

# 부하평준화 문제에서 국지적 탐색의 효율향상을 위한 이웃해 선정 기법

## (A Neighbor Selection Technique for Improving Efficiency of Local Search in Load Balancing Problems)

강병호<sup>†</sup>    조민숙<sup>\*\*</sup>    류광렬<sup>\*\*\*</sup>  
(Byoung-ho Kang)    (Minsook Cho)    (Kwang Ryel Ryu)

**요약** 일반적으로 국지적 탐색에서 최적해를 획득할 가능성은 가능한 많은 이웃해를 생성하면서 반복 수를 늘릴수록 높아지나 긴 탐색시간이 소요된다. 따라서 한정된 시간 내에 최적해를 효율적으로 찾기 위해서는, 적절한 수의 이웃해를 생성하되, 탐색의 질을 높일 수 있는 이웃해를 선별해서 생성하는 것이 요구된다. 본 논문에서는 국지적 탐색기법을 적용하여 부하평준화 문제를 해결할 때, 탐색의 효율을 향상시킬 수 있는 이웃해 선정 기법을 제안하고, 실제계 데이터를 대상으로 그 성능을 검증하였다. 본 논문에서 제안하는 이웃해 선정 기법은 확률적 선별에 기반한 방법으로서, 탐색의 질을 개선시킬 가능성에 대한 추정치를 기준으로 부여된 확률에 따라 이웃해를 선별하여 생성하는 기법이다. 대상 문제에 국지적 탐색기법으로 tabu 탐색과 simulated annealing을 적용한 실험에서, 무작위 또는 그리디 선별에 기반한 방법보다 우수한 성능을 보임을 확인하였다.

**키워드** : 일정계획, 부하평준화, 국지적 탐색, 이웃해 선정 기법

**Abstract** For a local search algorithm to find a better quality solution it is required to generate and evaluate a sufficiently large number of candidate solutions as neighbors at each iteration, demanding quite an amount of CPU time. This paper presents a method of selectively generating only good-looking candidate neighbors, so that the number of neighbors can be kept low to improve the efficiency of search. In our method, a newly generated candidate solution is probabilistically selected to become a neighbor based on the quality estimation determined heuristically by a very simple evaluation of the generated candidate. Experimental results on the problem of load balancing for production scheduling have shown that our candidate selection method outperforms other random or greedy selection methods in terms of solution quality given the same amount of CPU time.

**Key words** : scheduling, load balancing, local search, neighbor selection techniques

### 1. 서론

실제 현장의 최적화 문제들은 대부분 그 탐색공간의 규모가 방대하기 때문에 모든 해들을 열거하여 그 중 최적해를 구하는 방법으로는 해결하기 힘들다. 따라서 이런 문제들은 지금까지 정확한(exact) 알고리즘보다는 주로 휴리스틱 탐색을 통해 해결되어 왔다[1]. 실제로

문제 해결에 자주 적용된 대표적인 휴리스틱 탐색기법으로는 tabu 탐색[2,3,4], simulated annealing[5], genetic algorithm[6] 등이 있다.

일반적으로 탐색 과정에서 최적해를 획득할 가능성은 매 반복(iteration)에서 충분히 많은 수의 이웃해를 생성하고 평가함으로써 높아질 것이다. 그러나 이렇게 할 경우 탐색시간이 길어져서 탐색공간이 매우 큰 문제나 이웃해 평가에 긴 계산시간이 요구되는 문제에 대해서는 주어진 시간 내에 해결하기 어렵다[7]. 계산자원이 한정되어 있을 경우 이웃해 수를 늘리면 매 반복에서 국지적으로 보다 좋은 후보해를 찾을 가능성은 높아진다. 그러나 상대적으로 반복 수가 줄어들어 전체 탐색경로가 짧아지기 때문에 최종적으로 찾게 되는 해의 질이 만족스럽지 못하게 된다. 반복 수를 늘려서 전체 탐색경로가

· 본 연구는 부산대학교 연구(보조비)비(4년과제)에 의한 연구임

† 비회원 : 부산대학교 컴퓨터공학과

bhokang@pusan.ac.kr

\*\* 비회원 : ERP 컨설팅 프리랜서

minsuk21@dreamwiz.com

\*\*\* 종신회원 : 부산대학교 컴퓨터학부 교수

kr Ryu@hyowon.pusan.ac.kr

논문접수 : 2003년 6월 2일

심사완료 : 2004년 10월 18일

깊어지도록 함으로써 최적해를 찾을 가능성을 높이기 위해서는 결국 상대적으로 매 반복 시 생성하는 이웃해의 수를 적정 수준으로 제한할 수밖에 없다.

본 논문에서는 탐색공간이 매우 큰 일정계획을 위한 부하평준화 문제에 대해서, 이웃해 수를 적절히 유지하면서 탐색이 효율적으로 진행될 수 있도록 하기 위해, 이웃해를 선별하여 생성하는 방법을 제안한다. 일정계획 문제를 효율적으로 해결하기 위한 기존의 연구로는 문제를 계층적으로 분해하고 tabu 탐색의 다각화 전략을 적용[8]하거나, 유전 알고리즘과 휴리스틱 탐색 기법을 결합한 하이브리드 유전 알고리즘을 적용하는 시도[9]가 있었으나, 이웃해의 생성에 관해서는 단순히 무작위적 방법만 사용했을 뿐, 본 논문에서와 같이 적정 수의 이웃해를 선별하여 생성하려는 노력은 없었다. 본 논문에서 제안하는 이웃해 선정 기법은 국지적 탐색에서 이웃해 수를 적게 유지하면서도 탐색을 효율적으로 진행하기 위해서 좋아 보이는 이웃해를 선택적으로 생성하는 기법이다. 이 기법은 확률적 여파에 기반한 기법으로, 다음과 같이 두 단계로 구성된다. 우선 이웃해 후보(candidate)를 무작위로 선택한 뒤에, 매우 간단한 평가에 의해 이 후보가 탐색의 질을 얼마나 개선시킬 수 있는지를 추정한다. 그리고 이렇게 추정된 값을 확률로 변환하여, 이 확률에 따라 이 후보를 이웃해(neighbor)로 생성할 것인지 결정한다.

이웃해를 선별하여 생성하기 위해 쉽게 도입될 수 있는 다른 방법으로는 무작위로 선별하는 방법과 좋아 보이는 이웃해를 위주로 선별하는 그리디(greedy) 방법을 들 수 있다. 국지적 탐색기법으로 tabu 탐색과 simulated annealing를 적용한 실험에서, 무작위와 그리디 방법을 이용한 이웃해 생성 기법보다 본 논문에서 제안하는 확률적 선별에 기반한 방법이 우수한 성능을 보임을 확인하였다.

이하의 구성은 다음과 같다. 2장에서는 대상 문제인 일정계획에서 부하평준화 문제를 보다 상세히 소개하고, 3장에서는 탐색 알고리즘으로 tabu 탐색과 simulated annealing을 설명한다. 4장에서는 본 논문에서 제안하는 확률기반의 이웃해 생성 방법과 대상문제에 적용방법을 설명한다. 5장에서는 실험결과를 분석한다. 마지막으로 6장에서는 결론 및 향후 연구과제를 기술한다.

## 2. 대상문제

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

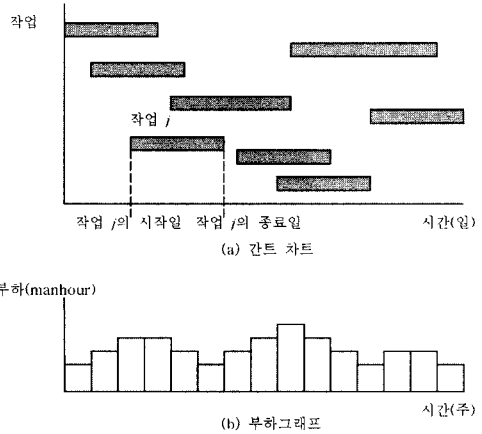


그림 1 한 작업장에서 수행되는 작업들의 일정과 부하그래프의 예

요약된다. 본 논문에서 대상으로 하는 부하평준화 문제는 하나의 장기 프로젝트(1)를 구성하는 작업들을 대상으로 각 작업장에서 주별로 본 부하가 평준화가 되도록 작업일정을 수립하는(작업별 시작일과 종료일을 결정하는) 문제이다.

보통 하나의 프로젝트를 구성하는 작업들은 여러 작업장에서 병렬적으로 수행되지만, 그림 1에서는 하나의 작업장에서 수행되는 작업들의 일정과 부하그래프만을 보이고 있다. 한 작업장에서 수행되는 각 작업은 그림 1의 (a)와 같이 간트 차트(Gantt chart) 상에서 하나의 바(bar)로 표현된다. 이 바의 시작과 끝은 해당 작업의 시작일과 종료일을 의미한다. 그리고 한 작업장에 대한 부하그래프는 그림 1의 (b)와 같이 이 작업장에서 수행되는 작업들의 일자별 부하를 주 단위로 합한 막대그래프로 표현된다. 주 단위로 합산을 하는 이유는 본 대상 문제의 경우 실제 현장에서 작업장마다 주별 능력에 비해 계획 상의 부하가 어느 정도인지를 확인하고 싶어 하기 때문이다. 부하의 단위는 manhour로서, 작업자 1인에 의한 1시간 동안의 노동력을 의미한다.

본 문제의 목표는 각 작업장에 대해 부하(그래프)가 주별로 가능한 평준화되도록 작업들의 일정을 수립하는 것이다. 따라서 대상문제의 목적함수는 식 (1)과 같이 각 작업장의 주별 부하가 능력에 비해 얼마나 차이가 나는지에 대한 오차의 가중치 합을 최소화하는 것이 된다.

$$\min \left( \sum_{k=1}^g w_k e_k^2 \right) \quad (1)$$

여기서  $g$ 는 작업장의 총 수이고  $w_k$ 는 작업장  $k$ 의 중

1) 보통 수주산업에서 주문주와의 계약을 통해 납품하는 단위로 공사 또는 사업을 말함.

요도를 나타내는 가중치이며  $\epsilon_k^2$ 은 작업장  $k$ 의 주별 능력 대비 정규화된 부하의 평균제곱오차로서 다음과 같이 계산된다.

$$\epsilon_k^2 = \frac{\sum_{i=1}^n \left( \frac{L_{ki}}{C_k} - 1 \right)^2}{n} \quad (2)$$

여기서  $L_{ki}$ 는 작업장  $k$ 의  $i$ 번째 주의 부하이고,  $C_{ki}$ 는 작업장  $k$ 의  $i$ 번째 주의 처리능력이며  $n$ 은 전체 작업기간 내에 포함되는 총 주의 수이다.

부하평준화를 위해서는 각 작업의 일정을 조정하여야 하는데 현장에서 실제로 작업이 원활히 이루어지기 위해서는 다음과 같은 제약조건들이 항상 만족되어야 한다.

- 선후관계 제약
- 공기 제약
- 프로젝트 시작 및 종료시점에 대한 제약
- 작업장에 대한 제약

한 프로젝트를 구성하는 작업들 사이에는 선후관계의 제약이 존재하며 이들 제약조건은 그림 2와 같이 작업들의 네트워크 형태로 표현된다. 각 노드(node)는 개별 작업을 나타내고, 노드  $i$ 에서  $j$ 로 향하는 아크(arc)는 작업  $i$ 가 작업  $j$ 에 선행하여 이루어져야 함을 표시한다. 그림에 표현되어 있지는 않지만 보통 선후관계 제약의 내용에는 선행작업 종료 후 후행작업이 시작되기까지 필요한 최소 여유기간이 명시되어 있다. 예를 들어 어떤 아크의 최소 여유기간이 1일이라면 그 아크와 관련된 선행작업이 종료된 후 최소한 1일 이후에야 후행작업을 시작할 수 있게 되는 것이다.

공기제약이란 각 작업의 시작일로부터 종료일까지의 기간이 작업별로 주어진 최소공기 이상, 최대 공기 이하의 범위 내에 있어야 함을 의미한다. 프로젝트의 시작일과 종료일은 주문주와의 계약사항이기 때문에 반드시 지켜야 한다. 따라서 프로젝트를 구성하는 모든 작업들의 일정은 해당 프로젝트 전체의 시작 시점과 종료 시점 이내로 결정되어야 한다. 그리고 각 작업은 그 작업이 이루어질 수 있는 소속 작업장이 있고 각 작업장은 주간 단위로 보아 처리 가능한 물량 즉 주간 능력이 정

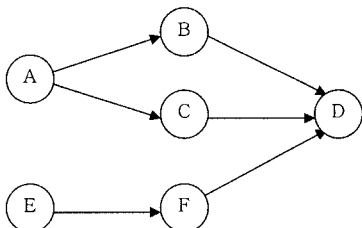


그림 2 작업들간의 선후 관계 제약을 표현한 작업들의 네트워크

해져 있다. 주간 능력은 투입가능 manhour와 대략 비례하는 것으로 과거의 운영 경험에 의해 결정되지만, 부하평준화를 통한 일정계획 수립 결과 필요하다고 판단되는 기간에 대해서는 추가의 인력 투입계획을 수립할 수도 있다. 사실 본 논문에서 취급하는 부하평준화 일정계획의 주요 용도 중의 하나도 바로 이 인력 투입계획 수립의 근거 자료로서 사용하는 데 있다.

대상문제를 요약하면 이상 설명한 제약조건을 어기지 않는 범위 내에서 작업장들의 기간별 부하불균형이 최소화되도록 각 작업의 일정을 조정하는 것이다. 본 논문에서는 이 문제를 한 프로젝트에 포함된 전체 작업을 대상으로 가능한 일정조정 조합들 중에서 제약조건을 만족하면서 부하불균형을 최소화하는 최적안을 찾는 최적화 문제로 보고 휴리스틱 탐색기법을 적용하여 해결하고자 한다. 본 대상문제의 특징 중 하나는 서로 다른 작업장에 소속된 작업들 간에도 서로 선후관계의 제약이 존재할 수 있기 때문에 어떤 작업장의 부하평준화를 위해서는 다른 작업장의 작업 일정을 조정하여야 하는 경우도 있다는 것이다. 따라서, 평균적으로 각 작업별로 시작일과 종료일을 조정하는 방법이  $m$ 가지이고 일정조정 대상 작업의 수가  $n$ 이라면 전체 탐색공간의 규모는  $m^n$ 으로 작업의 수에 따라 기하급수적으로 증가하게 된다. 실제로 본 논문에서 실험한 대상문제의 경우, 작업의 수는 총 467개이고, 각 작업별로 일정을 조정하는 방법의 수가 평균적으로 214개 정도인 매우 규모가 큰 문제로서 휴리스틱 탐색에 의하지 않고는 해결이 불가능하다.

### 3. 탐색 알고리즘의 개요

본 논문에서 사용하는 국지적 탐색 기법은 simulated annealing과 tabu 탐색이다. 이 두 탐색 기법은 언덕 오르기(hill climbing) 탐색기법의 문제점인 지역 최적화(local optima) 현상을 극복하기 위해 고안된 것이다. 이 중 simulated annealing은 현재해에서 임의로 이웃해를 생성하고 평가하여 현재해보다 좋으면 이 이웃해로 무조건 이동하고, 현재 해보다 좋지 않은 경우에는 이 이웃해로 이동할 지의 여부를 확률적으로 결정함으로써 지역 최적해에 빠지더라도 그로부터 벗어나 탐색을 계속할 수 있는 기회를 가지게 된다[5].

Tabu 탐색은 현재해의 여러 이웃해들 중에서 가장 좋은 해로(비록 현재해보다 좋지 않더라도) 항상 이동되 최근에 방문한 해들을 tabu list라는 메모리에 저장해 둬으로써 무한 루프를 막고 지역 최적해를 극복하여 탐색을 가능하게 한다[2-4]. Tabu 탐색 알고리즘의 개요는 그림 3과 같다.

대상문제에서 초기해는 현업의 전문가에 의해 작성된

```

Function TabuSearch(problem, tenure)
  inputs : problem, a problem  tenure, tabu tenure
  local variables : current, a solution  next, a solution
                  best, a solution  optimal, a solution
  current = InitialSolution(problem)
  best = current
  optimal = current
  loop until the termination condition is met {
    loop until enough neighbors are examined {
      Generate a neighbor, next, from current solution, current.
      if next is tabu
        then Evaluate next with a large penalty
        else Evaluate next with no penalty
      if next is better than best then best = next
    }
    current = best
    UpdateTabuList(current, tenure)
    if current is better than optimal then optimal = current
  }
  return optimal
    
```

그림 3 Tabu 탐색 알고리즘

다. 즉 한 프로젝트를 구성하는 모든 작업들의 표준공기와 선후관계 제약을 감안하여 과거의 경험에 따라 단순한 방법으로 초기 일정계획을 현업 전문가가 수립하여 일정계획의 초기해로 입력한다. 초기해로부터 이웃해를 생성하는 방법에는 여러 가지 방안이 있을 수 있겠으나 보통 하나의 작업을 선택하여 시작일 또는 종료일을 수정함으로써 하나의 이웃해가 생성된다. Tabu 탐색에서는 이들 이웃해 중 가장 좋은 해로 이동하는 탐색이 수행된다. 그러나, 이동해 간 이웃해가 현재해보다 나쁠 경우에는 (즉, 현재해가 지역 최적해일 경우) 탐색이 다시 지역 최적해로 되돌아 가는 방향으로 진행됨으로써 무한 루프에 빠질 위험이 있으므로 매 이동 시마다 이동 내역을 tabu list에 저장해 두고 이를 최근에 방문했던 해로의 이동을 방지하는데 사용한다. Tabu list에 저장되는 이동 내역은 소정의 tenure를 가지고 있어서 tenure가 만료되면 tabu list에서 삭제된다. 따라서 tabu list는 최근의 탐색 내역을 저장하고 있는 기록이라 할 수 있다. 이웃해 평가 시에는 항상 tabu list를 조사하여 그 이웃해의 내역이 tabu list에 포함되어 있으면 그 이웃해를 tabu 상태로 간주하여 벌점(penalty)을 부여한다. 이렇게 함으로써 지역 최적해로 탐색이 회귀하는 것을 막고 탐색을 다각화하여 궁극적으로 우수한 해를 찾을 수 있도록 탐색을 유도하게 된다.

본 논문에서는 이상 설명한 국지적 탐색기법을 적용함에 있어서 좋은 이웃해를 적절한 규모로 생성시킴으로써 부하평준화 일정계획을 효율적으로 수립할 수 있는 방안을 제시하고 있다. 다음 장에서는 본 논문에서

제안하는 확률 기반의 이웃해 생성 방법에 대해 상세히 설명한다.

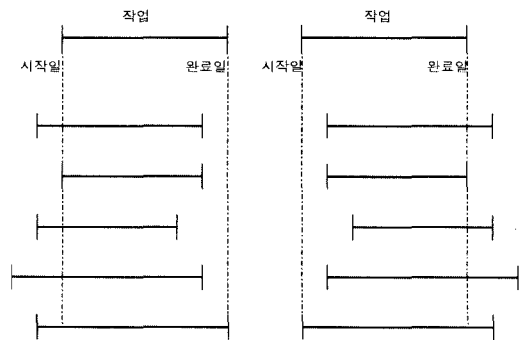
#### 4. 확률 기반의 이웃해 생성 방법

##### 4.1 해의 표현과 이웃해 생성 방법

대상 문제에서 해는 하나의 일정계획안으로서 대상 프로젝트의 모든 작업들에 대한 일정(시작일과 완료일)의 조합으로 표현된다. 탐색공간은 2장에서 설명한 제약 조건을 만족하는 모든 일정계획안들의 집합이 된다. 이웃해는 현재 해를 구성하는 작업들 중 하나를 선택한 다음 그 작업의 시작일과 종료일을 조정함으로써 생성된다.

본 논문에서는 대상문제에 적합한 이웃해 생성 방법을 고안하기 위해서, 우선 기본적인 이웃해 생성 전략 중에서 탐색공간을 분할함으로써 줄이는 효과가 있는 successive filtration strategy를 도입하여, 조정해야 할 작업을 선택하는 단계와 선택된 작업에 대해 일정을 조정하는 단계로 구분하였다[10]. 두 번째 단계인 일정을 조정하는 단계에서는 조정할 방향 - 예를 들면 작업의 일정을 앞으로 당길 지 또는 뒤로 밀 지에 대한 두 가지 방향 - 만을 결정하고, 조정량에 대해서는 제약조건이 만족되는 범위 내에서 무작위로 결정한다.

조정 방향별 조정량 결정 방법을 설명하면 다음과 같다. 우선 앞으로 조정할 경우에는 선행 작업들과의 최소 여유와 프로젝트 시작일을 고려하여, 가능한 가장 이른 시점으로부터 현재 시작일까지의 범위 내에서 무작위 추출에 의해 시작일을 구한다. 새로 구한 시작일을 기준으로부터 최소 및 최대 공기의 제약과 후행 작업들과의 최소 여유를 고려하여 허용 범위 내에서 무작위 추출에 의해 완료일을 구한다. 뒤로 조정할 경우에도 이와 유사한 방법으로 구한다. 그림 4의 (a)와 (b)는 선택된 작업을 앞으로 또는 뒤로 조정할 때 가능한 이웃해의 형태들을 보여준다.



(a) 앞으로 조정하는 경우

(b) 뒤로 조정하는 경우

그림 4 조정 방향별 작업 일정의 조정량 결정방법

4.2 여러 가지 이웃해 선별 전략

본 논문에서 비교 대상으로 한 이웃해 생성방법으로 무작위 방법, 그리디(greedy) 방법 그리고 확률적 선별에 기반한 방법을 소개하면 다음과 같다.

첫째, 무작위 방법은 일정조정 대상 작업과 조정 방향을 무작위로 선택(simple random sampling)하는 방법이다. 일정조정 대상 작업을 선택하는 단계에서는 비복원 추출 방법에 의해 무작위로 일정수의 작업을 선택한다. 일정을 조정하는 단계에서는 선택된 각 작업에 대해 조정 방향을 무작위로 결정한다.

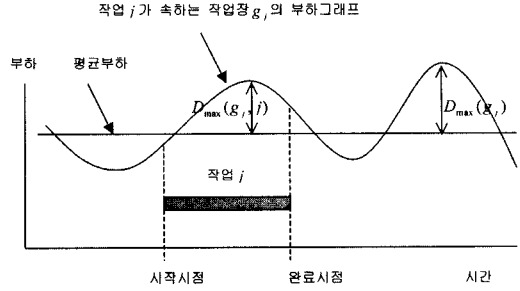
둘째, 그리디 방법은 그림 1의 (b)에서 설명한 작업장의 부하그래프를 이용하여 일정조정 대상 작업과 조정 방향을 선택하는 방법이다. 본 논문에서는 일정조정 대상 작업을 그리디로 선택하는 방법으로 부하그래프에서 평균 부하에 대한 편차를 이용한다. 즉, 평균 부하에 대한 편차가 큰 시점에 걸쳐 있는 작업일수록 조정할 때 부하의 불균형 정도를 낮출 가능성이 높다고 판단했다. 우선 작업장별 부하그래프에서 편차가 큰 시점들을 구하고, 편차가 가장 큰 시점에 걸쳐 있는 작업들을 우선적으로 선택한 뒤에, 다음으로 편차가 큰 시점 순서로 작업들을 선택한다. 선택된 작업에 대해 일정을 조정하는 단계에서도 해당 작업의 시작시점과 완료시점의 부하를 보고 그리디로 조정 방향을 결정한다. 만약 시작시점의 부하가 완료시점의 부하보다 낮으면 앞쪽으로, 높으면 뒤쪽으로 조정 방향을 결정한다. 시작시점과 완료시점의 부하가 같으면 무작위로 결정한다.

셋째, 확률적 선별에 기반한 방법은 그리디 방법에서 사용하는 편차와 시작시점과 완료시점의 부하 차이에 대한 각에 대해 확률을 부여한 방법이다. 일정조정을 수행할 작업을 확률적으로 선택하는 방법은 다음과 같다. 우선 일정조정을 수행할 작업을 무작위로 추출한다. 추출된 작업에 대해서 탐색을 개선시킬 가능성에 대한 추정치에 따라 일정조정 대상 작업으로 선택될 확률과 선택되지 않을 확률을 구하고, 이 확률에 기반한 무작위 추출(weighted random sampling)을 통해 선택여부를 결정한다.

4.3 확률적 선별에 기반한 이웃해 선정 기법

본 논문에서는 부하그래프에서 평균부하로부터 부하의 편차가 큰 기간에 있는 작업일수록 일정조정 시 탐색을 개선시킬 가능성이 크다고 판단하여, 이 가능성에 대한 추정치로 평균 부하에 대한 편차를 사용하였다. 그림 5는 무작위로 추출된 작업에 대해서 확률적으로 선택여부를 결정하는 방법을 보여준다.

어떤 작업  $j$ 의 소속 작업장  $g_j$ 의 부하그래프가 그림 5와 같을 때, 작업  $j$ 의 공기동안에 평균부하 대비 최대편차는  $D_{max}(g_j, j)$ 이고, 이 부하그래프에서 평균부하 대



$D_{max}(g_j)$  : 작업장  $g_j$ 의 부하그래프에서 평균부하 대비 최대 편차

$D_{max}(g_j, j)$  : 작업장  $g_j$ 의 부하그래프에서 작업  $j$ 의 공기 동안 평균부하 대비 최대 편차

그림 5 조정 대상 작업 선택 방법

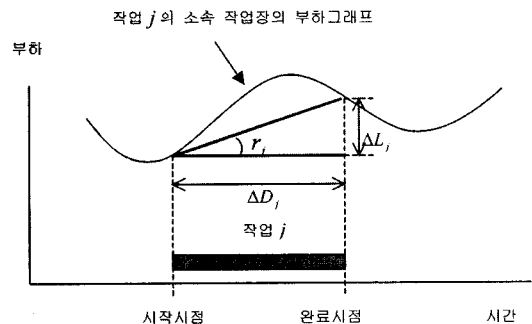
비 최대편차는  $D_{max}(g_j)$ 이다. 이 때 작업  $j$ 에 대해서 일정조정 대상 작업으로 선택할 확률  $P_{select}(j)$ 와 선택하지 않을 확률  $P_{not\_select}(j)$ 은 다음과 같이 구한다.

$$P_{select}(j) = \frac{D_{max}(g_j, j)}{D_{max}(g_j)} \tag{3}$$

$$P_{not\_select}(j) = 1 - P_{select}(j) \tag{4}$$

만약 특정 작업이 일정조정 대상으로 선택되면, 작업의 시작시점과 완료시점의 부하 차이에 대한 각을 기준으로 앞으로 조정할 확률과 뒤로 조정할 확률을 구해서, 역시 이 확률에 기반한 무작위 추출을 통해 조정 방향을 결정한다. 그림 6은 조정 방향을 결정하기 위해 선택된 작업의 시작시점과 완료시점의 부하로부터 각을 구하는 방법을 나타낸다.

선택된 작업  $j$ 와 이 작업의 부하그래프가 그림 6과



$\Delta L_j$  = 작업  $j$ 의 완료시점 부하 - 작업  $j$ 의 시작시점 부하

$\Delta D_j$  = 작업  $j$ 의 공기

그림 6 조정 방향 결정 방법

같다면, 작업  $j$ 의 공기  $\Delta D_j$ 와 시작일과 완료일의 부하 차이  $\Delta L_j$ 을 이용해서 조정 방향을 결정하는데 사용될 각(radian)은 다음과 같이 구한다.

$$r_j = \text{atan}\left(\frac{\Delta L_j}{\Delta D_j}\right), \quad \left(-\frac{\pi}{2} < r_j < \frac{\pi}{2}\right) \quad (5)$$

본 논문에서는 앞에서 구한 각의 절대값이 클수록 부하의 불균형 정도를 낮추기 위해 시간 축 상 앞으로 또는 뒤로 조정해야 할 확률이 커지도록 한다. 예를 들어 각  $r_j$ 가  $\pi/2$ 에 접근할수록 앞쪽(그림에서 왼쪽)으로 조정할 확률이,  $-\pi/2$ 에 접근할수록 뒤쪽(그림에서 오른쪽)으로 조정할 확률이 커지게 하고, 0에 접근할수록 앞 또는 뒤로 이동할 확률이 비슷해지도록 하는 것이다.  $r_j$ 가 양의 방향으로 크다는 것은 작업  $j$ 의 시작시점보다 완료시점 근처의 작업장 부하가 크다는 것을 의미하므로, 작업  $j$ 의 일정을 시작시점 방향으로 앞당겨 줄 경우 완료시점 근처의 작업장 부하를 경감시켜 부하평준화에 기여할 수 있게 된다고 판단할 수 있다.  $r_j$ 가 음의 방향으로 큰 값을 가질 때에는 반대의 이야기가 성립하며,  $r_j$ 가 0에 가까울 때에는 어느 방향으로 일정을 조정하든 거의 마찬가지일 것이므로 이동 확률은 1/2에 가깝게 되도록 하면 될 것이다. 이런 관점에서 각을 바탕으로 앞으로 이동할 확률  $P_{forward}(j)$ 과 뒤로 이동할 확률  $P_{backward}(j)$ 을 구하는 방법은 다음과 같다.

$$P_{forward}(j) = \left(\frac{1}{\pi} \times r_j + 0.5\right) \quad (6)$$

$$P_{backward}(j) = 1 - P_{forward}(j) \quad (7)$$

### 5. 실험 및 분석

본 논문에서는 실험 데이터로 국내 모회사의 실제 해양 플랜트 생산일정 데이터를 사용하였다. 작업의 수는 467개이고 전체 작업기간은 1년이다. 각 작업들이 속하는 작업장의 수는 2개이다. 대상 문제에서 최적해를 구하는 데 사용할 수 있는 CPU시간은 1시간으로 제한하였는데, 이는 현업 전문가가 대상 문제와 관련된 업무를 효율적으로 수행하기 위하여 요구한 시간을 반영한 것이다. 초기해는 현장의 전문가가 작성한 계획을 사용하였다.

Tabu 탐색 알고리즘으로는 tabu 리스트만을 사용하는 simple tabu search를 사용하였고, tabu tenure와 생성된 이웃해 수는 실험으로 구한 15와 23을 사용하였다. Tabu 탐색을 적용할 때, 이웃해는 4장에서 설명한 세가지 방법(무작위 방법, 그리디 방법, 확률적 방법)대로 생성된다.

Simulated annealing은 기본적으로 전체 이웃해에서

1개의 이웃해를 무작위로 생성하여 탐색을 진행하는 방법이다. simulated annealing을 적용할 때 이웃해 생성 방법은 다음과 같다. 우선 무작위 방법은 기본 알고리즘과 같이 일정 조정 대상 작업의 선택과 조정 방향에 대해서 무작위로 선별하여 1개의 이웃해를 생성하는 방법이다. 그리디 방법은 각 작업장별 부하그래프에서 평균 부하로부터 가장 편차가 큰 시점들(5곳)에 걸쳐있는 작업들을 미리 뽑은 뒤에, 이 작업들 중에서 무작위로 1개의 작업을 선택하고, 조정 방향은 (4장에서 설명한 바와 같이) 작업의 시작시점과 종료시점의 부하를 기준으로 결정하여 이웃해를 생성하는 방법이다. 확률적 방법은 우선 무작위로 1개의 작업을 뽑고, 부하의 편차에 따른 확률을 고려하여 일정조정 대상으로 선택할 것인지 결정한 뒤에, 만약 1개의 작업이 선택되면 조정 방향에 대한 확률을 구하여 이를 바탕으로 무작위 추출을 통해 조정 방향을 결정하여 이웃해를 생성하는 방법이다.

표 1은 tabu 탐색과 simulated annealing에 각 이웃해 생성 방법을 적용하여 5회 수행하였을 때 구한 가장 좋은 해에 대한 평가값의 평균을 보여준다. 그림 7과 그림 8은 각각 tabu 탐색과 simulated annealing에 대해서 1시간 동안 탐색의 진행과정에서 가장 좋았던 해의 추이를 각 이웃해 생성 방법별로 보여준다.

표 1과 그림 7 그리고 그림 8에서 나타난 바와 같이, 확률적 선별 방법이 가장 좋은 결과를 나타냈다. 그리디 방법이 탐색의 전체 진행과정에서 무작위 방법보다 모두 좋지 않게 나타난 이유를 분석해보면, 대상 문제에서 부하그래프에서 평균 부하 대비 편차를 기준으로 가장 편차가 큰 시점들의 작업을 위주로 선택해더라도, 작업들간의 선후관계에 의한 제약 때문에 탐색의 질을 잘 개선시키지 못하기 때문인 것으로 판단된다. 확률적 선별에 기반한 이웃해 생성 방법이 좋게 나타난 이유는 그리디 방법에서 사용하는 휴리스틱(부하그래프에서 평균 부하에 대한 편차가 큰 시점에 걸쳐 있는 작업일수록 조정 시 부하의 불균형 정도를 낮출 가능성이 높다는 것과 시작일과 완료일의 부하차이에 의한 기울기를 보고 조정 방향을 결정하는 것)을 완전히 신뢰한 상태에서 결정하지 않고, 확률적으로 결정하기 때문이라고 판단된다. 확률적으로 결정하면, 선후관계 제약에 있는 다른 작업을 조정하는 기회가 있고 그에 따라 주목 대

표 1 이웃해 생성 방법별로 구한 가장 좋은 해의 평가값(5회 평균값)

이웃해 생성 방법	Tabu 탐색	Simulated annealing
그리디 방법	7.38	8.74
무작위 방법	3.64	3.84
확률적 방법	1.57	3.29

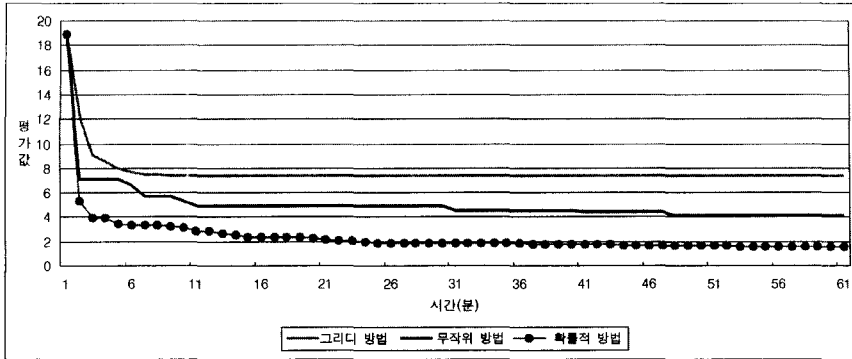


그림 7 Tabu 탐색의 각 이웃해 생성 방법별 가장 좋은 해의 추이

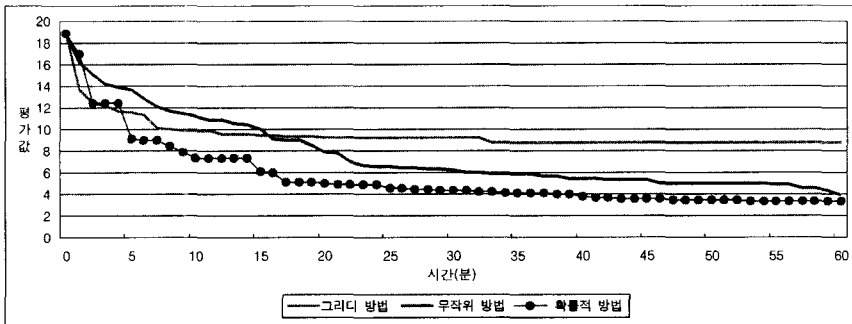


그림 8 Simulated annealing의 각 이웃해 생성 방법별 가장 좋은 해의 추이

상 작업의 실제로 일정 조정될 폭이 충분해질 수가 있기 때문이다. 사실 탐색기법을 이용한 대부분의 문제 해결에서, 탐색의 질을 개선시킬 수 있도록 하기 위해 이와 같은 휴리스틱을 고안하여 이웃해 생성 방법에서 사용한다. 그러나 이런 휴리스틱은 탐색과정에서 발생하는 다양한 상황들을 모두 파악하기 어렵기 때문에, 궁극적으로 볼 수 밖에 없는 한계가 존재한다. 따라서 탐색의 과정에서 이에 대한 신뢰의 정도를 바탕으로 확률적으로 결정하는 것은 이런 문제를 해결할 수 있는 대안이 된다고 사료된다.

대상 문제를 해결하기 위해 이웃해 생성 방법에서 사용하는 두 가지 정보 즉, 작업을 선택할 때 사용하는 평균 부하 대비 편차와 조정 방향을 선택할 때 사용하는 시작일의 부하와 완료일의 부하에 의한 기울기 정보가 탐색을 효율적으로 진행하는 데 얼마나 기여하는지를 분석하기 위해서 표 2와 같이 추가 실험을 하였다. 표 2는 이웃해 생성 과정에서 작업을 선택하는 방법과 조정할 방향을 선택하는 방법에 대해서 서로 다른 이웃해의 생성 방법들을 조합하여 실험한 결과를 나타낸다. 그리고 이 방법들에 대해 1시간 동안 탐색의 진행과정에서

표 2 Tabu 탐색의 이웃해 생성 방법별로 구한 가장 좋은 해의 평가값(5회 평균)

방향 선택 방법 \ 작업 선택 방법	그리디 방법(GD)	무작위 방법(RD)	확률적 방법(PD)
그리디 방법(GJ)	7.38 (GJGD)	7.62 (GJRD)	6.78 (GJPD)
무작위 방법(RJ)	5.18 (RJGD)	3.64 (RJRD)	1.88 (RJPD)
확률적 방법(PJ)	5.37 (PJGD)	2.18 (PJRD)	1.57 (PJPD)

가장 좋았던 해의 추이를 그림 9에서 보여준다<sup>2)</sup>.

작업을 선택하는 방법이나 조정 방향을 선택하는 방법 모두에서 1가지 경우(작업 선택으로 그리디 방법을 사용할 때, 조정 방향 선택 방법에서 확률적 방법, 그리디 방법, 무작위 방법 순으로 좋은 성능을 보였음)를 제외하고는 확률적 방법, 무작위 방법, 그리디 방법 순으로 높은 성능을 보였다.

그림 10과 그림 11은 표 2의 각 이웃해 생성 방법을

2) 추이 그래프들을 알기 쉽도록 하기 위해서 일부만 나타내었다.

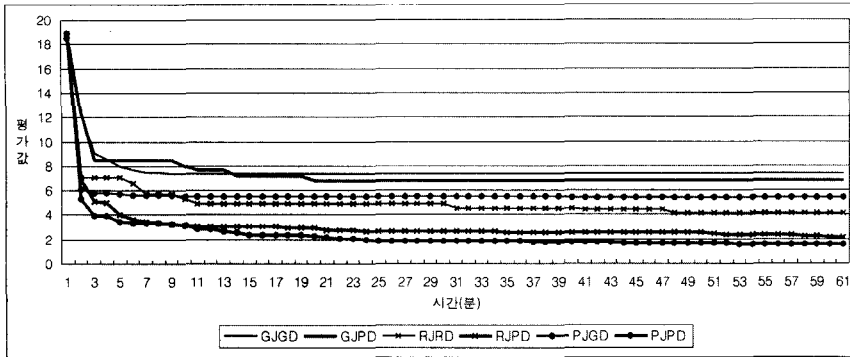


그림 9 Tabu탐색의 이웃해 생성 방법별 가장 좋은 해의 추이

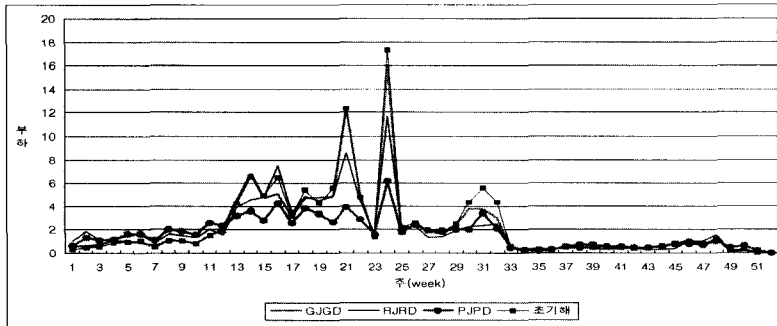


그림 10 첫 번째 작업장의 각 이웃해 생성 방법별 부하그래프

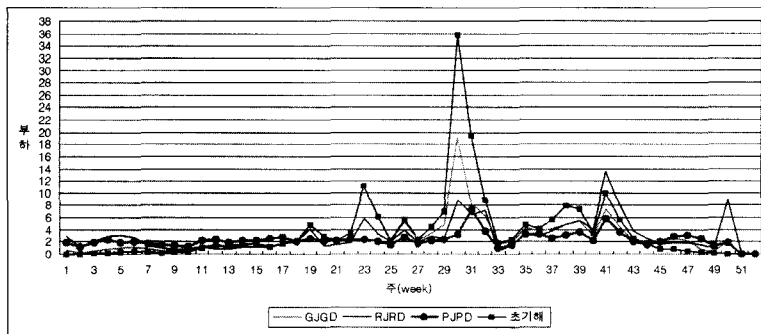


그림 11 두 번째 작업장의 각 이웃해 생성 방법별 부하그래프

적용하여 구한 가장 좋은 해의 부하그래프를 작업장 2 곳에 대해서 보여준다. 초기해에서 부하의 불균형 정도가 상당히 높았으나 확률적 선별에 기반한 이웃해 생성 방법이 가장 크게 부하평준화를 수행하였음을 보여준다.

### 6. 결론 및 향후 연구

본 논문에서는 국지적 탐색기법을 적용하여 부하평준화 일정계획을 수립할 때, 탐색의 효율을 향상시킬 수

있는 확률적 선별에 기반한 이웃해 선정 방법을 제시하였다. 이웃해를 선별하여 생성할 수 있는 다른 방법으로 무작위 선별에 기반한 방법과 그리디 선별에 기반한 방법을 함께 소개하였고, 탐색공간이 매우 큰 실제 현장의 부하평준화 문제를 대상으로 다양한 실험을 통해 그 성능을 비교 검증하였다. 국지적 탐색기법으로 tabu 탐색과 simulated annealing를 적용한 실험에서, 확률적 선별에 기반한 이웃해 선정 방법이 무작위와 그리디 방법



보다 우수한 성능을 발휘하는 것을 확인하였다.

이것은 탐색의 질을 개선시키는데 도움이 되는 이웃해를 생성하는데 사용되는 휴리스틱의 국지적인 성질 때문에 이를 지나치게 신뢰하는 그리디 방법은 좋지 못하고, 그렇다고 이를 완전히 무시하는 무작위 방법도 효율적이지 못함을 보여주는 것이다. 반면에 이웃해 선별 휴리스틱에 대한 신뢰의 정도를 바탕으로 확률적으로 결정하는 방법이 보다 효과적임을 확인할 수 있게 해주었다.

향후 확률적인 선별기법에 대한 보다 일반적인 모델을 수립하여 국지적 탐색기법에 반영하고 이것의 효과를 검증하는 작업이 필요하다. 또한 본 논문에서 실험한 대상 문제뿐만 아니라 다른 문제들의 이웃해 생성에서도 확률적인 선별 방식이 효과가 있는지 검증할 필요도 있을 것이다.

### 참 고 문 헌

- [1] Feo, T. A., Resende, M.G. C., "Greedy Randomized Adaptive Search Procedures," Journal of Global Optimization 6, 1995, pp.109~133.
- [2] Glover, F., Laguna, M., "Tabu Search," Kluwer Academic, Publishers, Boston, 1997.
- [3] Glover, F., "Tabu Search:Part1" ORSA Journal on Computing, 1(3), 1989, pp.190~206.
- [4] Glover, F., "Tabu Search Fundamentals and Uses," Graduate School of Business, University of Colorado, condensed version published in Mathematical Programming: State of Art, 1994, Birge & Murty, eds., 64~92.
- [5] Kirkpatrick, S., "Optimization by simulated annealing: Quantitative studies," Journal of Statistical Physics 34, 1984, pp.975~986.
- [6] Goldberg, D., "Genetic algorithms in search, optimization and machine learning," Addison-Wesley, 1989.
- [7] Ravindra, K., Ahuja, James, B., Orlin and Dushyant Sharma, "Very large-scale neighborhood search," International Transactions in Operational Research, Vol.7, No.4~5, pp.301~317, 2000.
- [8] 최우형, 류광렬, 황준하, 조규갑, "Tabu Search를 이용한 작업일정계획 및 재일정계획 수립", 한국전문가 시스템학회 '98 추계학술대회 논문집, pp.376~387, 1997.
- [9] 황준하, 류광렬, 최형림, 조규갑, "하이브리드 유전 알고리즘과 반복적 개선 탐색 기법에 의한 일정계획", 정보과학회논문지 A 제25권 제8호, pp.825~835, 1998.
- [10] Rangaswamy, B., Jain, A. S. and Glover, F. (1998a) "Tabu Search Candidate List Strategies in Scheduling," in Woodruff, D. L. (ed) 6th 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, pp.215~234.



강 병 호

1994년 부산대학교 컴퓨터공학과 학사  
1994년~1997년 (주)현대중공업 사원  
1997년~1999년 부산대학교 컴퓨터공학과 석사. 1999년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야는 최적화, 에이전트, 기계학습, 데이터 마이닝 등



조 민 숙

1992년 경상대학교 전자계산학과 학사  
1992년~1999년 한국중공업 전산실 근무  
2001년~2002년 부산대학교 컴퓨터공학과 석사. 2003년~현재 ERP 컨설팅 프리랜서. 관심분야는 최적화 에이전트 기계학습 데이터 마이닝 등



류 광 렬

1979년 서울대학교 전자공학과 학사  
1981년 서울대학교 전자공학과 석사  
1983년 3월~1984년 8월 충북대학교 컴퓨터공학과 전임강사. 1992년 University of Michigan 전기 및 컴퓨터공학과 박사. 1992년 3월~1993년 2월 Scientific Research Lab., Ford Motor Company 전임연구원. 1993년 3월~현재 부산대학교 정보컴퓨터 공학부 부교수. 관심분야는 기계학습, 데이터 마이닝, 정보 검색 최적화