

상이한 납기와 도착시간을 갖는 단일기계 일정계획을 위한 유전 알고리즘 설계

이동현* · 이경근* · 김재균** · 박창권** · 장길상***

A Genetic Algorithm for Single Machine Scheduling with Unequal Release Dates and Due Dates

Donghyun Lee* · Kyungkeun Lee* · Jaegyun Kim** ·
Changkwon Park** · Gilsang Jang***

■ Abstract ■

In this paper, we address a single machine non-preemptive n-job scheduling problem to minimize the sum of earliness and tardiness with different release times and due dates. To solve the problem, we propose a genetic algorithm with new crossover and mutation operators to find the job sequencing. For the proposed genetic algorithm, the optimal pair of crossover and mutation rates is investigated. To illustrate the suitability of genetic algorithm, solutions of genetic algorithm are compared with solutions of exhaustive enumeration method in small size problems and tabu search method in large size problems. Computational results demonstrate that the proposed genetic algorithm provides the near-optimal job sequencing in the real world problem.

1. 서 론

과거 수년동안 일정계획에 대한 연구는 정규평
가함수(regular measures)라고 언급되는 단일평가

척도(single performance measures)들에 대해서만
진행되었으며, 대부분의 문헌들은 평균처리시간,
평균지연시간, 총 처리시간, 납기 지연된 작업의
수 등의 최소화를 목적으로 하는 일정계획 문제들

* 부산대학교 산업공학과
** 울산대학교 산업공학과
***한국오라클㈜

을 다루고 있다. 따라서 이들 평가 기준들의 대부분은 납기의 준수여부에 달려있다. Baker & Scudder[1]는 동일한 납기를 갖는 단일기계에서 납기를 만족시키지 못했을 때 발생하는 비용이 작업의 완료시점에 종속 되어 있는 문제의 일정계획을 수립하였다.

이러한 개념은 JIT 제조 환경하에서는 간과할 수 없는 것이다. 왜냐하면 제품을 조기생산을 하는 경우와 지연생산을 하는 경우 모두 비용을 발생시키기 때문이다[3]. 그래서 이상적인 일정계획은 모든 작업들을 정확하게 그들의 납기에 맞추는 것이다. 이러한 문제들은 다양한 형태의 목적함수를 가지는 문제로 표현이 가능한데, 특히 납기와 완료시점 간의 편차를 최소화 하는 것이 목적이다. 이와 관련된 문헌들을 살펴보면 다음과 같다. Kim과 Yano[9]는 상이한 납기를 갖는 단일기계 문제에 대하여 최적해에 대한 특성들을 논의하였고, 이를 해결하기 위한 발견적 기법(heuristic algorithm)을 제시하였다. 또한, Garey와 Johnson[4]는 단일기계에서 도착시간과 납기가 일반적인 경우에 지연작업의 수를 최소화 하는 문제가 NP-hard임을 입증하였다. Lee와 Choi[11]는 단일기계 문제에서 납기가 상이하고 도착시간이 동일 할 때, 조기생산과 지연생산에 대한 비용이 일반적인 경우에 대하여 유전 알고리즘(genetic algorithm)을 제시하였다. Gupta et al.[9]는 완료시점에 따라 가중치가 변화하는 단일기계 문제에서 유전 알고리즘의 응용을 보여 주었다. 최근에 Sule[13]은 상이한 납기와 도착시간을 가지는 단일기계 일정계획 문제에서 목적함수가 가중치의 합을 최소화하는 경우에 대하여 2단계(two phase) 발견적 기법을 개발하였다.

이러한 연구들의 대부분에 적용된 알고리즘들은 일반적인 작업할당 규칙들을 변경 시켜보면서 그들 간의 유용성(efficiency) 및 효과성(effectiveness)을 비교 분석하여 기존의 알고리즘보다 효과적임을 입증하는 형태의 해법들을 제시하였다. 그러나 현실적인 일정계획을 수립하기 위해서는 불규칙적인 납기, 도착시간 등을 동시에 고려한 실현

가능 해(realizable solution)를 제시할 수 있는 해법이 바람직한 것이다.

본 연구에서는 단일기계에서 조기생산과 지연생산의 합을 최소로 하면서 도착시간과 납기를 동시에 고려한 일정계획 문제를 해결하기 위한 유전 알고리즘을 제안하고자 한다. 또한, 제안된 유전 알고리즘의 효과성을 검증하기 위하여, 작은 규모의 문제는 열거법(enumeration method)의 최적해와 비교하고, 규모가 큰 문제에 대해서는 지능형 탐색 기법의 하나인 타부서치(tabu search) 방법의 해와 비교한다. 본 논문의 구성은 다음과 같다. 2장에서는 문제정의에 필요한 기호 및 가정을 설명하며, 3장은 제안된 유전 알고리즘에 대하여 설명하고, 4장은 전산실험을 통하여 제안된 유전 알고리즘의 성능을 분석하고, 결론 및 추후 연구방향이 5장에서 설명된다.

2. 문제 정의

본 연구에서는 수주생산의 성격이 매우 강한 H사의 선박엔진 조립공장에 있어서 실질적인 일정계획 수립 문제를 고려하고자 한다. 제품의 특징이 고비용 기술집약형 이기 때문에 조기생산을 하면 제품의 재고비용 및 보관장소의 활용문제가 발생하고, 지연생산을 하게 되면 선박의 전체 건조일정에 영향을 줄 뿐 아니라 고객에게 지체 보상 비용을 지불하여야 한다. 따라서, 이 공장에서의 중요한 점은 각 작업들에 대한 납기를 얼마나 정확하게 맞추어 줄 수 있는가 라는 것이다. 각 작업들의 납기를 정확하게 맞추려면 가장 중요한 것이 조립공장에서의 조립착수 시점을 결정하여야 한다. 따라서 합리적인 조립착수 시점은 조립공장 내에 있는 조립장비(test-board)의 일정계획 수립에 의하여 결정되는 것이다[12].

본 연구에서는 고객들로부터 주문되는 오더를 작업으로 가정하였다. 또한 이들 작업들은 각각 상이한 도착시간과 납기를 가진다. 이러한 문제들은 각 작업들에 대하여 납기를 정확하게 맞추지 못하

면 즉, 조기생산 및 지연생산 모두에 대하여 비용이 초래 되기 때문에 높은 비용이 발생한다. 그러므로 납기와 작업의 완료시점 간의 편차를 최소화하는 것이 목적이다. 본 연구에서는 이 문제를 MINSUM-SM-ET 모델이라고 정의한다. Garey et al. [5]는 목적함수가 조기생산과 지연생산의 합을 최소화하고, 각 작업들의 납기가 서로 상이하고 도착시간이 동일 할 때, 단일기계에서 이들의 일정을 수립하는 것이 NP-hard문제임을 처음으로 입증하였다. 따라서, 본 논문에서 고려하는 단일기계에서의 각 작업들에 대한 상이한 도착시간과 납기를 가지면서 조기생산과 지연생산의 합을 최소로 하는 일정계획 문제는 Garey et al.의 문제에서 도착시간을 일반화한 것이므로 쉽게 NP-hard라는 것을 알 수 있다.

본 연구에서 다루는 MINSUM-SM-ET 모델은 아래의 가정을 갖는다.

- (1) 각 작업은 도착시간 이후의 시점부터는 언제든지 착수가 가능하다.
- (2) 긴급작업이 발생하더라도 현재 진행중인 작업을 중지하고 다른 작업을 시작할 수는 없다.
- (3) 모든 작업들에 대한 조기 및 지연생산에 대한 비용은 동일한 비율로 가정한다.

또한 본 논문에 사용한 기호들을 정의하면 다음과 같다.

- j : 작업의 인덱스 ($j = 1, 2, \dots, n$)
- $[i]$: i 번째로 배정된 작업 ($i = 1, 2, \dots, n$)
- p_j : 작업 j 의 조립시간(processing time)
- r_j : 작업 j 의 도착시간(release time)
- d_j : 작업 j 의 납기(due date)
- c_j : 작업 j 의 완료시간(completion time)
- a : 조기생산된 작업에 대한 비용(earliness cost)
- β : 지연생산된 작업에 대한 비용(tardiness cost)
- E_j : 작업 j 의 조기생산 기간(earliness time:

$$E_j = \max(0, d_j - c_j)$$

T_j : 작업 j 의 지연생산 기간(tardiness time:

$$T_j = \max(0, c_j - d_j)$$

위의 기호를 이용하여 MINSUM-SM-ET 문제를 정의하면 다음과 같다.

$$\text{Min } f(S) = \left\{ \sum_j (\alpha E_j + \beta T_j) \right\} \quad (1)$$

위의 식 (1)은 납기를 맞추지 못했을 때 발생하는 조기생산과 지연생산의 합을 최소화하는 목적함수를 나타낸다. 이것은 각 작업들의 완료시점이 납기에 근접할수록 값이 줄어들기 때문에 이것은 완료시점에 관한 함수로써 표현된다. 따라서 각 작업의 완료시점을 가능한 한 납기에 맞추는 것이 보다 바람직한 접근방법이다.

3. 유전 알고리즘 설계

유전 알고리즘은 일정계획, 그룹 테크놀로지, 외판원 문제 등과 같이 많은 분야에서 조합 최적화(combinatorial optimization) 문제의 최적 또는 유사 최적해를 구하는데 강력한 도구로써 사용되고 있다[6]. 이것은 모집단이라 불리는 랜덤하게 선택된 초기 해 집합으로부터 시작하여 해를 점차 개선시키기 위하여 생물학적인 진화과정과 유사한 방법으로 탐색해 나가는 것이다. 모집단을 구성하는 각 요소들을 염색체(chromosome)라 하고, 이들 염색체를 구성하는 요소를 인자(gene)라고 한다.

유전 알고리즘을 이용하여 다양한 문제를 해결함에 있어서 무엇보다 중요한 것은 주어진 문제의 해를 자연스럽게 표현하는 것과 유사 최적 해를 얻을 수 있도록 유전 연산자(genetic operators)들을 설계하는 것이다. 이것들을 세부적으로 나열 해 보면 해의 표현(representation), 해의 초기화(initialization), 평가함수(evaluation function), 교차변이(crossover), 돌연변이(mutation) 및 선택(selection)등이 있다. 또한 유전 파라미터(parameter)들로 모집단 크기(population size: pop_size), 최대 세대수(maximum number of generation: max_gen),

교차변이 확률(crossover probability: p_c) 및 돌연변이 확률(mutation probability: p_m)이 있으며 유전 알고리즘을 실행하기 이전에 반드시 정의 되어야 한다.

3.1 해의 표현과 초기화

본 연구에서는 작업 리스트로 표현되는 순열 표현을 사용한다. 초기 모집단을 구하는 데는 다양한 개체를 생산하기 위하여 난수 키(random keys)를 이용하였다. 이것은 [0,1]사이의 난수를 발생시켜서 이 값들을 정렬하여 해를 표현하는데 사용한다. 예를 들어 6개의 작업에 대한 염색체 구성은 다음과 같다.

난수 키 리스트 : [0.15, 0.89, 0.13, 0.97, 0.56, 0.78]

여기서 리스트 내의 위치 j 는 작업 j 를 표현하고, 위치 j 에 있는 난수는 작업 j 의 순서를 표현한다. 난수 키 리스트를 올림차순으로 정리하면 다음과 같은 작업리스트가 도출된다.

작업리스트 : [3, 1, 5, 6, 2, 4]

그리고 작업들은 정수로 표현된다. 초기화 과정은 모집단 크기만큼의 가능 해 집합을 랜덤하게 난수를 발생시켜 초기 가능 해를 구성한다.

3.2 해의 평가 함수

세대를 반복하는 동안 해를 개선하기 위해서는 적합도(fitness)라고 하는 평가척도를 사용하여 모집단의 염색체들을 평가한다. 대부분의 최적화 문제의 목적함수는 유틸리티를 최대화 하는 것이기 때문에 적합도의 평가척도를 원래 목적함수를 토대로 계산한다. 그러나 본 연구에서 다루는 문제는 목적함수를 최소화 하는 것이기 때문에 각 염색체에 대한 적합도의 평가척도를 계산하기 위하여 목적함수의 형태를 최대화 하는 문제로 변형을 해야 한다. 이러한 평가척도로서 다음과 같이 변형된 목

적합수를 본 논문에서의 적합도 함수(fitness function)로 이용한다.

$$fit(c_k) = 1/f(c_k) \quad (2)$$

$$k = 1, 2, \dots, pop_size$$

여기서, 식 (2)의 $fit(c_k)$ 는 k 번째 염색체의 적합도 함수이고, $f(c_k)$ 는 k 번째 염색체에 대한 목적함수 값을 나타낸다. 목적함수를 구하기 위해서는 각 작업들의 시점을 결정해야 하는데 그 방법은 다음과 같다. 랜덤하게 생성된 각 작업 리스트(염색체) 내의 인자(작업)들의 순서를 그대로 인정한 상태에서 현재의 리스트 내에 위치 해 있는 순서대로 각 작업의 납기에 완료시점을 맞추어서 후방전개(backward) 방식으로 착수 및 완료시점을 결정한다. 이 과정을 수식으로 표현하면 다음과 같다.

$$c_{[j]} = \begin{cases} c_{[i-1]} + p_{[j]}, & \text{if } d_{[j]} - c_{[i-1]} < p_{[j]} \\ d_{[j]}, & \text{otherwise} \end{cases} \quad (3)$$

3.3 유전 연산자

현 세대의 가능 해 집합으로부터 다음 세대에 더 적합하고 좋은 가능해 집합들을 구성하기 위한 유전 연산자는 교차변이 연산자(crossover operator)와 돌연변이 연산자(mutation operator)가 있다. 교차변이 연산자는 부모 염색체의 몇 개 인자를 서로 교환하여 새로운 해를 생성한다. 반면에 돌연변이 연산자는 염색체 구조에 아주 작은 변화를 줌으로써 해의 개선을 돕는 역할을 한다. 본 논문에서는 새로운 교차변이 연산자와 돌연변이 연산자를 개발하였다. 이것은 근본적으로 두 부모의 염색체로부터 두 자식의 염색체를 생성한다. 개발한 교차변이 연산자와 돌연변이 연산자의 절차는 다음과 같다.

교차변이 절차 (Crossover Procedure) :

단계 1 : 첫번째 부모로부터 랜덤하게 두 지점을 선택한다. 만일 선택된 두 지점의 위치가 동일

하면, 또다시 랜덤하게 한 지점을 선택해서 동일하지 않게 선택한다.

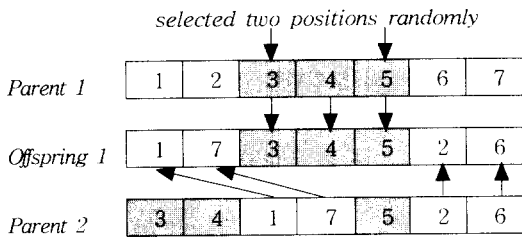
단계 2: 랜덤하게 선택된 두 지점 사이에 있는 작업들의 위치를 첫 번째 자식염색체에 그대로 상속 시킨다.

단계 3: 첫 번째 부모로부터 상속되지 못한 나머지 인자(작업)들은 두 번째 부모로부터 왼쪽에서 오른쪽으로 이동시키면서 인자들을 자식염색체에 상속하여 첫 번째 자식염색체를 생성한다.

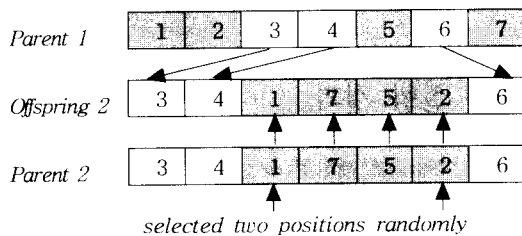
단계 4: 두 번째 자식염색체는 단계1의 첫 번째 부모 대신에 두 번째 부모를 기준으로 하고, 단계3의 두 번째 부모가 첫 번째 부모로 변경되고 나머지 과정은 동일하게 수행한다.

단계 5: $(pop_size \times p_c)$ 만큼 반복 생성한다.

두 부모 염색체로부터 두 자식 염색체를 생성하는 예제가 각각 [그림 1]과 [그림 2]와 같다. [그림 1]은 첫 번째 자식 염색체를 생성하는 과정이고, 두 번째 자식 염색체를 생성하기 위한 과정은 [그림 2]와 같다.



[그림 1] 첫 번째 자식 염색체를 생성하기 위한 교차변이 연산자의 예



[그림 2] 두 번째 자식 염색체를 생성하기 위한 교차변이 연산자의 예

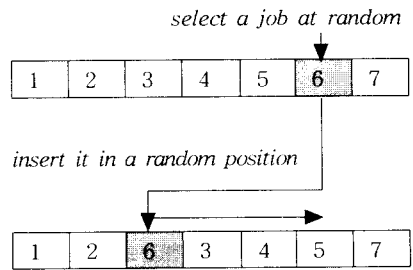
돌연변이 절차(Mutation Procedure) :

단계 1: 동일한 염색체 상에서 랜덤하게 두 지점을 선택한다. 첫 번째 위치의 인자를 선택된 인자라 하고, 두 번째 위치는 선택된 인자가 삽입(insertion)되어야 할 위치라 한다. 여기서, 선택된 인자의 위치와 삽입되어야 할 위치가 동일하면, 삽입되어야 할 위치를 다시 랜덤하게 선택한다.

단계 2: 선택된 인자의 위치와 삽입되어야 할 위치를 비교하여 만일, 선택된 인자의 위치가 삽입되어야 할 위치 보다 크면, 삽입되어야 할 위치에 선택된 인자를 삽입한 후, 나머지 인자들은 왼쪽에서 오른쪽으로 이동한다. 그렇지 않으면, 삽입되어야 할 위치에 선택된 인자를 삽입한 후, 나머지 인자들은 오른쪽에서 왼쪽으로 이동한다.

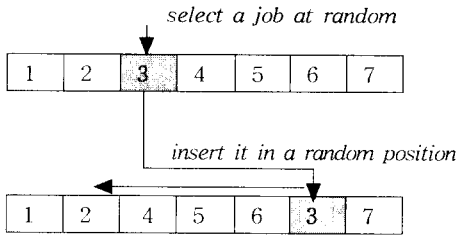
단계 3: $(pop_size \times p_m)$ 만큼 반복 생성한다.

돌연변이 절차에 대한 과정을 도식하면 [그림 3]과 [그림 4]와 같다. 이 연산자의 특징은 선택된 인자의 위치와 삽입될 인자의 위치에 따라 이동방향이 결정되는 양방향성(bi-direction)을 가지고 있다. [그림 3]은 선택된 인자의 위치가 6번째이고, 삽입되어야 할 위치는 3번째 일 때의 염색체 생성과정을 나타낸 것이다.



[그림 3] 돌연변이 절차

[그림 4]은 선택된 인자의 위치가 3번째이고, 삽입되어야 할 위치는 6번째 일 때의 염색체 생성과정을 나타낸 것이다.



[그림 4] 돌연변이 절차

3.4 선택 기준

선택기준은 다음세대의 자손을 생성하기 위하여 현재의 부모 모집단에서 염색체들을 어떻게 선택할까 하는 방법에 대한 것이다. 이것은 부모 염색체들의 적합도를 기준으로 선택된다. 본 논문에서는 룰렛휠(roulette wheel) 방법이 상대적으로 적합도가 높은 부분에서만 해를 선택하기 때문에 해의 다양성을 확보하는데 있어서 편견을 내포하고 있다. 따라서 이러한 통계적인 오류를 보완할 수 있는 선택 방법들 중의 하나인 *remainder stochastic sampling without replacement* 방법[8]과 이전 세대에서 가장 좋은 해를 발생시킨 염색체는 반드시 선택하여 다음 세대로 상속 시키는 엘리티즘(elitism) 방법을 조합하여 이용하였다. 여기서, *remainder stochastic sampling without replacement*에 대하여 절차를 간략하게 설명하면 다음과 같다. 각 염색체별로 적합도를 구한 후, 각 적합도에 대하여 정규화된 값을 구한다. 그러면 이들(정규화된 값)은 정수부분과 소수부분으로 표현이 되는데, 이것을 정수부분 만큼 염색체를 자손세대로 상속 시킨다. 이러한 과정을 적합도가 가장 좋은 염색체들로부터 순서대로 모집단 크기 만큼 반복한다. 만일, 정수부분만 가지고도 모집단 크기 만큼 생성이 되지 않으면, 적합도의 순서대로 정렬되어 있는 염색체들의 소수부분을 가지고 자손세대로 상속 시킨다.

3.5 유전 알고리즘

본 논문에서 제안된 유전 알고리즘의 전반적인

과정을 요약하면 다음과 같다.

단계 1 : 초기화 (Initialization)

초기해를 랜덤하게 *pop_size*만큼 생성한다.

단계 2 : 평가(Evaluation)

현재 모집단의 각 작업 리스트(염색체)들에 대하여 식 (1)의 목적함수를 계산한다. 그리고 식 (2)에 의해서 적합도를 평가한다.

단계 3 : 정규화(Computation of normalized values)

현 세대에서 다음세대의 작업 리스트들을 선택할 때 통계적인 오류를 보정하기 위하여 각 작업 리스트들에 대하여 정규화 수치(normalized values: η_k)를 계산한다.

$$\eta_k = \{fit(c_k) / \sum_{k=1}^{pop_size} fit(c_k)\} \times pop_size \quad (4)$$

단계 4 : 재생(Reproduction)

*remainder stochastic sampling without replacement*를 이용하여 다음 세대로 보다 우수한 작업 리스트들을 상속 시키기 위하여 현 세대에서 *pop_size* 만큼 재생성한다.

단계 5 : 재구성(Recombination)

단계 4로부터 재생된 모집단에서 다음 세대의 모집단을 생성하기 위하여 교차변이 확률 만큼 연산을 하고, 그리고 돌연변이 확률 만큼 연산을 한다. 또한, 현 세대에서 가장 좋은 작업 리스트를 다음 세대로 상속 시켜서 모집단을 형성한다.

단계 6 : 종료(Termination)

정해진 반복 수 및 시간을 만족하면 중지하고, 그밖에는 단계 2부터 계속 수행한다.

4. 실험 결과

본 절에서는 제안된 유전 알고리즘의 성능을 평가하기 위하여 H사의 선박용 엔진 조립공장의 상황을 고려하였다. 실제로 선박용 엔진을 조립하기 위하여 수행되는 조립시간은 엔진타입에 따라 약

간의 차이는 있겠지만 평균적으로 30일에서 40일 정도 소요된다. 그리고 작업의 도착은 일년동안 계속 발생하며, 작업이 조립공장에 도착한 후 완료되기 까지 조립시간을 제외하고 평균적으로 허용되는 여유시간은 20일 정도이다. 이러한 현장 상황을 고려하여 실험 조건들을 다음과 같이 설정하였다.

- (1) 작업들의 조립시간(p_j)은 [30, 40] 사이의 이산형 일양 분포(discrete uniform distribution)로부터 랜덤하게 발생시킨다.
- (2) 작업들의 도착시간(r_j)은 [1, 365] 사이의 이산형 일양 분포로부터 랜덤하게 발생시킨다.
- (3) 각 작업의 납기(d_j)는 [$r_j + p_j, r_j + p_j + k_j$] 사이의 이산형 일양 분포로부터 랜덤하게 발생시킨다. 여기서 k_j 는 허용치(allowance value)를 의미하는데, 이것의 범위는 [10, 30] 사이의 이산형 일양 분포로부터 랜덤하게 발생시킨다.

위의 실험조건을 적용하여 효율적으로 해를 탐색함을 입증하기 위하여 전산실험을 수행하였다. 유전 알고리즘은 IBM 워크스테이션인 RS/6000-R40에서 C언어로 프로그래밍 하였고, 이것을 랜덤하게 생성된 많은 문제에 대하여 전산실험을 하였다.

제안된 유전 알고리즘의 효율성을 입증하기 위하여 우선, 평가함수의 성능을 최적화 시키기 위하여 상이한 파라미터 집합(parameter set : (p_c, p_m)) 들에 대하여 작업의 수가 10개($n = 10$)인 문제를 랜덤하게 10회 생성하고, 파라미터 집합의 종류가 20개인 경우를 가지고 실험하여 보았다.

<표 1>은 실험에 사용된 파라미터 집합들의 종류와 그 결과를 보여주고 있다. 여기서 평균 반복수(average generation)는 제안된 유전 알고리즘이 열거법에 의해서 구한 최적 해를 생성하기까지의 평균 반복 수를 의미하고, 평균 CPU시간은 최적 해에 도달하기 까지 걸린 평균시간을 나타낸다. 파라미터 집합을 최적화하기 위하여 사용된 최대 세대수(max_gen)는 500이고, 모집단(pop_size)의

크기는 100이다. <표 1>로부터 파라미터 집합(p_c, p_m)이 (0.20, 0.70)의 조합을 가질 때 평균적으로 가장 짧은 시간 안에 최적 해를 찾는다 라는 것을 알 수 있었다. 이 실험 결과를 분석하여 보면, 돌연변이가 교차변이 보다 해의 질을 향상시키는 데 더 많이 기여한다는 것을 알 수 있었다. 또한 돌연변이 비율이 증가함에 따라 탐색 시간이 비교적 적게 걸린다는 것을 알 수 있었다. 따라서 본 연구에서는 가장 적합한 파라미터 집합(p_c, p_m)을 (0.20, 0.70)으로 고정하고, 작업의 수 6, 7, 8, 9, 10, 15, 20, 30, 40, 및 50개일 경우의 각각에 대하여 랜덤하게 문제를 10회씩 생성하고, max_gen이 1000이고 pop_size가 100으로 설정하여 실험을 수행하였다.

<표 1> 파라미터 집합 (p_c, p_m)에 대한 실험 결과

No.	p_c	p_m	평균 CPU 시간(second)	평균 반복수
1	0.00	1.00	3.40	119.8
2	0.05	0.85	3.80	124.2
3	0.15	0.85	4.40	141.1
4	0.20	0.70	3.00	97.8*
5	0.20	0.60	3.30	116.5
6	0.30	0.80	4.90	165.7
7	0.30	0.70	4.34	143.4
8	0.40	0.70	5.42	180.3
9	0.40	0.60	5.09	158.2
10	0.50	0.50	8.17	265.2
11	0.60	0.50	5.45	164.7
12	0.60	0.40	5.60	179.5
13	0.70	0.60	6.28	189.5
14	0.70	0.40	7.10	227.3
15	0.80	0.20	7.35	223.6
16	0.85	0.15	8.43	247.5
17	0.85	0.05	10.82	327.9
18	0.90	0.30	10.50	316.8
19	0.90	0.10	9.84	303.8
20	1.00	0.00	8.47	247.5

제안된 유전 알고리즘을 평가하기 위해서 작업

〈표 2〉 유전 알고리즘과 열거법 및 타부서치의 차이(GAP) 결과 비교

(단위: %)

작업의 수(n) 차이(GAP)		작업의 수(n)									
		6	7	8	9	10	15	20	30	40	50
OPT us. GA	최소	0.00	0.00	0.00	0.00	0.00	*	*	*	*	*
	평균	0.00	0.00	0.00	0.00	0.00	*	*	*	*	*
	최대	0.00	0.00	0.00	0.00	0.00	*	*	*	*	*
TS us. GA	최소	0.00	0.00	0.00	0.00	0.00	0.00	-0.27	-0.22	0.23	0.48
	평균	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.89	1.24	2.67
	최대	0.00	0.00	0.00	0.00	0.00	0.00	0.90	1.61	2.50	6.10

의 수가 6개부터 10개인 경우에 대하여 열거법으로 최적해를 구하여 유전 알고리즘에 의해서 구한 해와 비교하였다. 그러나 작업의 수가 15개 이상인 경우는 열거법으로 최적해를 구하기가 곤란하여 타부서치에 의해서 구한 해와 비교검증 하였다. 본 논문에서 사용된 타부서치의 알고리즘은 부록에 기술하였다[7]. 타부서치에 적용된 파라메타는 유전 알고리즘에서 생성된 동일한 문제에 대하여 최소 타부목록(minimum tabu list)의 크기는 20이고, 최대 타부목록(maximum tabu list)의 크기는 최소 타부목록 크기의 3배로 정의하였다. 그리고 열망수준(aspiration level)은 [1.0, 2.0]이고, 최대 반복 수는 GA와 동일한 크기의 개체 수를 생성하기 위하여 작업의 크기에 따라 다르게 지정하여 실험을 수행하였다.

〈표 2〉는 각 작업의 수에 대하여 랜덤하게 10문제씩 생성하여 수행한 실험의 결과를 요약한 것이다. 여기서, ‘*’ 표시는 열거법에 의해서 이항계산시간 (polynomial time)내에서 최적해를 도출할 수 없음을 의미한다. OPT는 열거법에 의해서 구한 최적해라고 정의하고, GA는 제안된 유전 알고리즘에 의해서 구한 해라고 정의한다. 그리고 TS는 타부서치에 의해서 구한 해라고 정의한다. OPT us. GA에 대한 차이(gap)는 $\{((OPT - GA)/GA) \times 100\}$ 로 계산하였고, TS us. GA에 대한 차이는 $\{((TS - GA)/GA) \times 100\}$ 로 계산한 것이다. 평균은 해당 작업의 크기 범위내의

각각 차이들에 대한 평균치를 의미하고, 최소와 최대는 이러한 범위내의 차이들 중에서 각각 가장 작은 차이와 가장 큰 차이를 의미한다.

위의 결과에서, 제안된 유전 알고리즘은 문제의 크기가 작은 경우에 대해서는 최적 해를 도출하였고, 문제의 크기가 증가함에 따라 타부서치에서 얻어진 해와 동일하거나 우위에 있음을 보여주었다. 이 결과로 제안된 유전 알고리즘을 적용하여 구한 해의 타당성이 검증되었다.

5. 결 론

본 논문에서는 상이한 납기와 도착시간을 고려한 단일기계 일정계획 문제를 해결하기 위하여 도착시간과 납기를 동시에 고려하여 실현가능 해를 찾을 수 있는 유전 알고리즘 해법을 개발하였다. 개발된 유전 알고리즘의 효과성을 입증하기 위하여, 규모가 작은 문제들은 열거법으로 구한 최적해와 비교하였으며, 규모가 큰 문제에 대해서는 타부서치 방법으로 구한 해와 비교하였다. 비교 결과, 제안된 유전 알고리즘이 상이한 납기와 도착시간을 고려한 단일기계 일정계획 문제에 대하여 최적 해 또는 유사 최적해를 제공하여 준다는 것을 보여주었다.

향후의 연구과제로는 비정규 평가척도를 갖는 병렬기계 일정계획 수립 문제로 확장하여 연구할 필요가 있다.

부록 : 타부서치 알고리즘

본 논문에서 사용한 타부서치 알고리즘의 절차를 기술하면 다음과 같다.

단계 1 : 초기화(Initialization)

- (1) 타부속성(tabu attribute)으로써 작업과 그 위치로 정의한다.
- (2) 타부목록(tabu list) 크기는 20으로 하였고, 열망수준은 1.0과 2.0 사이에서 일정한 증분치를 사용한다. 그리고 종료조건은 최대 반복 수로 정의한다.
- (3) 초기 가능해는 랜덤하게 생성하여 이를 현재해(incumbent solution)로 설정한다.

단계 2 : 이웃해 생성 (Neighborhood solution)

- (1) 현재해로부터 타부이동이 아니거나 타부이동이지만 열망수준을 만족하는 이웃해를 생성한다. 여기서 사용된 타부이동은 삽입이동(insert move)방법을 사용한다.

단계 3 : 현재해와 최선해(Best solution) 수정

- (1) 생성된 이웃해 중에서 가장 좋은 해를 현재해로 한다. 이때 평가기준은 본문의 식 (1)에 의해서 계산한다.
- (2) 현재해가 최선해 보다 좋으면 현재해를 최선해로 둔다. 만일 현재해와 최선해가 동일하면, 단계 5로 간다.

단계 4 : 타부목록 수정

- (1) 새로운 현재의 이동속성을 타부목록에 기록한다.
- (2) 만약 타부목록에 저장된 타부속성의 수가 타부목록 크기보다 크면 가장먼저 기록된 타부속성을 삭제한다. 단계 2로 간다.

단계 5 : 다각화(Diversification)

- (1) 타부목록에 이미 저장되어 있는 최선해의

값에 열망수준을 곱해서 새로운 임계치(threshold)를 계산한다.

- (2) 저장하려는 이웃해가 (1)에서 계산된 임계치와 비교하여 적거나 같으면 타부목록에 저장을 하지않고, 이것을 가지고 다시 이웃해를 생성한다.
- (3) (2)에서 생성된 이웃해가 (1)에서 계산된 임계치 범위내에 포함되면 타부목록에 기록을 한다. 그렇지 않으면, (2)의 과정으로 간다.
- (4) (2)와 (3)의 과정을 지정된 반복횟수 만큼 수행을 하여도 타부목록에 기록할 수 없다면 열망수준을 증가시켜서 새로운 임계치를 계산한다. 그리고 타부목록의 크기를 초기에 설정된 크기의 3배로 증가시킨다.
- (5) 현재의 최선해를 가지고 이웃해를 생성하여 (4)에서 증가된 임계치보다 작으면 타부목록에 기록하고, 이 과정을 증가된 타부목록의 크기만큼 반복한다.
- (6) 타부목록의 크기와 임계치의 값을 초기상태로 되돌린다. 이 때 타부목록을 비울때는 가장 오래된 정보부터 증가된 수 만큼 삭제한다.

단계 6 : 종료(Termination)

- (1) 종료 조건인 최대 반복수를 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

참 고 문 헌

- [1] Baker, K.R. and G.D. Scudder, "Sequencing with Earliness and Tardiness Penalties : a review," *Ops Res.*, Vol.38(1990), pp.22-36.
- [2] Cheng, R., M. Gen, and Y. Tsujimura, "A Tutorial Survey of Job-shop Scheduling Problems Using Genetic Algorithms-I. Representation," *Computers Ind. Engng.*, Vol.30, No.4(1996), pp.983-997.
- [3] Fry, T.D., R.D. Armstrong, and L.D. Rosen,

- "Single Machine Scheduling to Minimize Mean Absolute Lateness : a heuristic solution," *Computers and Ops Res.*, Vol.17(1990), pp.105-112.
- [4] Garey, M. R. and D. S. Johnson, *Computers and Intractability: a guide to the theory of NP-completeness*, W. H. Freeman, San Francisco, CA, 1979.
- [5] Garey, M.R., R.E. Tarjan, and G.T. Wilfong, "One-processor Scheduling with Symmetric Earliness and Tardiness Penalties," *Math Ops Res.*, Vol.13, No.2(1988), pp.330-348.
- [6] Gen M. and R. Cheng, *Genetic algorithms and engineering design*, A Wiley-Interscience Publication, 1997.
- [7] Glover, F., "Tabu Search-Part I," *ORSA Journal on Computing*, Vol.1, No.3(1989), pp.190-206.
- [8] Goldberg, D.E., *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley Publishing Co., 1989.
- [9] Gupta, Y.P., M.C. Gupta, and A. Kumar, "Minimizing Flow Time Variance in a Single Machine System Using Genetic Algorithms," *Europ. J. Opnl Res.*, Vol.70(1993), pp.289-303.
- [10] Kim, Y.D. and C.A. Yano, "Minimizing Mean Tardiness and Earliness in Single-Machine Scheduling Problems with Unequal Due Dates," *Naval Res. Logistics*, Vol.41 (1994), pp.913-933.
- [11] Lee, C.Y. and J.Y. Choi, "A Genetic Algorithm for Job Sequencing Problems with Distinct Due Dates and General Early-Tardy Penalty Weights," *Computers Ops Res.*, Vol.22, No.8(1995), pp.857-869.
- [12] Lee, D.H., J.G. Kim, and J. Hur, "Decision Support System for Dynamic Production Schedule in the Ship Engine Manufacturing System," *Proc. 20th Int. Conf. on Computers Ind. Engng.*, Vol.2, 6-9 Oct. (1996), pp.1115-1118.
- [13] Sule, D.R., *Industrial Scheduling*, PWS Publishing Co., 1997.