韓國經營科學會誌
第25卷 第2號
2000年 6月

납기와 조립가능 시점을 고려한 병렬기계의 스케쥴링을 위한 발견적 해법

이동현* · 이경근* · 김재균** · 박창권** · 장길상***

A Heuristic for Parallel Machine Scheduling with Due Dates and Ready Times

Dong-Hyun Lee* · Kyung-Keun Lee* · Jae-Gyun Kim** ·
Chang-Kwon Park** · Gil-Sang Jang***

Abstract

In this paper, we consider an n-job, non-preemptive and identical parallel machine scheduling problem of minimizing the sum of earliness and tardiness with different release times and due dates. In the real world, this problem is more realistic than the problems that release times equal to zero or due dates are common. The problem is proved to be NP-complete. Thus, a heuristic is developed to solve this problem. To illustrate its suitability, a proposed heuristic is compared with a genetic algorithm for a large number of randomly generated test problems. Computational results show the effectiveness and efficiency of proposed heuristic. In summary, the proposed heuristic provides good solutions than genetic algorithm when the problem size is large.

1. 서 론

JIT(Just In Time) 제조 환경에서는 고객의 납기를 얼마나 정확하게 만족시켜 줄 수 있는가가 중점 목표이다. 납기보다 조기생산을 하면 재고기간에 따른 재고비용이 발생하고, 지연생산을 하게 되면 지연에 따른 벌금(Penalty)을 부과하기 때문에 경제적 손실이 증가 할 뿐만 아니라 고객만족도의

* 부산대학교 산업공학과
** 울산대학교 산업공학부
*** 동국대학교 상경대학 정보산업학과

저하로 인한 기회손실비용이 증가하게 된다[1, 7]. 따라서, 기업의 입장에서는 항상 고객의 납기관리에 관심을 기울이고 있는 실정이다. 가장 이상적인 일정계획은 모든 작업들을 정확하게 그들의 납기에 맞추는 것이다. 그러나 현실적으로 여러 가지의 제약 조건들 때문에, 모든 작업들에 대하여 그들의 납기를 만족시켜주는 것은 어려운 문제이다. 이러한 문제는 조기생산(Earliness)과 지연생산(Tardiness)에 대한 손실비용을 동시에 고려한 일정계획 문제로 정의할 수 있다. 전형적으로 이것은 비정규 평가척도(Non-regular Performance Measure)를 가진 일정계획 문제라고 알려져 있다. 이러한 문제는 실질적으로 수주생산방식(Make-to-order)을 갖는 대부분의 제조업체에서 직면하고 있는 중요한 문제이며, 이를 해결하기 위하여 주로 경험적인 방법에 의존하고 있는 실정이다.

본 연구에서는 수주생산의 성격이 매우 강한 선박용 엔진을 생산하는 공장에서 효과적으로 사용될 수 있는 실질적인 일정계획 수립 문제를 고려한다. 고비용 기술집약형 성격을 갖는 선박용 엔진 생산에서는 일정계획 수립이 기업의 생산성과 연간 사업계획에 절대적인 영향을 미친다. 따라서, 제조현장의 상황을 적절히 반영한 효율적인 일정계획 수립이 요구된다. 선박용 엔진의 제조흐름은 소재공장(주조, 단조) - 가공공장 - 조립공장의 순서로 이루어지며, 엔진의 종류에 따라 전체 제작기간이 약 290일에서 340일 정도가 소요된다. 고객의 납기에 대하여 조기생산을 하면 제품의 재고비용 및 보관장소의 활용문제가 발생하고, 지연생산을 하게 되면 선박의 전체 건조일정에 영향을 줄 뿐 아니라 고객에게 지체 보상 비용을 지불하여야 한다. 고객의 납기와 관련하여 가장 중요한 공정은 조립공장에서의 생산일정이며, 선행공정의 소요시간에 따라 조립가능시점(Ready Time)이 주어지는 복잡한 문제이다. 따라서, 선박엔진의 합리적인 조립착수 및 완료시점은 조립공장 내에 있는 여러 대의 동일한 작업능력을 갖는 조립장비(Test-board)에서의 일정계획 수립에 의하여 결정된다[12]. 이러한 조립일정의 수립은 제품의 납기와 조립가능시점을 고려한 병렬기계에서의 조립일정계획을 수립하는 문제로 분류 될 수 있다.

조기생산과 지연생산을 모두 고려한 기존의 연구는 단일기계 문제와 병렬기계 문제로 분류되어진다. 또한, 작업의 납기는 모든 작업들에 대해서 동일한 경우와 각 작업의 납기가 상이한 경우를 대상으로 한다. Abdul-Razaq and Potts[2], Grey et al.[8],와 Ow and Morton[14]은 단일기계에서 납기가 상이하고 도착시간(Release Time)은 동일한 경우에 대하여 조기생산과 지연생산에 대한 비용의 합을 최소화하는 발견적 기법(Heuristic Algorithm: HA)을 제시하였다. Kim and Yano[11]는 납기가 상이하고 도착시간이 동일한 단일기계 문제에 대하여 최적해에 대한 특성과 발견적 기법을 제시하였다. Baker and Scudder[3]는 단일기계에서 동일한 납기를 갖는 경우와 상이한 납기를 갖는 문제를 다룬 기존의 논문들을 목적함수 형태별로 정리하였다.

Cheng and Chen[4]은 병렬기계에서 동일한 도착시간을 가진 경우에 대하여 조기생산과 지연생산의 합을 최소로 하는 작업의 순서와 공통납기를 결정하는 문제를 다루었다. Cheng et al.[5]은 병렬기계에서 조기생산과 지연생산의 최대 가중치를 최소화하는 공통납기를 찾는 문제를 해결하기 위한 유전(Genetic) 알고리즘을 개발하였다. Cheng and Sin[6]은 병렬기계에서 일정계획을 수립하는 문제를 연구한 기존의 논문들로부터 목적함수의 형태별로 정리하였다.

이와 같이 기존의 연구들은 모든 작업들에 대하여 도착시간이 동일한 경우를 고려하였다. 그러나 실제 조립공장에서는 선행 공장(제관, 가공공장)들의 소요시간에 따라 실제 조립착수 가능시점(Ready Time)이 결정된다. 이것을 조립공장에서 작업 도착시간으로 간주한다. 이러한 문제는 실제 산업현장의 스케쥴링 분야에서 빈번하게 발생하는 문제이고 또한 시급하게 해결되어야 할 과제중의 하나이다. 그러나 이와 같은 문제의 중요성에도 불구하고 기존

의 문헌고찰을 통하여 확인한 바로는 아직까지 다루어진 바가 없는 것으로 사료된다. 본 연구에서는 이와 같이 산업현장의 보다 현실적인 상황을 반영하기 위하여 선박용 엔진 조립공장을 대상으로 작업능력이 동일한 병렬기계에서 상이한 납기 및 도착시간을 동시에 고려한 조기생산과 지연생산의 합을 최소화하는 문제를 제시하고, 이를 해결하기 위한 효과적인 발견적 해법을 제안하고자 한다. 또한, 제안된 발견적인 해법의 효과성을 검증하기 위하여, 다양한 크기의 문제에 대하여 지능형 탐색 기법(Intelligent Search Method)인 유전 알고리즘(Genetic Algorithm: GA)과 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 문제정의에 필요한 기호 및 가정을 설명한다. 3장에서는 발견적 해법의 성질 및 절차에 대하여 설명하고 4장에서는 발견적 해법의 적합성을 검증하기 위하여 수치예제를 보여준다. 5장에서는 전산실험을 통하여 제안된 알고리즘의 성능을 분석하고, 결론 및 추후 연구과제가 6장에 언급된다.

2. 문제정의

본 연구에서는 n개의 작업($j=\{1,2,\cdots,n\}$)에 대하여 m개의 병렬기계($i=\{1,2,\cdots,m\}$)로 구성된 작업장에서의 일정계획 수립을 대상으로 한다. 고객들로부터의 주문을 작업(Job)으로 가정한다. 또한 이들 작업들은 각각 상이한 도착시간(Ready Time: r_j)과 납기(Due Date: d_j)를 가진다. 각 작업의 조립시간(Processing Time: p_j)은 어떤 장비에서도 동일하다. 여기서, $d_j \geq r_j + p_j$이다. 이러한 문제들은 각 작업들에 대하여 납기를 정확하게 맞추지 못하면 조기생산 및 지연생산 모두에 대하여 비용이 초래 되기 때문에 경제적 손실 및 신뢰도 저하로 인한 문제가 발생한다. 그러므로 납기와 작업의 완료시점 간의 편차를 최소화 하는 것이 목적이다.

Garey et al.[8]은 각 작업들의 납기가 서로 상이하고 도착시간이 동일할 때, 단일기계에서 조기생산과 지연생산의 합을 최소화로 하기 위하여 이들의 일정을 수립하는 것이 NP-complete문제임을 처음으로 입증하였다. 본 논문에서 고려하는 작업능력이 동일한 병렬기계에서의 각 작업들에 대한 상이한 도착시간과 납기를 가지면서 조기생산과 지연생산의 합을 최소로 하는 일정계획 문제는 도착시간과 기계의 수를 일반화한 것이므로 NP-complete 문제로 분류된다.

본 연구에서는 다음과 같은 상황을 가정하고 있다.

(1) 긴급작업이 발생하더라도 현재 진행중인 작업을 중지하고 다른 작업을 시작할 수는 없다.(Non-preemption)
(2) 각 기계는 한번에 하나의 작업만을 처리할 수 있다.

본 논문에 사용한 기호들을 정리하면 다음과 같다.

기호정의 :

i : 기계의 인덱스($i=1,\cdots,m$)
j : 작업의 인덱스($j=1,\cdots,n$)
r_j : 작업 j의 도착시간(Ready Time)
p_j : 작업 j의 조립시간(processing Time)
d_j : 작업 j의 납기(Due Date)
C_j : 작업 j의 조립 완료시간(Completion Time)
E_j : 작업 j의 조기생산 기간(Earliness Time: $E_j = \max(0, d_j - C_j)$)
T_j : 작업 j의 지연생산 기간(Tardiness Time: $T_j = \max(0, C_j - d_j)$)
J_i : 기계 i에 배정된 작업들의 부분집합(Subset)

본 연구에서 다루고자 하는 문제의 목적은 각 작업들의 납기를 만족시키지 못했을 때 발생되는 조기생산과 지연생산의 합을 최소화하는 것이다. 그래서 목적함수를 표현하면 다음과 같이 정의할

수 있다.

$$\begin{aligned} Minimize\ f(S) &= \sum_i \{ \sum_{j \in J_i} (E_j + T_j) \} \\ &= \sum_i f_i(s) \end{aligned} \quad (1)$$

식 (1)의 $\sum_{j \in J_i}(E_j + T_j)$는 기계 i의 부분 일정(Partial Schedule)에 대한 목적함수를 의미하며, $f_i(s)$로 정의한다. 여기서 s는 부분일정을 의미한다. 또한, 본 논문에서는 모든 작업들에 대한 조기 및 지연생산에 대한 단위비용을 동일한 것으로 하여, 해의 특성과 해법을 제안하고자 한다.

3. 발견적 해법 설계

본 연구에서 다루는 상이한 납기와 도착시간을 고려한 병렬기계 일정계획수립 문제는 NP-complete 문제이기 때문에, 이 문제의 특별한 구조를 이용한 발견적 해법의 개발이 요구된다. 먼저, 제안되는 알고리즘 설계에 사용되는 기호들을 정의하면 다음과 같다.

$C_{i[k]}$: 기계 i에서 k번째 위치로 배정되는 작업의 완료시점

$S_{i[k]}$: 기계 i에서 k번째 위치로 배정되는 작업의 착수시점

$p_{i[k]}$: 기계 i에서 k번째 위치로 배정되는 작업의 조립시간

$d_{i[k]}$: 기계 i에서 k번째 위치로 배정되는 작업의 납기

제안된 알고리즘은 다음의 다섯 가지의 원칙들을 기반으로 개발되었다.

원칙 1 : 납기가 빠른 작업을 우선적으로 배정한다.

원칙 2 : 가장 먼저 배정되는 작업은 어느 기계든지 배정이 가능하다.

원칙 3 : 식 (1)의 목적함수를 만족하기 위하여 기계 i에서 k번째 위치로 배정되는 작업 j의 완료시점을 초기에는 가능한 한 작업 j의 납기에 맞춘다. 즉, $C_{i[k]} \rightarrow d_j$이다.

원칙 4 : $S_{i[k]}$를 구하기 위하여 작업이 이동할 수 있는 범위는 기계 i에서 $k-1$번째 위치에 배정되어 있는 작업의 완료시점과 현재 k번째 위치로 배정하려는 작업의 도착시점 중에서 큰 값을 가지는 시점보다 크거나 같다.

원칙 5 : 기계 i에서 작업을 배정할 때, 납기와 완료시점 간의 편차를 최소화 하기 위하여 가능하면 납기에 근접하게 배정한다.

제안된 알고리즘은 다음과 같은 성질을 내포하고 있다.

성질 I. 작업 j를 기계 i에서 k번째 위치에 배정하려고 할 때, 이미 배정되어 있는 작업들 사이에 삽입(Insertion)하려고 하는 경우, 이미 $k-1$번째 위치에 배정되어 있는 작업의 전방과 $k+1$번째 위치에 배정되어 있는 작업의 후방에 연속된 작업이 없을 때, 작업 j의 완료시점을 자신의 납기에 일치시키는 것이 최적해를 보장한다. 이것의 증명은 부록 A에 상세하게 기술하고 있다.

본 연구에서 제안된 발견적 기법은 모든 작업들에 대하여 여유시간을 계산하여 올림차순(Ascending Order)으로 정렬을 한다. 그리고 가장 작은 여유시간을 갖는 작업 j를 선택하여 이용 가능한 기계에 배정하며, 이 때 목적함수를 만족하기 위하여 작업 j의 완료시점을 가능한 한 초기에는 작업 j의 납기에 일치시킨다. 다음단계로 각 기계에 대하여 조기생산 비용과 지연생산 비용의 합을 최소화 하기 위해서 각 작업들의 완료시점을 전후방(Backward-Forward)으로 이동시켜 보면서 배정 될 위치를 찾는다. 이러한 과정을 조기생산 비용과 지연생산 비용의 합이 최소화 되는 기계를 찾을 때 까

지 반복한다. 마지막 단계로 일정계획 대상인 모든 작업들의 배정이 완료되었을 때 끝낸다.

앞에서 언급한 발견적 기법을 단계별로 기술하면 다음과 같다.

단계 1 : 모든 작업들에 대하여 납기를 올림차순으로 정렬한다. 만일 동일한 경우가 발생하면 p_j가 큰 작업 순서로 정렬하고, 그래도 동일하면 r_j가 큰 작업 순서로 정렬하고, 여전히 동일하면 임의로 작업을 정렬한다.

단계 2 : 단계 1에서 정렬된 작업을 순차적으로 선택한다. 선택된 작업 j를 배정할 기계 i를 결정하기 위하여 다음과 같은 절차를 따른다. 선택된 작업 j를 배정하기 위하여 모든 기계 i에 대하여 탐색하여 목적함수 증가를 최소화하는 기계에 배정한다. 만일 $p_{i[k]} \leq (d_j - C_{i[k-1]})$이고 $d_j \leq S_{i[k+1]}$이면, 단계 2.1을 수행하고, 그렇지 않으면, 단계 2.2를 수행한다.

단계 2.1 : 작업 j를 기계 i에 배정한다. 즉, $J_i \leftarrow J_i \cup \{j\}$로 변경된다. 그리고 부분일정(Partial Schedule)의 목적함수 $f_i(s)$를 계산한다.

단계 2.2 : 다음의 4가지 이동전략(Moving Strategy)을 이용하여 목적함수 증가를 최소화하는 기계 i에 작업 j를 배정한다.

① 연속된 작업의 수가 적은 방향으로 작업 j의 완료시점을 이동시킨다.

② 작업 j의 완료시점을 이미 배정되어 있는 $k+1$번째 위치의 작업 착수시점에 일치시킨 후, {$k-1$번째 위치의 작업, $k-2$번째 위치의 작업, $k-3$번째 위치의 작업, …}들을 전방으로 이동시킨다.

③ 작업 j의 착수시점을 이미 배정되어 있는 $k-1$번째 위치의 작업 완료시점에 일치시킨 후, {$k+1$번째 위치의 작업, $k+2$번째 위치의 작업, $k+3$번째 위치의 작업, …} 들을 후방으로 이동시킨다.

④ 작업 j의 완료시점을 납기(d_j)에 일치시키고 이미 배정되어 있는 작업들을 전후 방향으로 이동시킨다.

단계 2 : 만일 배정해야 할 작업이 없다면 종료하고, 그렇지 않으면 단계 2를 반복 수행한다.

4. 수치 예

본 논문은 제안된 알고리즘의 성능을 수치 예를 들어서 설명한다. <표 1>은 기계의 대수가 2대($m=2$)이고, 작업의 수가 9개($n=9$)일 때 각 작업별 도착시간, 조립시간 및 납기에 관한 자료를 보여 준다. 알고리즘 전개를 용이하게 하기 위하여 납기가 빠른 순서로 정렬을 사전에 한 결과 작업 4가 가장 빠른 납기를 가진다.

〈표 1〉 수치 예($m=2, n=9$인 경우)

	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9
r_j	11	56	14	5	22	46	47	37	55
p_j	44	45	50	33	38	45	52	34	40
d_j	71	114	83	55	89	113	125	87	115
τ_j	16	13	19	17	29	22	26	16	20

먼저, 작업 4에 대하여 고려해보자. 그것의 착수시점, 완료시점 및 목적함수는 다음과 같이 계산된다. 기계 1에 대해서, $S_{1[1]} = d_{1[1]} - p_{1[1]} = 55-33 = 22$, $C_{1[1]} = d_{1[1]} = 55$이고 $f_1(s) = 0$. 기계 2에 대해서, $S_{2[1]} = d_{2[1]} - p_{2[1]} = 55-33 = 22$, $C_{2[1]} = d_{2[1]} = 55$이고 $f_2(s) = 0$. 작업 4는 기계 1 또는 기계 2 둘 중 어느 곳에 배정이 되어도 목적함수가 동일하기 때문에 인위적(Arbitrarily)으로 기계 1에 배정을 한다.

두 번째로 고려되는 작업은 작업 1이다. 기계 1

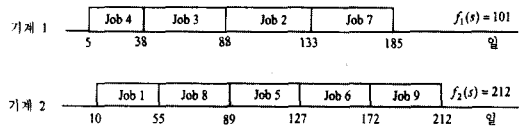
에 대해서, $S_{1[2]} = d_{1[2]} - p_{1[2]} = 71-44 = 27$, $C_{1[2]} = d_{1[2]} = 71$이다. 이런 경우에는 기계 1에 이미 작업 4가 배정 되어있기 때문에 단계 2.2를 수행해야 한다. 그러면 $S_{1[2]} = C_{1[1]} = 55$이고 $C_{1[2]} = S_{1[2]} + p_1 = 55+44 = 99$이다. 기계 1에서 작업 1에 대한 지연생산비용(T_1)은 $C_{1[2]} - d_{1[2]} = 99-71 = 28$이다. 따라서 기계 1의 부분일정에 대한 목적함수 $f_1(s) = 28$. 기계 2에 대해서, $S_{2[1]} = d_{2[1]} - p_{2[1]} = 71-44 = 27$, $C_{2[1]} = d_{2[1]} = 71$이다. 이런 경우 기계 2에는 이미 배정되어 있는 작업이 없기 때문에 조기생산 및 지연생산 어느 쪽도 발생하지 않는다. 따라서, $f_2(s) = 0$. 그래서 작업 1은 기계 2에 배정을 한다. 이와 유사한 방법으로 9개의 작업이 모두 배정될 때까지 반복한다.

제안된 알고리즘의 우월성을 보이기 위해서, 본 논문에서는 지능형 탐색 기법인 유전 알고리즘을 이용하여 비교하였다. 수치예제를 풀기 위한 유전 알고리즘의 파라메터들은 최대 세대수(max_*gen*)는 500이고, 모집단(*pop_size*)의 크기는 100이다. 교차변이율(p_c)은 0.4이고, 돌연변이율(p_m)은 0.2로 하였다.

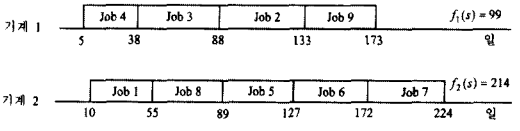
〈표 2〉 수치 예의 조기생산과 지연생산의 현황

		Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	$f(S)$
HA	E_j	16			17						313
	T_j		19	5		38	39	60	2	97	
GA	E_j	16			17						313
	T_j		19	5		38	39	99	2	38	

<표 2>는 수치 예제에 대하여 제안된 알고리즘과 유전 알고리즘을 수행한 결과 각 작업별 조기생산과 지연생산에 따른 비용 및 목적함수 값을 보여준다. 계산결과 목적함수가 313으로 동일하게 계산되었다. 비교결과, 제안된 알고리즘과 유전 알고리즘에 의해서 구한 해 사이에 차이(Gap)가 없기 때문에, 제안된 알고리즘의 효과성이 입증되었다. [그림 1]과 [그림 2]는 각각 제안된 알고리즘과 유전 알고리즘에 의한 작업배치 현황을 보여준다.



[그림 1] 제안된 알고리즘(HA)에 의한 작업 배치현황



[그림 2] 유전 알고리즘(GA)에 의한 작업 배치현황

5. 전산 실험

본 절에서는 제안된 알고리즘의 성능을 평가하기 위하여 H사의 선박용 엔진 조립공장의 상황을 고려하였다. 실제로 선박용 엔진을 조립하기 위하여 수행되는 조립시간(Day)은 엔진타입에 따라 평균적으로 30일에서 45일 정도 소요된다. 그리고 작업의 도착은 일년동안 계속 발생하며, 작업이 조립공장에 도착한 후 완료되기까지 조립시간을 제외하고 평균적으로 허용되는 여유(Allowance)시간은 20일 정도이다. 동일한 작업능력을 가진 조립장비는 12대가 설치 되어있다. 경험적으로 1대의 조립장비에서 처리되는 선박엔진의 수는 연간 7대에서 12대 정도이다. 이러한 현장 상황을 고려하여 실험조건들을 다음과 같이 설정하였다.

(1) 작업들의 조립시간(p_j)은 [30, 45] 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_j는 허용치(Allowance Value)를 의미하는데, 이것의 범위는 [10, 30] 사이의 이산형 일양 분포로부터 랜덤하

게 발생시킨다.

위의 실험조건을 적용하여 제안된 알고리즘이 효과적으로 해를 탐색하는지를 검증하기 위하여 다양한 크기의 문제들에 대하여 전산실험을 수행하였다. 제안된 알고리즘은 IBM 워크스테이션인 RS/6000-C10에서 C언어로 프로그래밍 되었다. 장비의 대수와 작업의 수는 <표 3>과 같은 조합(Combination)으로 구성하고, 각 조합에 대하여 랜덤하게 10문제씩 생성하여 실험하였다.

〈표 3〉 실험 결과 요약

기계의 수(m)	작업의 수(n)	평균비율차이(APG: %)				발견적 해법의 CPU 시간 (Second)
		$p_c = 0.2$, $p_m = 0.4$	$p_c = 0.2$, $p_m = 0.8$	$p_c = 0.4$, $p_m = 0.2$	$p_c = 0.8$, $p_m = 0.2$	
3	20	1.40	1.58	1.17	1.26	0.09
	30	0.72	1.11	0.64	1.25	0.10
	50	1.53	1.37	1.42	1.11	0.09
5	20	0.99	0.89	1.75	1.45	0.08
	30	1.08	1.08	0.33	0.88	0.10
	50	0.67	0.62	0.17	0.62	0.09
7	50	1.05	0.75	0.83	0.26	0.09
	70	-0.17	0.14	0.28	0.31	0.09
	100	-0.22	-0.11	-0.21	-0.95	0.10
10	50	-1.33	-1.32	-1.41	-1.81	0.09
	70	0.42	-0.20	-0.51	-0.47	0.09
	100	-0.68	-0.42	-1.40	-1.45	0.10
15	50	-1.18	-2.59	3.23	-4.49	0.10
	70	-1.78	-2.47	-3.49	-3.93	0.08
	100	-2.02	-2.26	-3.36	-4.36	0.09

제안된 알고리즘을 평가하기 위해서 유전 알고리즘에 의해서 구한 해와 비교검증 하였다. 유전 알고리즘은 최근에 비정규 평가척도를 갖는 일정계획의 근사 최적해를 제공해 주는 유용한 해법으로 알려지고 있다. 본 논문에 사용한 유전 알고리즘은 부록 B에 상세하게 기술되어 있다[9, 10]. 본 논문에서, 유전 알고리즘을 수행하기 위하여 사용한 기본 파라메터 정보는 다음과 같다. 최대 세대수(max_*gen*)는 2000이고, 모집단(*pop_size*)의 크기는 100이다. 교차변이율(p_c)과, 돌연변이율(p_m)은 다양한 파라메터 집합을 적용하기 위해 4가지 조합으로 실험을 하였다. 본 논문에서는 제안된 알고리즘과 유전 알고리즘을 비교하는 척도로 평균비율차이(Average Percent Gap: APG)를 사용한다. 여기서, H_A는 제안된 알고리즘에 의한 목적함수 값이고, G_A는 유전 알고리즘에 의한 목적함수 값이라고 정의하자. 그러면 평균비율차이 즉, $APG = \{(H_A - G_A)/G_A\} \times 100$으로 정의된다. 이것은 유전 알고리즘에 의하여 얻어진 근사 최적해를 기준으로 제안된 알고리즘에 의해서 구한 해와의 편차를 추정하는 것이다[13].

<표 3>은 실험결과를 요약 정리한 것이다. 여기서 제안된 알고리즘의 해가 평균적으로 유전 알고리즘의 1.75% 범위 이내에 존재한다는 것을 알 수 있다.

따라서 제안된 알고리즘과 유전 알고리즘의 다양한 파라메터 집합들에 대하여 실험을 수행 해 본 결과, 이항계산시간(Polynomial Computation Time) 내에서 제안된 알고리즘이 좋은 해를 제공하여 준다는 것을 알 수 있었다. 제안된 알고리즘의 계산량은 $O(n \log n + mn + n)$이고, CPU 시간은 평균적으로 0.10초가 소요되었다. <표 3>에서 알 수 있듯이, 문제의 크기 즉, 작업의 수와 기계 수가 증가함에 따라 제안된 알고리즘이 효과적이라는 것을 알 수 있다. APG 열에서 '-' 값은 주어진 최대 세대수(max_*gen* = 2000)의 범위 내에서 제안된 알고리즘의 해가 유전 알고리즘에 의해서 구한 해보다 좋다는 것을 의미한다.

6. 결 론

본 논문에서는 작업능력이 동일한 병렬기계에서 상이한 납기와 도착시간을 가지는 작업들의 일정계획 문제를 다루었다. 이 문제를 효과적으로 해결하기 위하여, 근사 최적해를 찾을 수 있는 발견적 해법을 개발하였다. 논문 고찰을 통하여 확인한 바로는 상이한 납기와 도착시간을 동시에 고려한 병렬기계 일정계획 문제의 해법을 제시한 것은 본 논문이 처음이라고 사료된다. 제시한 발견적 해법의

효과성을 입증하기 위하여, 다양한 크기의 문제에 대하여 유전 알고리즘의 해와 비교하였다. 실험 결과, 우리의 발견적 해법이 상이한 납기와 도착시간을 갖는 병렬기계의 일정계획 문제에 대하여 근사최적해를 제공하여 준다는 것을 보여주었다.

향후의 연구과제로는 비정규 평가척도를 갖는 2기준(Bi-criteria) 병렬기계 일정계획 문제로 확장하는 것과 기계그룹(Machine Group) 개념을 고려한 문제로 확장해 볼 가치가 있을 것이다. 현재 이러한 문제들에 대하여 연구가 진행 중에 있다.

부록 A

(증명) 여기에서 사용되는 기호를 정의하면 다음과 같다.

$a_{i[k]}$: 기계 i에서 작업 j를 k번째 위치에 배정하기 위하여 작업 j의 완료시점을 d_j에 일치시킬 때, k번째 위치의 작업과 $k-1$번째 위치에 있는 작업과의 중복(Overlap)되는 구간 즉, $(C_{i[k-1]} - S_{i[k]})$이다.

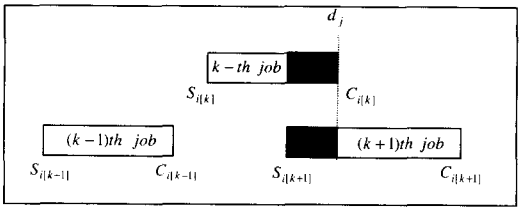
$b_{i[k]}$: 기계 i에서 작업 j를 k번째 위치에 배정하기 위하여 작업 j의 완료시점을 d_j에 일치시킬 때, k번째 위치의 작업과 $k+1$번째 위치에 있는 작업과의 중복(Overlap)되는 구간 즉, $(d_{i[k]} - S_{i[k+1]})$이다.

$f_i^p(s)$: 기계 i에서 작업 j가 배정되기 직전까지의 부분일정에 대한 목적함수

δ_E, δ_T : 초기에 작업 j의 완료시점을 d_j에 일치시킬 때, 중복되는 $b_{i[k]}$, $a_{i[k]}$보다는 작거나 같은 양수

다음은 본 논문에서 제시한 알고리즘에 내포되어 있는 속성들 중에 하나인 $d_{i[k-1]} = C_{i[k-1]}$이고 $d_{i[k+1]} = C_{i[k+1]}$인 경우일 때, 성질 I이 성립함을 보여준다.

Case 1 : $p_{i[k]} \le S_{i[k+1]} - C_{i[k-1]}$이고 $a_{i[k]} \le 0$인 경우



[그림 3] $a_{i[k]} \le 0$이 발생하는 경우

(Case 1.1) $C_{i[k]}$를 d_j에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + b_{i[k]}.$$

(Case 1.2) $C_{i[k]}$를 $S_{i[k+1]}$에 일치시키는 경우:

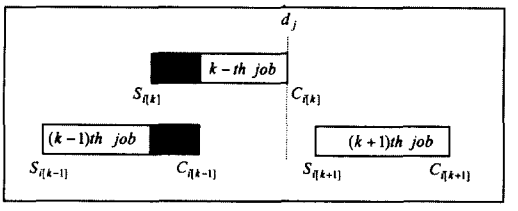
$$f_i(s) = f_i^p(s) + b_{i[k]}.$$

(Case 1.3) $S_{i[k]}$를 $C_{i[k-1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{b_{i[k]} + \delta_E\}.$$

따라서 Case 1.1이 최소값을 가진다.

Case 2 : $p_{i[k]} \le S_{i[k+1]} - C_{i[k-1]}$이고 $b_{i[k]} \le 0$인 경우



[그림 4] $b_{i[k]} \le 0$이 발생하는 경우

(Case 2.1) $C_{i[k]}$를 d_j에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + a_{i[k]}.$$

(Case 2.2) $C_{i[k]}$를 $S_{i[k+1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{a_{i[k]} + \delta_T\}.$$

(Case 2.3) $S_{i[k]}$를 $C_{i[k-1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + a_{i[k]}.$$

따라서 Case 2.1이 최소값을 가진다.

Case 3 : $p_{i[k]} > S_{i[k+1]} - C_{i[k-1]}$이고 $a_{i[k]} \leq 0$인 경우

(Case 3.1) $C_{i[k]}$를 d_j에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + b_{i[k]}.$$

(Case 3.2) $C_{i[k]}$를 $S_{i[k+1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{|a_{i[k]}| + b_{i[k]}\}.$$

(Case 3.3) $S_{i[k]}$를 $C_{i[k-1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{b_{i[k]} + \delta_E\}.$$

단, Case 3.2에서의 $a_{i[k]}$의 값은 [그림 3]에서 알 수 있듯이 초기에는 음수 값을 가지는데 작업의 완료시점을 찾기 위하여 이동을 함에 따라 이것은 양수 값을 가지기 때문에 작업 j의 완료시점을 d_j에 일치시키는 Case 3.1이 최소값을 가진다.

Case 4 : $p_{i[k]} > S_{i[k+1]} - C_{i[k-1]}$이고 $b_{i[k]} \leq 0$인 경우

(Case 4.1) $C_{i[k]}$를 d_j에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + a_{i[k]}.$$

(Case 4.2) $C_{i[k]}$를 $S_{i[k+1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{a_{i[k]} + \delta_T\}.$$

(Case 4.3) $S_{i[k]}$를 $C_{i[k-1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{a_{i[k]} + |b_{i[k]}|\}.$$

단, Case 4.3에서의 $b_{i[k]}$의 값은 [그림 4]에서 알 수 있듯이 초기에는 음수 값을 가지는데, 작업 j의 완료시점을 찾기 위하여 이동을 함에 따라 이것은 양수 값을 가지기 때문에 작업 j의 완료시점을 d_j에 일치시키는 Case 4.1이 최소값을 가진다.

Case 5 : $p_{i[k]} > S_{i[k+1]} - C_{i[k-1]}$이고 $a_{i[k]}$, $b_{i[k]} > 0$인 경우

(Case 5.1) $C_{i[k]}$를 d_j에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{a_{i[k]} + b_{i[k]}\}.$$

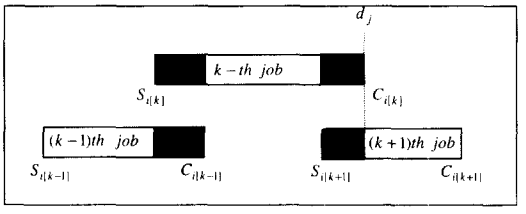
(Case 5.2) $C_{i[k]}$를 $S_{i[k+1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{a_{i[k]} + 2b_{i[k]}\}.$$

(Case 5.3) $S_{i[k]}$를 $C_{i[k-1]}$에 일치시키는 경우:

$$f_i(s) = f_i^p(s) + \{2a_{i[k]} + b_{i[k]}\}.$$

따라서 Case 5.1이 최소값을 가진다.



[그림 5] $a_{i[k]}$, $b_{i[k]} > 0$이 발생하는 경우

상기의 Case 1에서 Case 5까지의 결과를 종합하면 전체적으로 성질 I을 만족한다.

부록 B

본 논문에서 사용한 유전 알고리즘의 전반적인 과정을 기술하면 다음과 같다.

단계 1 : 변수 초기화(Set Parameters)

n, m, r_j, p_j 및 d_j를 랜덤하게 생성한다. 그리고 max_gen, pop_size, p_c 및 p_m값을 결정한다.

단계 2 : 해의 초기화(Initial Population Generation)

특정한 크기의 작업리스트(Chromosome: c_l)를 모집단 크기 만큼 랜덤하게 생성하고, 각 작업 리스트들은 순열표현(Permutation Representation)으로 구성된다. 그리고 작업리스트를 구성하고 있는 각 작업(Gene)들을 랜덤하게 기계에 배정한다.

단계 3 : 평가(Evaluation)

각 작업 리스트들(c_l)에 대한 목적함수($f(c_l)$)를 계산한다. 각 작업 리스트들에 대하여 적합도(Fitness Value: $fit(c_l)$)를 계산한다.

$$fit(c_t) = \{1/f(c_t)\}$$

단계 4 : 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.

$$\eta_k = \{fit(c_t)/\sum_{t=1}^{pop\_size} fit(c_t)\} \times pop\_size$$

단계 5 : 재생(Reproduction Operation)

*remainder stochastic sampling without replacement*를 이용하여 다음 세대로 보다 우수한 작업 리스트들을 상속시키기 위하여 현 세대에서 *pop_size*만큼 재 생성한다

단계 6 : 재구성(Recombination)

두 부모(Parent)의 염색체로부터 두 자식(Offspring)의 염색체를 생성한다.

① **Crossover Operation**: 교차변이 비율 만큼 연산을 수행한다. 랜덤하게 두 개의 작업리스트를 선택한다. 그리고 수정(Modified)된 Order Crossover(OX) 연산자를 이용하여 교차변이를 수행한다.

② **Mutation Operation**: 랜덤하게 선택된 두 개의 작업이 서로 다른 기계에 배정되어 있을 때 돌연변이 연산을 수행한다. 그리고 돌연변이 비율 만큼 연산을 수행하는데 Reciprocal Exchange Mutation(REM) 연산자를 이용하여 돌연변이를 수행한다.

③ **Evaluation Operation**: 각 작업 리스트들에 대하여 평가를 한다.

단계 7 : 대체(Replacement)

이전 세대에서 가장 좋은 해를 발생시킨 작업리스트는 반드시 선택하여 다음 세대로 상속시키는 엘리티즘(elitism) 전략을 이용한다.

단계 8 : 종료(Termination)

최대 세대 수(max_gen) 및 시간을 만족하면 중지하고, 그밖에는 단계 2부터 반복 수행한다.

참 고 문 헌

[1] 강용혁, 이홍철, 김성식, "서로 다른 납기를 갖는 작업에 대한 이종 병렬기계에서의 일정계획 수립", 「대한산업공학회지」, 제24권, 제1호, pp.37-50, 1998.

[2] Abdul-Razaq, T.S. and C.N. Potts, "Dynamic Programming State-Space Relaxation for Single Machine Scheduling", *J. Opnl. Res. Soc.*, Vol.39, No.2(1988), pp.141-152.

[3] Baker, K.R. and G.D. Scudder, "Sequencing with Earliness and Tardiness Penalties: A Review", *Operations Research*, Vol.38(1990), pp.22-36.

[4] Cheng, T.C. and Z.L. Chen, "Parallel-Machine Scheduling Problems with Earliness and Tardiness Penalties", *Journal of the Operational Research Society*, Vol.45(1994), pp.685-695.

[5] Cheng, R., M. Gen and T.Tozawa, "Minmax Earliness/Tardiness Scheduling in Identical Parallel Machine System Using Genetic Algorithms", *Computers and Industrial Engineering*, Vol.29(1995), pp.513-517.

[6] Cheng, T.C. and C.C. Sin, "A State-of-the-Art Review of Parallel-Machine Scheduling Research", *European Journal of Operational research*, Vol.47(1990), pp.271-292.

[7] Fry, T.D., R.D. Armstrong and L.D. Rosen, "Single Machine Scheduling to Minimize Mean Absolute Lateness: A Heuristic Solution", *Computers and Operations Research*, Vol.17(1990), pp.105-112.

[8] 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.

[9] Gen M. and R. Cheng, *Genetic Algorithms and Engineering Design*, A Wiley-Interscience Publication, 1997.

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

[11] 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.

[12] Lee, D., J. Kim and J. Hur, "Decision Support System for Dynamic Production Schedule in The Ship Engine Manufacturing System", *Proceedings of 20th International conference on Computers and Industrial Engineering*, Vol.2(1996), pp.1115-1118.

[13] Lee, H. and G. Jang, "File and Workload Allocation on A Local Multi-Access Computer Network: Incorporating Local Processing and Communication Overhead", *International Journal of Systems Science*, Vol.27 (1996), pp.831-837.

[14] Ow, P.S. and T.E. Morton, "The Single Machine Early-Tardy Problem", *Mgmt. Sci.*, Vol.35(1989), pp.177-191.