

병렬기계에서 납기지연 가중 합을 최소화하기 위한 유전 알고리즘*

박문원

전북대학교 산업공학과

A Genetic Algorithm for the Parallel-Machine Total Weighted Tardiness Problem

Moon-Won Park

This paper considers the problem of scheduling a set of n jobs on m parallel machines to minimize total weighted tardiness. For the problem a genetic algorithm is proposed, in which solutions are encoded using the random key method suggested by Bean and new crossover operators are employed to increase performance of the algorithm. The algorithm is compared with the Modified Due-Date (MDD) algorithm after series of tests to find appropriate values for genetic parameters. Results of computational tests on randomly generated test problems show that the suggested algorithm performs better than the MDD algorithm and gives good solutions in a reasonable amount of computation time.

1. 서론

이 연구에서는 n 개의 작업(job)을 m 대의 동일한 병렬기계(identical parallel machines)에 스케줄링 하는 문제를 다룬다. 이때, 각 작업은 어느 기계에서든지 처리될 수 있지만, 단 한대의 기계에서만 처리된다. 각 기계에서는 동시에 여러 개의 작업이 처리될 수 없으며, 일단 하나의 작업이 시작되면 그 작업이 완료될 때까지 처리가 계속된다. 작업 i ($i=1, 2, \dots, n$)의 처리 시간 p_i 는 모든 기계에서 동일하며, 납기 및 납기 가중치는 각각 d_i 와 w_i 로 알려져 있다. 이때, 작업 i 의 완료시간을 C_i 라 하면, 이 연구에서의 스케줄링 목표는 납기지연 가중 합(total weighted tardiness; $\sum_i w_i \cdot \max\{0, C_i - d_i\}$)을 최소화하는 것이다. 만일, 모든 작업의 가중치가 같으면 (즉, $w_i = 1$, for all i) 이 문제는 총 납기지연(total tardiness)을 최소화하는 문제가 된다. (이 연구에서는 이후부터 납기지연 가중 합을 최소화하는 문제를 TWT 문제로, 총 납기지연을 최소화하는 문제를 TT 문제로 부르도록 한다.)

오늘날 하드웨어의 발달로 기계가 범용화되고 다기능화되면서 병렬기계들로 구성된 제조 시스템의 수는 증가추세에 있

다. 또한, 주문생산방식이 확산되고 기업간 경쟁이 치열해지면서 납기준수가 중요한 관리목표가 되고 있다. 따라서, 병렬기계에서의 TWT 및 TT 문제는 상당히 많은 응용분야가 존재하지만 (Woolsey, 1990), 이에 비해 기존연구는 많지 않은 실정이다. Azizogly and Kirca (1998)는 동일한 병렬기계에서 TT 문제의 최적해 (optimal solution)를 제공하는 분지한계(branch and bound) 알고리즘을 제안한 바 있다. 그러나, 이 알고리즘은 작은 크기의 문제 ($n \leq 15, m \leq 3$)에 대해서만 적절한 시간 내에 최적해를 제공할 수 있으며, 이보다 더 큰 문제에서는 계산시간의 급속한 증가로 인해 사용이 힘들다. 실제로 동일한 병렬기계에서의 TT 문제는 Lenstra *et al.* (1977)에 의해 NP-hard로 알려져 있으며, 단일기계 문제에서도 NP-hard이다 (Du and Leung, 1990). 따라서, 작업의 수와 기계의 수가 큰 현실적 문제에 대해서 최적해를 찾는다는 것은 사실상 불가능하다.

이러한 이유로 대부분의 연구들이 실제 제조시스템에서 발생할 수 있는 크기의 문제들에 대해 최적해는 아니지만 우수한 해를 적절한 시간 내에 찾아줄 수 있는 발견적 알고리즘들 (heuristic algorithms)을 제안하고 있다. Dogramaci and Surkis (1979)는 병렬기계에서의 TT 문제를 해결하기 위하여 최단 작

* 이 논문은 1998년도 전북대학교 지원 연구비에 의해 연구되었음.

업시간 우선 (shortest processing time; SPT), 최소 납기일 우선 (earliest due date; EDD) 및 최소 여유시간 우선 (least slack time; LST) 규칙들을 이용하는 3개의 스케줄링 알고리즘을 실험하였다. Alidaee and Rosa (1997)는 TWT 문제와 TT 문제에 대하여 수정된 납기일 우선(modified due date; MDD) 규칙을 이용한 스케줄링 알고리즘을 개발하였는데, 계산실험 결과 MDD 규칙이 기존의 다른 우선순위 규칙보다 더 좋은 결과를 준다고 발표하였다. Koulamas (1997)는 TT 문제에 대해 시뮬레이티드 어닐링(simulated annealing; 이하 SA) 알고리즘을 제안하였는데, 여기서는 m 대의 병렬기계 문제를 m 개의 단일기계 문제로 분할하여 각각의 분할된 문제를 기존에 개발된 효율적인 단일기계 스케줄링 알고리즘을 적용하여 해결하였다. 이때, SA는 병렬기계 문제를 단일기계 문제로 분할하는 역할을 한다.

이 연구에서는 동일한 병렬기계에서의 TWT 문제를 위한 유전 알고리즘(genetic algorithm)을 제안한다. 이 연구의 구성은 다음과 같다. 2장에서는 이 연구에서 다루는 문제를 위한 유전 알고리즘을 소개한다. 3장에서는 제안한 유전 알고리즘에 사용되는 유전 파라미터(genetic parameter) 값들을 설정하는 과정을 소개하고, 기존 알고리즘과의 비교를 통해 제안한 유전 알고리즘의 성능을 평가한다. 마지막으로 4장에서는 결론과 추후 연구과제에 대해 논의한다.

2. 유전 알고리즘

유전 알고리즘은 생물의 유전적 메커니즘과 적자생존(survival-of-the-fittest)의 원리를 모방하여 세대를 진행시키면서 진화하는 탐색기법으로서 Holland (1975)에 의해 처음으로 소개되었다. 이 알고리즘은 타부 서어치(taboo search; 이하 TS)나 SA 등 다른 탐색기법들과 달리 복수의 개체집단을 운용하는데, 지금까지 다양한 조합최적화 문제들에 응용되어 그 실효성이 입증되고 있다. 이 알고리즘은 스케줄링 분야에서도 활발히 응용되고 있는데, 특히 Cheng and Gen (1997), Liu *et al.* (1998), 이문규와 이승주(1999) 등은 병렬기계에서 납기와 관련된 스케줄링 문제를 해결할 수 있는 유전 알고리즘을 제안한 바 있다. 이들 연구에서는 병렬기계에서의 스케줄링 문제를 i) 작업들을 각 기계에 할당하는 문제와, ii) 각 기계에 할당된 작업들의 작업순서를 결정하는 문제로 나누어 해결한다. 이때, 유전 알고리즘은 작업들을 각 기계에 할당하는 역할만을 하며, 할당된 작업들의 작업순서는 단일기계 스케줄링 문제를 위한 발견적 알고리즘을 사용한다. 이렇게 추가의 다른 알고리즘을 유전 알고리즘과 병행하여 사용하는 유전 알고리즘을 혼합형 유전 알고리즘(hybrid genetic algorithm)이라 부른다.

실제로 유전 알고리즘은 TS나 SA에 비해 문제의 다양한 해공간을 탐색하는 광역탐색에는 강점을 가지지만, 좁은 범위의 해공간을 효율적으로 탐색할 수 있는 국소최적화에는 약점이 있는 것으로 알려져 있다. 따라서, 국소탐색을 효율적으로 수

행할 수 있는 별도의 효과적인 알고리즘을 유전 알고리즘과 병행하여 사용하는 혼합형 유전 알고리즘이 Moscato and Norman (1992) 이후 각광을 받고 있다. 그러나 이러한 혼합형 유전 알고리즘은 문제의 특성이 다소 변경되면 (예를 들어 병렬기계 스케줄링 문제에서 목적함수가 바뀌거나, 작업순서에 따라 작업준비시간이 달라지거나, 기계들이 동일하지 않아 기계별로 작업시간이 달라지는 경우가 발생하면) 현재의 국소탐색 알고리즘을 더 이상 사용할 수 없게 된다는 문제가 발생한다. 이 연구에서는 이러한 변화에도 쉽게 대처할 수 있도록 작업할당과 작업순서결정을 동시에 수행할 수 있는 유전 알고리즘을 설계한다. 이러한 설계는 문제의 특성이 다소 변경되더라도, 적합도 함수(fitness function)만을 바꾸어주면 사용할 수 있다는 장점이 있다.

이 연구에서 제시하는 유전 알고리즘은 다음과 같은 7단계의 절차를 따른다.

단계 1. (모집단 생성) P 개의 개체를 임의로 발생시켜 초기 모집단을 구성한다.

단계 2. (평가) 모집단 개체들의 적합도를 계산한다.

단계 3. (새로운 모집단 구성) 단계 3.1, 3.2와 3.3을 차례로 적용하여 현재의 모집단으로부터 진화된 다음 세대의 모집단을 구성한다.

단계 3.1. (선별) 현재의 모집단에서 적합도가 우수한 $S \times P$ 개의 개체를 선정한다. (이 연구에서는 S 를 선별율이라 부른다.)

단계 3.2. (교차) 현재의 모집단에서 임의로 선정된 2개의 개체로부터 1개의 새로운 개체를 생성한다. 이 절차를 반복하여 총 $C \times P$ 개의 새로운 개체를 생성한다. (이 연구에서는 C 를 교차율이라 부른다.)

단계 3.3. (돌연변이) $M \times P$ 개의 개체를 임의로 발생시킨다. (이 연구에서는 M 를 돌연변이율이라 부른다.)
(단, $S+C+M = 1, 0 \leq S, C, M \leq 1$)

단계 4. (반복) 유전 알고리즘이 종료조건에 도달할 때까지 단계 2와 3을 반복수행 한다.

이 절차는 전형적인 유전 알고리즘 절차와는 약간의 차이가 보이는데, 그 이유는 이 연구에서 제안하는 유전 알고리즘이 작업할당과 작업순서결정을 동시에 해결하는 데 적합한 기법들을 채용하기 때문이다. 이에 대한 자세한 설명은 다음과 같다.

2.1 해의 표현

유전 알고리즘을 이용하여 주어진 문제를 해결하기 위해서는 그 문제의 해(solution)를 개체(염색체; chromosome)로 표현하여야 한다. 이때, 개체의 각 원소는 유전자(gene)라 부르며 개체의 유전자 값들로부터 원래 문제의 해를 해석할 수 있다. 스

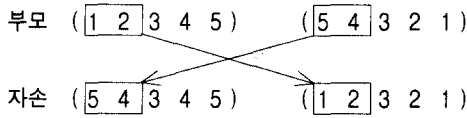


그림 1. 순열표현에서의 일점교차.

케줄링 문제나 외판원 문제 (traveling salesman problem)와 같은 순서결정 문제에 대한 기존연구에서는 해를 주로 순열 (permutation)로 표현하여 왔다. 순열표현 방법에서는 예를 들어 단일기계에서 5개의 작업들에 대한 처리순서가 1→2→3→4→5와 5→4→3→2→1인 2개의 해를 각각 (1, 2, 3, 4, 5) 및 (5, 4, 3, 2, 1)인 개체로 표현한다. 이러한 순열표현에서는 대표적인 유전 연산자인 일점교차(one-point crossover)나 이점교차(two-point crossover) 방법을 사용할 경우 <그림 1>과 같이 비가능해 (infeasible solution)가 생성될 수 있다. 따라서, 순열로 표현된 개체들에 대해 일점교차나 이점교차를 할 경우, 비가능해를 가능해(feasible solution)로 바꾸어 줄 수 있는 보수방법(repair method)이 필요한데, 보수를 할 경우 부모가 갖는 중요한 유전적 성질들이 파괴될 수 있다(김여근 외, 1997).

이러한 문제점을 극복하기 위하여 개체들이 교차된 후에도 가능해(feasible solution)가 될 수 있도록 고안된 매우 다양한 교차방법들이 개발되었다(Goldberg and Lingle, 1985; Oliver *et al.*, 1987; Cleveland and Smith, 1989; Whitley *et al.*, 1989; Grefenstette, 1987; Davis, 1991; Nakano, 1991; Syswerda, 1991). 이러한 교차방법들은 적용하는 문제와 그 문제의 목적함수에 따라 성능의 우열이 달라지므로 문제의 특성을 잘 반영할 수 있는 교차방법을 신중히 선택할 필요가 있다(Starkweather *et al.*, 1991; 김여근 외, 1997). 또한, 병렬기계 스케줄링 문제와 같이 작업의 기계할당정보와 기계에서의 작업순서정보를 동시에 표현하여야 할 경우, 개체로부터 해를 해석하기 위한 특별한 방법이나 규칙이 추가로 필요하게 된다. 이러한 단점을 극복하기 위하여 Bean (1994)은 다양한 순서결정 문제에서 추가의 노력 없이 전통적인 일점교차나 이점교차를 하더라도 생성되는 자손이 가능해가 될 수 있는 random key라는 새로운 개체표현 방법을 고안하였다.

Random key 표현이란 개체의 유전자들을 일정범위의 실수값으로 표현하는 방법이다. 예를 들어, 단일기계 스케줄링 문제에서 개체의 유전자들이 0과 1 사이의 실수값을 가진다고 하자. 이 유전자들은 해당 작업들의 작업순서정보를 담고 있으며, 유전자들을 오름차순으로 정렬하여 작업들의 순서를 해석해낼 수 있다. 예를 들어 작업 수가 5인 단일기계 문제에서 개체 (0.46, 0.91, 0.33, 0.75, 0.51)의 작업순서는 3→1→5→4→2로 해석된다. 초기해(initial solution)는 각 유전자에 U(0, 1)으로부터 발생된 확률변수 값을 할당하여 생성할 수 있는데, 여기서 U(a, b)는 a와 b를 모수로 갖는 균등분포(uniform distribution)를 뜻한다. 교차는 작업순서가 아니라 개체를 대상으로 직접 실행하면 된다. 이 표현방법에서는 유전자 값들이 정렬되어 작업

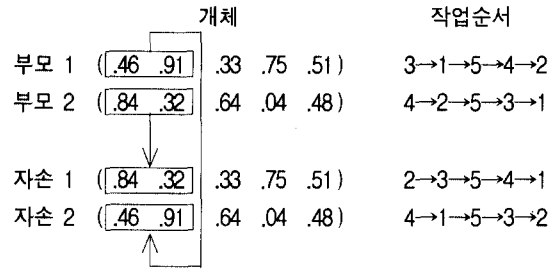


그림 2. Random key 표현에서의 일점교차.

순서로 해석되는 과정을 거치므로 교차로 생성되는 모든 자손들이 <그림 2>와 같이 가능해가 된다. 드물지만 동일한 값을 가지는 유전자가 둘 이상 존재하면, 그 작업들에 대해서는 적절한 tie-breaking 방법으로(예: 작은 인덱스 값의 작업을 먼저 할당) 순서를 결정하면 된다.

Bean은 병렬기계 문제에 대해서도 random key 표현을 확장하여 적용한 사례를 예시하였는데, 이 연구에서도 이 방법을 사용한다. 즉, n개의 작업을 m대의 병렬기계에 스케줄링 하는 문제의 경우, 각 작업에 해당하는 유전자는 1이상 m+1 미만의 실수값을 갖는다. 이때, 정수부분은 해당 작업이 할당된 기계정보를, 소수부분은 그 기계에서의 작업순서정보를 의미한다. 예를 들어 n=5, m=2인 병렬기계 문제의 경우, 개체 (2.46, 1.91, 2.33, 2.75, 1.51)은 기계 1과 2에서의 작업순서가 각각 5→2와 3→1→4인 해로 해석된다. 물론 이 경우도 일점교차나 이점교차로 생성되는 모든 자손들이 가능해가 된다.

2.2 선별

유전 알고리즘에서 새로운 모집단을 구성하기 위한 첫번째 단계는 현재의 모집단에서 우수한 개체들을 선별하는 과정이다. 이 연구에서 사용한 선별방법은 Goldberg (1989)에 의해 소개된 바 있는 elitist 전략인데, 이는 현재 모집단의 P개의 개체들을 적합도가 우수한 순(목적함수 값이 작은 순)으로 정렬하여 이들 중 상위 S×P개의 개체들을 새로운 모집단에 그대로 복사하는 방법이다. 이 방법은 기존의 확률적 선별방법들(확률바퀴방법, 토너먼트 선별, 순위선별 등)과는 다른 확정적 선별방법이다. 이 방법은 목적함수 값을 그대로 적합도 함수로 사용할 수 있다는 장점과 세대가 진행됨에 따라 모집단 내의 우수한 개체들이 지속적으로 개선된다는 장점이 있다.

2.3 교차

새로운 모집단 구성을 위한 다음 단계는 현재 모집단 개체들을 교차하여 새로운 자손들을 생성하는 과정이다. 이 연구에서는 전통적인 일점교차나 이점교차 대신 Spears and DeJong (1991)이 제안한 parameterized uniform 교차방법(교차1)과 이를 이 연구에서 다루는 문제에 적합하도록 수정한 새로운 2개의

작업	작업순서				
	1	2	3	4	5
동전	앞	앞	뒤	앞	뒤
부모1	2.46	1.91	2.33	2.75	1.51
부모2	2.71	1.95	1.83	1.29	2.64
자손	2.46	1.91	1.83	2.75	2.64

그림 3. 교차1에 의한 자손생성.

교차방법(교차2 및 교차3)을 병행하여 사용한다. 교차1에서는 현재 모집단의 P 개의 개체들 중 임의로 두 개체(부모1과 부모2)를 선택한 후, 각 유전자 위치에 대해 동전 던지기를 하여 어떤 부모의 유전자 값을 자손에 상속할 것인지를 확률적으로 결정하는 방법이다. 즉, 동전 던지기 결과 앞면이 나오면 부모1의 유전자를, 뒷면이 나오면 부모2의 유전자를 자손에 상속한다. 예를 들어, 부모1과 부모2의 특정위치의 유전자 값이 각각 1.34와 2.54인 경우, 동전 던지기 결과 앞면이 나오면 자손은 1.34의 유전자 값을, 뒷면이 나오면 2.54의 유전자 값을 상속받는다. 단, 이때 사용하는 동전은 앞면이 나올 확률이 p_H ($0 < p_H < 1$)인 동전(biased coin)이다. <그림 3>은 교차1을 통해 생성된 자손의 예이다.

교차1만을 사용할 경우 각 유전자에 해당되는 작업은 하나의 부모로부터 기계정보와 순서정보를 동시에 상속받으므로 생성된 자손이 부모와는 이질적일 수 있다. 이러한 단점을 극복하기 위하여 이 연구에서는 교차2 및 교차3을 병행하여 사용한다. 교차2에서는 하나의 부모(예: 부모1)로부터 기계정보(유전자 값의 정수부분)를 전부 상속받는다. 순서정보(유전자 값의 소수부분)는 교차1과 마찬가지로 동전 던지기를 통해 부모1과 부모2로부터 확률적으로 상속받는다. 예를 들어 부모1과 부모2의 특정위치의 유전자 값이 각각 1.34와 2.54인 경우, 동전 던지기 결과 앞면이 나오면 자손은 1.34를, 뒷면이 나오면 1.54를 상속받는다. 교차3은 교차2와 반대로 순서정보가 부모1로부터 전부 상속되며 기계정보만이 확률적으로 상속된다. 예를 들어 부모1과 부모2의 특정위치의 유전자 값이 각각 1.34와 2.54인 경우, 동전 던지기 결과 앞면이 나오면 자손의 해당 유전자 값은 1.34가 되며 뒷면이 나오면 2.34가 된다. <그림 4>와 <그림 5>는 각각 교차2와 교차3을 통해 생성된 자손의 예이다.

이 연구에서는 총 $C \times P$ 개의 자손을 교차를 통해 생성하는데, 하나의 자손을 생성할 때마다 어떤 교차방법을 사용할 것인지를 확률적으로 결정한다. 즉, 확률 p_{C1} , p_{C2} 및 p_{C3} (단, $p_{C1} + p_{C2} + p_{C3} = 1$)로 각각 교차1, 교차2 및 교차3을 선택하는데, 예를 들

작업	작업순서				
	1	2	3	4	5
동전	앞	앞	뒤	앞	뒤
부모1	2.46	1.91	2.33	2.75	1.51
부모2	2.71	1.95	1.83	1.29	2.64
자손	2.46	1.91	2.83	2.75	1.64

그림 4. 교차2에 의한 자손생성.

작업	작업순서				
	1	2	3	4	5
동전	앞	앞	뒤	앞	뒤
부모1	2.46	1.91	2.33	2.75	1.51
부모2	2.71	1.95	1.83	1.29	2.64
자손	2.46	1.91	1.33	2.75	2.51

그림 5. 교차3에 의한 자손생성.

어 $U(0, 1)$ 로부터 발생시킨 확률변수 값이 0과 p_{C1} 사이면 교차1을, p_{C1} 과 $p_{C1} + p_{C2}$ 사이면 교차2를, $p_{C1} + p_{C2}$ 와 1 사이면 교차3을 사용한다. (이 연구에서는 p_{C1} , p_{C2} 와 p_{C3} 을 각각 교차1 선택확률, 교차2 선택확률 및 교차3 선택확률이라 부른다.)

2.4 돌연변이

새로운 모집단 구성을 위한 마지막 단계는 돌연변이를 통해 현재 모집단의 개체와는 상이한 $M \times P$ 개의 새로운 개체를 생성하는 것이다. 이 연구에서의 돌연변이는 아주 작은 확률로 개체들의 몇몇 유전자 값들을 바꾸는 전통적인 방법이 아니라, 일정 정도의 새로운 개체를 임의로 생성하여 모집단에 삽입하는 이민(immigration)방법을 사용한다. Random key에 의한 해의 표현방법에서 이민방법은 전통적인 돌연변이 방법보다 더 효과적인 것으로 알려져 있다 (Bean 1994).

2.5 세대의 진행과 해의 탐색

이 연구에서의 유전 알고리즘은 <그림 6>과 같이 현재 세대의 모집단으로부터 선별과 교차, 돌연변이를 통해 새로운 세대의 모집단을 구성하고, 이 과정을 종료조건에 도달할 때까지 반복하면서 해공간을 탐색한다. 단, 초기 모집단은 P 개의 개체를 임의로 발생시켜 구성하는데, 개체들의 각 유전자에 $U(1, m+1)$ 로부터 발생시킨 확률변수 값을 할당하는 방법을 사용한다. (돌연변이 개체도 이와 동일한 방법으로 생성한다.)

탐색 알고리즘이 효율적이고 효과적으로 우수한 해를 탐색하기 위해서는 해공간(solution space)에서 다양한 해들을 생성해낼 수 있는 광역탐색 전략과 좋은 해의 성질을 이용하여 그

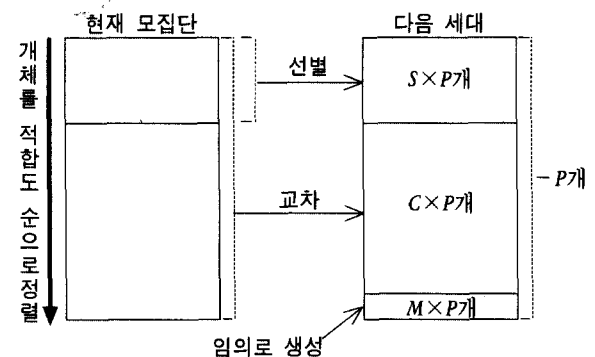


그림 6. 다음 세대 모집단의 구성.

해의 인접공간을 집중적으로 탐색할 수 있는 국소최적화 전략이 조화를 이루어야 한다. 이 연구에서 제안하는 유전 알고리즘에서의 광역탐색 전략은 교차1과 이민에 의한 돌연변이 발생방법을 들 수 있다. 앞서서도 언급한 바와 같이 교차1은 교차2와 교차3에 비해 부모와는 다소 이질적인 자손을 생성시키므로 알고리즘이 다양한 해공간을 탐색할 수 있도록 한다. 또한, 이민방법을 통해 일정정도의 새로운 개체들을 모집단에 지속적으로 공급해줌으로써 개체들의 다양성을 확보한다.

한편, 이 연구에서는 국소최적화를 위하여 별도의 problem-specific한 국소최적화 알고리즘을 사용하지는 않지만 elitist 선별과 교차2 및 교차3을 사용하여 국소탐색능력을 강화한다. 즉, 우수한 개체들을 일정정도 모집단에 늘 유지시키고, 부모와 유사한 자손들을 발생시킴으로써 국소탐색능력을 향상시킨다. 이러한 광역탐색 전략과 국소최적화 전략을 조화롭게 적용하기 위하여 이 연구에서는 계산실험을 통해 이 전략들과 관계된 유전 파라미터(genetic parameter) 값들을 설정하도록 한다.

3. 계산실험 및 분석

3.1 실험문제의 생성

제한한 알고리즘의 유전 파라미터 설정과 이 알고리즘의 성능평가를 위하여 이 연구에서는 다양한 실험문제를 생성하였다. 목적함수를 2종류(TWT, TT)로 하고, 작업의 수(n)를 3종류(20, 30, 50)로, 기계의 수(m)를 3종류(3, 5, 7)로, 납기의 분포정도를 2종류(tight, loose)로 하여 36가지($2 \times 3 \times 3 \times 2$)의 문제조합에 대해 각각 10문제씩, 총 360문제를 임의로 생성하였다. 각 실험문제의 생성방법에 대한 자세한 설명은 다음과 같다.

- 1) 작업들의 작업시간($p_i, i=1, 2, \dots, n$)은 $DU(1, 100)$ 로부터 발생시킨다. 여기서 $DU(a, b)$ 는 a 와 b 를 모수로 가지는 이산형 균등분포(discrete uniform distribution)를 의미한다.
- 2) 작업들의 납기($d_i, i=1, 2, \dots, n$)는 $U(T_i \times M_L, T_i \times M_L)$ 로부터 발생시킨다. 여기서, M_L 은 총 작업소요시간(makespan)의 이론적 하한값인 $\sum_i p_i / m$ 을 의미하며 tight와 loose한 문제에 대해서 (T_i, T_i) 값을 각각 (0.4, 0.8)과 (0.5, 1.0)으로 한다.
- 3) TWT 문제에서 작업들의 납기 가중치($w_i, i=1, 2, \dots, n$)는 $U(1, 10)$ 로부터 발생시키며, TT 문제에서는 모두 1로 할당한다.

3.2 유전 파라미터(genetic parameter)의 설정

제한한 유전 알고리즘이 주어진 문제에 대해 좋은 해를 제공하기 위해서는 적합한 유전 파라미터 값들을 설정하여야 한다. 이 연구에서는 유전 파라미터들이 알고리즘 성능에 미치는 영향을 알아보기 위하여 먼저 예비실험을 수행하였는데, 이 실험에서는 유전 파라미터 값들의 변화에 따른 유전 알고리즘의

성능변화와 수렴추이를 알아보기 위하여 충분히 긴 시간동안 알고리즘을 구동하였다. 이 실험을 통해 나타난 결과들을 정리해보면 다음과 같다.

- 1) 모집단의 크기(P)는 클수록 좋은 결과를 주지만, 300 이상인 경우 계산시간의 증가에 비해 알고리즘 성능에는 큰 변화가 없다.
- 2) 선별율(S)은 모집단 크기의 10~20%에서 좋은 결과를 주며, 모집단의 크기가 커짐에 따라 선별율을 작게 하는 것이 더 좋은 결과를 준다.
- 3) 돌연변이율(M)은 모집단 크기의 1%로 하는 것이 좋은 결과를 준다.
- 4) 교차방법은 교차1,2,3선택확률을 다음과 같이 설정할 때 좋은 결과를 준다.

$$0.6 \leq p_{C1} \leq 0.7, p_{C1} > p_{C2} \geq p_{C3} \geq 0.1$$
- 5) 교차에서 유전자상속을 위해 동전 던지기를 할 때, 앞면이 나올 확률(p_H)을 0.5로 하는 것이(즉, unbiased coin을 이용할 때) 좋은 결과를 준다.
- 6) 작업의 수(n)가 증가함에 따라 유전 알고리즘의 수렴속도는 늦어지지만, 기계의 수(m)는 증감하더라도 유전 알고리즘의 수렴속도에는 거의 변화가 없다.

이러한 예비실험결과를 바탕으로 유전 파라미터 설정을 위한 이 실험을 수행하였는데 이 실험은 3.1절에서 기술한 방법으로 생성한 360개의 실험문제들을 대상으로 수행하였다. 이때 사용된 유전 파라미터 값들의 조합은 <표 1>과 같으며, 서로 다른 유전 파라미터 값을 가지는 27개의 유전 알고리즘들을 비교하였다. 알고리즘들의 공정한 비교를 위해서는 사용되는 유전 파라미터 값들이 다르더라도 알고리즘 수행시간이 동일한 종료조건이 필요하다. 또한, 유전 알고리즘들이 대부분의 실험문제에 대해 수렴치에 도달할 정도의 충분한 세대진행이 보장되어야 한다. 따라서, 이 연구에서는 유전 알고리즘들이 30000/ mP 세대가 진행된 후 종료하도록 하였는데, 이 값은 예비 실험결과를 바탕으로 결정하였다.

생성된 실험문제들의 최적해는 적절한 시간 내에 찾아질 수 없으므로, 이 연구에서는 각각의 실험문제에 대해 아래에 정의된 relative deviation index (RDI)를 이용하여 알고리즘들의 성능을 비교하였다.

$$RDI = (Z_a - Z_B) / (Z_w - Z_B)$$

표 1. 실험된 유전 파라미터 조합

유전 파라미터	실험값
P	200, 250, 300
(S, C, M)	(.1, .89, .01), (.15, .84, .01), (.2, .79, .01)
p_H	.5
(p_{C1}, p_{C2}, p_{C3})	(.6, .2, .2), (.6, .3, .1), (.7, .2, .1)
종료조건	30000/ mP 세대가 진행된 후 종료

표 2. 교차방법에 따른 평균 RDI

(p_{C1}, p_{C2}, p_{C3})	모집단의 크기 (P)		
	200	250	300
(.6, .2, .2)	.402	.348	.334
(.6, .3, .1)	.348	.353	.265
(.7, .2, .1)	.302	.331	.255

여기서 Z_a, Z_B 및 Z_W 는 주어진 실험문제에 대해 각각 알고리즘 a 가 찾은 해의 값, 가장 좋은 결과를 주는 알고리즘이 찾은 해의 값 및 가장 나쁜 결과를 주는 알고리즘이 찾은 해의 값을 뜻한다. 따라서, RDI는 0에서 1사이의 값을 가지며, 알고리즘의 성능이 우수할수록 0에 가까운 값을 가진다.

실험결과 교차방법은 <표 2>와 같이 모집단의 크기에 상관없이 $(p_{C1}, p_{C2}, p_{C3}) = (0.7, 0.2, 0.1)$ 일 때 가장 우수하였으며, 유의수준 5% 하에서 각 수준 간의 차이가 있는 것으로 나타났다. 따라서, 이 연구에서는 최종적으로 (p_{C1}, p_{C2}, p_{C3}) 값을 (0.7, 0.2, 0.1)로 고정하여 모집단의 크기와 선별율과의 관계를 분석하였다. 그 결과 <그림 7>과 같이 모집단의 크기와 선별율과는 교호작용이 있으며 모집단의 크기가 커짐에 따라 선별율을 작게 하는 것이 더 좋은 결과를 준다는 사실을 발견하였다. 또한, 선별율이 10%와 15%인 경우 모집단의 크기가 300인 유전 알고리즘이 가장 우수하였으며 선별율이 20%인 경우 모집단의 크기가 200인 유전 알고리즘이 가장 좋은 결과를 주었다. 3가지 선별율에 대해서 모집단의 크기가 250인 유전 알고리즘은 파레토 최적성(Pareto optimality)에 의해 탈락되었다.

따라서, 이 연구에서는 최종적으로 2개의 유전 알고리즘(GA200과 GA300)을 비교 평가하였으며, 이때 사용한 유전 파라미터 값은 <표 3>과 같다. 두 알고리즘의 비교를 위하여 이 연구에서는 각각의 실험문제에 대해 performance ratio (PR; $PR = Z_{200} / Z_{300}$)를 정의하였는데, 여기서 Z_{200} 과 Z_{300} 은 주어진 실험문제에 대해 각각 GA200과 GA300이 찾은 해의 값을 의미한다. 이

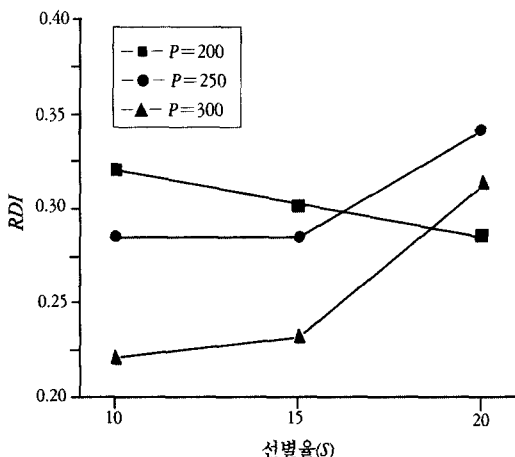


그림 7. 모집단의 크기와 선별율과의 관계.

표 3. GA200과 GA300에 사용된 유전 파라미터 값

알고리즘	유전 파라미터			
	P	(S, C, M)	p_H	(p_{C1}, p_{C2}, p_{C3})
GA200	200	(.2, .79, .01)	.5	(.7, .2, .1)
GA300	300	(.1, .89, .01)	.5	(.7, .2, .1)

표 4. GA200과 GA300의 성능비교

n	m	PR	
		TWT	TT
20	3	0.981	1.018
	5	0.992	1.022
	7	1.023	0.999
30	3	0.991	0.982
	5	0.997	1.000
	7	0.954	1.001
50	3	1.000	1.010
	5	1.027	0.988
	7	1.060	1.027
Average		1.003	1.005
p-value		0.386	0.020

때, PR 값이 1보다 작으면 GA200이, 1보다 크면 GA300이 더 좋은 해를 찾았다는 것을 뜻한다. <표 4>에 나타난 PR 값에서 알 수 있듯이 GA300이 GA200보다는 약간 더 좋은 결과를 주었으며, 두 알고리즘의 성능 차이는 paired-t test 결과 유의수준 5% 하에서 TWT 문제에 대해서는 차이가 없는 것으로, TT 문제에 대해서는 차이가 있는 것으로 나타났다. 이러한 실험결과를 바탕으로 이 연구에서는 병렬기계에서 스케줄링 문제를 위한 유전 알고리즘으로 GA300을 사용하기로 한다.

3.3 비교 알고리즘

제시한 유전 알고리즘의 성능을 평가하기 위하여 이 연구에서는 Alidaee and Rosa (1997)가 제안한 수정된 납기일 우선(modified due date; 이하 MDD) 규칙을 이용한 알고리즘과 비교하였다. MDD 규칙은 원래 Baker and Bertrand (1982)가 단일기계에서의 TT 문제를 해결하기 위하여 제안하였는데, 이 규칙은 각 시점 t 에서 아래에 정의된 수정된 납기일(d_i')이 가장 빠른 작업을 다음 작업으로 선택하는 방법이다.

$$d_i' = \max\{C_i, d_i\} = \max\{t + p_i, d_i\}$$

실제로 목적함수 $\sum_i w_i \cdot \max\{0, C_i - d_i\}$ 를 최소화하는 TWT 문제는 아래 식과 같이 $\sum_i w_i \cdot \max\{C_i, d_i\}$ 를 최소화하는 문제와 동일해진다(TT 문제에서의 $w_i = 1$, for all i).

$$\begin{aligned} \sum_i w_i \cdot \max\{0, C_i - d_i\} &= \sum_i w_i \cdot \max\{0, C_i - d_i\} + \sum_i w_i d_i - \sum_i w_i d_i \\ &= \sum_i w_i \cdot \max\{C_i, d_i\} - \sum_i w_i d_i \end{aligned}$$

따라서, MDD 규칙은 목적함수의 증가를 최소화하는 작업을 다음 작업으로 선택하는 일종의 greedy 알고리즘이다. 한편, Rachamadugu (1987)은 단일기계 TWT 문제에서 아래의 성질을 규명하였다.

Proposition. 단일기계 TWT 문제의 최적해는 연속하는 2개의 작업 i 와 j (단, $i \rightarrow j$)에 대해 아래의 조건을 만족하거나, 혹은 아래의 조건을 만족하도록 연속하는 2개의 작업들의 작업 순서를 바꾸어 줌으로써 또 다른 최적해를 달성할 수 있다.

$$\frac{w_i}{p_i} \left(1 - \frac{\max\{0, d_i - C_i\}}{p_i} \right) \geq \frac{w_j}{p_j} \left(1 - \frac{\max\{0, d_j - C_j\}}{p_j} \right)$$

이러한 특성을 반영하는 단일기계 TWT 문제를 위한 MDD 알고리즘 절차는 다음과 같다.

(단일기계에서의 MDD 알고리즘)

단계 0. 시점 t 를 0으로 하고 현재까지 선택되지 않은 작업들의 집합 u 를 $\{1, 2, \dots, n\}$ 으로 한다.

단계 1. 집합 u 를 아래의 방법으로 집합 u_1 과 집합 u_2 로 분리한다.

$$u_1 = \{i \in u: C_i = t + p_i > d_i\}, u_2 = u - u_1 = \{i \in u: C_i = t + p_i \leq d_i\}$$

단계 2. 아래의 식을 만족하는 작업들의 집합 γ 와 λ 를 찾는다.

$$\gamma = \{i \in u_1: w_i/p_i = \max_{k \in u_1} \{w_k/p_k\}\},$$

$$\lambda = \{i \in u_2: d_i = \min_{k \in u_2} \{d_k\}\}$$

단계 3. 아래의 식을 만족하는 작업 a 와 b 를 찾는다.

$$w_a C_a = \min_{k \in \gamma} \{w_k C_k\}, w_b C_b = \min_{k \in \lambda} \{w_k d_k\}$$

단계 4. 아래의 식을 만족하는 작업 $g \in \{a, b\}$ 를 선택한다.

$$\delta_g = \min\{w_a C_a, w_b d_b\}$$

단계 5. 다음 작업으로 작업 g 를 선택하고, $t = t + p_g$, $u = u - \{g\}$ 로 한다. 만일 $u = \phi$ 이면 알고리즘을 종료하고 $u \neq \phi$ 이면 단계 1로 간다.

이 알고리즘은 Alidaee and Rosa에 의해 병렬기계 TWT 문제를 위한 알고리즘으로 다음과 같이 확장되었다.

(병렬기계에서의 MDD 알고리즘)

단계 0. 시점 t_k ($k = 1, 2, \dots, m$)를 0으로 하고, 현재까지 선택되지 않은 작업들의 집합 u 를 $\{1, 2, \dots, n\}$ 으로 한다.

단계 1. 단일기계에서의 MDD 알고리즘의 단계 1~4를 각각의 기계에 대해 차례로 적용하여 작업 g_k 와 이에 해당하는 δ_{g_k} ($k = 1, 2, \dots, m$) 값을 구한다.

단계 2. $\delta_{g_i} = \min_{1 \leq k \leq m} \{\delta_{g_k}\}$ 를 만족하는 작업 g_i 를 기계 i 의 다음 작업으로 선택한다.

단계 3. $u = u - \{g_i\}$ 로 하고, 기계 k ($k = 1, 2, \dots, m$)에 이미 할당된 작업들의 완료시점 t_k ($k = 1, 2, \dots, m$)를 계산한다. 만일 $u = \phi$ 이면 알고리즘을 종료하고 $u \neq \phi$ 이면 단계 1로 간다.

이 알고리즘은 각각의 기계에 대해 단일기계를 위한 MDD 알고리즘을 적용하여 작업들을 선정해 보고, 이들 중 목적함수의 증가를 최소로 하는 작업을 선택하여 해당 기계에 할당하는 방법이다. Alidaee and Rosa은 계산실험결과 MDD 알고리즘은 다른 우선순위 규칙 알고리즘들에 비해 TWT 문제뿐 아니라 TT 문제에서도 우수한 성능을 보인다고 발표하였다.

3.4 유전 알고리즘의 성능평가

이 연구에서는 GA₃₀₀의 성능을 MDD 알고리즘의 성능과 비교하였으며, 또한 GA₃₀₀에서 교차2와 교차3을 사용하지 않고 교차1만을 사용하는 유전 알고리즘(GA_{300,C1})의 성능과도 비교하였다. 알고리즘들의 성능비교를 위하여 이 연구에서는 각각의 실험문제에 대해 알고리즘들의 RDI 값과 relative error (RE; $RE = 100 \times (Z_a - Z_B) / Z_B$) 값을 계산하였는데, TWT와 TT 문제에 대한 결과를 각각 <표 5>와 <표 6>에 정리하였다.

<표 5>로부터 알 수 있듯이 TWT 문제의 경우 GA₃₀₀이 MDD보다 RDI와 RE 측면에서 훨씬 우수하였으며, 이 표에는 나타나있지는 않지만 모든 실험문제에 대해 GA₃₀₀이 MDD보다 좋은 결과를 주었다. 또한, GA₃₀₀은 납기분산 정도가 tight한 문제보다는 loose한 문제에 대해 MDD와 더 큰 성능차이를 보이는 것을 알 수 있다. 한편, 교차1만을 사용하는 GA_{300,C1}과의 비교에서는 GA₃₀₀이 좀 더 좋은 결과를 주었다. 이 표에는 나타나있지 않지만 교차1,2,3을 병행하여 사용하는 것이 교차1만을 사용하는 것에 비해 평균 4% 정도의 해의 개선효과가 있는 것으로 나타났으며, paired-t test 결과 유의수준 5% 하에서 두 알고리즘 간의 성능의 차이가 있는 것으로 나타났다.

TT 문제의 경우에서도 GA₃₀₀이 MDD보다 우수한 결과를 주지만, 그 차이는 TWT 문제에서 비해 작은 것을 알 수 있다. 그 이유는 크게 2가지인 것으로 분석된다. 즉, i) TT 문제가 TWT 문제에 비해 해 값들의 분산정도가 작고; ii) MDD 알고리즘이 TWT 문제보다는 TT 문제에서 더 경쟁력이 있기 때문에 이러한 결과가 나오는 것으로 판단된다. 실제로 Alidaee and Rosa의 계산실험 결과에서도 MDD 알고리즘이 TWT 문제에 비해 TT 문제에서 다른 알고리즘들보다 더 좋은 성능을 보이고 있다.

TWT 문제의 경우와 마찬가지로 TT 문제에서도 GA₃₀₀은 납기분산 정도가 tight한 문제보다는 loose한 문제에 대해 MDD와 더 큰 성능차이를 보이고 있다. 하지만, TWT 문제의 경우와는 달리 작업의 수가 증가할수록 두 알고리즘의 성능차이의 폭이 작아지는 경향을 보인다. 한편, 교차1만을 사용하는 GA_{300,C1}보다 GA₃₀₀이 좀 더 좋은 결과를 주었으며, 이 표에는 나타나있지 않지만 TT 문제에서도 교차1, 2, 3을 병행하여 사용하는 것이

표 5. TWT 문제에 대한 알고리즘들의 성능비교

<i>n</i>	<i>m</i>	납기분포	RDI			RE		
			GA ₃₀₀	GA _{300,C1}	MDD	GA ₃₀₀	GA _{300,C1}	MDD
20	3	tight	0.00	0.01	1.00	1.1	1.6	268.1
		loose	0.00	0.01	1.00	0.6	9.0	1321.6
	5	tight	0.00	0.01	1.00	1.0	2.4	229.5
		loose	0.01	0.07	1.00	6.2	17.2	789.2
	7	tight	0.00	0.11	0.91	0.5	2.6	180.3
		loose	0.02	0.01	1.00	4.9	1.8	302.8
30	3	tight	0.00	0.00	1.00	0.3	1.1	399.5
		loose	0.01	0.01	1.00	8.5	16.6	2039.5
	5	tight	0.00	0.01	1.00	1.2	2.3	264.9
		loose	0.01	0.01	1.00	5.6	10.2	752.5
	7	tight	0.00	0.02	1.00	0.7	3.9	242.4
		loose	0.01	0.03	1.00	4.9	17.6	511.7
50	3	tight	0.00	0.00	1.00	1.0	0.9	342.9
		loose	0.02	0.01	1.00	20.8	12.7	2512.8
	5	tight	0.00	0.01	1.00	0.5	1.3	252.6
		loose	0.02	0.02	1.00	9.3	21.8	956.6
	7	tight	0.01	0.00	1.00	2.6	2.0	271.8
		loose	0.01	0.01	1.00	13.9	3.8	671.7
Average			0.01	0.02	0.99	4.6	7.2	683.9

표 6. TT 문제에 대한 알고리즘들의 성능비교

<i>n</i>	<i>m</i>	납기분포	RDI			RE		
			GA ₃₀₀	GA _{300,C1}	MDD	GA ₃₀₀	GA _{300,C1}	MDD
20	3	tight	0.09	0.24	0.98	0.4	0.8	5.5
		loose	0.00	0.05	1.00	0.1	1.7	58.0
	5	tight	0.27	0.15	0.85	1.4	0.7	5.5
		loose	0.05	0.03	1.00	1.5	2.7	54.6
	7	tight	0.03	0.13	1.00	0.1	0.4	3.8
		loose	0.14	0.06	1.00	1.5	3.0	21.1
30	3	tight	0.08	0.29	0.93	0.1	0.9	2.6
		loose	0.07	0.10	1.00	7.2	0.7	65.0
	5	tight	0.44	0.14	0.83	2.6	0.8	4.4
		loose	0.09	0.12	0.90	1.7	1.0	30.6
	7	tight	0.46	0.37	0.55	1.3	1.2	2.2
		loose	0.13	0.15	0.90	1.8	2.7	34.2
50	3	tight	0.29	0.51	0.70	1.4	1.5	2.7
		loose	0.13	0.10	0.90	3.8	4.4	50.2
	5	tight	0.45	0.29	0.76	1.0	0.5	2.0
		loose	0.13	0.21	0.95	2.1	7.2	35.2
	7	tight	0.34	0.51	0.36	1.0	2.0	1.6
		loose	0.02	0.38	0.94	0.3	3.3	17.8
Average			0.18	0.21	0.86	1.6	2.0	22.1

교차1만을 사용하는 것에 비해 평균 1% 정도의 해의 개선효과가 있는 것으로 나타났다. 하지만 유의수준 5% 하에서 GA₃₀₀과 GA_{300,C1}의 차이는 없는 것으로 밝혀졌다.

결론적으로 이 연구에서 제안한 유전 알고리즘 GA₃₀₀은 성능 면에서 기존의 알고리즘 MDD보다 우수하였으며, TWT 문제의 경우 이 연구에서 제안한 3가지 교차방법을 병행하여 적용

하는 것이 Spears and DeJong (1991)이 제안한 parameterized uniform 교차방법(교차1)만을 적용하는 것보다 더 좋은 결과를 주는 것을 알 수 있었다. 이 연구에서 실험된 모든 알고리즘은 Pascal 언어로 프로그래밍 되었으며, Pentium-500Mhz 프로세서가 장착된 개인용 컴퓨터에서 계산실험이 수행되었다. 이때의 GA₃₀₀의 수행시간은 <표 7>에 정리되어 있다. 이 표에서 알

표 7. GA300의 수행시간

n	m	Average CPU time (sec.)	
		TWT	TT
20	3	61	59
	5	64	63
	7	68	66
30	3	142	138
	5	154	146
	7	160	157
50	3	421	415
	5	441	440
	7	462	461

수 있듯이, TWT 문제와 TT 문제의 수행시간의 차이는 거의 없으며, 큰 문제(n=50, m=7)에 대해서도 적절한 시간(8분) 내에 해를 제공하는 것을 알 수 있다.

4. 결론

이 연구에서는 병렬기계에서 납기지연 가중 합을 최소화하기 위한 유전 알고리즘을 제안하였다. 병렬기계 스케줄링 문제의 해를 유전 알고리즘에서의 개체로 표현하기 위하여 이 연구에서는 random key 방법을 사용하였는데, 이 방법은 작업의 기계 할당정보와 각 기계에서의 작업순서정보를 동시에 개체에 표현할 수 있다는 장점이 있다. 또한, 선별을 위하여 elitist 전략을 사용하였으며, 교차방법으로는 기존의 parameterized uniform 교차방법과 이를 응용한 새로운 2개의 교차방법을 병행하여 사용하였다. 제안한 알고리즘은 현재까지 우수하다고 알려진 MDD 알고리즘과 성능을 비교하였는데, 그 결과 이 연구에서 제안한 유전 알고리즘이 기존의 MDD 알고리즘에 비해 납기지연 가중 합 문제에 대해서는 월등히 좋은 결과를 주었으며, 총 납기지연 문제에 대해서도 더 좋은 결과를 주었다. 또한, 이 알고리즘은 작업의 수가 큰 문제에 대해서도 적절한 시간 내에 양질의 해를 제공하였다.

이 연구에서 제시한 유전 알고리즘은 다른 병렬기계 스케줄링을 위한 유전 알고리즘에서 대부분 채용하고 있는 별도의 problem-specific한 국소최적화 알고리즘을 사용하지 않는다. 따라서, 비록 이 연구에서는 실험되지 않았지만 이중 병렬기계 (unrelated parallel machine) 문제, 순서 의존적인 작업준비시간이 존재하는 문제, 납기지연 작업 수를 최소화하는 문제 등 문제의 특성이 다른 병렬기계 환경에서도 알고리즘의 큰 수정 없이 목적함수 계산과정만을 수정하여 사용할 수 있다는 장점이 있다.

이 연구에서는 problem-specific한 국소최적화 알고리즘을 채용하지 않은 대신 국소탐색 능력을 다소 강화할 수 있는 방안 (새로운 교차연산자의 채용과 elitist 전략의 사용)을 알고리즘 상에 반영하였다. 하지만 이러한 방안은 problem-specific한 국소

최적화 알고리즘을 사용할 경우에 비해 알고리즘의 효율이 다소 떨어질 것으로 판단된다. 만일 problem-specific한 국소최적화 알고리즘을 채용한 혼합형 유전 알고리즘의 성능과 이 연구에서 제시한 유전 알고리즘의 성능을 직접 비교한다면, 제시한 유전 알고리즘의 효율을 정량적으로 분석할 수 있을 것이다. 하지만, 지금까지의 조사에 의하면 이 연구에서 다루는 TWT 문제나 TT 문제를 위한 혼합형 유전 알고리즘을 기존연구에서 발견할 수 없었다. 따라서, 혼합형 유전 알고리즘과의 비교를 통한 정량적 분석과 국소탐색 능력을 향상시킬 수 있는 추가의 방안에 대한 연구를 추후과제로 남겨둔다.

참고문헌

김여근, 윤복식, 이상복 (1997), *메타휴리스틱*, 영지문화사.
 이문규, 이승주 (1999), 병렬의 동일기계에서 처리되는 순서의존적인 작업들의 스케줄링을 위한 유전알고리즘, *대한산업공학회지*, 25, 360-368.
 Alidaee, B. and Rosa, D. (1997), Scheduling parallel machines to minimize total weighted and unweighted tardiness, *Computers and Operations Research*, 8, 775-788.
 Azizogly, M. and Kirca, O. (1998), Tardiness minimization on parallel machines, *International Journal of Production Economics*, 55, 163-168.
 Baker, K. and Bertrand, J. (1982), A dynamic priority rule for sequencing against due-dates, *Journal of Operational Management*, 3, 37-42.
 Bean, J. (1994), Genetic algorithms and random keys for sequencing and optimization, *ORSA Journal on Computing*, 6, 154-160.
 Cheng, R. and Gen, M. (1997), Parallel machine scheduling problems using mimetic algorithms, *Computers and Industrial Engineering*, 33, 761-764.
 Cleveland, G. and Smith, S. (1989), Using genetic algorithms to schedule flow shop releases, *Proceedings of the 3rd International Conference on Genetic Algorithms*, 160-169.
 Davis, L. (1991), *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York.
 Dogramaci, A. and Surkis, J. (1979), Evaluation of a heuristic for scheduling independent jobs on parallel identical processors, *Management Science*, 23, 1208-1216.
 Du, J. and Leung, J. (1990), Minimizing total tardiness on one machine is NP-Hard, *Mathematics of Operations Research*, 15, 483-495.
 Goldberg, D. (1989), *Genetic algorithms in search optimization and machine learning*, Addison Wesley, Reading, Mass.
 Goldberg, D. and Lingle, R. (1985), Alleles, loci, and the traveling salesman problem, *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, 154-159.
 Grefenstette, J. (1987), Incorporating problem specific knowledge into genetic algorithms, in Darig, L. (ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufman, San Mateo, CA, 42-60.
 Holland, J. (1975), *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor.
 Koulamas, C. (1997), Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem, *Naval Research Logistics*, 44, 109-125.
 Lenstra, J., Rinooy Kan, A. and Brucker, P. (1997), Complexity of machine scheduling problems, *Annals of Discrete Mathematics*, 1, 343-362.
 Liu, M., Wu, C. and Jiang, X. (1998), Genetic algorithm method for

- minimizing the number of tardy jobs in identical parallel machine scheduling problem, *Chinese Journal of Electronics*, 7, 188-192.
- Moscato, P. and Norman, M. (1992), A memetic approach for the traveling salesman problem: implementation of a computational ecology for combinatorial optimization on message-passing systems, *Proceedings of the International Conference on Parallel Computing and Transportation Applications*, 177.
- Nakano, R. (1991), Conventional genetic algorithm for job shop problems, *Proceedings of the 4th International Conference on Genetic Algorithms*, 474-479.
- Oliver, I., Smith, D. and Holland, J. (1987), A study of permutation crossover operators on the traveling salesman problem, *Proceedings of the 2nd International Conference on Genetic Algorithms*, 224-230.
- Rachamadugu, R. (1987), A note on the weighted tardiness problem, *Operations Research*, 35, 450-452.
- Spears, W. and DeJong, K. (1991), On the virtues of parameterized uniform crossover, *Proceedings of the 4th International Conference on Genetic Algorithms*, 230-236.
- Starkweather, T., McDaniel, S., Mathias, K., Whitley, C. and Whitley, D. (1991), A comparison of genetic sequencing operators, *Proceedings of the 4th International Conference on Genetic Algorithms*, 69-76.
- Syswerda, G. (1991), Scheduling optimization using genetic algorithms, in *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 332-349.
- Whitley, D., Starkweather, T. and Fuquay, D. (1989), Scheduling problems and traveling salesman: the genetic edge recombination operator, *Proceedings of the 3rd International Conference on Genetic Algorithms*, 133-140.
- Woolsey, R. (1990), Production scheduling quick and dirty method for parallel machines, *Production Inventory Management Journal*, 31, 84-87.