韓國經營科學會誌
第17卷 第3號
1992年 12月

작업 일정계획문제 해결을 위한 유전알고리듬의 응용

김석준* · 이채영*

Application of Genetic Algorithms to a Job Scheduling Problem

Seok Jun Kim* and Chae Y. Lee*

Abstract

Parallel Genetic Algorithms (GAs) are developed to solve a single machine n-job scheduling problem which is to minimize the sum of absolute deviations of completion times from a common due date. (0,1) binary scheme is employed to represent the n-job schedule. Two selection methods, best individual selection and simple selection are examined. The effect of crossover operator, due date adjustment mutation and due date adjustment reordering are discussed. The performance of the parallel genetic algorithm is illustrated with some example problems.

1. GENETIC ALGORITHM 개요

GAs이란 자연계의 적응 과정과 유전자의 변이 과정을 도입하여, 이와 유사한 인공적인 유전자 해를 만들어 거듭된 적응 과정을 통하여 해를 구하려는 발견적 기법(heuristic technique)이다. John Holland [9]에 의해서 제시된 GAs는 기존의 전통적인 방법과는 많은 차이가 있다.

첫째로 GAs는 parameter를 그대로 사용하는 것이 아니라 GAs의 구조에 맞는 유전자의 형태로 변환된 parameter의 집합으로 사용한다. 즉, 표현형이 십진수 하나로 표현된 것이라고 한다면, 인자형은 다섯개의 bit로 구성된 이진수로 표현된다. 둘째, GAs에서는 하나의 점으로 이동하면서 탐색해 나가는 것이 아니라 여러 점을 동시에 변화시키며 탐색해 나간다.

* 한국과학기술원 경영과학과

개체 하나가 해집합에서 하나의 점을 나타내는 것이라면, GAs에서는 개체군들의 군집을 동시에 변화시켜 나간다는 것이다. 셋째, GAs에서는 목적함수 값만을 사용할 뿐이지, 미분값이나 그 밖의 다른 방법에 의한 값을 필요로 하지 않는다. 마지막으로 GAs는 다음 단계로 넘어가는 전이 규칙이 어떤 값의 계산에 의한 결정된 변환이 아닌 확률적인 값에 의한다. 그러나 적자 생존에 의한 방법이기 때문에 Random Search와는 근본적으로 다르다.

이러한 GAs의 전형적인 응용 분야로는 travelling salesman problem[7], graph coloring[4], job shop 스케쥴링[3] 등과 같은 combinatorial optimization 문제가 포함된다.

GAs에서 해의 군집은 환경에의 적응 과정을 거치면서 더 좋은 해를 향해 나아가게 되는데, 변이 과정에서 사용되는 기본 연산자는 복제(Reproduction), 교차 (Crossover), 그리고 돌연 변이 (Mutation)이다.

병렬 GA은 위의 세 연산자 중 교차 연산에서 큰 차이가 있다. 즉, 병렬적인 방법은 교차 변이를 할 때, 동료관계를 구성하는데 있어서 전체 집단을 대상으로 하지 않고, 일정한 규칙에 의해 규정된 부집단 속에서만 동료를 찾는 방법이다. 이 방법은 어떤 규정을 가지느냐에 따라 다음의 두 가지 모형으로 구분된다[6,11].

① 고립 부집단 모형

② 인접 동료 모형

고립 병렬 부집단 모형은 최초 집단의 구성 때 부터 완전히 고립된 부집단을 구성하고, 하나의 부집단이 독립적인 알고리듬을 수행하는 방법이다. 큰 규모의 문제는 전체 집단의 개체수가 많아야 하기 때문에 여러 과정을 수행하는 데에 많은 노력과 시간이 든다. 이때 부집단을 구성하고 각 부집단이 독립적인 알고리듬을 수행한다면 시간적인 절약을 얻을 수 있다. 또한 부집단의 특성에 따라 전체집단이 다양성을 유지할 수 있다는 장점도 있다. 이 방법을 간단히 기술하면 〈표 1〉과 같다. 여기에서 NGA는 부분 부집단이 수행하는 알고리듬을 의미하며, 정보 교환(Communication)은 부집단 끼리의 개체 교환 과정이다. 정보 교환은 보통 그 부집단의 최우수 개체를 다른 부집단으로 보내는 방법이 쓰여지고 있다.

인접 동료 모형은 전체 집단을 고립적인 부집단으로 구성하지 않고, '거리에 의한 고립[11]'에 의해 구성하는 방법이다. 동료로 선택될 수 있는 개체는 인접한 개체에서만 가능한 것이고, 어느 한 개체가 정확히 어느 집단에 소속해 있다는 표현을 할 수는 없다.

본 논문에서는 고립 부집단 모형을 이용한 작업 스케쥴링 문제를 다루고자 한다.

〈표 1〉 부분 부집단의 수행과정 알고리듬

```
NGA
    begin
        initialization;
        evaluation;
        while (not done)
        begin
            communication;
            selection;
            recombination;
            evaluation;
        end;
    end.
```

2. 문제의 정의 및 특성

앞으로 풀고자 하는 문제는 다음과 같이 정의된다.

"하나의 기계에 n개의 작업이 있고, 이 각각의 작업들은 공통의 만기일을 가지고 있으며, 만기일 보다 빨리 끝나는 작업과 만기일보다 늦게 끝나는 작업이 모두 같은 비중의 페널티가 주어진다고 할 때, 이 페널티의 합을 최소로 하는 것."

이러한 문제에서 사용되는 용어를 다음과 같이 정의 한다.

P_i : 작업 i의 작업 시간(processing time)

d : 만기일 (due date).

C_i : 작업 i의 작업 종료 시간(completion time).

MS : $\Sigma_i P_i$, i=1……n

Δ : $P_1+P_3+\cdots\cdots+P_n$ n이 홀수일 경우

$P_2+P_4+\cdots\cdots+P_n$ n이 짝수일 경우

여기에서 작업은 다음과 같이 작업 시간이 짧은 순서로 번호가 매겨져 있다고 가정한다.

$$P_1 \le P_2 \le \cdots\cdots \le P_n$$

이러한 문제는 Δ에 의해 두가지 유형으로 구분된다. 만약 $\Delta \le d$이면 기존의 연구에 의해 쉽게 풀리는 알고리듬이 개발되어 있다. $\Delta > d$가 되면 이 문제는 NP-complete [8]문제로서 쉽게 풀리지 않는다. 그러나 기존의 연구[1,2,12]에 의하면 최적해는 다음과 같은 특성을 갖는다.

특성 1 : 최적스케쥴에서는 작업과 작업사이에 유휴 시간이 없다.

특성 2 : 최적해는 최단 작업을 중심으로 그 이전은 LPT (longest processing time), 그리고 그 이후는 SPT (shortest processing time)의 순으로 정리된다.

특성 3 : 최적해는 만기일 (d)을 중심으로 그 이전 또는 만기일에 끝난 작업은 LPT, 그 이후에 끝난 작업은 SPT의 순으로 정리된다.

특성 4 : 최적해는 다음의 식을 만족하는 u, v 보다 작은 작업은 u의 앞에 수행하거나, v의 뒤에 수행될 수 없다.

$H_u=(P_1+P_u)/2+\Sigma_{k=2..u-1} P_k < d$

$H_{u+1} \ge d$

$T_v=(P_1+P_v)/2+\Sigma_{k=2..v-1} P_k < MS-d$

$T_{v+1} \ge d$

또한 본 연구에서는 d가 상당히 제한적인 경우 일반적으로 택하여진 '스케쥴의 첫번째 작업이 시각 0에서 시작함'을 가정한다. 여기서 제한적인 경우라 하더라도 최적해가 시각 0에서 시작되지 않는 경우가 있으므로 위 가정이 현실성을 고려하지 않은 점은 Baker and Scudder[2]에 의해 지적된 바 있다.

3. 병렬 Genetic 알고리듬의 적용

3.1 문제의 표현 방법

작업 스케쥴링문제에 있어서 최적해의 기본적인 특성을 살펴 보면 해의 모양이 최단 작업을 중심으로 V-형이 됨을 알 수 있다. 즉, 작업 순서를 (E, 1, T)라고 한다면, T집합은 오름차순으로 정리되어 있고, E집합은 내림차순으로 정리되어 있다.

예를 들면

5 4 1 2 3 6 (숫자는 작업 번호)

일때, E=(5, 4) T=(2, 3, 6)과 같이 내림차순과 오름차순으로 되어있다. 따라서 하나의 작업 i가 T,E 두개의 어느 집합에 속하느냐에 따라서 다음과 같은 값을 가지는 (0,1) 이진 변수로 변환할 수 있다.

$a_i=0$ (if $a_i \in T$)
$=1$ (if $a_i \in E$)

위의 이진 변수에 의한 표현 방법은 문제의 해를 항상 V-형이 되게 한다. 만일 다음과 같은 인자형의 해가 있다고 하자.

작업번호 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
인자형 해 0 1 1 0 1 1 0 1 1 1 0 1 0 1 0

이런 인자형 해에서 작업 번호가 큰 작업으로부터 시작해서 1의 값을 가지는 작업을 찾아서 그 작업 번호를 내림차순으로 나열하고 모두 끝나면 1번 작업 부터 0의 값을 가지는 작업들을 오름차순으로 배열하면 같은 표현형 해를 구할 수 있다. 이 해의 의미는 처음 14번 작업을 수행하고, 다음 12번, 10번의 순서로 작업한다는 것을 의미한다. 즉, 위 인자형에 대응하는 작업 순서는 다음과 같다.

표현형 해 14 12 10 9 8 6 5 3 2 1 4 7 11 13 15

해집합의 크기를 비교해 보면, 인자형 해로 나타날 수 있는 전체 해집합의 원소의 갯수는 2^n개이다. 여기에서 가능한 해의 수를 따져 본다면, 1번 작업의 경우는 0이 되나 1이 되나 똑같은 작업 순서를 나타내기 때문에 결국 가능한 해의 갯수는 2^{n-1}개가 된다. 그러나, 표현형 해의 경우는 총 갯수가 최악의 경우 n!이 된다. 그러므로 인자형 해를 이용할 경우 최적을 찾기 위한 해집합의 크기를 줄이고 시작하는 결과가 되어 그만큼 해공간을 좁혀주는 효과를 얻을 수 있다.

3.2 병렬 부집단의 구성

본 스케쥴링 문제의 경우 작업 순서에서 최초 작업에 따라 전체 해의 값이 큰 영향을 받게 된다. 2장의 특성 4에 의하면 첫 두식을 만족하는 u보다 작은 작업은 최적해에서 최초 작업이 될 수 없다. 따라서 전체 해집단에서 u보다 작은 작업으로 시작하는 작업 일정은 제외할 수 있다.

또한, 부집단을 구성하게 되면, GA의 초기 수행과정에서 나타나는 초개체 (Super Individual)의 영향을 줄일 수 있다. 단순히 전체 집단을 하나의 집단으로 구성하였을 경우는, 초기 초개체의 영향에 의해서 다음 세대의 개체들이 그 모양과 거의 같은 모습으로 수렴하게 되고, 초개체의 복제 개체들의 비율이 커지게 되므로 부분 최적 (local optimum)에서 벗어나지 못할 가능성이 크다.

그러나, 처음부터 서식 환경에 따라 부집단을 구성하게 되면, 한 부집단에서 초개체가 나타나더라도, 다른 부집단에는 그리 큰 영향을 끼치지 못하게 되고, 다른 부집단의 서식 환경 내에 개체들 중에 전체 최적해가 있다면 그것을 찾을 가능성이 크다는 것이다.

본 논문에서는 서식 환경을 도식적인 의미에서 구분하고, 또 작업 시간이 큰 작업에 더 큰 비중을 두어서 부집단을 구성하기로 한다. 즉, 작업 순서의 최초 작업에 따라 부집단을 구성한다. 최초 작업을 모두 같은 작업으로 구성하

〈표 2〉 작업 순서 중 최초 작업이 될 수 있는 작업

부집단	최초 작업 번호
1	n
2	n-1
3	n-2, n-3
.	.
.	.
k	$n-2^{k-2}$, …, $n-2^{k-1}+1$

게 되면 부집단의 수가 엄청나게 늘어나기 때문에 다음과 같은 구조를 사용하게 되면, 부집단의 수를 줄이면서 구분할 수 있다.

위에서 k는 부집단의 수이며 마지막 구성 원소는 특성 4를 만족하는 u의 값에 따라 결정된다.

3.3 연산자

3.3.1 교차변이

본 논문에서는 교차변이 연산중 최적화문제의 응용에서 주로 다루어지고 있는 이중점 교차변이를 고려한다. 교차점이 2인 교차, 즉 이중점 교차변이의 예가 [그림 1]과 [그림 2]에 나타나 있다.

[그림 1] 교차 변이 이전의 인자형 해와 표현형 해

(교차 변이 이전의 해)

인자형 해

개체 1 : 1 0 0 | 1 0 0 0 1 1 0 1 | 0 0 1
개체 2 : 0 1 1 | 1 1 0 0 0 1 0 1 | 1 0 1
교차점 1 교차점 2

표현형 해

개체 1 : 14 11 9 8 4 1 2 3 5 6 7 10 12 13
개체 2 : 14 12 11 9 5 4 3 2 1 6 7 8 10 13

[그림 2] 교차변이 이후의 인자형 해와 표현형 해의 변화

(교차변이 이후의 해)

인자형 해

개체 1 : 1 0 0 | 1 1 0 0 0 1 0 1 | 0 0 1
개체 2 : 0 1 1 | 1 0 0 0 1 1 0 1 | 1 0 1
교차점 1 교차점 2

표현형 해

개체 1 : 14 11 9 5 4 ① 1 2 3 6 7 8 10 ② 12 13
개체 2 : 14 12 11 9 8 4 ① 3 2 1 5 6 7 10 ② 13
(밑줄이 그어진 곳은 변화가 일어난 곳)

교차점의 위치가 결정되면 전체 작업 순서는 [그림 3]과 같이 세 부분으로 구분된다.

[그림 3] 교차 변이시 작업순서의 구분

개체 ├──────────────────────────┤
1번 작업 교차점1 교차점2 n번 작업
| A | B | C |

C부분은 인자형 해에서나 표현형 해에서나 모두 변화가 일어나지 않는다. C의 작업들은 작업 시간이 긴 비중이 큰 작업들인데 이들은 변화하지 않고, 그대로 전 세대의 양극 구조를 유지한다. 특히 n번 작업의 경우는 어떠한 경우에도 교차변이를 통해서는 변화하지 않는다.

B부분은 실제로 두 개체간에 유전자 교환이 일어나는 부분이다. 이 교환의 형태는 표현형 해에서는 두 부분으로 구분되어 나타난다. A부분 중에 0의 값을 가지는 작업들과, 1의 값을 가지는 부분의 두개로 구분 되어 두 개체간에 집합의 단위로 교환된다. [그림 2]에서 개체 1의 ①은 원래 개체 2에 있던 유전자들이고, 개체 1에 있는 ②는 개체 2에 있던 것들이다.

A부분은 개체간의 유전자 교환이나 구조 변화는 일어나지 않으나, B부분의 유전자 갯수 차이에 의해 왼쪽 쏠림이나 오른쪽 쏠림 현상이 일어나는 부분이다.

만약 교차점이 2이상이라고 한다면, 2일때보다 더 큰 변화를 일으키게 될 것이다. 인자형 해에서 교차점이 2일때는 표현형 해에서는 교차점이 4가 된다. 그러므로, 다중점 교차변

이에서 교차점이 둘 이상이 되면 표현형 해에는 네개 이상의 교차점이 생기게 되므로, 다음 세대의 개체에는 그 전 세대의 특질이나 구조를 유지하기가 힘들게 되어 그 이전부터 유지해 온 정보를 다음 세대에 이용하지 못하게 된다. De Jong[5]의 연구에 따르면 다중점 교차변이는 교차점이 많을수록 수렴 속도가 느리고 성과가 떨어지는 것으로 나타나 있다. 이것은 교차점이 커질수록 교차변이에 의해 그 이전의 구조나 정보들이 파괴될 가능성이 커지기 때문이다.

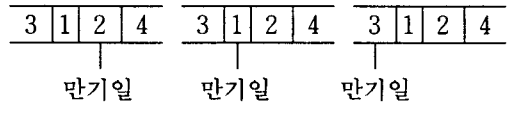
3.3.2 돌연변이와 순위 재조정(Mutation and Reordering)

돌연변이는 단순 GA에서는 임의의 유전 형질 하나를 다른 인자형으로 만드는 것을 말한다. 돌연변이가 일어나는 유전자도 무작위로 선택하며, 만일 유전 인자형이 $\{a_1, a_2, \cdots, a_n\}$과 같이 두개 이상이면 바뀌는 인자형도 임의로 선택된다. 이러한 돌연변이를 주는 이유는 개체의 다양성을 유지하기 위해서이다. 부분 최적해에 머무를지도 모르기 때문에, 계속 임의로 구조를 바꾸게 되면 전체 집단의 다양성도 증가되고, 이 다양성의 증가로 전체 최적해에 도달할 수 있는 가능성이 커지기 때문이다. 그러나 이 돌연변이도 너무 많이 사용하게 되면, 단순 무작위 탐색법과 유사하게 되어 수렴속도가 떨어지게 된다.

순위 재조정의 방법도 돌연변이와 같이 구조의 변화를 주어 다양성을 유지하고 최적해에 더 가까운 해를 얻기 위한 방법으로 쓰여진다.

이 논문에서는 돌연변이 방법과 순위 재조정의 방법을 원래의 단순 GA에서와 같은 다양성을 유지하기 위한 방법으로 쓰지 않고, 만기일을 이용한 복합 방법(Hybrid Method)의 의미로 이용한다. 본 문제의 특성에 의하면 최적해에 있어서 1번 작업은 [그림 4]와 같이 만기일에 걸쳐서 나타나게 된다.

[그림 4] 만기일과 1번 작업의 관계



즉, 1번 작업은 최적 작업 순서에서 만기일의 부근에 있게 된다는 것이다. 그런데 무작위로 복제하고 무작위 교차점을 잡고 교차 변이를 하게 되면, 이러한 특성을 만족시키지 못하는 개체들이 많이 늘어나게 된다. 이러한 개체들을 인위적인 돌연변이와 순위 재조정을 통하여 1번 작업을 만기일 부근으로 옮기는 효과를 얻고자 한다.

예로서 인자형 표현형이 다음과 같은 작업 일정계획을 보자.

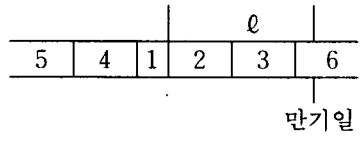
[그림 5] 1번 작업과 만기일의 차이

인자형 해

1 0 0 1 1 0 0 0 1 0 1 0 0 1

표현형 해

14 11 9 5 4 1 2 3 6 7 8 10 12 13



위 일정 계획에서 [그림 5]와 같이 1번 작업이 E 집합에 포함 되어 있다고 가정하면 1번 작업이 끝나는 시점에서 만기일 까지가 우리가 줄여야 할 구간으로 정의된다. [그림 5]에서는 ℓ구간이 이를 나타내고 있다. 위의 경우는 1

번 작업이 끝나는 시각이 만기일 보다 앞에 있는 경우로서, 1번 작업이 끝나는 시점을 뒤로 밀어야 한다. 따라서 1번 작업보다 뒤에 하기로 한 작업들 중에 어떤 것을 1번 작업의 앞에 위치하게 하면 차이 구간이 좁아지게 된다. 돌연변이에서는 이 때 작업 선택의 기준으로 다음의 수식을 도입한다.

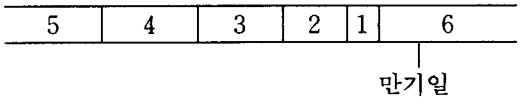
$$\min_i \{ | C_i-(\text{만기일}-C_1) | \} \; i \in T,\ i=2,\cdots,n$$

순위 재 조정의 방법에 있어서는 두가지의 방법을 고려할 수 있다.

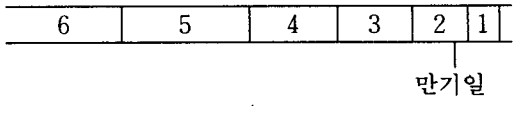
1) 만기일에 걸친 작업을 제외하고 그 작업을 중심으로 내림차순과 오름차순에 의해 재조정하는 방법.
2) 만기일에 걸친 작업을 포함하여 재조정하는 방법.

위의 두가지 방법을 그림으로 나타내면 [그림 6]과 [그림 7]과 같다. 본 연구에서는 위 두가지 방법을 비교하여 우수한 일정계획을 택하는 것으로 한다.

[그림 6] 만기일을 제외한 순위 재조정



[그림 7] 만기일을 포함한 순위 재조정



3.4 정보 교환(Communication)

병렬 GA의 기본적인 특성으로 정보 교환을 들 수 있다. 일반적으로 고립부집단 모형(island model)에서는 각각의 부집단에서 GA를 수행한 후 그 부집단의 가장 좋은 개체를 다른 부집단으로 전달하는 방식을 택하고 있다.

정보 교환도 부집단의 선택에 따라 다음의 두가지로 분류할수 있다. 첫째는 모든 부집단들이 연결되어서 각 부집단에서 다른 모든 부집단의 가장 좋은 개체를 받아들이는 방법이고, 둘째는 부분적인 연결만 허용하는 방법이다. 이 두가지 방법을 비교해 보면 모든 부집단이 연결되어 있는 경우는 수렴 속도가 빠르다고 볼 수 있다. 특히, 전체적으로 가장 좋은 개체의 경우, 다음 세대에서는 모든 부집단의 가장 좋은 개체가 된다. 그러나 빠른 속도의 수렴은 해를 부분최적에 그치게 하는 단점이 따르게 된다. 본 논문에서는 부집단을 ring의 형태로 연결하여 이웃한 두 부집단에게 최적의 개체를 복사 전달하는 방법을 택하기로 한다.

3.5 적자생존(Selection)

단순 적자생존의 경우는 복제의 과정이 적자생존의 과정이 된다. 그러나 병렬 GA은 정보교환의 과정이 포함되기 때문에 계속 집단 크기를 유지하려면 복제의 과정 이외에 또다른 적자 생존의 과정이 포함되어야 한다. 이 논문에서는 두가지의 적자 생존 과정을 고려한다. 첫째는 단순 적자생존(Simple Selection)으로 단순 GA와 같은 방법으로 부집단 안에서 복제, 교차, 돌연변이를 수행하고, 정보 교환에서 넘어온 개체를 비교하여 선택하는 방법이다.

둘째, 우수개체 적자생존 과정(Best Individual Section)은 앞의 방법이 단순 무작위 탐색법과 같은 과정이 될 수 있기 때문에 이러한

문제를 해결하기 위하여 고려한 방법이다. 처음의 집단에서 복제해서 새로운 집단을 구성하고 이 새로운 집단에서 교차와 돌연변이를 거친 다음 이 개체들과 이전의 세대에 있었던 개체, 그리고 정보 교환에 의한 개체의 세가지 개체군을 모두 고려하여 생존 능력이 더 나은 개체를 다음 세대의 개체로 선택하는 방법이다. 이러한 방법은 결코 부집단의 평균 생존 능력이 떨어지는 경우가 없으며 수렴하는 속도도 빠르다고 볼 수 있다.

4. 결과분석

병렬 GA에 대한 문제 해결의 우수성을 평가하기 위하여 우선 적자생존의 방법에 있어서 단순 적자생존과 우수개체 적자생존을 비교한다. 본 논문에서는 모든 알고리즘은 C언어를 이용하여 프로그램되었으며, 실험은 IBM PC 386에서 Turbo C compiler에 의해 이루어졌다. 10개의 20-job문제를 만들어 풀어본 결과가 〈표 3〉에 나타나 있다. 여기서 각 job의 처리시간 Pi는 1-100 사이의 정수로 uniform 분포를 따르는 것으로 하였으며, 만기일 d의 tightness는 Δ의 값이 0.3, 0.6, 0.9인 경우로 분류하였다. 〈표 3〉에서 알 수 있듯이 오차율의 면에서는 단순 적자생존이 더 우월한 결과를 보이고 있으며, 수렴속도 면에서는 우수개체 적자생존이 더 효율적인 것으로 나타났다. 본 연구에서는 수렴속도 보다 오차율이 적은 최적에 가까운 작업 일정계획을 세우기 위하여 이후의 분석에서는 단순 적자생존의 방식을 택하기로 한다.

병렬 GA의 성능을 분석하기 위하여 20, 30, 50, 80문제를 실험한 결과가 〈표 4, 5, 6, 7〉에 각각 나타나 있다. 표에서 최적 페널티는 Branch & Bound의 방법[12]에 의하여 얻어

〈표 3〉 단순 적자생존과 우수개체 적자생존의 비교

문 제	단순 적자생존		우수개체 적자생존	
	평균 오차율(%)	평균 세대수	평균 오차율(%)	평균 세대수
20-1	0.18	9.0	0.14	4.3
20-2	0.45	16.0	0.54	4.3
20-3	0.39	14.7	0.59	4.3
20-4	0.11	7.7	0.11	4.3
20-5	0.47	13.0	0.40	4.3
20-6	0.46	15.7	0.47	4.3
20-7	0.39	10.7	0.44	4.3
20-8	0.28	9.3	0.23	4.3
20-9	0.47	9.3	0.56	4.3
20-10	0.25	11.7	0.27	5.0
총 평 균	0.247		0.299	

진 결과이다. 〈표 4, 5〉에서와 같이 병렬 GA는 그 오차율이 0.5% 미만이므로 Branch & Bound 방법에 의해 얻어진 해에 상당히 근접하는 것으로 나타났다.

작업 수가 50, 80으로 늘어나면 Branch & Bound의 해법은 계산 시간이 기하급수로 늘어나 실험에 의하면 40개의 작업의 경우 이미 24시간을 넘는 것으로 나타났다. 따라서 최적해와의 비교는 불가능하며 기존의 연구[10]에 의한 해를 비교한다. [10]에서는 표현 방법이 (0,1)이 아닌 order 표현 방법을 이용하였으며 병렬 GA이 아닌 단순 GA을 쓴 것이 차이점이다.

〈표 4〉 20-job 문제의 풀이과정

문 제	병렬 GA		Branch & Bound		오차율
	최우수 페널티	CPU in Sec.	최적해	CPU in Sec.	(%)
1	4187	21.50	4178	1.21	0.215
2	3248	30.97	3235	7.25	0.401
3	2817	36.83	2806	10.0	0.392
4	5599	14.62	5594	1.43	0.089
5	4314	22.19	4301	10.65	0.302
6	3732	35.46	3721	22.69	0.295
7	4683	25.59	4666	1.04	0.364
8	3614	40.98	3606	8.35	0.221
9	3129	17.05	3115	11.69	0.449
10	4997	13.12	4992	1.65	0.100

〈표 5〉 30-job 문제의 풀이 결과

문 제	병렬 GA		Branch & Bound		오차율
	최우수 페널티	CPU in Sec.	최적해	CPU in Sec.	(%)
1	10743	9.29	10728	35.16	0.140
2	8346	65.59	8324	675.2	0.264
3	8197	46.26	8180	7059.8	0.207
4	13661	11.44	13650	52.57	0.081
5	10606	28.44	10593	1690	0.388
6	9244	33.77	9226	6287	0.195
7	10600	25.68	10565	41.19	0.331
8	8235	30.98	8218	1015	0.207
9	7203	31.78	7182	4643	0.292
10	12187	10.47	12153	61.85	0.280

〈표 6과 7〉은 본 연구에서의 병렬 GA이 수렴 속도 면에서는 단순 GA보다 평균적으로 많은 시간을 요구하나, 해의 질적인 측면에서는 항상 우월함을 보여주고 있다.

이상의 결과에서, 복잡한 작업 일정계획 문제에 병렬 GA이 상당히 효율적인 휴리스틱으로 증명되었으며 무엇보다 GA을 위한 해의 표현 과정에서의 (0,1) 이진 표현 방법이 order 표현 방법에 비해 해의 탐색공간 및 계산 시간을 줄이는 결정적인 역할을 한 것으로 판단된다.

마지막으로 본 논문에서 다루어진 병렬 유전

〈표 6〉 50-job 문제의 풀이 결과

문 제	병렬 GA		단순 GA	
	최우수 페널티	CPU in Sec.	최우수 페널티	CPU in Sec.
1	29127	18.14	29223	21.19
2	22801	24.34	23098	18.36
3	20068	60.25	20140	27.96
4	27625	12.24	27806	31.47
5	21679	30.81	21782	30.81
6	19107	24.99	19391	20.88
7	29835	42.96	30012	19.79
8	23394	74.08	23579	20.98
9	20657	18.69	20725	19.10
10	30062	58.52	30188	14.16

〈표 7〉 80-job 문제의 풀이 결과

문 제	병렬 GA		단순 GA	
	최우수 페널티	CPU in Sec.	최우수 페널티	CPU in Sec.
1	74541	14.38	74905	34.69
2	58643	58.06	59779	19.02
3	52120	41.11	52300	21.64
4	93228	13.12	93865	14.17
5	73040	28.69	73379	22.80
6	64468	33.84	64564	24.76
7	79757	14.82	80057	35.96
8	62490	35.87	62835	30.31
9	55285	40.03	55465	21.79
10	69821	9.64	70161	33.80

알고리즘의 계산 복잡도(computational complexity)에 대한 연구 및 분석, 단순 알고리즘과의 비교, 그리고 보다 일반적인 작업 일정계획 문제에의 적용이 추후 연구과제로 추천된다.

참 고 문 헌

[1] Bagchi, U., R. S. Sullivan and Y. L. Chang, "Minimizing Mean Absolute Deviation of Completion Times about a Common Due Date," *Naval Research Logistics Quarterly*, Vol 33 (1986), pp 227-240.

[2] Baker, K. R. and G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties:A Review," *Operations Research*, Vol 38 (1990), pp 22-27.

[3] Biegel, J. E. and J. J. Davern, "Genetic Algorithms and Job Shop Scheduling," *Proceedings of the 12th Annual Conference on Computers & Industrial Engineering*, Vol 19 (1990), pp 81-91.

[4] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[5] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[6] Gorges-Schleuter, M., "ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategе," *Proceeding of the 3rd International Conference on Genetic Algorithms and Their Applications* (1989), pp 422-427.

[7] Grefenstette, J. J., R. Gopal, B. J. Rosmaita and D. Van Gught, "Genetic Algorithms for the Traveling Salesman Problem," *Proceeding of the First International Conference on Genetic Algorithms and Their Applications* (1985), pp 160-168.

[8] Hall, N. G., W. Kubiak and S. P. Sethi, "Earliness Tardiness Scheduling Problems, Ⅱ:Deviation of Completion Times about a Restrictive Common Due Date," *Operations Research*, Vol 39 (1991), pp 847-856.

[9] Holland, J. H., "Adaptation in Natural and Artificial Systems," Ann Arbor :The University of Michigan Press, 1975.

[10] Lee, C. Y., "Genetic Algorithms for Single Machine Job Scheduling with Earliness and Tardiness Penalties," Submitted paper.

[11] Mühlenbein, H., "Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization," *Proceeding of the 3rd International Conference on Genetic Algorithms and Their Applications* (1989), pp 416-421.

[12] Szwarc, W., "Single-Machine Scheduling to Minimize Absolute Deviation of Completion Times from a Common Due Date," *Naval Research Logistics*, Vol.36 (1989), pp 663-673.