

# 국지적 탐색의 효율향상을 위한 확률적 여과 기법

## (A Probabilistic Filtering Technique for Improving the Efficiency of Local Search)

강 병 호 <sup>†</sup>      류 광 렬 <sup>\*\*</sup>  
(Byoung-ho Kang)      (Kwangryel Ryu)

**요약** 국지적 탐색 알고리즘들은 최적해를 찾기 위해서 이웃해를 생성하여 평가한 뒤에 좋은 해로 이동하는 과정을 반복한다. 본 논문에서는 생성된 이웃해를 원래의 목적함수로 평가하기 전에 간단한 예비 평가 휴리스틱을 이용하여 미리 평가함으로써, 좋지 않아 보이는 이웃해를 확률적으로 여과할 수 있는 기법을 소개한다. 이 확률적 여과 기법은 결국에 버려질 이웃해를 엄밀하게 평가하는데 낭비되는 시간을 절약하고, 이 시간 동안 보다 좋아 보이는 이웃해를 더 많이 탐색할 수 있게 함으로써 탐색 효율을 높이는 기법이다. 대규모의 실 세계 최적화 문제인 교통망에서의 교통 신호 최적화 문제와 작업 일정 계획에서의 부하평준화 문제를 대상으로 한 실험에서 확률적 여과를 적용한 경우가 적용하지 않은 경우에 비해 주어진 탐색시간 동안 더 좋은 질의 최적해를 얻을 수 있는 것으로 확인되었다.

**키워드** : 이웃해 생성, 국지적 탐색, 최적화

**Abstract** Local search algorithms start from a certain candidate solution and probe its neighborhood to find ones with improved quality. This paper proposes a method of probabilistically filtering out bad-looking neighbors based on a simple low-cost preliminary evaluation heuristics. The probabilistic filtering enables us to save time wasted on fully evaluating those solutions that will eventually be trashed, and thus improves the search efficiency by allowing us to spend more time on examining better looking solutions. Experiments with two large-scaled real-world problems, which are a traffic signal control problem in traffic network and a load balancing problem in production scheduling, have shown that the proposed method finds better quality solutions, given the same amount of CPU time.

**Key words** : Neighbor generation, local search, optimization

### 1. 서론

Simulated annealing, hill climbing, tabu search와 같은 휴리스틱 탐색 알고리즘들은 특정 해에서 시작하여 보다 좋은 해를 찾기 위해 이웃해를 조사하며 이동하는 과정을 반복한다. 이 탐색 알고리즘들의 수행 시간은 총 평가 수, 즉 매 반복에서 생성되어 평가된 이웃해들의 총 수에 비례한다. 그런데 탐색 과정에서 평가 후 선택되지 못하고 버려지는 이웃해의 경우에는 그 평가에 소요되는 시간이 낭비된다는 측면이 있다. 특히 평가

비용이 높은 복잡한 최적화 문제의 경우 이렇게 낭비되는 시간이 상대적으로 커져 주어진 시간 내에 충분한 수의 해를 탐색할 수 없게 되고, 결과적으로 좋은 해를 찾지 못하게 된다. 본 논문에서는 이런 점에 착안하여, 주어진 목적함수를 이용한 본 평가 이전에 간단한 예비 평가 휴리스틱을 적용하여 좋아 보이지 않는 이웃해를 여과함으로써 결국에 버려질 이웃해를 엄밀하게 평가하는데 낭비되는 시간을 절약하고, 이 시간 동안 보다 좋아 보이는 해를 더 많이 탐색할 수 있게 함으로써 탐색 효율을 높일 수 있는 기법을 제안한다.

본 연구에서 사용하는 예비 평가 휴리스틱은 대상 문제에 관한 지식을 바탕으로 원래의 목적함수를 완화(relax)하여 만들어지는 것으로, 원천적으로 부정확성을 내포하고 있다. 따라서 이 예비 평가 휴리스틱 값에 근거한 이웃해의 여과는 다음과 같이 확률적으로 이루어진다. 먼저, 하나의 이웃해 후보<sup>1)</sup>를 무작위로 생성한 후

· 본 연구는 해양수산부의 '지능형 항만물류 시스템 기술개발' 사업의 일환으로 수행되었습니다.

<sup>†</sup> 정 회 원 : 부산대학교 컴퓨터공학과  
bhokang@pusan.ac.kr

<sup>\*\*</sup> 종신회원 : 부산대학교 컴퓨터공학과 교수  
krryu@pusan.ac.kr

논문접수 : 2006년 6월 2일

심사완료 : 2006년 11월 24일

예비 평가 휴리스틱을 적용하여 그 평가 값  $h$ 를 구한다.  $h$ 가 취하는 값의 구간이  $[a, b]$ 라 하면 이를  $[0, 1]$ 의 구간으로 사상(mapping)한 값  $p$ 를 구한 다음,  $p$ 의 확률로 이 후보를 이웃해로 선택한다. 그런데, 이웃해 후보들에 대한 예비 평가 값  $h$ 가 특정 값 주변으로 편중되어 분포할 경우에는 이 값으로부터 사상된 여과 확률  $p$  역시 비슷한 분포를 가져서 여과 과정의 변별력이 상실될 수 있다. 따라서 본 연구에서는 이웃해 후보들을 잘 변별할 수 있도록 예비 평가 값의 분포를 조정해 주는 스케일링 함수를 고안하여 적용한다. 스케일링 함수는  $p$ 의 값이  $[0, 1]$ 의 구간에 균일(uniform)하게 분포되도록 준비해 주는 기능을 한다. 그러나, 예비 평가 휴리스틱의 신뢰도가 상당히 높거나 아니면 낮은 것으로 판단될 경우에는 의도적으로  $p$ 의 분포를 불균일하게 함으로써 탐색이 그리디(greedy)한 방향 혹은 무작위적인(random) 방향으로 진행되도록 유도할 필요가 있다. 이를 위해 본 연구에서는  $p$ 의 분포를 의도적으로 조정하는 bias 함수를 추가로 사용한다. 대상 탐색 알고리즘이 hill climbing이나 tabu search일 경우에는 미리 정해진 수의 이웃해가 모일 때까지 이상의 이웃해 후보들에 대한 여과 과정을 반복하며, simulated annealing의 경우에는 하나의 이웃해가 확보될 때까지만 한다.

제안 기법의 효과를 검증하기 위해서 대규모 실 세계 최적화 문제인 교통망에서의 교통 신호 체계 최적화 문제와 작업 일정 계획에서의 부하 평준화 문제를 대상으로 실험한 결과, 확률적 여과 기법을 적용한 경우가 적용하지 않은 경우에 비해 주어진 탐색시간 동안 보다 우수한 해를 얻을 수 있는 것으로 확인되었다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴보고, 3장에서는 본 논문에서 제안하는 확률적 여과 기법을 보다 상세히 소개한다. 4장에서는 대상 문제들 각각에 적용한 실험결과를 보여주고 분석한다. 마지막으로 5장에서는 결론 및 향후 연구과제를 기술한다.

## 2. 관련 연구

지금까지 탐색 기법에서 확률적 결정은 탐색 성능을 향상시키기 위해서 주로 탐색을 다각화하기 위한 수단으로 도입되었다. 국지적 탐색에서 이동할 이웃해를 확률적으로 결정하는 대표적인 예로 stochastic hill climbing과 simulated annealing을 들 수 있다. Stochastic hill climbing [1]은 이웃해들을 목적함수로 평가한 후, 그 결과값으로부터 확률을 구하여 무작위 추출을 통해 이동할 이웃해를 선택한다. 이 알고리즘은 항상 가장 좋

은 이웃해로 이동하는 simple hill climbing에 비해 탐색 초기에 수렴하는 속도가 늦지만, 확률적 선택을 통해 탐색을 다각화 함으로써 보다 좋은 해를 찾는다. Simulated annealing[2, 3]은 한 개의 이웃해를 무작위로 생성하여 목적함수로 평가한 후, 현재해보다 좋으면 이동하고, 좋지 않으면 물리적 어닐링으로부터 별도로 마련된 함수( $e^{AE/T}$ )에 의해 확률적으로 이동한다. 이 알고리즘은 온도  $T$ 를 통해 탐색의 다각화를 조정하는데, 탐색 초반에는 온도  $T$ 를 크게 하여 좋지 않아도 이동할 수 있는 확률을 높여서 탐색이 다양하게 진행되고, 탐색 후반으로 갈수록 온도  $T$ 를 낮춰줌으로써 좋지 않았을 때는 이동할 확률을 줄여서 탐색이 그리디하게 진행된다. 이들 기법들은 목적함수의 본 평가 후, 그 결과 값을 바탕으로 탐색을 다각화한 방법이다.

이웃해를 생성하는 단계에서 확률적인 결정을 도입한 연구도 있었는데, 대표적인 예가 mixed random walk strategy[4]이다. 이 기법은 사전에 이웃해를 무작위로 생성할 확률( $q$ )과 그리디로 생성할 확률( $1-q$ )을 정해두고, 탐색의 매 반복에서 이웃해를 생성할 때 어떤 방법을 사용할 지 무작위 추출을 통해 결정한다. 이 기법은 사전에 정해진 일정 확률  $q$ 의 값을 조정하여 탐색의 다각화를 조정하는데, 확률  $q$ 의 값을 크게 하면 이웃해를 무작위로 생성할 확률이 높아져 다양한 이웃해가 생성되고, 반대로  $q$ 의 값을 작게 하면 탐색이 보다 그리디하게 진행된다. 이 기법은 그리디 탐색 기법인 GSAT[5]으로 SAT을 해결할 때 자주 발생하는 지역적 최적해(plateaux)를 벗어나기 위한 수단으로 적용되었다[6].

국지적 탐색 기법뿐만 아니라 구성적 휴리스틱 알고리즘에서도 부분해를 확장할 때 확률적 결정을 도입하여 탐색을 다각화한 연구들도 있었는데, 대표적인 방법으로 HBSS(Heuristic-Biased Stochastic Sampling) [7]이 있다. 이 알고리즘은 완전해를 구하기 위해 부분해를 확장할 때, 부분해 평가 값을 기준으로 확률적 선별을 통해 확장할 부분해를 결정하는데, 원하는 해를 구할 때까지 완전해를 구하는 과정을 반복한다. HBSS에서 부분해를 확장할 때는 확장 가능한 모든 부분해에 대해서 부분해 평가 휴리스틱으로 평가한 뒤에, 그 평가 값을 기준으로 순위를 정한다. 그리고 이 순위에 따라 확률을 부여하고 무작위 추출을 통해 확장될 변수를 결정한다. 이후 Smith는 HBSS의 일부 단점을 보완하여 WHISTLING(Wasp behavior Inspired Stochastic sampling[8])을 제안했다. Smith는 HBSS와 같이 순위에 따라 고정적인 확률을 사용할 때 부분해 평가 휴리스틱이 제공하는 후보들간의 선호 정도를 활용할 수 없다는 문제점을 제기하고, 순위 대신 평가값으로부터 확률을 구하는 방식을 사용하였다.

1) 이하에서는 현재 해로부터 생성은 되었으나 여과 여부가 결정되지 않은 이웃해를 이웃해 후보라고 하고, 확률적 여과를 통과하여 선택된 해를 이웃해라고 부른다.

이들 연구에서는 탐색을 그리디하게 진행하는 것 보다 다각화시키는 것이 더 좋은 해를 구할 수 있다는 점에 착안하여 탐색을 다각화하는 수단으로 확률적 결정을 도입하였다. 반면, 본 논문에서는 목적함수에 의한 본 평가 전에 예비 평가를 통해 이웃해 생성 단계에서 좋지 않아 보이는 이웃해를 여과함으로써 좋아 보이는 해를 더 많이 탐색할 수 있도록 하는데 초점을 두고 있다. 이와 유사한 연구로서 [9, 10]에서는 이웃해 생성을 할 때, 예비 평가 휴리스틱의 평가 값을 바탕으로 확률적 선택을 하는 것이 무작위 또는 그리디하게 선택하는 것보다 부하 평준화 문제를 대상으로 탐색 성능이 좋다는 것을 보여준다. 본 논문에서는 [9, 10]에서 제안된 확률적 선택에 의한 이웃해 생성 방법을 일반화하여 예비 평가 휴리스틱의 평가 값을 스케일링하여 좋지 않은 이웃해 후보를 보다 잘 변별할 수 있고, 확률  $p$ 의 분포를 인위적으로 조정하는데 보다 적합한 bias 함수를 마련한 확률적 여과 기법을 제안한다.

### 3. 확률적 여과 기법

본 논문에서 제안하는 확률적 여과 기법의 알고리즘은 그림 1과 같다.

```

Algorithm ProbabilisticFiltering (current, n,  $\rho$ )
    inputs: current, current solution
              n, the number of neighbors to be generated
               $\rho$ , bias factor
    outputs: S, set of neighbors selected
    variables: c, a candidate solution generated
                 h, preliminary evaluation value ( $0 \leq v \leq 1$ )

    count  $\leftarrow$  0, S  $\leftarrow$  {}
    while count < numberneighbor, Do
        c  $\leftarrow$  a candidate solution generated by simple random sampling
        h  $\leftarrow$  PreliminaryEvaluation(c)
        p  $\leftarrow$  Scaling&Truncation(h)
        p  $\leftarrow$  Bias( $\rho$ , p)
        if c is selected by random sampling with respected to its probability p
            S  $\leftarrow$  S  $\cup$  {c}, count  $\leftarrow$  count + 1
        else c is filtered
    end while
    return S
    
```

그림 1 확률적 여과 기법 알고리즘

이 알고리즘은  $n$ 개의 이웃해를 생성할 때까지 다음의 과정을 반복한다. 우선 현재 해로부터 하나의 이웃해 후보  $c$ 를 무작위로 생성하고, 후보  $c$ 가 얼마나 좋은지를 예비 평가 함수(PreliminaryEvaluation())로 간단히 평가하여 예비 평가 값  $h$ 를 구한다. 그리고  $h$ 를  $[0, 1]$ 의 구간으로 사상하되, 생성될 수 있는 이웃해 후보들의 예비 평가 값들이 0과 1사이에 균일하게 분포되어 여과

과정에서 이웃해 후보를 잘 변별할 수 있도록 하기 위해 스케일링(Scaling&Truncation())한다. 이웃해 후보  $c$ 의 예비 평가 값  $h$ 로부터 스케일링 된 평가 값  $p$ 를 구한 뒤에는 bias 함수(Bias())를 사용하여 탐색이 무작위 또는 그리디 방향으로 진행될 수 있도록 선택 확률  $p$ 의 분포를 조정한다. 마지막 단계에서는 확률  $p$ 에 기초하여 무작위 추출을 통해 이웃해 후보  $c$ 를 이웃해로 선택할 지 또는 여과할 지 결정한다.

[9, 10]에서 제안된 확률적 여과 기법에서는 예비 평가 휴리스틱에서 평가 값으로 단순히 0과 1사이의 값을 반환하고, 이 값 자체를 확률적 여과에 사용하였다. 그러나 예비 평가 휴리스틱은 이웃해 후보에 대해 좋은 정도를 간단히 평가만 하기 때문에, 이웃해 후보들에 대한 평가 값의 분포가 특정 값으로 치우치는 경우가 발생될 수 있다. 만약 예비 평가 값이 특정 값으로 치우쳐 있으면, 어떤 이웃해 후보가 좋은지 잘 구별할 수 없게 된다. 예를 들면, 대부분의 이웃해 후보들의 평가 값들이 1 주변에 있고, 극히 소수만 0 주변의 값을 가진다면 이후 확률적 여과 과정에서 대부분의 이웃해 후보들이 선택될 것이고, 반대의 경우에는 대부분 여과될 것이다. 이와 같이 이웃해 후보에 대한 변별력에 대한 문제를 해결하기 위해서는 이웃해 후보들이 0과 1사이에서 균일하게 분포되도록 예비 평가 값을 조정하는 기능이 필요한데, 본 논문에서는 수식 1과 같이 평균과 편차를 사용하여 스케일링과 절단(truncation)을 할 수 있는 스케일링 함수를 마련하였다.

$$p(h) = \begin{cases} 1 & (h \geq \mu + k\sigma) \\ \frac{h - (\mu - k\sigma)}{2k\sigma} & (\mu - k\sigma < h < \mu + k\sigma) \\ 0 & (h \leq \mu - k\sigma) \end{cases} \quad (1)$$

- $h$ : 예비 평가 값 ( $a \leq h \leq b$ )
- $\mu$ : 이웃해 후보들의 예비 평가 값의 평균
- $\sigma$ : 이웃해 후보들의 예비 평가 값의 표준편차
- $k$ : truncation factor

식 (1)의 스케일링 함수는 그림 2의 좌측 그림에서 나타낸 바와 같이 예비 평가 값들의 평균으로부터 일정 편차 내에 있는 평가 값들을 스케일링하고 그 외에 있는 값들은 절단하여 0과 1로 사상한다. 스케일링의 효과에 대한 예는 그림 2의 우측 그림에서 보여준다. 이 예에서 예비 평가 값의 분포는 실선으로 표현되는데 0쪽으로 치우쳐 있다. 이것을 수식 1에 따라 스케일링하면 점선과 같이 0과 1사이에서 이웃해 후보들이 비교적 균일하게 분포하게 된다. 이웃해 후보들의 예비 평가 값의 평균  $\mu$ 와 표준 편차  $\sigma$ 를 구하기 위해서는 탐색의 매 반복에서 모든 이웃해 후보들에 대해서 구해야 하나, 실제 실험에서는 일정 수의 후보들에 대해서 구한 값들을 사

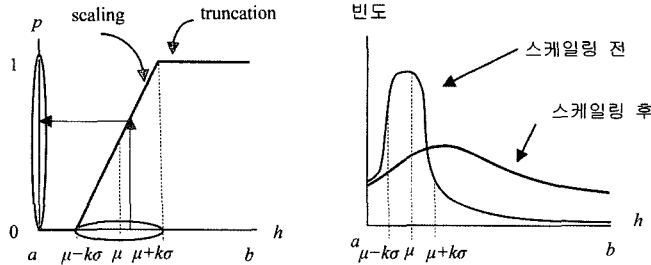


그림 2 스케일링 함수와 그 효과에 대한 예

용하였다.

Truncation factor  $k$ 에 대한 적절한 값을 구하기 위해서 본 논문에서는 우선 초기해를 대상으로 이웃해를 다수 생성해서 예비 평가를 한 다음, 평가 값을 0과 1사이에서 10등분하여 스케일링 했을 때 각 등분의 빈도에 대한 편차가 적도록  $k$ 값을 조정하여 찾았다. 정확한  $k$  값을 찾기 위해서는 탐색의 때 반복에서 이와 같은 과정을 해야 하지만, 본 논문의 실험에서는 초기해에 대해 구한  $k$ 값을 탐색 전반에 사용하였다.  $k$ 값의 조정 범위와 조정 값은 다음과 같이 대략 결정하였다. 이웃해 예비 평가값의 분포가 표준 정규 분포라고 가정하면  $k$ 가 1.0과 2.0일 때, 절단에 의해 예비 평가 값이 0과 1로 되는 부분은 각각 약 31.7%와 4.6%이다. 본 논문에서는 실험에서 1.0과 2.0 사이에서 0.1씩 조정하면서 각 등분의 편차가 적은 지점을 찾도록 하였다.

위 식 (1)과 같이 스케일링하면 예비 평가 값  $h$ 가 매우 좋지 않은 이웃해 후보의  $p$ 는 0으로 사상되고, 이런 후보들은 확률적 여과 과정에서 무조건 여과된다. 본 논문에서는 부정확한 예비 평가 휴리스틱의 평가에 의해 좋지 않게 평가된 이웃해 후보를 무조건 여과하는 것을 방지하기 위해서 식 (2)와 같이 Laplace smoothing constant  $\tau$ 를 도입하여 정규화하는 과정을 추가하였다.

$$p \leftarrow \frac{p + \tau}{1 + \tau} \quad (2)$$

$p$ : 이웃해로 선택할 확률  
 $\tau$ : 상수 ( $0 < \tau < 1$ )

이와 같이 예비 평가와 스케일링을 통해 무작위로 선택된 이웃해 후보에 대한 평가 값으로부터 (0, 1]로 사상된  $p$ 를 구할 수 있는데, 간단히 이 값을 이웃해로 선택할 확률  $p$ 로 보고, 여과할 확률을  $1-p$ 로 둘 수 있다. 그러나 이렇게 하여 이웃해를 여과하면 탐색의 때 반복에서 예비 평가 휴리스틱에 의해 좋게 평가된 이웃해들을 위주로 생성하게 된다. 확률적 여과를 하더라도 예비 평가 휴리스틱은 탐색의 전 과정에서 탐색을 안내하는 데 영향을 주기 때문이다. 따라서 해의 공간에서 탐색을

진행할 때, 예비 평가 휴리스틱의 안내를 어느 정도 따라야 좋을지에 대한 문제가 있다. 만약 탐색 과정에서 예비 평가 휴리스틱에 너무 의존하여 그리디하게 탐색이 진행되면(more greedy) 지역 최적해에 빠져 실제 최적해를 획득할 가능성은 오히려 떨어질 수 있고, 반대로 예비 평가 휴리스틱을 완전히 무시하여 무작위로 이웃해를 생성하면(less greedy) 이 휴리스틱에 의해 높일 수 있는 탐색의 효율은 기대하기 힘들기 때문이다. [9, 10]에서는 이런 점을 파악하여 예비 평가 함수의 분포를 인위적으로 조정하기 위한 수단으로 그림 3의 (a)와 같은 bias 함수를 제안한 바 있다. 그러나 이 bias 함수에서는 bias가 커질 경우(less greedy) 이웃해 후보에 대한 분별력을 크게 잃어버리는 문제점이 있다. 본 논문에서는 이 점을 해결하기 위해서 식 (3)과 같이 보다 개선된 bias 함수를 제안하였다.

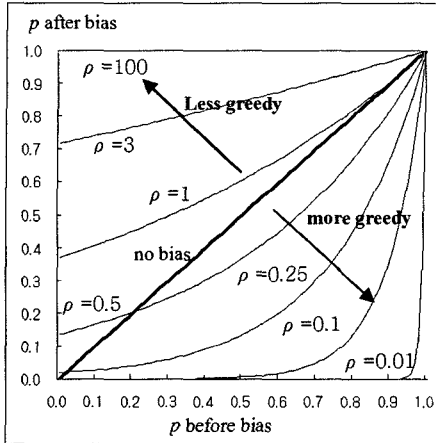
$$Bias(\rho, p) = \begin{cases} (1 - (1 - p)^\rho)^{\frac{1}{\rho}} & (\rho \geq 1) \\ 1 - (1 - p^{-\rho})^{-\frac{1}{\rho}} & (\rho < -1) \end{cases} \quad (3)$$

$p$ : 이웃해로 선택할 확률 ( $1-p$ : 여과 확률)  
 $\rho$ : bias ( $|\rho| \geq 1$ )

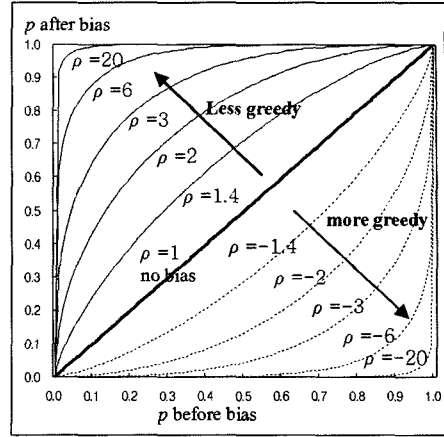
그림 3의 (b)와 같이 본 논문에서 제안된 bias 함수에서 bias  $\rho$ 가 1일 때는 bias를 주지 않은 경우로 예비 평가 값으로부터 스케일링하여 구한  $p$ 를 그대로 확률로 사용한다. Bias  $\rho$ 를 크게 하면, 대부분의 이웃해 후보들의 선택 확률  $p$ 도 커져 무작위로 선택된 이웃해 후보를 이웃해로 선택할 가능성이 커진다(less greedy). 그러나 이전 bias 함수와는 다르게 매우 좋지 않은 이웃해 후보들은 선택될 가능성은 여전히 낮다. 반대로 bias  $\rho$ 를 작게 하면, 이웃해 후보들 중에서 점차 예비 평가 값이 높은 후보들만 선택하게 된다(more greedy). Bias  $\rho$ 에 대한 적절한 값은 사전에 알기 힘들기 때문에 실험으로 결정된다.

#### 4. 실험 및 분석

확률적 여과 기법의 성능을 검증하기 위하여 본 논문



(a) 이전 bias 함수 ( $e^{-(1-p)/\rho}$ )



(b) 제안된 bias 함수

그림 3 이전 bias 함수와 제안된 bias 함수

에서는 실 세계 최적화 문제로 교통망에서의 교통 신호 최적화 문제와 작업 일정 계획에서의 부하 평준화 문제에 대해서 실험하였다.

4.1 교통 신호 최적화 문제

교통 신호 최적화 문제는 대상 교통망에서 원활한 교통 소통을 위해 각 차량들이 겪는 지체(delay)의 합을 최소화할 수 있는 신호 체계를 수립하는 문제로 정리된다[11]. 지체는 특정 도로를 이 도로의 설계 기준 속도로 통과하지 못할 때 발생되는데, 이 속도로 해당 도로를 통과할 때 걸리는 시간보다 더 발생되는 지연 시간을 지체라고 한다[12, 13]. 교통망에서 지체는 모든 차량들이 겪는 지체의 합으로 표현된다. 교통망에서 지체를 최소화하기 위해서는 차량의 흐름에 따라 각 교차로에서의 신호 길이와 주기 그리고 오프셋이 적절히 조정되어야 한다[11]. 그러나 변화되는 교통류에 대응해서 이들에 대한 적합한 값을 구하는 것은 교통망의 특성상 매우 어려운 문제이다. 교통망에서 특정 도로의 교통류를 개선하기 위해 어느 한 신호를 조정할 경우, 그와 연결된 다른 도로로 신호조정의 영향이 연쇄적으로 파급되어 모든 도로의 교통류에 영향을 주기 때문이다.

본 논문에서 해는 최적화 대상 시간 동안 가능할 수 있는 신호조정안으로서 모든 교차로에서의 신호 길이의 조합으로 보았다. 이웃해는 현재 해, 즉 현재 신호 계획에서 두 신호 사이의 경계 지점을 선택하는 단계와 선택된 경계 지점을 조정하는 단계를 거쳐 새로운 신호 계획으로 생성된다. 예를 들어 두 신호 A, B 사이의 경계 지점은 신호 A가 끝나는 시점이고, 신호 B가 시작되는 시점을 말한다. 이 경계 지점은 두 신호 A와 B의 신호길이에 대한 제약은 만족하는 범위 내에서 앞 또는 뒤로 조정함으로써 신호 A와 신호 B의 길이를 동시에

변화시킬 수 있다. 이렇게 두 단계로 구분한 것은 해의 공간을 분할함으로써 줄이는 효과를 기대할 수 있기 때문이다. 또한 각 단계에 유리한 휴리스틱을 잘 고안할 수 있다는 부가적인 장점도 있다. 보통 복잡한 최적화 문제를 탐색으로 해결할 때, 이와 같이 해의 공간을 분할하는 방식을 취하는데 이런 기법을 successive filtration strategy[14]라고 한다. 목적함수의 평가 값으로는 대상 교통망에서 최적화 대상 시간 동안 도로에 있거나 통과한 모든 차량들의 총 지체의 합을 사용하였다. 이것을 보다 정확히 파악할 수 있도록 본 논문에서는 차량 단위의 교통류 시뮬레이터를 만들어서 본 평가 함수로 사용하였다.

4.1.1 확률적 여과 기법의 적용

이웃해 생성 과정에서 확률적 여과 기법을 적용한 곳은 두 신호 사이의 경계 지점을 선택하는 단계에서만 적용하였다. 경계 지점의 조정은 어떤 값이 좋은지 평가하기가 매우 어렵기 때문에, 간단히 조정 범위 내에서 무작위로 결정하였다. 본 논문에서는 인접한 두 신호에서 발생하는 지체의 차이가 큰 경계 지점을 우선적으로 조정하는 것이 지체를 줄이는데 효과적이라 판단하여, 경계 지점의 예비 평가 값으로 인접한 두 신호에서의 지체 차이를 예비 평가 값으로 사용하였다. 각 신호별로 발생하는 지체를 정확히 구하기 위해서는 개별 차량 단위로 정교한 시뮬레이션을 통해 파악해야 하나, 이를 완화하여 한 신호가 끝났을 때 이 신호가 있는 교차로로 유입되는 차로 당 잔류 차량 수를 이 신호의 지체로 사용하였다. 따라서 경계 지점에 대한 예비 평가 휴리스틱은 이 경계 지점과 인접한 두 신호에 대해서 신호가 끝난 시점의 유입 차로 당 잔류 차량수의 차이를 구하는 함수로 표현된다.

4.1.2 교통 신호 최적화 문제에 대한 실험 및 분석

본 논문에서 실험한 대상 문제는 그림 4와 같이 8개의 교차로로 구성된 교통망에서 15분 동안 지체를 최소화할 수 있도록 최적의 신호길이와 음셋을 구하는 문제<sup>2)</sup>이다. 각 교차로에서의 신호와 교통량 데이터는 대상 교통망으로부터 수집된 데이터를 사용하였다.

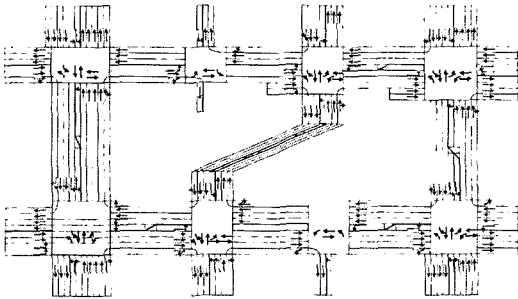


그림 4 대상 교통망

대상 문제에 대한 실험 결과로 simulated annealing에 확률적 여과를 적용하지 않은 경우와 적용한 경우에 대한 성능은 표 1과 같다. 각 경우에 대해서 최적해의 평가값은 5회 실험하여 평균한 값으로 교통망의 전체 지체(초)를 나타낸 값이다. 표 1에서 no filtering은 확률적 여과 기법을 적용하지 않고 simulated annealing만 수행한 경우이다. Filtering은 확률적 여과 기법을 적용한 경우인데, optimal bias는 bias factor  $\rho$  값을 바꿔가면서 실험하여 가장 좋았을 때를 나타낸 경우로  $\rho = 4$ 일 때가 가장 좋았다. No bias는 bias를 주지 않은 경우( $\rho = 1$ )이다. Greedy bias는 매우 그리디하게  $\rho$  값을 설정한 경우로  $-20$ 을 사용하였다. 실험결과를 보면 확률적 여과 기법을 적용하지 않은 경우에 비해 적용한 것이 더 좋은 성능을 보여준다. 그리고 확률적 여과 기법을 적용한 경우에서도 bias를 적절히 설정했을 때( $\rho = 4$ )는 bias를 주지 않았을 때보다 성능이 대략 2.4%의 개선<sup>3)</sup>이 있었다. 반면 no bias의 경우에는 0.7%의 개선이 있었다. 마지막에 bias를 매우 그리디하게( $\rho = -20$ ) 주면 성능이 매우 좋지 않음을 볼 수 있다. 각 경우에 대해 탐색의 (5회 평균) 추이는 그림 5와 같다. Bias를 적절하게 준  $\rho = 4$ , no bias, no filtering, 매우 그리디한  $\rho = -20$  순으로 탐색이 진행됨을 보여 준다.

표 1 교통 신호 최적화 문제에서 확률적 여과 기법의 성능

이웃해 여과 방법	no filtering	filtering		
		optimal bias ( $\rho = 4$ )	No bias ( $\rho = 1$ )	greedy bias ( $\rho = -20$ )
최적해의 평가값	493,648	481,860 (2.4%)	490,068 (0.7%)	521,415 (-5.6%)

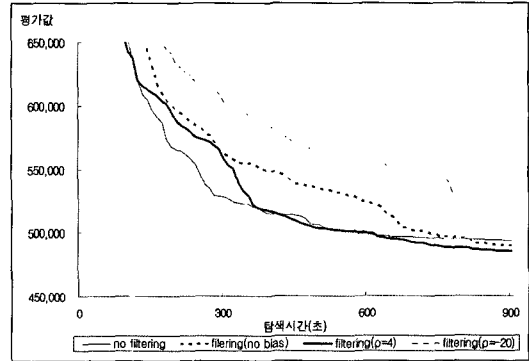


그림 5 교통 신호 최적화 문제에서의 탐색 추이 비교

표 2는 각 경우에 대해 해의 평가 수를 비교한 표이다. 확률적 여과 기법을 적용하지 않았을 때에 비해, 적용한 경우는 추가 비용이 들기 때문에 총 평가 수는 적다. 반면 보다 그리디로 갈수록 여과되는 횟수가 증가함을 볼 수 있다. No bias와 optimal bias에 해당하는  $\rho = 4$ 의 경우에는 평가수가 비슷하고 매우 그리디하게  $\rho$  값을 설정한  $\rho = -20$ 의 경우에는 여과를 너무 많이 하기 때문에 본 평가에 비해 예비 평가를 매우 많이 하고 있음을 나타낸다.

표 2 교통 신호 최적화 문제에서 확률적 여과 기법의 평가 수 비교

이웃해 여과 방법	no filtering	filtering		
		optimal bias ( $\rho = 4$ )	no bias ( $\rho = 1$ )	greedy bias ( $\rho = -20$ )
본 평가 수	64,644	40,258	43,447	22,565
여과 수	0	937	57,720	$9.1 \times 10^8$

4.2 부하 평준화 문제

부하평준화 문제는 주로 생산 현장에서 작업일정계획을 수립할 때 가장 빈번하면서도 중요하게 다루는 문제로서, 여러 가지 제약조건을 만족시키면서 작업장간의 부하불균형 혹은 각 작업장에서의 단위 기간별 부하불균형이 최소화되도록 작업들의 일정을 수립하는 문제로 요약된다[9,10]. 본 논문에서 대상으로 하는 부하평준화 문제는 하나의 장기 프로젝트<sup>4)</sup>를 구성하는 작업들을 대

2) 본 논문에서 신호주기는 교통류에 따라 변할 수 있고 신호들의 길이가 정해지면 신호주기는 구할 수 있기 때문에 특별히 고정된 값을 사용하도록 별도의 제약을 주지 않았다.  
3) 교통 신호 최적화 문제에 대한 실험 결과 분석을 통해 no filtering에 비해 optimal filtering이 유의수준 10%에서 개선이 있음을 확인하였다.

상으로 각 작업장에서 주별로 본 부하가 평균화가 되도록 작업일정을 수립하는(작업별 시작일과 종료일을 결정하는) 문제이다. 부하평준화를 위해서는 각 작업의 일정을 조정하여야 하는데 현장에서 실제로 작업이 원활히 이루어지기 위해서는 선후관계, 공기, 프로젝트 시작과 종료 시점, 작업장 등에 관한 제약조건들이 항상 만족되어야 한다.

해는 대상 작업들에 대한 일정계획으로 모든 작업들의 시작일과 종료일로 표현된다. 이웃해 생성 과정은 교통 신호체계 최적화 문제와 비슷하게 두 단계로 구분되는데, 일정을 조정해야 할 작업을 선택하는 단계와 선택된 작업에 대해 일정을 조정하는 단계를 거친다. 이 문제의 목표는 각 작업장에 대해 부하(그래프)가 주별로 가능한 평준화되도록 작업들의 일정을 수립하는 것이다. 따라서 대상문제의 목적함수는 각 작업장의 주별 부하 분산의 가중치 합으로 표현된다[9,10].

#### 4.2.1 확률적 여과 기법의 적용

이 문제에서도 확률적 여과 기법은 일정을 조정할 작업을 선택하는 단계에만 적용되었다. 부하그래프에서 평균부하로부터 부하의 편차가 큰 기간에 있는 작업일수록 일정조정 시 부하 평준화의 개선 효과가 클 것으로 보고, 작업을 선택할 때 예비 평가 값으로 전체 계획 기간 평균 부하 대비 대상 작업의 작업기간 동안의 부하의 최대 편차<sup>5)</sup>를 사용하였다[9,10]. 부하 평준화를 위해 어떤 작업의 일정을 조정할 때는 이 작업의 시작일과 완료일의 부하를 보고 상대적으로 적은 쪽으로 이동하는 것이 부하의 편차를 줄이는데 도움을 줄 수 있다. 따라서 일정 조정 단계에서는 선택된 작업의 시작시점과 완료시점의 부하의 차이를 이용하여 조정할 방향을 결정하고, 제약조건이 만족되는 범위 내에서 무작위로 조정량을 결정하였다[9,10].

#### 4.2.2 실험 및 분석

본 논문에서는 [10]에서 소개한 부하평준화 문제를 대상으로 실험하였다.<sup>6)</sup> 작업의 수는 467개이고 전체 작업 기간은 1년이다. 각 작업들이 속하는 작업장의 수는 2개이다. 초기해는 현장의 전문가가 작성한 계획을 사용하였다. 표 3은 simulated annealing에 확률적 여과 기법을 적용하지 않은 경우와 적용한 경우에 대해서 CPU 시간 30분 동안 5회 실험한 평균 성능을 보여준다. 이

문제에서도 확률적 여과 기법을 적용한 경우가 적용하지 않은 경우에 비해 더 좋은 성능을 보여준다. 그러나 이 문제에서는 확률적 여과 기법을 적용할 때, bias를 적절히 설정한 경우( $\rho=2$ )는 6.9%를 개선시킨 반면, 오히려 bias를 주지 않은 경우( $\rho=1$ )의 성능이 16.1%로 더 좋음을 보여준다. 사실 이 문제는 작업장에서 부하의 편차에 대한 분포가 한 쪽으로 매우 많이 치우쳐져 있는 문제였는데, no bias가 가장 좋음을 볼 때, 스케일링 만으로도 충분한 효과가 있음을 보여준다. 이런 개선은 스케일링 함수를 사용하지 않는 이전 [10]의 확률적 여과 방법에 비해서도 향상되었음을 볼 때 스케일링의 역할이 중요함을 알 수 있다. 이 문제에서도 역시 bias를 매우 그리디하게 설정( $\rho=-20$ )하면 성능이 매우 좋지 않다. 각 경우에 대한 탐색의 추이는 그림 6과 같다.

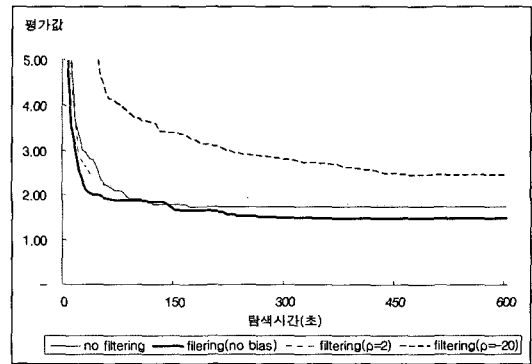


그림 6 부하 평준화 문제에서 이웃해 생성 방법에 따른 탐색 추이

각 경우에 대해 해의 평가 수는 표 4에서 나타내었는데, 교통 신호 최적화 문제와 비슷한 양상을 보여준다. 확률적 여과 기법을 적용하지 않았을 때에 비해, 적용한 경우는 추가 비용이 들기 때문에 평가 수가 적고, 반면 보다 그리로 갈수록 여과되는 횟수가 증가함을 볼 수 있다. No bias와 optimal bias에 해당하는  $\rho=2$ 의 경우에는 평가 수가 비슷하고, greedy bias라고 볼 수 있는  $\rho=-20$ 의 경우에는 여과를 너무 많이 하기 때문에 본 평가에 비해 예비 평가를 많이 하고 있음을 나타낸다. 그리고 본 논문에서 제안하는 확률적 여과 기법은 [10]에서 소개한 확률적 여과에 비해 스케일링 비용이 추가로 들기 때문에 본 평가 수와 여과 수에서 작음을 알 수 있다.

앞에서 살펴본 바와 같이 두 개의 실 세계 최적화 문제에 대해서, 확률적 여과 기법이 적용되었을 때 주어진 탐색 시간에 탐색 성능을 높일 수 있음을 실험으로 검증하였다. 탐색의 성능 개선의 정도는 대상 문제에 따라

4) 보통 수주산업에서 주문주와의 제약을 통해 납품하는 단위로 공사 또는 사업을 말함

5) [9,10]에서 예비 평가 값을 0과 1사이의 값으로 만들기 위해서 이 값을 다시 전체 계획 기간 동안의 평균 부하 대비 최대 부하로 나누었으나, 본 방법론에서는 일반화된 스케일링 함수가 추가 되어있기 때문에 예비 평가 값이 0과 1사이의 값이 될 필요가 없음.

6) 본 논문에서 실험한 데이터는 [9]와는 다르고 [10]와 같은 데이터가 없기 때문에 [10]의 실험결과와 비교하였다.

표 3 부하 평준화 문제에서 확률적 여과 기법의 성능

이웃해 여과 방법	no filtering	previous filtering	filtering		
			less greedy bias ( $\rho = 2$ )	no bias ( $\rho = 1$ )	greedy bias ( $\rho = -20$ )
최적해의 평가값	1.74	1.61 (7.5%)	1.62 (6.9%)	1.46 (16.1%)	2.27 (-30.5%)

표 4 부하 평준화 문제에서 확률적 여과 기법의 평가 수 비교

이웃해 여과 방법	no filtering	previous filtering	filtering		
			less greedy bias ( $\rho = 2$ )	no bias ( $\rho = 1$ )	greedy bias ( $\rho = -20$ )
본 평가 수	567,359	730,000	425,832	447,369	275,990
여과 수	0	$1.4 \times 10^7$	124,945	452,587	$3.3 \times 10^7$

그 의미가 다르기 때문에 일괄적으로 같이 비교할 수 없지만, 두 문제에 대해서 확률적 여과 기법을 적용하지 않았을 때에 비해 상대적으로 수%~수 십%를 개선시키고 있음을 볼 수 있다. 또한 각 문제에 대해서 실험을 통해 찾은 optimal bias 값을 볼 때, 부하 평준화 문제에서는 no bias가 좋았으나, 교통 신호 최적화 문제에서는 no bias에 비해 조금 무작위 방향으로 bias를 줄 때 탐색의 성능을 더 개선시킬 수 있었다. 그리고 매우 그리디하게 bias를 주면 두 문제 모두 확률적 여과 기법을 사용하지 않은 경우보다 더 낮은 성능을 보였다.

해의 평가 수와 관련하여 정리를 해보면 확률적 여과 기법을 적용할 때는 이 기법의 수행에 소요되는 시간이 있기 때문에 실제 해를 본 평가한 수는 적용하지 않았을 때보다 적지만, 예비 평가한 횟수를 함께 보면 더 많은 해를 평가하고 있음을 알 수 있다. 그리고 매우 그리디하게 bias를 주게 되면 아주 좋은 해만 선택되고 거의 대부분의 해를 여과하기 때문에 상대적으로 훨씬 많은 해를 여과하는데 시간을 보내고 있음을 알 수 있다. 따라서 제안하는 확률적 여과 기법을 효과적으로 적용하기 위해서는 매우 그리디하게 bias를 주는 것은 피하는 것이 좋겠다고 판단된다.

5. 결론

본 논문에서는 평가 비용이 큰 문제를 대상으로 국지적 탐색의 효율을 높일 수 있도록, 이전의 확률적 여과 기법에 스케일링 함수를 추가하고 개선된 bias 함수를 도입하여 일반화된 확률적 여과 기법을 제안하였다. 제안한 확률적 여과 기법의 효과를 검증하기 위해서 대규모 실 세계 최적화 문제로 교통망에서의 교통 신호 최적화 문제와 작업 일정 계획에서의 부하 평준화 문제를 대상으로 실험하였다. 대상 문제들을 simulated annealing로 해결할 때, 본 논문에서 제안된 확률적 여과 기법을 사용한 것이 사용하지 않았을 때보다 주어진 탐색

시간 내에서 더 좋은 해를 찾을 수 있음을 확인하였고, 이전의 확률적 여과 방법보다 개선된 결과를 얻을 수 있음을 확인하였다.

향후 연구로서 다음과 같은 세가지를 제안한다. 우선 확률적 여과 기법이 탐색의 효율을 높이기 위해서는 예비 평가 휴리스틱이 얼마나 정확해야 하는지와 그 평가 시간이 본 평가함수에 비해 얼마나 작아야 하는지를 실험적으로 분석할 필요가 있다. 둘째, 확률적 여과 기법의 탐색을 보다 효율적으로 안내할 수 있는 방안에 관한 연구가 필요하다. 탐색과정에서 초반에는 무작위로 후반으로 갈수록 그리디하게 탐색을 안내할 수 있도록 bias factor  $\rho$  값을 변화시킬 수 있는 방안을 마련하면 보다 탐색의 효율을 높일 수 있을 것으로 예상된다. 셋째, 제안하는 방법이 평가시간에 제약이 있는 대표적인 문제로 실시간 최적화 문제 해결에 효율적으로 적용할 수 있는 방안이 되는지 실험적 검증을 통해 확인하고, 이런 문제에 적합하기 위해 필요한 보완점들을 마련할 필요가 있다.

참고 문헌

[1] Stuart J. Russell, Peter Norvig, "Artificial Intelligence : A Mordern Approach," 111-113, 1995.  
 [2] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," Science, 220, 671-680, 1983.  
 [3] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative studies," Journal of Statistical Physics 34, 975-986, 1984.  
 [4] Papadimitriou, C.H., "On Selecting a Satisfying Truth Assignment," Proceedings of the Conference on the Foundations of Computer Science, 163-169, 1991.  
 [5] Bart Selman, Hector Levesque, David Mitchell, "A New Method for Solving Hard Satisfiability Problems," Proceedings of the Tenth National Conference on Artificial Interlligence (AAAI-92),



- 440-446, 1992.
- [6] Bart Selman, Henry Kautz, "Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems," Proceedings of the International Joint Conference on Artificial Intelligence, 290-295, 1993.
- [7] Bresina, J.L.: Heuristic-Biased Stochastic Sampling. Proceedings of AAAI-96, 271-278, 1996.
- [8] V.A. Cicirello, S.F.Smith, "Amplification of search performance through randomization of heuristics," In P. Van Henteryck, editor, Principles and Practice of Constraint Programming - CP 2002: 8<sup>th</sup> International Conference, Proceedings, volume LNCS 2470 of Lecture Notes in Computer Science, 124-138. Springer-Verlag, 2002.
- [9] 강병호, 조민숙, 류광렬, "부하평준화 문제에서 국지적 탐색의 효율 향상을 위한 이웃해 선정 기법", 정보과학회논문지 제31권 제2호, 164-172, 2004.
- [10] Kang, B., Ryu, K. R., "Neighborhood Selection by Probabilistic Filtering for Load Balancing in Production Scheduling," In Bob Orchard, Chunsheng Yang, Moonis Ali, editor, Innovations in Applied Artificial Intelligence - IEA/AIE 2004: 17<sup>th</sup> International Conference, Proceedings, volume LNAI 3029 of Lecture Notes in Artificial Intelligence, 533-542, Springer-Verlag, 2004.
- [11] I. Porche, M. Sampath, R. Sengupta, Y.-L. Chen, S. Lafortune, "A Decentralized Scheme for Real-Time Optimization of Traffic Signals," Proceeding of the 1996 IEEE International Conference on Control Applications, Dearborn, MI, USA, 1996.
- [12] 도철웅, 교통공학원론, 청문각, 1997.
- [13] 원제무, 최재성, "교통공학", 박영사 1990.
- [14] Ranganwamy, B., Jain, A. S. and Glover, F., "Tabu Search Candidate List Strategies in Scheduling," in Woodruff, D. L. (ed) 6<sup>th</sup> INFORMS Advances in Computational and Stochastic Optimization, Logic Programming and Heuristic Search: Interfaces in Computer Science and Operations Research Conference, January 7-9, Monterey Bay, California, Kluwer Academic Publishers, chapter 8, 215-234, 1998.



류 광 렬

현재 부산대학교 정보컴퓨터공학부 교수로 재직 중이다. 서울대학교 전자공학 학사(1979년), 서울대학교 전자공학과 석사(1981년), University of Michigan 컴퓨터공학 박사학위(1993년)를 취득하였다. 1983년 3월부터 1년 반 동안 충북대학교 컴퓨터공학과 전임강사를 역임하였으며, 1992년 3월부터 1년간 Ford Motor Company의 Scientific Research Lab.에서 선임연구원으로 재직하였다. 주요 연구분야는 인공지능, 기계학습, 지능형물류시스템, 정보검색 등이다.



강 병 호

현재 부산대학교 지능형시스템연구실 박사후 연구원으로 재직 중이다. 부산대학교 컴퓨터공학과 학사학위(1994년)를 취득한 후, 현대중공업에서 1994년에서 1997년까지 근무하였고, 부산대학교에서 컴퓨터공학과 석사학위(1999년)와 박사학위(2006년)를 취득하였다. 주요 연구분야는 최적화, 기계학습, 지능형물류시스템 등이다.