

병렬휴리스틱법의 고속화모델을 적용한 생산 스케줄링 문제의 해법

홍 성 찬[†] · 조 병 준^{††}

요 약

기존의 많은 논문에서는 병렬휴리스틱법(Parallel Heuristics) 또는 몇 개의 휴리스틱법을 결합하는 하이브리드 시스템(Hybrid System)이 보다 양질의 탐색 결과를 얻을 수 있음을 보고하고 있다. 그러나 단일 프로세스를 가진 범용 컴퓨터상에서의 병렬화와 하이브리드화는 지나치게 많은 탐색시간을 요구한다. 이러한 경우 탐색의 고속화를 위해서는 탐색법간의 보다 적절한 결합법이 요구된다. 이를 위한 방법론으로서 병렬휴리스틱법의 고속화모델(Fast Model of Parallel Heuristics : FMPH)을 제안하고 있다. 유전적알고리즘(Genetic Algorithms)의 섬모델(Island Model)에 의거, 다양한 탐색 공간에서 선택된 우수한 엘리트해에 대해서만 선택적으로 국소탐색(local search)의 능력이 뛰어난 타부탐색법을 도입한다. 본 논문에서는 NP-hard문제 중에서도 가장 어려운 문제로 평가되는 잡숍 스케줄링 문제(Job-Shop Scheduling Problem)에 대해 적용한 병렬휴리스틱법의 고속화 모델을 소개하며 모델의 범용성을 입증하기 위해 유명한 벤치마크 문제에 적용하여 얻은 뛰어난 결과를 보고한다.

A Solution of Production Scheduling Problem Adapting Fast Model of Parallel Heuristics

Sung-Chan Hong[†] · Byung-Jun Cho^{††}

ABSTRACT

Several papers have reported that parallel heuristics or hybrid approaches combining several heuristics can get better results. However, the parallelization and hybridization of any search methods on the single CPU type computer need enormous computation time. That case, we need more elegant combination method. For this purpose, we propose Fast Model of Parallel Heuristics(FMPH). FMPH is based on the island model of parallel genetic algorithms and takes local search to the elite solution obtained from each island(sub group). In this paper we introduce how can we adapt FMPH to the job-shop scheduling problem notorious as the most difficult NP-hard problem and report the excellent results of several famous benchmark problems.

1. 서 론

실세계에는 NP-hard문제에 속하는 많은 문제들이

존재하며 이러한 문제들에 대해서는 다항식 계산 시간 오더 내에 문제를 해결할 수 있는 방법은 존재하지 않게 된다. 이러한 경우 실용적인 면에서 받아들일 수 있는 해를 발견적인 방법으로 탐색해 나가는 방법인 휴리스틱(Heuristics)법을 적용할 수 있다. 그러나 많은 어려운 NP-hard문제에 대해 이러한 휴리스틱법은 최

* 이 연구는 1997년도 한국학술진흥재단의 공모과제(1997-003-e00321) 연구비 지원에 의한 결과임.

† 종신회원 : 한신대학교 컴퓨터정보통신학부 교수

†† 정 회 원 : 영진전문대학 경영정보학과 교수

논문접수 : 1998년 1월 15일, 심사완료 : 1999년 1월 18일

적해를 발견한다는 보장은 할 수 없으며 최적해 근방에서 더 이상의 해의 개선이 이루어지지 않게 될 때가 존재한다. 탐색과정이 이러한 국소 최적해의 계곡에 빠져 탈출할 수 없게 되는 상황을 극복하기 위해 여러 가지 방법론이 제안되고 있다. 여러 휴리스틱법에 대한 성능을 개선하기 위한 방법론으로서 병렬(Parallel)화와 하이브리드(Hybrid)화를 생각할 수 있다. 먼저 병렬화를 통해 보다 양질의 해를 얻을 수가 있을 것이다. 그러나, 병렬화는 많은 탐색 시간을 요구하며, 특히 단일 프로세스를 사용하는 일반유저의 컴퓨터상에서 이루어지는 병렬화 프로그램의 시뮬레이션은 보다 많은 탐색 시간을 필요로 하게 된다는 문제점으로 인해 실용성을 갖추기가 어렵다는 단점이 있다. 또한 여러 휴리스틱법이 가지는 본질적인 특성을 이용하기 위해 다양한 방법들을 통합 또는 교대로 적용하는 하이브리드화도 성능의 개선을 기대할 수가 있을 것이다. 그 중에서도 유전적알고리즘과 같은 전역탐색법과 시뮬레이티드 어닐링법 또는 타부탐색법과 같은 1점 탐색법의 하이브리드화는 각 방법이 지니는 본질적인 장점을 상호 이용하여 자신의 단점을 보완할 수 있다는 점에서 매우 적절한 결합이라 할 수 있다. 이러한 점을 바탕으로 전역탐색법과 국소탐색법의 하이브리드화를 피할 경우 필연적으로 병렬국소탐색법이 만들어진다. 유전적알고리즘과 같은 전역탐색법은 복수의 가능해에 의해 구성되며 본질적으로 병렬프로그램의 성격을 지닌다. 이들 전역탐색법에서 얻어지는 복수의 가능해에 대해 1점탐색법을 적용하고자 하는 경우 평가치가 낮을 수도 있는 복수의 가능해에 대해 모두 1점탐색법을 적용하고자 한다면 지나친 탐색 시간의 낭비를 초래할 수 있다. 비록, 현 시점에서의 평가치가 나쁘다 하더라도 그 후의 탐색을 통한 평가치의 개선은 얼마든지 기대할 수가 있다. 그러나 그러한 해가 차후 현재의 다른 우수한 해보다 나은 해로 개선되어갈 것이라는 보장은 아무도 할 수가 없다. 이러한 경우 현재까지 탐색된 우수해를 보다 효과적으로 이용할 수 있는 기구가 필요하게 된다.

본 논문에서는 국소 최적해와 최적해 근방에서의 해의 개선과 탐색의 속도를 향상시키기 위한 방법으로서 섬모델을 기본으로 한 유전적알고리즘을 도입하여 엘리트해를 추출하고 이들 해에 개별적으로 국소탐색법을 적용하는 병렬휴리스틱법의 고속화모델에 대한 방법을 제시한다. 서로 다른 경로를 통해 진화되어 나

은 엘리트해에 대해서만 국소탐색법을 적용함으로써 병렬화에 따른 탐색 시간의 급증을 방지하고 이들 해의 정보를 다른 섬으로의 이주(migration)를 통해 상호 공유함으로써 탐색의 과정을 고속화 할 수 있다. 이주되어온 최우수해에 의해 집단의 다양성이 상실되지 않도록 하기 위해 서로 다른 경로에 의해 개선된 해들간의 교차를 제어하는 기구를 지닌다.

본 논문에서는 병렬휴리스틱법의 고속화모델의 성능상의 우수성을 확인함과 동시에 모델의 일반성을 제시하기 위해 NP-hard문제 중에서도 가장 어려운 문제에 속하는 잡숍 스케줄링 문제(Job-Shop Scheduling Problem : JSP)를 풀기 위한 방법론을 제시함과 동시에 기타 다른 방법과의 비교를 행한다. 이 실험결과는 종래의 단일 휴리스틱법을 적용한 연구와 비교할 때 보다 뛰어난 성능을 제공함으로써 본 모델의 효율성을 입증하고 있다.

2. 병렬휴리스틱법의 고속화모델

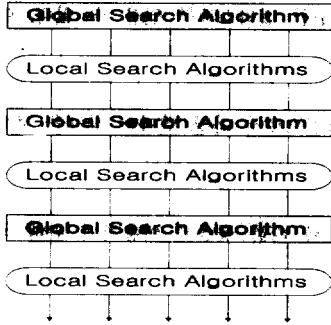
여러 탐색법을 결합하는 하이브리드 시스템을 고려할 경우 각각의 탐색법의 특성을 잘 고려한 결합이 필요하다. 탐색법은 그 성격에 따라 하나의 해로부터 다음해로 이동해 가면서 탐색을 진행하는 국소탐색법과 복수의 해에 대한 해공간을 동시에 탐색하는 전역탐색법으로 구분할 수 있다. 국소탐색법으로는 시뮬레이티드 어닐링[15][16], Hill-Climbing법, 타부탐색법 등이 있으며, 전역탐색법으로는 유전적알고리즘을 들 수 있다. 위의 각각의 방법을 결합하여 하이브리드 시스템을 구축할 수 있으나 성격이 다른 탐색법을 결합함으로써 보다 넓은 탐색공간에 대하여 집중적인 탐색이 가능하게 된다.

일반적으로 전역탐색법과 국소탐색법의 하이브리드화는 (그림 1) 같은 구조를 가진다.

이러한 형태의 하이브리드화는 전역탐색법의 각각의 해에 대해, 국소탐색법을 적용함으로써 결과적으로는 국소탐색법의 병렬모델로서의 성격을 지니며 다음과 같은 단점이 있다.

- 예상 가능한 결과로서 단일 프로세서상에서의 이러한 모델의 구현은 전체 해에 대한 병렬 탐색으로 인해 지나치게 많은 탐색 시간을 요구하며 불필요하거나 상대적으로 열등한 해의 탐색에 많은 시간을 보내게 된다.

- 적용되는 전역탐색법으로서 유전적알고리즘이 사용될 경우 집단의 다양성을 상실하기 쉽다.
- 다양성이 상실된 해 집단에 대한 탐색은 불필요한 동일한 탐색이 반복될 가능성이 많다.



(그림 1) 일반 하이브리드 시스템의 구조
(Fig. 1) Structure of General Hybrid System

이러한 문제점을 극복하기 위해서는 다음과 같은 사항의 고려가 필요하다.

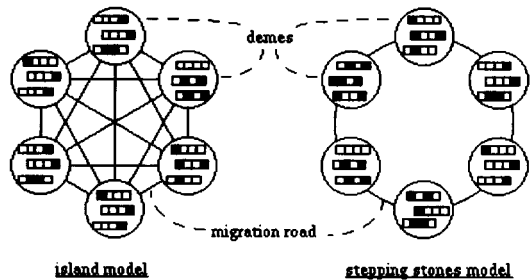
- 국소탐색법의 병렬화로 인한 탐색 시간의 증가를 방지할 수 있어야 한다.
- 전체 집단 중에서 소수의 엘리트해에 대한 탐색은 전체 탐색 시간을 줄일 수 있다.
- 동일한 탐색을 방지하기 위해 선택되는 엘리트해는 유전 형질상 다른 구조를 가질 필요가 있으며 이를 위한 구조적 조정이 필요하다.
- 탐색과정에서 발견한 우수 해에 대한 정보를 이용하기 위한 구조가 필요하다.

본 논문에서는 유전적알고리즘과 같은 전역탐색 알고리즘과 시뮬레이티드 어닐링, 타부탐색법과 같은 국소탐색법의 하이브리드화 또는 병렬화[13]에 대하여 논한다. 이때, 국소탐색법으로서 사용되는 방법으로서 무엇을 선택하는가 하는 것은 중요하지 않다. 본 논문에서는 다른 특성을 가지는 탐색법의 결합을 통해 전체 탐색 시간의 증대를 방지하기 위한 방법에 관해 논한다. 특히, 타부탐색법은 국소탐색법 중에서 최고의 성능을 나타내고 있음이 여러 연구 결과를 통해 밝혀지고 있다[3][12].

이러한 점에서 본 논문에서는 전역탐색법으로서 유전적알고리즘의 변형인 섬모델 유전적알고리즘과 최고 성능의 국소탐색법인 타부탐색법의 하이브리드화에 대하여 집중적으로 논한다.

2.1 섬모델 유전적알고리즘(Island Model of Genetic Algorithm)

본 논문에서는 위에서 언급한 사항을 포함하는 모델로서 병렬 유전적알고리즘의 섬모델(Island Model)을 이용한 병렬휴리스틱법의 고속화모델(FMPH)을 제안한다. 유전적알고리즘에 있어 집단의 다양성을 확보하기 위한 방법으로 섬모델이 있으며 (그림 2)에 두가지 기본적 형태인 섬모델(island model)과 디딤돌모델(steping stones model)에 대한 구조가 나타나 있다.



(그림 2) 병렬 유전적알고리즘의 구조
(Fig. 2) Structure of Parallel Genetic Algorithm

섬모델에 있어서는 전체 집단(population)은 몇 개의 섬(Island) 또는 서로 다른 부분 집합(sub-population or demes)으로 구성되며, 각각은 지리적으로 분리되어 서로 다른 경로를 통해 진화해 나간다. 섬모델이라 명명한 이유는 쉽게 추측 가능하며, 각각의 섬은 서로 다른 최적값을 향해 진화해 나간다. 섬의 각 해는 섬내의 다른 해와의 교배만이 허락된다. 이에 따라 각 섬의 유전자는 각각 다른 경로를 통해 발전된 유전 형질을 지니게 된다. 이 모델은 다른 몇 개의 국소 최적값을 찾아낼 수 있으며 전체 탐색 공간상의 어디에 우수한 해가 있는가를 판단하고자 할 때 도움이 된다. 또 다른 하나의 이점으로서 각 섬의 해는 다른 기준에 따라 진화해 나갈 수 있다는 것이다. 이러한 점은 다목적 최적화 문제에 사용될 수 있음을 뜻한다. 각 섬에서는 다른 기준을 바탕으로 선택될 수 있으며 다른 평가함수와 표현법을 사용할 수 있다.

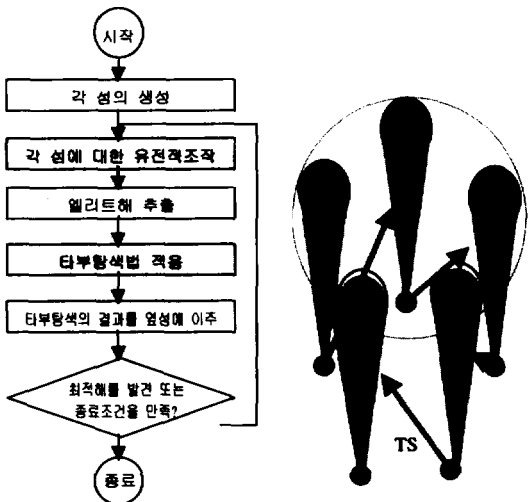
2.2 타부탐색법

Tabu search(TS)는 Glover[5][6][7]에 의해 처음으로 제시되었으며, 그때부터 점차 사용되게 되었다. 이 기법은 scheduling[3][8][9][12], timetabling, TSP와 같

은 문제에 성공적으로 적용되어 최적 또는 준최적해를 얻을 수 있었다[8][12]. 이 기법의 기본적 아이디어는 일련의 이동(move)을 통해 모든 가능해의 탐색 공간을 탐색하게 된다. 한 해에서 다른 해로의 이동은 가능해 중에서 최우수해를 택한다. 그러나, 국소해로부터의 탈출과 순화(cycling)를 방지하기 위해서 어떤 정해진 횟수 내에서 몇몇 이동을 금지 또는 타부(tabu)로 분류한다. 타부로 설정되는 이동은 일련의 이동에 대한 단기 또는 장기 기억에 그 기본을 둔다. 그 예로서, 반대 방향으로의 이동이 곧 또는 빈번하게 발생할 때 그러한 이동을 타부로 할 수 있다. 때로 그러한 타부가 증가하면, 이동이 극도로 제한되는 경우가 있다(overridden). 타부였던 이동을 해제하는 만족기준(aspiration criteria)은 탐색을 지금까지 발견되지 않았던 가장 좋은 해로 인도하게 된다. 이와 같이 타부탐색법은 탐색공간 상의 한 점을 바탕으로 탐색을 진행해 나가는 1점 탐색법 또는 국소탐색법에 속한다.

2.3 알고리즘

국소탐색법인 타부탐색법의 병렬화와 전역탐색법인 유전적알고리즘의 하이브리드 시스템을 구축하기 위하여, 양 알고리즘의 장점을 활용하면서 탐색을 고속화하는 병렬휴리스틱법의 고속화모델(Fast Model of Parallel Heuristics: FMPH)에 관한 알고리즘이 (그림 3)에 나타나 있다.



(그림 3) 병렬휴리스틱법의 고속화모델의 알고리즘 (Fig. 3) Algorithm of Fast Model of Parallel Heuristics

FMPH에서는 먼저 GA를 적용하기 위한 전체 해의 집단을 생성한 다음 섬모델을 적용하여 이를 몇 개의 섬으로 나눈다. 각각의 섬에 대해 유전적알고리즘을 적용하여 각각 다른 경로와 기준을 바탕으로 최적해를 향해 진화해 나가도록 한다. 이러한 GA를 통해 각 섬의 가장 우수한 엘리트해를 얻을 수 있다. 이때 이들 엘리트해는 섬모델을 통해 진화함으로써 서로 다른 형질을 보유할 수 있게 된다. 이들 엘리트해에 대해 다시 국소탐색법으로서 가장 뛰어난 성능을 보이고 있는 타부탐색법을 적용한다. 이러한 조작을 통해 엘리트해는 국소 최적해에 도달하게 된다. 또한 각 섬에서 얻은 엘리트해에 대한 형질을 이용하기 위해 국소 최적해를 옆의 섬의 가장 나쁜 해와 교체하는 조작을 통해 이주시킨다. 이주해 온 엘리트해를 새로운 구성원으로 하여 최적해를 발견하거나 정지 조건을 만족 할 때까지 다시 GA와 타부탐색법의 처리를 반복한다.

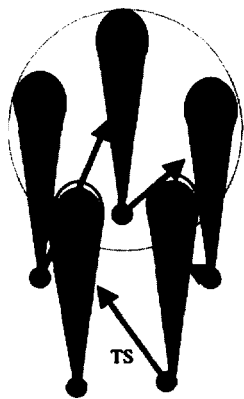
이때, 한 섬에서 GA를 통해 얻어진 엘리트해는 오른쪽에 있는 다른 한 섬에만 이주해간다. 이러한 이주의 제한을 통해 뛰어난 하나의 엘리트해가 전체 집단에 급속하게 퍼져나가는 것을 방지할 수 있다. 또한 GA 과정에서도 성능이 뛰어난 동일한 해가 집단중에 차지할 비율을 조정할 필요가 있다.

3. 잡숍(Job-Shop) 스케줄링 문제

본 논문에서는 이론의 유용성을 입증하기 위해 NP-hard문제 중에서도 가장 어려운 문제에 속하는 잡숍 스케줄링 문제(Job-Shop Scheduling Problem : JSP) [1][10]에 병렬휴리스틱법의 고속화모델을 적용하고자 한다. 먼저 잡숍 스케줄링 문제에 대해 살펴 보도록 한다.

고전적인 JSP는 다음과 같이 설명할 수 있다. 작업 순서가 결정되어야 하는 m 개의 다른 기계와 n 개의 다른 잡이 있다고 하자. 각 잡은 일련의 작업으로 구성되어지며 기계상에서의 작업 순서는 미리 결정되어 있다. 각 작업은 사용되는 기계와 고정된 작업시간에 의해 특정 지어진다. 잡과 기계에 관해서는 다음과 같은 가정을 바탕으로 한다.

- 하나의 기계는 동일한 잡에서는 한번만 사용된다.
- 다른 잡의 작업간에는 우선순위는 존재하지 않는다.
- 작업은 중단될 수 없다.
- 각 기계는 한번에 하나의 잡(job)만 처리할 수 있다.
- 유휴시간과 완료시간은 결정되어 있지 않다.



3.1 유전자 표현, 평가와 유전자 조작

유전적알고리즘에 의한 JSP의 해법[2][11][13]은 상당한 성과를 보이고 있으나 일반적으로 타부탐색법이나 시뮬레이티드 어닐링과 비교할 때는 탐색의 성능과 속도가 떨어진다 할 수 있다. JSP의 테스트 문제로 자주 이용되는 Muth와 Thompson[10]의 6 잡, 6 기계 문제(MT6)가 <표 1>에 나타나 있다. 여기에서 J1은 처리되어야 할 잡1을 나타내며, 하나의 잡에 대한 처리를 완료하기 위해서는 6종류의 다른 기계를 사용하는 6단계의 작업(Op1, Op2, , Op6)을 수행하여야 하며, 이때 기계의 사용은 잡에 따라 결정되어 있는 순서대로 사용되어야 한다. 잡 J1에 대한 첫번째 작업(Op1)의 값 3(1)은 기계3을 이용하여 1단위 시간 작업이 이루어짐을 의미한다.

본 논문에서는 유전자를 표현하기 위해 전작업 순서법(Total Operation Order Method: TOOM)이라 부르는 표현법을 사용한다[2]. TOOM에서는 수행되어야 하는 전체 작업의 순서를 유전자로 구성한다. MT6에는 모두 36개의 작업이 존재하며 이들 각 작업의 순서가 유전자의 구성요소가 된다. <표 2>는 <표 1>의 각 작업이 수행되어야 할 순서를 나타내고 있다. TOOM에서는 <표 2>의 각 행을 연결한 1차원 구조의 배열을 유전자로 하고 있으며, <표 2>는 MT6에 대한 하나의 최적해를 이루고 있다.

<표 1> MT6 문제
<Table 1> MT6 Problem

| c | J1 | J2 | J3 | J4 | J5 | J6 |
|-----|------|-------|------|------|------|-------|
| Op1 | 3(1) | 2(8) | 3(5) | 2(5) | 3(9) | 2(3) |
| Op2 | 1(3) | 3(5) | 4(4) | 1(5) | 2(3) | 4(3) |
| Op3 | 2(6) | 5(10) | 6(8) | 3(5) | 5(5) | 6(9) |
| Op4 | 4(7) | 6(10) | 1(9) | 4(3) | 6(4) | 1(10) |
| Op5 | 6(3) | 1(10) | 2(1) | 5(8) | 1(3) | 5(4) |
| Op6 | 5(6) | 4(4) | 5(7) | 6(9) | 4(1) | 3(1) |

<표 2> 최적해
<Table 2> One Optimal Solution

| | J1 | J2 | J3 | J4 | J5 | J6 |
|-----|----|----|----|----|----|----|
| Op1 | 6 | 1 | 3 | 2 | 9 | 4 |
| Op2 | 8 | 7 | 10 | 11 | 13 | 5 |
| Op3 | 16 | 15 | 14 | 17 | 18 | 12 |
| Op4 | 23 | 22 | 19 | 21 | 27 | 20 |
| Op5 | 26 | 25 | 28 | 30 | 29 | 24 |
| Op6 | 31 | 33 | 32 | 35 | 36 | 34 |

<표 3> 최적해의 간트 차트
<Table 3> Gantt Chart of Optimal Solution

| Machine | 10 | 20 | 30 | 40 | 50 | 55 |
|---------|---|------------------|------------------|------------------|------------------|----|
| m1: | 111 | 4444433333333333 | 6666666666666666 | 2222222222222222 | 2555 | |
| m2: | | | | 3 | | |
| m3: | 333331 | 2222255555555555 | 44444 | | 6 | |
| m4: | 3333 | 666 | 4441111111 | | 22225 | |
| m5: | | 2222222222 | | | | |
| m6: | | 33333333 | 6666666666 | 2222222222 | 1115555444444444 | |
| m0: | 123456789+123456789+123456789+123456789+123456789+12345 | | | | | |

JSP에서는 사용되는 기계에는 우선 순위가 결정되어 있기에 이러한 기술적 순서를 만족시키기 위해 유전자상의 각 요소는 잡별로 선행작업이 빠른 작업순서를 지녀야 할 필요가 있기에 하나의 해 S의 작업 s_{ij} 는 다음 조건을 만족시켜야 한다.

$$s_{ij} < s_{i'j} \text{ if } i' < i'', \forall s_{ij} \in S.$$

JSP는 Gantt Chart를 이용해서 전체 스케줄을 구성, 평가 할 수가 있다. 3에는 <표 2>로 표현된 가능해의 Gantt Chart가 표현되어 있다. 각 숫자는 잡의 종류를 나타내며 기계 3의 행에 있어서의 33333은 잡 3이 5단위 시간 동안 처리됨을 의미한다. <표 3>에 의하면 잡 3에 의한 기계사용의 순서는 <표 1>에 나타난 것처럼 3-4-6-1-2-5의 순서를 지니게 된다.

3.2 JSP에서의 FMPH

JSP에 FMPH를 적용하기 위한 절차에 대한 의사코드가 C언어의 표현법에 따라 (그림 4)에 나타나 있다. JSP에서는 전체집단을 몇 개의 소규모 그룹(섬)으로 분할하고 이들 섬에 대해 유전적조작을 한다. 본 연구에서는 전체 집단을 100개의 유전자로 구성하고 이들을 20개씩 묶어 5개의 소그룹(섬)을 구성하였다. 이들 섬에 대해 다음과 같은 유전적조작을 한다.

- ① 교차 : <표 2>를 참조할 때 하나의 해S1에서의 각 잡(Job)에 대한Op1의 순서는 (6 1 3 2 9 4)가 된다. 교차의 대상이 되는 다른 해S2에 대해서도 Op1의 순서를 결정할 수 있고 그 순서가 (1 2 3 4 5 6)이라고 하자. S1을 S2의 순서를 기준으로 재배열하는 조작을 교차조작으로 하였다. 이러한 교차조작에 의해 S1은 (1 2 3 4 6 9)의 순서로 재열된다.

② 동일한 해의 제거와 돌연변이 : <표 3>의 간트 차트에는 크리티칼 패스(Critical Path)가 박스안에 나타나 있다. 간트 차트가 결정되면 크리티칼 패스를 알 수 있고 이 크리티칼 패스를 이용하여 해의 동질성 여부를 결정하였다. 해가 동일한 평가치를 갖고 동일한 크리티칼 패스를 가진 경우 동일한 해로 판단하여 해집단에서 제거하였으며 이를 보충하기 위해 새롭게 만들어지는 해를 유전적조작에 있어서의 돌연변이 조작으로 보았다.

```

// 유전자에 대한 구조체 정의
int IslandNum = 5; // GA에서 구성되는 섬의 수
int gnum = 20; // 한 섬을 구성하는 유전자의 수
typedef struct gene {
    int _gene[gmax]; // 유전자의 코드화
    int fit; // 유전자의 평가치
    int cpV; // 유전자의 크리티컬 패스 값
} GENE;

GENE GAPopu[IslandNum][gnum]; // 유전자들로 구성된 섬의 배열

// FMPH 처리과정
void FmpOper()
{
    int tabulter = 500 // TS에서의 1회 탐색 수
    GENE best; // GA에서 얻어진 최우량 유전자
    GENE bestGene[IslandNum]; // 각 섬에서의 최우량 유전자 배열

    for(int i=0; i < IslandNum; i++)
        for(j = 0; j < gnum; j++)
            MakeGene(GAPopu[i][j]); // 모든 유전자의 초기화

    for(j = 0; j < StopCondition; j++) { // 종료 조건을 만족할 때 까지
        for(int i=0; i < IslandNum; i++) { // 각 섬의 수 만큼
            best = GaOper(GAPopu[i]); // GA에서 얻어진 각 섬의 최우량치
            TabuOper(best, tabulter); // 최우량 유전자에 대한 TS
            bestGene[i] = best; // TS에서 얻어진 각 섬의 최우량치
        }
        for(int i=0; i < IslandNum; i++)
            // 각 섬의 최일등 유전자를 옆의 섬의 최우량 유전자로 대체
            GAPopu[i][gnum-1] = bestGene[(i+1)%IslandNum]; // 유전자
            //의 이주
    }
}
    
```

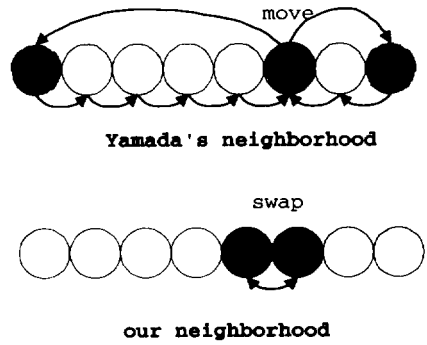
(그림 4) FMPH의 의사코드
(Fig. 4) pseudo code of FMPH

이러한 유전적조작을 통해 각 섬마다 얻어지는 하나의 엘리트해에 대해 타부탐색법을 적용한다. 타부탐색법은 하나의 해에 대해 국소탐색을 진행해나가기 때문에 탐색의 연속성과 깊이를 확보해 줄 필요가 있다. 이를 위해 본 JSP에 대해서는

① 타부탐색이 카운터 종료 500회 동안 새로운 해를 발견하지 못하면 그 탐색을 종료하도록 하였다. 그러나 탐색중에 새로운 해를 발견하게 되면 다시 카

- 운터를 0으로 설정하여 탐색의 연속성을 꾀했다.
- ② 각 섬에서 유전적조작과 타부탐색을 통해 얻어진 엘리트해를 옆의 다른 한 섬에 이주시킨다. 이주해은 엘리트해는 섬의 구성원과는 다른 유전적 형질을 지니고 있을 가능성이 높다.
- ③ 이주해은 엘리트해를 포함한 섬의 해에 대해 유전적 조작을 행한다.
- ④ 유전적조작과 해의 이주 타부탐색법을 최적해의 발견 또는 정지조건이 만족될 때까지 행한다.

유전적알고리즘을 통해 추출된 엘리트해에 대해 국소탐색법으로서의 타부탐색법을 적용하고자 하는 경우 근방의 구조와 타부 리스트의 구조에 대한 적절한 선택이 성능에 많은 영향을 끼친다. (그림 5)에는 Yamada의 근방구조[16]와 우리가 채택한 근방구조의 비교가 나타나 있다.



(그림 5) 해의 근방 구조
(Fig. 5) Neighborhood of Solution

본 논문에서는 각 작업으로 표현된 인접한 유전자의 구성요소를 교환함으로써 근방이 얻어진다. 이들 근방 해중 가장 평가치가 좋은 해가 다음번의 탐색의 대상으로 선택되며 선택된 근방 해의 교환된 작업의 번호가 타부 리스트로서 기억된다. 기억된 타부 리스트는 다음 탐색의 방향을 제어하게 된다.

이러한 전역탐색법과 국소탐색법의 이원 구조를 가지는 FMPH는 다음과 같은 단점을 지닌다.

FMPH의 단점

① 이주해은 엘리트해의 우량 형질이 집단의 다양성을 잃지 않도록 하기 위한 다양성 유지 정책이 필요하다. 이를 위해 첫째, 섬의 수를 증가시키는 것을 고

려할 수 있으며 이는 탐색 시간의 증가를 유발한다. 둘째, GA의 탐색 구조 내에서 각 섬마다 다른 평가 기준을 사용함으로써 다른 경로를 통해 해를 발견시켜 나갈 수 있으며 이러한 구조는 다양성 유지를 위한 구조가 될 수 있다.

- ② 기 획득된 우량 유전자가 GA에서의 유전적 조작에 의해 치사되지 않도록 하기 위한 세심한 주의가 필요하다. 이를 위해 FMPH의 유전적 조작은 일반 GA의 한 변형인 정상 상태(Steady State) GA의 구조를 취하고 있다.
- ③ 탐색의 후기단계 즉, 국소해 근방에 있어서 전역탐색법으로서의 GA의 탐색능력은 극도로 저하된다. 이는 탐색이 진행됨에 따라 각 섬에서의 해의 다양성이 상실되고, 형질의 변화를 기대하기가 어려워지기 때문이다. 이로 인해 해의 개선을 위해서는 타부탐색법에 대한 의존도가 심화되어 간다.
- ④ 실험을 위해서는 전역탐색법과 국소탐색법 양자를 모두 구현해야 하는 부담이 있다.

위의 단점에도 불구하고 FMPH는 다음과 같은 많은 장점을 가진다.

FMPH의 장점

- ① 전역탐색법과 국소탐색법의 내부 구현방식에 의존하지 않고 각각의 방법의 장점을 융합할 수 있기에, 해 탐색의 견고성을 확보할 수 있다.
- ② 탐색의 초기단계에서는 GA의 전역탐색능력을 활용하여 가능성이 있는 해공간을 보다 신속하게 선택할 수 있으며 이를 통해 전체 탐색 시간을 줄일 수 있다.
- ③ GA에서 추출된 우량해에 대해 국소탐색법을 적용함으로써 우량해 근방에 대한 집중적인 국소 탐색이 가능해지고 탐색의 깊이를 더 할 수 있게 된다. 이는 우량해의 진화 경로를 변경하여 형질구조의 변화를 유발하고 해의 다양성을 확보하는 계기가 된다.
- ④ 후기단계에서는 타부탐색법의 국소탐색능력을 통하여 GA만으로는 탈출하기 어려운 국소해로 부터의 탈출이 가능하게 된다.
- ⑤ 각 섬에서 얻어진 엘리트해 만을 바탕으로 국소탐색이 이루어지기 때문에 병렬화와 비교할 때 전체적인 탐색의 고속화를 꾀할 수 있다.
- ⑥ 전체 집단이 아닌 각 섬의 엘리트해에 대해서만 선

별적으로 집중탐색을 함으로써 불필요한 탐색을 없애고 탐색을 고속화 할 수 있다.

- ⑦ 선택적 병렬화의 단점을 전역탐색법으로서의 GA에 의한 유전적 조작을 통하여 흡수하고 있다.
- ⑧ 단순한 구조를 통하여 전역탐색법과 국소탐색법의 장점을 융합하고 있다.

4. 실험과 결과

고속 병렬 타부탐색법의 성능을 평가 하기위해 가장 많이 알려진 잡속 스케줄링 문제의 벤치마크로서 Muth와 Thompson[10]의 MT10 (10 job × 10 machine)과 MT20(20 job × 5 machine), 그리고 Lawrence의 문제 중에서도 난문에 속하는 7개의 벤치마크 문제(LA21, LA24, LA25, LA27, LA29, LA38, LA40)를 대상으로 선택했다. 실험의 환경은 다음과 같다. GA와 TS를 반복함에 있어 1회 탐색에 있어서는 GA를 100세대분 진화시킨 다음 TS의 과정으로 이동하여 500회 탐색하며 각섬에 대한 탐색에서 새로운 해의 발견이 이루어지지 않는 경우 TS의 과정을 500회씩 추가해 나가는 과정을 되풀이 하였다. 전체 탐색시간은 MT10을 10분, MT20을 30분으로 고정하였다. 실험의 환경으로는 펜티엄 프로 200MHz의 컴퓨터에서 Free-BSD에서 C언어로 프로그램 하였다. 다음 <표 4>는 각 문제를 위의 조건하에서 100번 반복한 결과이다.

<표 4> MT10과 MT20 문제에 대한 실험 결과
<Table 4> Experimental Results of MT10 and MT20

| | | MT10 | | 최적값 | | | MT20 | | 최적값 |
|------|----|------|------|-----|------|-----|------|------|-----|
| 방법 | 시간 | 평균 | 발견횟수 | | 방법 | 시간 | 평균 | 발견횟수 | |
| GA | 5분 | 952 | 0 | | GA | 20분 | 1206 | 0 | |
| PTS | 5분 | 939 | 15 | | PTS | 20분 | 1187 | 6 | |
| FMPH | 5분 | 933 | 72 | | FMPH | 20분 | 1174 | 23 | |

FMPH를 적용함으로써 단위 시간당 최적해의 발견 횟수 및 평균 탐색 값이 GA 또는 병렬타부탐색법(PTS)를 적용할 때보다 월등히 양호하게 되었다. MT10에 있어서는 최적해 930을 72번이나 발견할 수 있었다. 이것은 10분이라는 제한된 시간에서의 결과라는 것을 생각할 때 보다 장시간에 걸쳐 탐색을 행한다면 보다 나은 결과를 얻을 수 있을 것이다. MT20에 대해서도 최적해의 발견 횟수가 23번에 달하고 평균치

도 최적해 1165에 근사한 1174에 달하고 있다.

<표 5>는 Thompson과 Lawrence의 9문제에 대해 다른 연구결과와 비교한 표이다. <표 5>에서 얻어진 값은 탐색시간을 특정하지 않고 장시간(약 3시간 내외) 실험을 통해 얻어진 결과이다.

Thompson과 Lawrence의 9문제는 JSP문제중에서는 가장 유명하고 어려운 문제이다. 각 문제에 대해 20회씩 실험을 하였다. 이러한 문제에 대해 아래와 같은 양호한 결과를 얻을 수 있었음은 본 FMPH 구조의 강건성(Robustness)를 입증한다 할 수 있을 것이다.

이상의 결과를 바탕으로 판단할 때 FMPH는 보다 양질의 해를 보다 고속으로 탐색해 나갈 수 있는 매우 우수한 휴리스틱법이라 결론 지을 수 있을 것이다.

<표 5> 실험 결과의 비교
<Table 5> Comparison of Experimental Results

| Problem | Size | Optimal | FMPH | | RKGA | DT | LAL | YRN |
|---------|-------|------------|------|------|------|------|------|------|
| | | | Avg. | Best | | | | |
| MT10 | 10×10 | 930 | 933 | 930 | 937 | 930 | 930 | 930 |
| MT20 | 20×5 | 1165 | 1172 | 1165 | 1165 | 1165 | 1165 | - |
| LA21 | 15×10 | 1047, 1053 | 1052 | 1046 | 1055 | 1048 | 1063 | 1050 |
| LA24 | 15×10 | 935 | 947 | 939 | 966 | 941 | 941 | 943 |
| LA25 | 15×10 | 977 | 989 | 985 | 987 | 979 | 979 | 985 |
| LA27 | 20×10 | 1235, 1269 | 1268 | 1260 | 1256 | 1242 | 1242 | 1262 |
| LA29 | 20×10 | 1160, 1195 | 1191 | 1183 | 1179 | 1182 | 1182 | 1188 |
| LA38 | 15×15 | 1184 | 1218 | 1210 | 1217 | 1203 | 1203 | 1209 |
| LA40 | 15×15 | 1222 | 1232 | 1229 | 1234 | 1233 | 1233 | 1235 |

RKGA : Norman and Bean[11] LAL : Laarhoven's SA[15]
DT : Dell'Amico's TS[3] YRN : Yamada's SA[16]

5. 결 론

단일 CPU구조를 사용하는 일반 범용 컴퓨터 상에서 휴리스틱 알고리즘의 병렬화 또는 하이브리드화를 도모할 때, 잘 조직화된 구조를 이용한다면 전체 탐색시간의 단축 뿐만 아니라 보다 더 나은 결과를 얻을 수가 있다. 이 논문에서 제안하는 병렬휴리스틱법의 고속화모델은 기존의 1점 탐색법으로서의 타부탐색법, 시물레이티드 어닐링법 등의 메타 휴리스틱법과 다점, 전역탐색법으로서의 유전적알고리즘의 융합을 꾀하는 하이브리드화와 그에 따르는 병렬화에 대한 연구로서, 1점 탐색법으로서의 타부탐색법을, 다점 탐색법으로는 유전적알고리즘을 채택하였다. 다수의 하이브리드 모

델이 제안되고 있으나 이 논문에서 강조하고자 하는 점은 보다 효율적이며 고속으로 양질의 해를 탐색할 수 있는 하이브리드 모델의 구조에 그 초점을 두고 있다. 잡속 스케줄링 문제에 적용한 병렬휴리스틱법의 고속화모델은 매우 뛰어난 성능을 보여 주고 있으며 모델의 유용성을 대변하고 있다.

참 고 문 헌

- [1] Applegate, D. and Cook, W., A computational Study of the Job-Shop Scheduling Problem, *ORSA Journal on Computing*, Vol.3, pp.149-156, 1991.
- [2] Cho, B.J., Hong, S.C., Okoma, S., Jop Shop Scheduling using Genetic Algorithm, *Critical Technology*, Vol.2. pp.351-358, 1996.
- [3] Dell'Amico, M. and Trubian, M., Applying Tabu Search to the Job-Shop Scheduling, *Annals of Operations Research*, Vol.41, pp.231-252, 1993.
- [4] Gen, M. And Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley, New York, 1997.
- [5] Glover, F., Tabu Search-Part I, *ORSA Journal on computing*, Vol.1, pp.190-206, 1989.
- [6] Glover, F., Tabu Search-Part II, *ORSA Journal on computing*, Vol.2, pp.4-32, 1990.
- [7] Glover, F., Taillard, E., and Werra, D., A user's guide to tabu search. *Annals of Operations Research*, Vol.41, pp.3-28, 1993.
- [8] Laguna, M., Barnes, J. W., and Glover, F., Tabu search methods for a single machine scheduling problem, *Journal of Intelligent Manufacturing*, Vol.2. pp.3-74, 1991.
- [9] Laguna, M., and Glover, F., Bandwidth packing: A tabu search approach, *Management Science*, Vol.39(4). pp.492-500, 1993.
- [10] Muth, J.F. and Thompson, G.L., *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, NJ, 1963.
- [11] Norman, B.A. and Bean, J.C., Random Keys Genetic Algorithm for Job-Shop Scheduling, Vol.3-2, pp.145-156, 1997.
- [12] Nowicki, E. and Smutnicki, C., A Fast Tabu Search Algorithm for the Job Shop problem,

Management Science, Vol.42, pp.797-813, 1996.

- [13] Taillard, E., Parallel Tabu Search Techniques, *ORSA Journal on Computing*, Vol.6, pp.108-117, 1994.
- [14] Tsujimura, Cheng, R., and Y., Gen, M., Improved Genetic Algorithms for Job-Shop Scheduling Problems, *Engineering Design & Automation*, Wiley, New York, Vol.3, No.2, pp.133-144, 1997.
- [15] Van Laarhoven, P.J.M., Arts, E.H.L., and Lenstra, J.K., Job Shop Scheduling by Simulated Annealing, *Operations Research*, Vol.40, pp.113-125, 1992.
- [16] Yamada, T., Rosen, B.E., and Nakano, R., Critical Block Simulated Annealing for Job Shop Scheduling, *T.IEE, Japan*, Vol.114-C, No.4, 1994.



홍 성 찬

e-mail : schong@hucc.hanshin.ac.kr
 1982년 고려대학교 통계학과 졸업 (학사)
 1990년 게이오대학교 관리공학과 졸업(공학석사)
 1994년 게이오대학교 관리공학과 졸업(공학박사)

1994년~1995년 (주)LG-EDS시스템 컨설팅부문 부장
 1995년~1997년 상명대학교 정보과학과 전임강사
 1997년~현재 한신대학교 컴퓨터정보통신학부 조교수
 관심분야 : 소프트웨어공학, 최적화, CALS/EC



조 병 준

e-mail : bjcho@yeungjin-c.ac.kr
 1985년 경북대학교 경영학과 졸업 (학사)
 1995년 일본 게이오대학교 관리공학과 졸업(이공학 석사)
 1998년 일본 게이오 대학교 관리공학과 박사과정 수료

1986년~1987년 삼성반도체통신 근무
 1998년~현재 영진전문대학 산업정보계열 전임강사
 관심분야 : 분산 객체 컴퓨팅, 최적화, 웹 응용