라 하면 타부 탐색의 반복적 탐색과정에 생성되는 해는 s_0, s_1, \dots, s_n 이 된다. 탐색의 반복과정에서 s_{j+1} 는 $N(s_j)$ 에서 선택된다. 이러한 선택 과정은 먼저 타부 리스트를 T 라 하면 s_j 와 관련된 타부 움직임들을 적용한 타부 집합 $T(s_j)(T(s_j) \subseteq N(s_j))$ 를 결정한다. 그러면 s_{j+1} 는 s_j 의 이웃 해 $N(s_j)$ 중 타부 집합 $T(s_j)$ 에 속하지 않고 해의 좋고 나쁨을 판단하는 기준이 되는 적합도 함수($f(s_{j+1})$)값이 최대(maximize 문제일 때)인 것을 선택한다. 즉, 모든 $s'(s' \in N(s_j) - T(s_j))$ 에 대해서 $f(s_{j+1}) \geq f(s')$ 가 성립되는 s_{j+1} 를 선택한다[1]. 다음은 타부 탐색에 대한 간단한 알고리즘이다.

단순 타부 탐색 알고리즘 (maximize 문제일 경우)

1. 초기값 $s_0(s_0 \in S)$ 를 선택, $s^* = s_0$ 라 놓고 s^* 는 현재까지의 가장 좋은 값, 반복 계산횟수 $j=0$, 타부 리스트 $T=\emptyset$.
2. 만약 $N(s_j) - T(s_j) = \emptyset$ 이면 단계 4로 간다
 $f(s_{j+1}) \geq f(s')$ 인 s_{j+1} 를 선택($s' \in N(s_j) - T(s_j)$).
3. 만약 $f(s_{j+1}) > f(s^*)$ 이면 $s^* = s_{j+1}$, $j=j+1$.
4. 종료 조건을 만족하면 멈추고 그렇지 않으면 T 를 수정하고(가장 오래된 타부 움직임을 제외시키고 선택된 움직임이 존재하면 T 에 포함 시킨다) 단계 2로 간다.

위의 단순한 타부 탐색을 좀 더 개량하기 위해서는 단기 메모리로서 4간단히 타부 조건이나 타부 리스트 외에도 열망수준(aspiration criteria)을 사용해서 타부이지만 가치 있는 움직임이라면 받아들이거나, 중장기 메모리로서 강화와 다양화를 이용해 역사적으로 발견된 좋은 해의 구조적 성질을 이용하거나 탐색되지 않은 지역으로의 이동을 가능하게 하는 방법들이 사용된다[5].

본 연구에서 적용한 반응적 타부 탐색(reactive tabu search: RTS)[6]은 타부 리스트의 크기를 기존의 타부 탐색(strict tabu search)처럼 고정시키지 않고 증장 기적으로 어느 범위 안에서 사이클의 발생을 기록하여 사이클 발생에 대해 유연하게 타부 리스트의 적당한 사이즈를 자동적으로 학습하게 된다. 즉 탐색 해들이 장기적으로 반복될 때는 리스트 사이즈를 증가시키며, 그렇지 않을 때 감소시킨다. 이러한 방법은 기존의 고정 리스트 사이즈에 기초한 타부 구조가 타부 리스트의 특정 크기에 따라서 계속되는 사이클을 포함하거나 너무 많은 탐색을 만드는 것을 해결할 수 있다. 즉, 반응적 타부 탐색이 강조하는 것은 전체적인 최적해에 가까운 해를 찾기 위해서 지역 최적해에서 효율적으로 벗어날 수 있도록 하는 것과, 탐색의 궤도가 사이클을 이루면서 그 안에서 비효율적으로 반복되는 경우를 방지하는 것이다. 기존의 반응적 타부 탐색은 탐색이 제한된 공간에서 갇히는 경우를 무작위적인(random) 이동으로 해결한다. 즉, 중장기 메모리(mid and long term memory)를 이용한 다양화 메커니즘(diversification mechanism)으로 무질서한 유인(chaotic attractor : CA)개념을 이용하여 제한된 탐색공간의 탈출을 유도한다[6]. 여기서 무질서한 유인은 제한된 탐색공간에서 탐색이 이루어지면 같은 해를 다시 탐색하는 경우가 발생하게 되는데 이를 장기적 메모리를 사용하여 탐지하여, 현재 탐색 과정의 평균 사이클 길이의 임의의 배만큼 무작위 이동을 하여 현재의 탐색공간을 벗어나는 탈출 메커니즘이다. 이러한 탈출 메커니즘에서 탐색이 제한된 공간에서 이루어 지는지를 감지하기 위해서는 기존의 탐색한 해를 기억하고 있어야 하는데, 이로 인한 과도한 메모리의 사용을 해결하기 위하여 주로 해싱(hashing) 기법을 이용한다.

3. 이웃 해 전략 전환 메커니즘

3.1 이웃 해 전략 전환의 아이디어

본 연구에서 제시하는 이웃 해 전략 전환 메커니즘(neighborhood strategy switching mechanism: NSSM)은 타부 탐색의 증장기적 다양화를 위한 방법으로 반응적 타부 탐색의 탈출 메커니즘으로 사용되었던 무질서한 유인 개념[6]을 대체하는 효과적인 방법이다. 이웃 해 전략 전환 메커니즘의 기본 아이디어는 반응적 타부 탐색에서 사용하는 이웃 해 전략을 하나만 사용하지 않고 복수 개를 사용하고, 증장기 메모리의 도움으로 적당한 때 이웃 해 전략을 바꾸는 것으로 다양화를 유도하는 것이다. 즉, 같은 가능 해에서 탐색을 시작하더라도 다른 이웃 해 전략을 쓰면 다른 탐색 공간을 만들어내

며, 다른 탐색 공간은 이웃 해 구조가 달라지므로 새로운 탐색 공간으로 탐색을 시도하여 장기적인 해의 다양화를 유도할 수 있다는 것이다. 그림 1과 2는 무질서한 유인 개념을 이용한 타부 탐색과 이웃 해 전략 전환 메커니즘을 이용한 타부 탐색의 진행과정을 간단히 도식을 통하여 비교 설명한 것이다. 그래프의 가로축은 가능한 탐색 공간을 나타내며, 세로축은 해들의 적합도를 나타낸다. 설명의 간결성을 위하여 모든 해는 4개의 이웃 해를 가지는 경우를 고려한다.

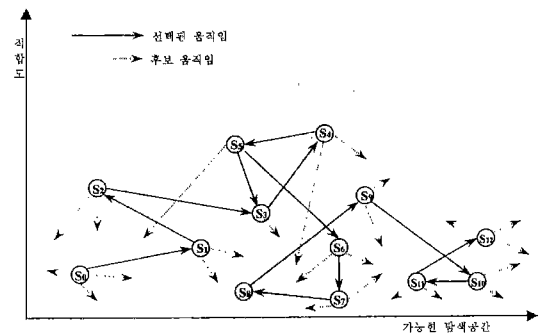


그림 1 무질서한 유인 개념을 이용한 타부 탐색

그림 1에 나타난 무질서한 유인 개념을 이용한 반응적 타부 탐색의 탐색과정을 살펴보면 무작위로 선정된 초기해 S_0 에서의 후보 움직임 중에서 가장 좋은 움직임(적합도를 가장 좋게 하는 움직임) $S_0 \rightarrow S_1$ 을 선택하고 움직임 $S_1 \rightarrow S_0$ 을 타부 리스트에 포함시킨다. 새로 찾아진 해 S_1 에서 시작해서 같은 방법으로 S_2, S_3, S_4, S_5 를 탐색하게 된다. 여기서 타부 리스트의 제약만을 가지고 해결되지않는 S_3, S_4, S_5 를 포함하는 사이클이 발생한 것을 알 수 있다. 무질서한 유인 개념에서는 이러한 사이클을 벗어나기 위해 발견된 사이클 지름의 임의의 배만큼을 무작위 이동을 한다. 그림에서는 사이클 지름 “3”의 2배인 “6”회를 무작위 이동을 하여 S_{11} 로 탈출을 하였다. S_{11} 부터는 다시 정상적인 타부 탐색을 시작하여 후보 움직임 중에서 가장 좋은 움직임인 $S_{11} \rightarrow S_{12}$ 를 선택한다. 즉 특정 탐색공간의 탈출을 발생한 사이클 지름의 임의의 배 만큼을 무작위 이동하는 방법으로 시도한다.

그림 2는 이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색을 보여주고 있다. 설명의 편리를 위하여 탈출이 일어나기 전의 탐색들을 “같은 phase에 있다”라고 정의하자. 즉 $S_0, S_1, S_2, S_3, S_4, S_5$ 는 같은 phase에 있다. 이웃 해 전략 전환 메커니즘의 과정은 S_0 에서 시작해서 그림 1에서와 같은 방법으로 S_1, S_2, S_3, S_4, S_5

를 탐색하게 된다. 여기서 S_3, S_4, S_5 를 포함하는 사이클이 발생한 것을 알 수 있다. 이웃 해 전략 전환 메커니즘은 이러한 사이클을 벗어나기 위해 현재 phase의 특정 해에서 이웃 해 전략 전환을 하게 된다. 이러한 이웃 해 전략 전환은 제한된 탐색 공간 어디에서 시도되어도 되겠지만 본 연구에서는 현재의 phase에서 가장 좋은 해(S_4)에서 전환을 하는 전략을 사용한다. 그림에서 볼 수 있듯이 S_4 의 이웃 해 전략 A를 통한 이웃 해에 비해서 이웃 해 전략 B를 통한 이웃 해 들은 전혀 새로운 해들임을 알 수 있다. 즉, 타부 탐색은 이웃 해 전략 전환 후 새로운 탐색공간으로의 탈출이 가능하게 된다. 이웃 해 전략 전환 후 타부 탐색은 새로운 이웃 해 전략 B를 이용하여 새롭게 진행한다.

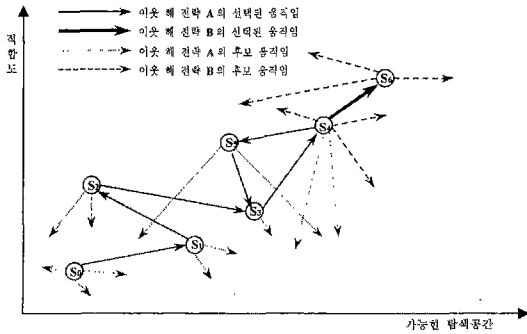


그림 2 이웃 해 전략 전환 메커니즘을 이용한 타부 탐색

이러한 이웃 해 전략 전환 메커니즘은 다른 방향으로의 탐색을 유도하게 하여 제한된 탐색 공간에서의 탈출을 하는데 있어 매우 효율적인 방법이다. 특히 성질이 상이한 이웃 해 전략을 사용함으로써 인하여 전혀 다른 탐색공간으로의 진입을 유도한다. 또 이웃 해 전략 전환이 현재 phase의 탐색 공간 내에서 가장 좋은 해에서 이루어진다면, 전환이 이루어지는 탈출점은 기존의 탐색 곡선의 기울기가 0인 지점으로 기존의 탐색 전략에서는 지역 최적해이다. 그러나 다른 이웃 해 전략에서는 이 지점의 탐색곡선의 기울기가 0이 아닌 값을 가질 확률이 높다. 즉, 이때 양의 기울기를 갖는다면 탈출점은 새로운 탐색공간 상에서는 더 이상 지역 최적해가 아니므로 더 좋은 이웃 해로의 움직임이 가능 할 수 있다. 따라서 본 연구에서 제시하는 이웃 해 전략 전환 메커니즘은 새로운 탐색 공간으로의 탈출과 기존의 지역 최적해가 다른 이웃 해 전략 사용 시에는 더 이상 지역 최적해가 아닐 수 있음을 사용하는 두 가지의 긍정적 역할을 동시에 수행하게 한다. 제4장의 실험에서 보듯이

기존의 반응적 타부 탐색에서의 무질서한 유인 탈출은 탈출 이후 적합도가 현저히 나빠지고 이후의 단기 메모리의 사용을 통한 타부 탐색을 통해 적합도가 다시 개선될 수 있는데 비교해, 이웃 해 전략 전환 메커니즘에서는 현재의 지역 최적해가 다른 이웃 해 전략의 해 공간에서는 더 이상 지역 최적 해가 아닐 수 있으므로 더 좋은 새로운 지역 최적 해를 찾거나 탐색이 더 진행된 후 더 좋은 해로의 이동을 쉽게 가능하도록 한다.

여러 개의 이웃 해 전략을 사용하는 것은 또 다른 매우 중요한 장점을 가지고 있다. 즉, 이웃 해 전략들은 문제의 구조적 특성에 기인한 각각 고유한 해 특성들을 가지는데 이러한 특성들을 다양한 이웃 해 전략들을 이용함으로써 최대한 이용할 수 있게 된다. 예를 들어 강의 시간표 문제의 경우 강의과목에 대한 시간배정이라는 scheduling적인 성질과 교수 또는 학생별 요구 조건이 갖는 graph coloring적인 두 가지 다른 속성을 각각 반영하도록 이웃 해 전략을 만들 수 있다. 특히 다양하고 복잡한 제약 조건들을 가지는 문제에 있어 한가지 이웃 해 전략으로 문제를 해결하는 것보다 문제의 제약 조건에 따라 복수개의 효율적인 이웃 해 전략들을 사용함으로써 더 다양한 이웃 해 공간의 탐색을 통한 좋은 해의 탐색이 가능할 수 있게 된다.

3.2 이웃 해 전략 전환 메커니즘을 적용한 반응적 타부 탐색 알고리즘

본 연구의 이웃 해 전략 전환 메커니즘을 활용한 반응적 타부 탐색의 알고리즘은 Battiti와 Tecchiolli[6]가 제안한 반응적 타부 탐색의 기본 골격을 사용하되, 탈출 메커니즘은 Battiti와 Tecchiolli가 사용한 무질서한 유인(chaotic attractor) 개념 대신 본 연구에서 제안한 이웃 해 전략 전환 메커니즘(NSSM)을 이용하였다. 초기 해의 선정은 불가능(infeasible) 해를 허락하는 임의의 초기값으로 배정하였으며, 타부 리스트의 크기는 사이클의 발생에 대해서 증가비율 또는 감소비율에 따라 반응적으로 조절이 된다. 다음은 알고리즘의 주요 부분을 보여준다. 알고리즘에 사용된 상수는 4장의 제어상수에서 상세히 설명하였다.

NSSM RTS 알고리즘(maximize 문제)

1. 초기화
 "이웃 해 전략을 전략 A"로 초기 설정
 초기해 $s_0(s_0 \in S)$ 를 선택, $s^*=s_0, s^{(-)}=s_0$
 $T=\emptyset, H=\emptyset, C=\emptyset, Tsize$ 를 초기화,
 $k=0, Csize=0, L=0, Steps=0$.
2. if s_k 가 H에 존재하지 않으면
 s_k 에 대한 해시테이블 값 $H(s_k).data=s_k$

```

    sk의 탐색 반복횟수 H(sk).repeat=1.
    sk를 탐색한 최종탐색시기 H(sk).last=k.
else
    H(sk).repeat=H(sk).repeat+1.
    sk의 반복주기 R=k-H(sk).last.
    H(sk).last=k.
    if H(sk).repeat > REPT 이면
        sk가 C에 존재하지 않으면 C에 sk를 추가;
        Csize=Csize+1
        if Csize>CHAOS면
            Csize=0, T=∅, swap(sk, s(*)).
            Neighborhood Strategy Switching.
            단계 5로 간다.
    if R <= CYCLE_MAX 이면(사이클의 발생시)
        L=L×0.9+Steps×0.1 (단 L=0 이면 L=Steps).
        Tsize=Tsize×INCREASE.
        Steps=0.
3. if Steps >L
    Tsize=Tsize×DECREASE (단 Tsize는 1 이상).
    Steps=0.
4. if N(sk)-T(sk)≠∅.
    모든 s'에 대해서 f(sj+1)≥f(s')인 sj+1를 선택 (s'∈N(sj)-T(sj)).
    T를 수정한다(T에 추가 공간이 없을 경우 첫 원소를 삭제하고, mk를 T에 추가).
    if f(sj+1)>f(s*)이면 s*=sj+1.
    if f(sj+1)>f(s(*))이면 s(*)=sj+1.
    k=k+1.
    Steps=Steps+1.
else
    f(sj+1)≥f(s')인 sj+1를 선택 (s'∈N(sj)).
    if sj+1가 열망조건(f(sj+1)>f(s*))을 만족하면
        s* = sj+1.
        s(*)=sj+1.
        k=k+1.
        Steps=Steps+1.
    else
        Tsize=Tsize×DECREASE. (if Tsize < 1 then
        Tsize=1).
        Steps = 0.
5. if k<MAX_ITERATION
    go to step 2.
6. stop.

```

여기서 각 변수들이 나타내는 의미는 s*는 현재까지의 가장 좋은 값을 나타내고, s^(j)는 현재 phase에서 가장 좋은 값, T는 타부 리스트, Tsize는 타부 리스트의 크기, H는 해시 테이블, C는 자주 반복되는 해 테이블을 나타내며, Csize는 C가 가지는 원소의 수, L은 사이클의 평균 발생 스텝 수, Steps는 타부 리스트의 크기 변경 후 지금까지의 스텝 수를 나타낸다.

4. 실험

본 연구에서 제시한 NSSM RTS의 실험적 검증을 위하여 대학 강의 시간표 작성 문제와 외판원 문제

(traveling salesman problem : TSP) 두 가지 문제에 대하여 적용 비교 분석을 하였다. 대학 강의 시간표 작성 문제는 매우 넓은 해공간과 수많은 제약 조건들을 가진 문제로서 최근 국내외에서 스케줄링 분야의 고전적 휴리스틱이나 유전자 알고리즘 등의 메타 휴리스틱(meta heuristic)이 많이 시도되고 있는 분야이다[7, 8]. 또한 TSP는 최소 경로를 찾는 대표적인 순열문제로서 도시의 수 N에 대하여 N!의 탐색공간을 갖는 NP-Hard이다.

4.1 실험 1 - 대학 강의 시간표 작성 문제

4.1.1 문제의 정의

일반적인 대학 강의 시간표 작성 문제는 하나의 강의 class를 일주일 중 어느 시간에, 어느 장소에서, 누구에 의해서 강의 되는가를 여러 가지 제약 조건들을 만족시키도록 할당하는 스케줄링 문제로 볼 수 있다. 본 논문에서 강의 class란 하나의 시간대에 할당되는 강좌 한 단위를 지칭하며 특정 과목이 1주일에 3시간 강의가 된다면 3개의 강의 class로 이루어 진다고 해석한다. 시간표 배정이 필요한 강의 class들의 집합을 $E = \{e_1, e_2, \dots, e_v\}$, 강의시간대의 집합을 $T = \{t_1, t_2, \dots, t_s\}$, 강의 장소의 집합을 $P = \{p_1, p_2, \dots, p_n\}$, 강사 또는 교수의 집합을 $A = \{a_1, a_2, \dots, a_m\}$ 으로 표현한다면 한 강의 class의 시간표 배정을 $(e \in E, t \in T, p \in P, a \in A)$ 인 4-tuple (e, t, p, a) 로 나타낼 수 있다. 이것은 강의 class e가 시간대 t에 배정되고, 장소 p에서 교수 a에 의해서 강의된다는 것을 나타낸다. 시간표 배정을 개개의 E에 대한 4-tuple의 할당이라고 정의하자[9]. 시간표 배정은 모든 강의 class들에 대해 강의시간대, 강의 장소, 교수를 하나씩 할당하는 것으로 해석되며 이를 통해 강의 시간표 작성이 가능하다. 하나의 시간표 배정이 유용한 강의 시간표가 되기 위해서는 각 대학이 갖고 있는 많은 제약 조건들을 가능한 만족하여야 한다. 본 실험에서는 많은 제약 조건들의 만족 정도를 계량화하여 최대값을 갖는 시간표 배정을 가장 좋은 시간표 배정으로 해석하도록 한다.

본 실험에서는 일반적인 강의 시간표 작성 문제의 4개의 요소 중 과목별 강의장소와 강의교수는 전공학부/학과나 교양학부 단위에서 자율적으로 결정되는 대학 환경에서의 강의 시간표 작성 문제를 다룬다. 이 경우는 강의장소와 강의교수는 강의마다 미리 지정이 되어 있거나, 강의 class별 시간대 배정이 해결된 후 결정되므로, 강의 시간표 작성 문제는 강의장소의 집합 P와 교수 집합 A를 제외한 2개의 tuple(e, t) (여기서 $e \in E, t \in T$)로 문제가 제한된다. 즉, 본 실험에서 다루는 강의

시간표 작성 문제는 모든 강의 class에 대한 두개의 tuple의 집합 중 제약 조건을 최대한 만족하는 시간표 배정을 찾는 문제로 정의한다. 이와 같은 2개의 tuple로 의 제한은 모형화 및 제안하는 방법론의 간결함을 보장하면서도 강의 장소 결정이나 교수 배정 결정의 자율권을 학부나 학과 단위에 보장하는 최근의 대학 현실을 반영하므로 실험의 일반성을 크게 상실하지 않는다고 믿어진다.

대학 강의 시간표 작성 문제는 많은 제약 조건들을 가질 수 있다. 본 실험에서는 특정대학에서 경험한 교수와 학생들의 요구사항을 기준으로 중요한 제약 조건을 선택하였다. 제약 조건은 반드시 지켜져야 강의가 진행될 수 있는 필수적 제약 조건(hard constraint)과 많이 지켜질수록 좋은 시간표가 되므로 가능한 적게 위반하고 싶은 부수적 제약 조건(soft constraint)으로 나눌 수 있다. 본 실험에서는 필수적 제약 조건에는 매우 무거운 벌점을 부여하였으며, 부수적 제약 조건에는 중요도에 따라 적당한 벌점을 부여하였다. 즉, 필수적 제약 조건은 전통적인 수리계획 최적화 문제의 제약 조건에 해당되어 반드시 만족되어야 하고, 부수적 제약 조건은 최적화 문제의 목적함수에 해당하여 가능한 많이 만족될수록 좋은 것이 된다. 본 실험에서는 많은 메타 휴리스틱 방법[6, 8, 9, 10]에서와 같이 제약 조건들을 벌점함수를 사용하여 모두 목적함수에 포함시켜 제약 조건이 없는 최적화 문제로 바꾸었다. 표 1에는 본 실험에서 적용한 강의 시간표 작성의 제약 조건들과 이들 중 각각의 제약 조건을 위반할 때 부과하는 벌점이 나타나 있다.

표 1 제약 조건

구분	번호	제약 내용	벌점
필수적 제약 조건	1	같은 학과 같은 학년 강의시간이 중복되면 안 된다.	20
	2	같은 교수가 강의하는 강의시간이 중복되면 안 된다.	20
	3	지정된 시간에 특정과목이 배정되면 안 된다. (교양강의 시간 지정 등)	20
	4	한 강의 과목이 하루에 일정시간 이상 강의되어서는 안 된다.	20
부수적 제약 조건	5	학생의 수업이 있는 날은 가능한 하루에 4시간 이상 되도록 한다.	0.01
	6	학생의 점심시간이 가능한 보장되어야 한다.	0.02
	7	교수의 강의가 있는 날은 가능한 하루에 2시간 이상 되도록 한다.	0.01
	8	교수의 점심시간이 가능한 보장되어야 한다.	0.02

필수적 제약 조건들인 제약 조건 1, 2, 3, 4는 모든 교수 또는 학과, 강의 과목들이 반드시 만족해야 하는 필수적 조건을 반영하였고, 부수적 제약 조건들인 제약 조건 5와 7은 교수와 학생의 강의가 너무 분산되지 않도록 하는 역할을 하며, 제약 조건 6과 8은 교수와 학생의 점심시간 확보와 함께 하루에 최대 강의시간이 지나치게 많지 않게 하는 역할을 한다. 본 실험에서는 제약 조건 6과 8은 제약 조건 5와 7보다 중요한 제약 조건으로 간주하였다. 각 제약조건에 대한 벌점은 부수 제약조건이 최대로 위반되더라도 필수 제약조건이 하나라도 위반되는 경우 보다 작은 값이 나오도록 하여 가능한 강의 시간표와 불가능 강의 시간표를 구분 할 수 있도록 하는 전제조건 하에 실험적으로 부여하였다.

본 실험에서는 해에 대한 평가를 계량화하고 탐색과정에서 좋은 이웃 해로의 이동을 평가하기 위해 적합도 함수를 이용한다. 적합도 함수는 특정해가 제약 조건을 얼마나 만족하느냐의 기준이 되며 많은 메타 휴리스틱 방법에서 목적함수 역할을 수행한다[9]. 본 실험에서는 표 1에서 제시한 제약 조건을 위반할 때마다 그에 해당하는 벌점을 부여하고 이러한 벌점을 모두 합한 것에 대해 역수를 취하는 다음과 같은 적합도 함수를 정의하였다.

$$f(g) = 1 / (1 + \sum_{i \in C} P_i \cdot V_i(g))$$

g : 하나의 시간표 배정(a time table)

C : 제약 조건들의 집합

P_i : 제약 조건 i 와 관련된 벌점

$V_i(g)$: 시간표 배정 g 가 제약 조건 i 를 위반한 횟수

적합도 함수 $f(g)$ 에서 g 는 하나의 해, 즉 하나의 시간표 배정을 의미하며 결정변수(decision variable)가 된다. $f(g)$ 는 해당 시간표의 시간표 배정이 위반한 제약 조건 개수 하나마다 벌점이 부과된다. 즉, 시간표가 X학과 2학년에 대해 강의 class e_j 와 강의 class e_k 가 시간이 중복되고, Y교수의 강의 class e_l 과 강의 class e_m 이 시간이 중복되었다면 $2 \times 20 = 40$ 점의 벌점이 적합도 함수의 분모에 반영된다. 적합도 함수 $f(g)$ 는 정의에 의해 그 값은 0과 1 사이의 수치가 되며, 필수적 제약 조건과 부수적 제약 조건을 하나도 어기지 않는 완전한 시간표 배정이 발견 된다면 1의 적합도를 가지며, 위반되는 제약 조건이 많을 수록 낮은 적합도를 갖게 된다. 발견된 시간표 배정이 필수적 제약 조건을 하나라도 어긴 경우는 시간표 배정에서 유도되는 강의 시간표를 적용 불가능 (infeasible) 시간표로 간주하고, 필수적 제약 조건을 하나도 어기지 않은 경우를 적용 가능한 (feasible) 시

간표로 간주하면, 별점이 20이하인 경우, 즉 적합도 함수 값이 $1/21 = 0.047619$ 보다 큰 값을 가지는 시간표 배정을 찾는 경우에는 적용 가능 시간표를 찾는 것으로 해석할 수 있다.

4.1.2 문제 표현 및 이웃 해 생성 방법

본 실험에서 해 개체의 표현은 각 강의 과목에 직접 강의 시작 시간을 배정하는 방법을 사용하였다. 총 강의 과목수는 n 개이며, 각 강의 과목에 배정되는 시간은 대학에서 가능한 강의 시간대의 집합으로 표현된다. 그림 3은 이러한 표현 방식을 보여주고 있다. 여기서 강의 과목 r 은 시간 t_r 에 시작된다는 것을 의미한다. 이러한 표현 방식은 간단하며 별점함수의 적용의 편리함 때문에 강의 시간표 작성 문제에서 많이 사용되어 왔다.

강의 강좌	0	1	2	...	r	...	$n-2$	$n-1$
강의 시작시간	3	20	9	...	t_r	...	24	29

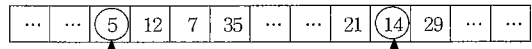
그림 3 해 개체의 표현

위의 그림 3에서 만일 매일 8시간의 강의가 월요일부터 금요일까지 5일간 가능한 경우라면 강의 시작시간의 값은 1에서 40까지의 정수 값을 가질 수 있다. 일반적으로 각 강의 과목은 연속해서 강의하는 시간이 미리 정해질 수 있다. 어느 과목이 2시간 연속 강의와 1시간 강의의 3시간으로 이루어진다고 하고, 그 과목의 2시간 강좌를 $r = 0$, 과목의 1시간 강좌를 $r = 1$ 로 표현한다고 하자. 만일 그림 3과 같이 배정되었다면 과목의 $r = 0$ 인 강의 class1, class2는 월요일 3, 4교시 연속 강의로 배정되며, 그 과목의 $r = 1$ 인 강의 class3은 수요일 4교시(20번째 강의 시간대는 하루 8시간씩 강의가 가능하므로 수요일 4교시)에 배정된다. 또 다른 예로 2시간 강의인 한 과목이 2시간 연속 강의이며 그림 3에 $r = 2$ 로 표현되었다면, 해당 과목의 강의 class1, class2는 화요일 1, 2교시에 강의가 배정된다는 의미이다.

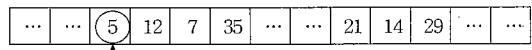
이웃 해 전략 전환 메커니즘을 사용하기 위해서는 여러 개의 이웃 해 생성 전략이 필요하게 되며 이러한 이웃 해 생성 전략은 상이한 이웃 해 구조를 가질수록 좋으며 문제의 제약 조건을 고려해서 선택되어야 한다. 본 실험에서는 많은 이웃 해 전략의 사용이 가능하지만 다음 두 가지 전략을 사용하여 이웃 해 전략 전환의 유용성을 검증하였다.

첫번째 이웃 해 생성 전략은 두 개의 강의 class의 강의 시간대를 서로 교환하는 방법을 사용하였다(그림 4 (a) 참조). 이는 대학에서 기존에 수동으로 시간표 작성 시, 교수 또는 학생의 제약 조건이 충돌할 때 흔히 사용

해은 교환기법으로 강의 시간표문제의 graph coloring적인 속성을 반영하는 이웃 해 교환 전략이다. 두 번째 이웃 해 생성 전략은 특정한 강의 class의 강의 시간대를 새로운 시간대로 바꾸는 방법을 이용하였다(그림 4 (b) 참조). 이는 특정 강의 class가 새로운 시간을 배정 받는 시간이동 기법으로 강의 시간표 문제의 scheduling 속성을 반영하는 이웃 해 이동 전략이다.



(a) 강의 시간의 상호 교환 (이웃 해 전략 A)



(b) 새로운 강의 시간의 배정(이웃 해 전략 B)

그림 4 사용된 이웃 해 전략

4.1.3 제어 상수

타부 탐색은 몇 가지 제어 상수들을 조정하여야 한다. 본 실험에서 채택한 반응적 타부 탐색을 위해서는 단기 메모리에서 사용되는 제어상수로서 사이클의 발생 여부 에 따라 타부 리스트 사이즈의 증가와 감소 비율을 결정하는 상수가 필요하며, 또 최대 사이클 상수보다 적은 탐색 경로에 사이클이 발생했을 경우에만 사이클이 발생한 것으로 판단하므로, 사이클 발생을 판단하기 위한 최대 사이클을 제한하는 상수가 필요하다. 증장기 메모리를 위한 제어상수는 특정한 해에 대한 반복 탐색의 최대 허락 제어상수가 필요하며, 이러한 최대 반복 허락 기준을 어기는 경우가 자주 발생할 때 이웃 해 전략을 전환하는 기준이 되는 제어 상수가 필요하다. 마지막으로 타부 탐색의 종료 조건을 위한 제어상수가 필요하다. 본 실험에서는 종료 조건은 최대 반복 횟수로 하였다. 표 2는 본 실험에서 사용된 제어 상수들의 내용과 많은

표 2 제어 상수

제어 상수	실험 수치	내 용
INCREASE	1.1	타부 리스트 크기 증가 비율
DECREASE	0.9	타부 리스트 크기 감소 비율
CYCLE_MAX	50	사이클 판단 기준
REPT	3	반복 탐색 최대 허락 기준
CHAOS	3	이웃 해 전략 전환 기준
MAX_ITERATION	500	최대 반복 횟수

사전실험을 통해 조정한 후 본 실험에 실제 사용된 제어 상수의 값을 보여주고 있다.

4.1.4 실험 환경

제안된 이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색을 기존에 제안된 반응적 타부 탐색과의 성능 비교를 위하여 각각에 대한 알고리즘을 구현하고 비교 실험을 실시하였다. 성능 비교의 객관성을 위해서 동일한 시스템 환경과 같은 데이터, 적합도 함수를 두 가지 알고리즘에 같이 적용하였다. 시스템 환경은 하드웨어 환경으로 Pentium II 300Mhz CPU를 가진 PC를 사용하였고, 소프트웨어 환경으로 OS는 MS Windows 98을, 구현 언어는 Visual C++를, 데이터 입출력용 데이터 베이스는 MS Access를 이용하였다.

적용 데이터는 모 대학교의 특정 단과대학의 5개 학과의 실제 1999학년도 1학기 개설 과목, 강의에 참여한 교수 및 강사, 월요일부터 금요일까지의 1시간 단위(50분 강의, 10분 휴식)의 하루 8시간 강의 시간대를 사용하였다. 적용 데이터에 대한 요약은 표 3에 나타나 있다.

표 3 실험에 사용된 데이터

내용	데이터 크기	설명
개설 과목	85	해당 학기에 개설된 과목 수
강의 class 수	260	개설 과목별 강의시간 수의 총합
학과 수	5	단과 대학내 학과 수
과목 수강 학생 그룹 수	20	학과 수 × 학년 수
담당 교수	28	해당 학기 강의에 참여하는 교수
강의 시간대	40	월-금요일 9:30 ~ 17:30

4.2 실험 II - 외판원 문제(traveling salesman problem : TSP)

4.2.1 문제의 정의

외판원 문제는 외판원이 특정 도시에서 출발하여 n개의 도시를 단 한번씩 반드시 방문하고 다시 출발한 도시로 돌아오는 최소비용의 경로를 찾는 문제이다. 여기서의 비용이란 경로상에서 인접해있는 도시들간의 비용(거리)의 전체 합을 말한다. 본 실험에서는 다음과 같은 비용 계산식을 적용하였다.

$$C = \sum_{r=0}^{n-2} d(c_r, c_{r+1}) + d(c_{n-1}, c_0)$$

C : 외판원이 n개의 도시를 방문하는데 필요한 전체 비용

d(c_r, c_{r+1}) : 외판원이 r번째 도시에서 r+1번째 도시를 방문하는데 필요한 비용

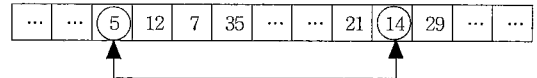
4.2.2 문제 표현 및 이웃 해 생성 방법

본 실험에서 해 개체의 표현은 외판원이 방문하는 도시의 순서를 표현하는 경로표현(path representation)을 사용하였다. 그림 5는 외판원이 “3”이란 도시에서 출발하여 처음으로 “20”이라는 도시를 방문한 후 9라는 이름의 도시에 방문하고 r 번째에는 “c_r”이라는 도시를 방문하고 마지막으로 “29”라는 도시를 방문한 후 다시 도시 “3”으로 돌아오는 경로를 표현한 것이다.

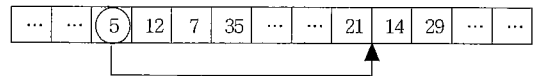
방문순서	0	1	2	...	r	...	n-2	n-1
도시이름	3	20	9	...	c _r	...	24	29

그림 5 해 개체의 표현

본 실험에서는 다음 두 가지 이웃 해 생성 전략을 사용하여 이웃 해 전략 전환의 유용성을 검증하였다. 첫번째 이웃 해 생성 전략은 두 개의 도시를 서로 교환하는 방법을 사용하였다(그림 6 (a) 참조). 두 번째 이웃 해 생성 전략은 특정 도시의 방문 순서를 바꾸는 방법을 이용하였다(그림 6 (b) 참조).



(a) 두 도시의 상호 교환 (이웃 해 전략 A)



(b) 특정 도시의 방문 순서 교체 (이웃 해 전략 B)

그림 6 사용된 이웃 해 전략

4.2.3 제어 상수

본 실험에 실제 사용된 제어 상수의 값을 표 4는 보여주고 있다. 실험 I과 같이 종료 조건은 최대 반복 횟수로 하였다.

표 4 제어 상수

제어 상수	실험 수치	내용
INCREASE	1.1	타부 리스트 크기 증가 비율
DECREASE	0.9	타부 리스트 크기 감소 비율
CYCLE_MAX	50	사이클 판단 기준
REPT	3	반복 탐색 최대 허락 기준
CHAOS	3	이웃 해 전략 전환 기준
MAX_ITERATION	1000	최대 반복 횟수

4.2.4 실험 환경

본 실험의 기본 환경은 실험 I과 동일하며 실험에 사용된 데이터는 방문해야 할 전체 도시 수는 100개로 하였으며 도시들간의 거리는 1부터 500사이의 임의의 정수가 되도록 생성하였으며 실험의 정확성을 위하여 5회 모두 서로 다른 데이터를 이용하였다.

4.3 실험 결과

비교 실험은 무질서한 유인을 탈출 메커니즘으로 사용하는 반응적 타부 탐색(CA RTS)과 이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색(NSSM RTS)에 대한 비교를 수행하였다. CA RTS는 무질서한 유인을 탈출 메커니즘으로 사용하는 반응적 타부 탐색 방법으로 [6]의 연구에서 QAP(Quadratic Assignment Problem)에 적용된 적이 있는 알고리즘을 실험 문제에 변형 적용하였다. 두 알고리즘 모두 동일한 적합도 함수(비용계산 식)를 사용하였고, 초기 해는 임의의 초기 해를 이용하였으며, 비교분석의 정확성을 위하여 5회 실험을 통한 평균값을 이용하였다. 그림 7, 8은 반복 수

증가에 따라 각 알고리즘이 그때까지 찾아 낸 가장 좋은 해의 최대 적합도 또는 최소 비용의 5회 실험 평균 값 변화를 나타낸다. 실험 I에서는 각 알고리즘은 모두 반복 수 300 이내에 최고 적합도의 변화가 거의 나타나지 않으므로, 500 반복 수의 제한을 두었으며, 실험 II에서는 두 알고리즘은 모두 반복 수 750회 이후에 최저 비용의 변화가 거의 나타나지 않으므로, 1000회 반복 수의 제한을 둔 것은 충분한 수치임을 알 수 있다. 그림 7에서 CA RTS는 500회의 반복 수에서 최대 적합도 평균이 0.8168582인 강의 시간표를, NSSM RTS는 500회의 반복 수에서 최대 적합도 평균이 0.9025982인 강의 시간표를 찾았다. 그림 8에서 CA RTS는 1000회의 반복 수에서 최소 비용 평균이 3359 인 경로를, NSSM RTS는 1000회의 반복 수에서 최소 비용 평균이 2777 인 경로를 찾았다. 즉, 두 실험 모두에서 NSSM RTS가 CA RTS에 비해 좋은 결과를 보이고 있다. 실험 I에서 각 알고리즘의 결과를 구할 때까지 걸린 시간의 평균은 CA RTS 11분 45초, NSSM RTS는 10분 19초

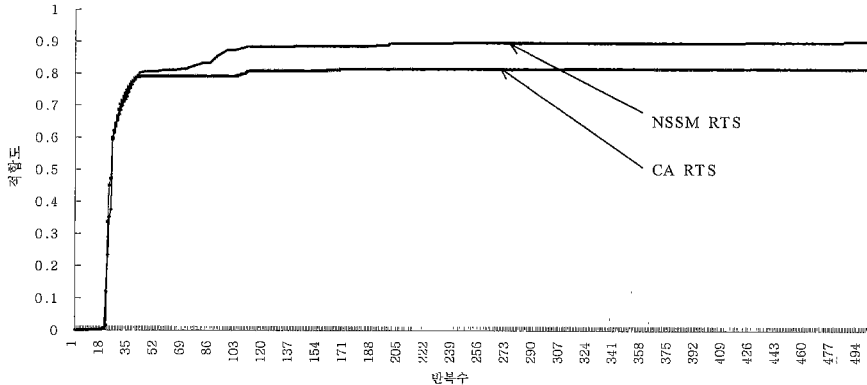


그림 7 최대 적합도 변화 추이 실험 I

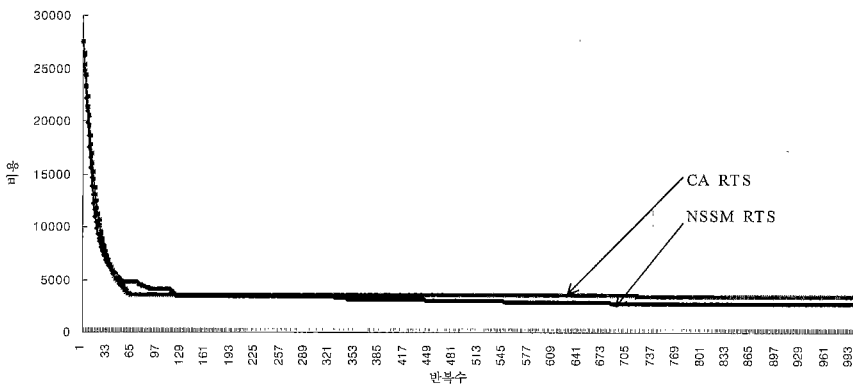


그림 8 최소 비용 변화 추이 실험 II

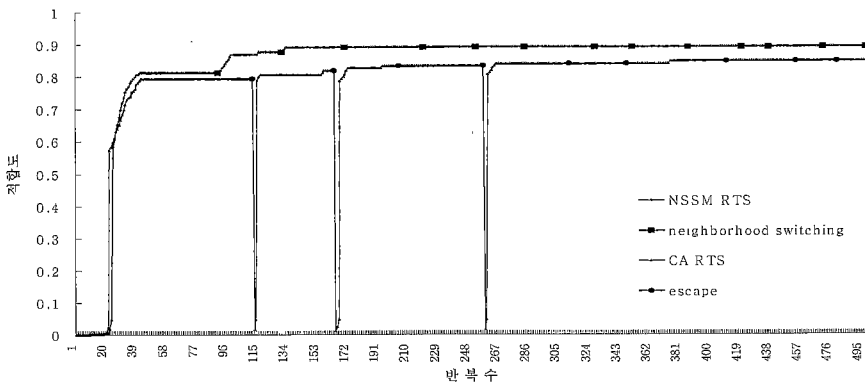


그림 9 CA RTS와 NSSM RTS의 탐색과정에서의 적합도 변화 실험 I

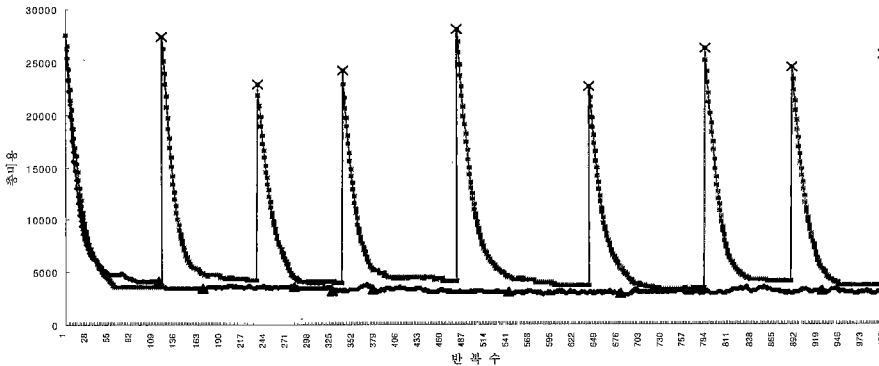


그림 10 CA RTS와 NSSM RTS의 탐색과정에서의 비용 변화 실험 II

이었으며, 실험 II에서 걸린 시간의 평균은 CA RTS 1분 48초, NSSM RTS는 1분 43초이었다. 즉 수행시간 면에서도 CA RTS에 비해서 NSSM RTS가 더 적은 시간이 걸리는 것을 알 수 있다.

NSSM RTS가 CA RTS보다 좋은 성능을 나타내는 이유, 즉 이웃 해 전략 전환의 유용성을 밝히기 위해 그림 9, 10을 통해 CA RTS와 NSSM RTS의 탐색 과정을 좀 더 자세히 살펴보았다. 그림 9, 10은 하나의 동일한 초기 해에서 알고리즘이 진행되는 동안 적합도(비용)의 변화를 나타낸다.

그림에서 나타나듯이 CA RTS는 새로운 공간으로의 탈출 메커니즘이 수행된 후 대부분의 경우에 매우 나쁜 해로부터의 탐색부터 시작된다. 즉 탈출 이후에는 다른 공간에서의 탐색만이 이루어지는 것이지 지금까지 찾은 좋은 해의 정보는 사용되지 않으므로, 더 이상의 개선에 대한 보장은 없는 것이다. 반면에 NSSM RTS는 새로운 공간으로의 단순한 탈출 이외에도 현재의 이웃 해

공간내의 가장 좋은 해(현재의 이웃 해 공간에서는 지역 최적해)에서 탈출하고, 새로운 이웃 해 전략을 사용하므로 탈출 지점인 기존의 지역 최적해가 새로운 이웃 해 공간에서는 더 이상 지역 최적해가 아닐 수 있으므로, 더 좋은 이웃 해를 가질 가능성이 증대한다. 따라서 제안한 NSSM RTS는 기존의 phase에서의 최대 적합도를 가진 해를 탈출 후 추가로 개선 시킬 가능성이 큰 반면, CA RTS는 탈출을 통해 새로운 해 공간을 탐색은 하되 급격히 나빠진 해에서 단기 메모리를 이용한 개선에 의존해야 하는 약점이 있다.

6. 결론 및 추후 연구 과제

본 연구는 조합 최적화 문제를 위한 반응적 타부 탐색에 이웃 해 전략 전환 이라는 새로운 개념의 반응적 타부 탐색을 개발하고, 대학 강의 시간표 작성 문제와 TSP에서 기존의 무질서한 유인을 사용한 반응적 타부 탐색 알고리즘보다 더 짧은 시간에 더욱 좋은 해를 찾

을 수 있음을 보였다. 본 연구의 탈출 메커니즘은 증장기 메모리를 이용해 이웃 해 전략을 바꾸는 메커니즘을 사용해 좋은 해를 갖는 다른 탐색 공간으로의 효율적인 탈출을 가능하게 하였다. 특히 이러한 전환 메커니즘은 탐색공간이 매우 넓은 문제에 대해서 기존의 임의의 다른 탐색공간으로 탈출을 사용하는 CA RTS 보다 매우 효과적임을 밝혔다.

타부 탐색은 거의 모든 조합 최적해 문제에 적용이 가능하다. 따라서 본 연구에서 제시한 이웃 해 전략 전환 메커니즘을 이용한 반응적 타부 탐색이 본 논문에서 적용한 문제 뿐 아니라 다른 많은 문제에 대해서 좋은 결과를 얻어내리라 기대한다. 특히 이웃 해 전략 전환 메커니즘은 제약 조건이 매우 상이한 탐색공간을 갖게 하는 문제에 대해서 효율적인 것으로 믿어진다. 향후에는 이러한 다양한 이웃 해 전략을 가진 다른 문제들에 대해서 다양한 이웃 해 전략의 개발, 응용, 분석 등에 관한 연구가 필요하다.

참 고 문 헌

[1] Glover, F., "Tabu Search, Part I," *ORSA Journal on Computing*, Vol.1, No.3, pp.190-206, 1989.

[2] Glover, F., "Tabu Search, Part II," *ORSA Journal on Computing*, Vol.2, No.1, pp.4-32, 1990.

[3] Hansen, F. and B. Jaumard, "Algorithms for the Maximum Satisfiability Problem," *Computation*, Vol.44, pp.279-303, 1990.

[4] 김여근, 윤복식, 이상복, 메타 휴리스틱, p.452, 영지문화사, 서울, 1999.

[5] Skorin-Kapov, J., "Tabu Search Applied to the Quadratic Assignment Problem," *ORSA Journal on Computing*, Vol.2, No.1, pp.33-45, 1990.

[6] Battiti, R. and G. "Tecchiolli, The Reactive Tabu Search," *ORSA Journal on Computing*, Vol.6, No.2, pp.126-140, 1994.

[7] Burke, E.K., D.G. Elliman, and R.F. "Weare, A University Timetabling System Based on Graph Colouring and Constraint Manipulation," *Journal of Research on Computation in Education*, Vol.27, Issue 1, pp.1-18, 1994.

[8] Colnri, A., M. Dorigo, and V. Maniezzo, "Genetic Algorithms and Highly Constrained Problem: The Time-table Case," *Parallel Problem Solving From Nature*, pp.55-59, 1990.

[9] Ross, P., D. Corne, and H.L. Fang, "Successful Lecture Timetabling with Evolutionary Algorithm," *DAI Research Paper*, No.707, 1994.

[10] Melcio, F., P. Caldeira, and A. Rosa, "Solving the Timetabling Problem with Simulated Annealing,"

First Int Conf on Enterprise Information Systems ICEIS' 99, pp.272-279, 1999.



김 재 호
1996년 한국외국어대학교 전자기산학(학사). 2000년 한국외국어대학교 컴퓨터공학과(석사). 2000년 ~ 현재 전자부품연구원(KETI) 고속네트워크연구센터 연구원. 관심분야는 메타 휴리스틱, Wireless Mobile Network



이 희 상
1983년 서울대학교 산업공학(학사). 1985년 서울대학교 산업공학(석사). 1991년 미국 Georgia Institute of Technology school of Industrial and Systems Engineering(박사). 1991년 ~ 1995년 한국통신 통신망 연구소. 1995년 ~ 현재 한국외국어대학교 산업정보시스템공학부 부교수. 관심분야는 정수 계획법, 최적화 이론, 통신 네트워크 및 시스템 최적화



한 현 구
1980년 서강대학교 수학과(학사). 1986년 University of South Florida(석사). 1990년 Auburn University(박사). 1992년 한국통신 멀티미디어 연구소. 1992년 ~ 현재 한국외국어대학교 컴퓨터공학과 교수. 관심분야는 Intelligent Software Agent Systems, Planning Systems, Expert Systems