

작업투입시점과 순서의존적인 준비시간이 존재하는 병렬기계 일정계획을 위한 개선 개미군집 시스템

주철민[†]

동서대학교 시스템경영공학과

An Improved Ant Colony System for Parallel-Machine Scheduling Problem with Job Release Times and Sequence-Dependent Setup Times

Cheol-Min Joo

Dept. of System and Management Engineering, Dongseo University

This paper considers a parallel-machine scheduling problem with job release times and sequence-dependent setup times. The objective of this problem is to determine the allocation policy of jobs and the scheduling policy of machines so as to minimize the weighted sum of setup times, delay times, and tardy times. A mathematical model for optimal solution is derived and a meta heuristic algorithm based on the improved ant colony system is proposed in this paper. The performance of the meta heuristic algorithm is evaluated through compare with optimal solutions using randomly generated several examples.

Keywords: Parallel-Machine Scheduling, Ant Colony System

1. 서론

본 연구에서는 다수의 작업(job)을 다수의 기계에 할당하는 병렬기계 일정계획(Parallel-Machine Scheduling) 문제를 다룬다. 각 기계는 한 번에 하나의 작업만 수행할 수 있으며, 각 작업은 하나의 기계에서만 수행된다. 각 작업 i 는 투입가능시점(job release time) r_i 와 완료요구시점(due time) c_i 를 가지며, 작업 준비 및 작업 시작시간 x_i 가 투입가능시점 이후일 경우 발생하는 지연시간(이하 delay 시간이라 한다)과 작업 완료시간이 완료요구시점 이후일 경우 발생하는 지연시간(이하 tardy 시간이라 한다)의 2가지 벌과 시간(penalty time)을 유발한다. 각 작업 i 의 가공시간 p_i 는 작업들이 가공되는 순서에 독립적으로 주어지지만, 작업준비시간 s_{hi} 는 이전 작업 h 에 따라 달라진다($s_{hi} \neq s_{ih}$). 즉 작업준비시간은 그 작업들이 가공되는 순서에 따라 달리 주어진다. 예를 들어 <Figure 1>에서 보는바와

같이 작업 h , i 와 j 가 같은 기계 k 에 할당되고 $h \rightarrow i \rightarrow j$ 순서로 작업이 수행된다면, 작업 i 를 위한 작업준비시간은 선행 작업 h 에 따라 s_{hi} 이고 작업 j 를 위한 작업준비시간은 선행 작업 i 에 따라 s_{ij} 가 된다. 작업 i 의 일정은 작업준비 및 작업 시작시간 x_i 가 투입가능시점 r_i 보다 늦어 $(x_i - r_i)$ 만큼의 delay 시간을 유발하고, 작업 i 의 완료시간도 완료요구시점 c_i 보다 늦으므로 $((x_i + s_{hi} + p_i) - c_i)$ 만큼의 tardy 시간을 유발한다. 작업 j 의 투입가능시점 r_j 이전에 작업 j 를 시작할 수 없으므로 작업 j 의 일정은 delay 시간을 유발하지 않으나, 기계 k 에 유휴시간이 발생하게 된다.

순서의존적인 준비시간(sequence-dependent setup time)은 유리제조공정, 종이제조공정, 반도체공정, 화학공정, 도장공정 및 사출공정 등에서 발생하는데 그 작업일정에 따라 설비의 효율과 납기 준수율 등에 많은 영향을 미친다. 작업 투입가능시점이 주어져 있는 일정계획 문제(Garay and Johnson,

[†] 연락저자 : 주철민, 617-716 부산광역시 사상구 주례동 산 69-1 동서대학교 시스템경영공학과, Tel : 051-320-1715, Fax : 051-320-1750,
Email : cmjoo@dongseo.ac.kr

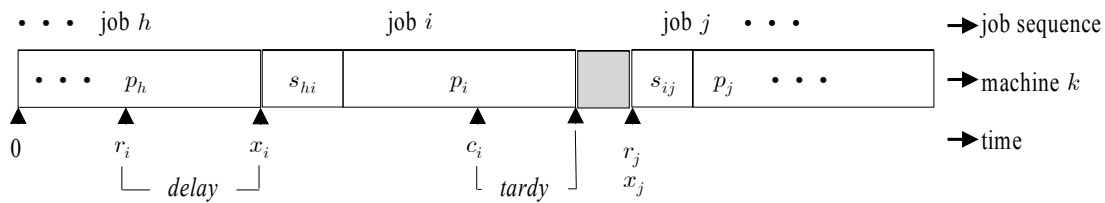


Figure 1. Example of job sequence in parallel-machine scheduling

1979)와 순서의존적인 작업시간을 갖는 일정계획 문제(Pinedo, 1995)는 각기 단일기계에서도 NP-hard 문제로 알려져 있다. 병렬기계의 경우 이러한 제약이 없는 경우에도(Garay and Johnson, 1997) NP-hard 문제로 알려져 있다. 따라서, 본 연구의 작업투입시점과 순서의존적인 준비시간이 존재하는 병렬기계 일정계획문제는 매우 NP-hard한 문제이다. Frederickson *et al.* (1978)은 순서의존적인 준비시간을 갖는 병렬기계 일정계획을 위한 휴리스틱 알고리즘을 처음으로 제시하였다. 이후 Flynn (1987), Hahn *et al.* (1989), Bobrowski and Kim(1994, 1997) 등이 이 문제를 다루었다. Mendes *et al.* (2002)은 순서의존적인 준비시간을 갖는 병렬기계 일정계획문제를 해결하기 위해 TS (Tabu Search) 방법을 적용하였고, Behnamian *et al.* (2009)은 ACO(Ant Colony Optimization), SA(Simulated Annealing), 그리고 VNS(Variable Neighborhood Search)를 결합한 하이브리드 알고리즘을 제시하였다. 순서의존적인 준비시간과 작업투입시점이 존재하는 병렬기계 일정계획문제는 Nessah *et al.* (2005)에 의해 다루어 졌다. 그들은 총 가중 작업완료시간을 최소화 하는 해를 구하기 위한 휴리스틱 알고리즘을 제시하였다. Tahar *et al.* (2006)은 이 문제에 대해 최대 완료시간을 최소화하기 위한 휴리스틱 알고리즘을 제시하였고, Gharehgozli *et al.* (2009)은 가중 flow time의 최소화 와 가중 tardiness의 최소화 두가지 목적을 갖는 MIGP(Mixed Integer Goal Programming) 모델과 해법을 제시하였다.

본 연구에서는 작업투입가능시점과 완료요구시점을 동시에 가지며 순서의존적인 준비시간이 존재하는 병렬기계 일정계획문제를 다룬다. 주어진 일정계획문제는 두 가지 벌과 시간(delay 시간과 tardy 시간)과 작업준비시간의 가중합을 최소로 하는 작업별 할당 기계 및 기계별 작업순서를 동시에 결정하는 것이 목적이다. 이를 위해 본 연구에서는 최적해를 도출하기 위한 수리모형을 제시하고, 보다 효율적으로 해를 도출하기 위해 개미군집시스템(Ant Colony System; ACS)에 기초한 메타휴리스틱 알고리즘을 제시한다. ACS 알고리즘은 실제 개미들이 먹이에서 집까지 가장 짧은 경로를 찾는 능력을 모방한 메타 휴리스틱 기법으로 조합 최적화 문제를 해결하기 위해 Dorigo *et al.* (1996)에 의해 소개되었고 Dorigo and Caro (1999)에 의해 정립되었다. 이후 Traveling Salesman Problem (Dorigo and Gambardella, 1997; Cheng and Mao, 2007), Assignment Problem(Solimanpur *et al.*, 2004), Scheduling Problem (Ying and Liao, 2004; Shyu *et al.*, 2004), Vehicle Routing

Problem(Bell and McMullen, 2004)등의 문제로 확대 적용되고 있다. ACS 알고리즘은 까다로운 제약조건을 반영하기 쉽고, 우수한 해들의 특성을 종합적으로 반영하기 때문에 NP-hard 문제의 다양한 해 공간 탐색에 있어서 큰 장점을 가지고 있다 (Dorigo *et al.*, 2005, 2006). 그러나 ACS 알고리즘을 적용하기 위해서는 기본적으로 주어진 문제를 네트워크 구조로 표현하여야 하는 제약과, 탐색을 진행함에 따라 국부 최적해(Local Optimum)로의 수렴 현상이 발생하는 단점이 있다. 이에 본 연구에서는 ACS 알고리즘을 적용할 수 있도록 병렬기계 일정계획문제의 네트워크 구조를 제시한다. 또한, 국부 최적해로의 수렴 현상을 보완할수 있는 개선 개미군집시스템(Improved Ant Colony System; IACS)을 제시하고 여러 실험을 통해 그 효과를 검증한다.

2. 최적화 모형

본 절에서는 작업투입가능시점과 완료요구시점을 동시에 가지며 순서의존적인 준비시간이 존재하는 병렬기계 일정계획 문제 대해 delay 시간, tardy 시간과 작업준비시간의 가중합을 최소로 하는 작업별 할당 기계 및 기계별 작업순서를 동시에 결정하기 위한 최적화 모델을 제시한다. 제시된 최적화 모델은 Gharehgozli *et al.* (2009)을 비롯한 여러 기존 연구에서처럼 병렬기계 일정계획문제에 대한 기본 모델을 본 연구의 목적과 상황에 맞게 변형 하였다.

<파라미터>

- r_i : 작업 i 의 투입가능시점(job release time)
- c_i : 작업 i 의 완료요구시점(due time)
- p_i : 작업 i 의 작업시간(processing time)
- s_{ij} : 작업 i 이후 작업 j 를 작업하기 위한 준비시간 (setup time)
- s_{Oi} : 작업 i 가 최초 작업일때 준비시간
- MS : 전체 기계 집합
- JS : 전체 작업 집합
- O : 각 기계별 최초작업을 위한 가상의 작업
- E : 각 기계별 최후작업을 위한 가상의 작업
- M : 양의 큰 수 (무한대)
- w_D, w_T, w_S : 가중치

<의사결정 변수>

x_i : 작업 i 의 작업준비 및 작업 시작시간

$y_{ik} = \begin{cases} 1 & \text{기계 } k \text{에 의해 작업 } i \text{가 작업되어지면} \\ 0 & \text{otherwise} \end{cases}$

$y_{ijk} = \begin{cases} 1 & \text{기계 } k \text{에 의해 작업 } i \text{이후 작업 } j \text{가} \\ & \text{작업되어지면} \\ 0 & \text{otherwise} \end{cases}$

d_i^+ : 작업 i 가 완료요구시점 이전에 작업완료 되어지는 경우 그 차이 시간

d_i^- : job i 가 완료요구시점 이후에 작업완료 되어지는 경우 그 차이 시간

$y_{Okk} = \begin{cases} 1 & \text{기계 } k \text{에 의해 작업 } i \text{가 첫번째로} \\ & \text{작업되어지면} \\ 0 & \text{otherwise} \end{cases}$

$y_{iEk} = \begin{cases} 1 & \text{기계 } k \text{에 의해 작업 } i \text{가 마지막으로} \\ & \text{작업되어지면} \\ 0 & \text{otherwise} \end{cases}$

$y_{OE_k} = \begin{cases} 1 & \text{기계 } k \text{에 의해 작업되어지는 작업이} \\ & \text{없으면} \\ 0 & \text{otherwise} \end{cases}$

주어진 기호를 사용하여 두 가지 별과 시간과 작업준비시간의 가장합을 최소화하는 작업별 할당 기계 및 기계별 작업순서 결정을 위한 최적화 모형은 다음과 같다.

Minimize

$$z = w_D \cdot \sum_{i \in JS} (x_i - r_i) + w_T \cdot \sum_{i \in JS} d_i^- + w_S \cdot \sum_{i \in JS} \sum_{j \in JS, j \neq i} \sum_{k \in MS} s_{ij} \cdot y_{ijk}$$

subject to

$$x_i \geq r_i \quad \text{for } \forall i \in JS. \quad (1)$$

$$x_i + \left(\sum_{k \in MS} s_{Ok} \cdot y_{Okk} + \sum_{h \in JS, h \neq i} \sum_{k \in MS} s_{hi} \cdot y_{hik} \right)$$

$$+ p_i + d_i^+ - d_i^- = c_i \quad \text{for } \forall i \in JS. \quad (2)$$

$$x_i + \left(\sum_{k \in MS} s_{Ok} \cdot y_{Okk} + \sum_{h \in JS, h \neq i} \sum_{k \in MS} s_{hi} \cdot y_{hik} \right) + p_i$$

$$\leq x_j + M \cdot \left(1 - \sum_{k \in MS} y_{ijk} \right) \quad \text{for } \forall i, j \in JS; j \neq i \quad (3)$$

$$\sum_{k \in MS} y_{ik} = 1 \quad \text{for } \forall i \in JS \quad (4)$$

$$\sum_{j \in JS, j \neq i} y_{jik} + y_{Okk} = y_{ik} \quad \text{for } \forall i \in JS; \forall k \in MS \quad (5)$$

$$\sum_{j \in JS, j \neq i} y_{ijk} + y_{iEk} = y_{ik} \quad \text{for } \forall i \in JS; \forall k \in MS \quad (6)$$

$$\sum_{j \in JS} y_{Ojk} + y_{OE_k} = 1 \quad \text{for } \forall k \in MS \quad (7)$$

$$\sum_{j \in JS} y_{jEk} + y_{OE_k} = 1 \quad \text{for } \forall k \in MS \quad (8)$$

$$x_i, d_i^+, d_i^- \geq 0 \quad \text{for } \forall i \in JS \quad (9)$$

$$y_{ik} = 0 \text{ 또는 } 1 \quad \text{for } \forall i \in JS; \forall k \in MS \quad (10)$$

$$y_{ijk} = 0 \text{ 또는 } 1 \quad \text{for } \forall i, j \in JS; \forall k \in MS \quad (11)$$

$$y_{OE_k} = 0 \text{ 또는 } 1 \quad \text{for } \forall k \in MS \quad (12)$$

$$y_{OEk}, y_{iEk} = 0 \text{ 또는 } 1 \quad \text{for } \forall i \in JS; \forall k \in MS \quad (13)$$

제약식(1)은 모든 작업 i 는 투입가능시점 이후에 작업준비 및 작업을 시작할 수 있음을 나타내며, (2)는 tardy 시간을 산출하기 위한 식이다. 제약식(3)은 동일 기계에 할당되는 선후 작업들의 작업시작 시간을 제약하며, (4)는 각 작업은 반드시 하나의 기계에만 할당되어야 됨을 나타낸다. 제약식(5)~(8)은 동일한 기계에 할당된 모든 작업들은 반드시 그리고 한번만 그 기계의 작업순서상에 나타나야 됨을 제약한다. 실제로 이 제약식들은 단순 할당 제약으로 sub-tour 없는 작업순서 생성을 장담할 수 없으나, 제약식(3)과 결합함으로써 이런 문제를 해결할 수 있다.

3. 개선 개미군집 시스템

실제 개미들은 자신이 이동한 경로 위에 페로몬(Pheromone)이라는 화학 물질을 분비하고, 또한 앞선 개미로부터 분비된 페로몬 흔적(Trail)에 의해 다음 이동경로를 선택함으로써 개미집으로부터 먹이까지 최단경로를 선택하여 이동하는 행동습성을 갖고 있다. 개미군집시스템(ACS)은 이러한 실제 개미의 행동습성을 모방하는 인공개미(Artificial Ant)라는 에이전트(Agent)를 구성하여 해를 탐색한다. 각 에이전트들은 탐색한 전체 경로의 길이(해)들로 만들어진 페로몬을 지역 업데이트 규칙(Local Updating Rule)에 따라 방문한 각 링크에 분비(변경)하게 된다. 모든 에이전트들이 탐색과정을 마치게 되면 가장 좋은 해를 탐색한 에이전트의 해로 전역 업데이트 규칙(Global Updating Rule)에 따라 페로몬 양을 변경하게 된다. 이들 페로몬은 에이전트들이 경로를 탐색하는 과정에 상태전이 규칙(State Transition Rule)의 기본 정보로 활용된다. ACS에서는 경로를 탐색하는 과정에서 페로몬뿐만 아니라 가시성의 척도인 휴리스틱 정보도 활용함으로써 최적해 탐색 효율을 높인다. 일반적인 ACS에서는 상태전이규칙으로 페로몬과 휴리스틱 정보에 대한 최대값 선택과 확률적 선택 두 가지를 사용한다. 본 연구에서는 이들 두가지 규칙에 랜덤 선택 규칙을 추가한 개선 개미군집시스템(IACS)을 제시한다. IACS에서 추가된 랜덤 선택 규칙은 유전알고리즘(Genetic Algorithm)의 돌연변이(Mutation)와 같은 기능을 수행함으로써 개미군집시스템의 국부 최적해(Local Optimum)로의 수렴 현상을 보완한다. 본 연구에서는 병렬기계 일정계획 문제에 ACS를 적용하기 위해 계획대상 작업들(최초작업을 위한 가상의 작업 포함)을 노드로 간주하고, 모든 노드쌍간의 링크가 작업 선후관계를

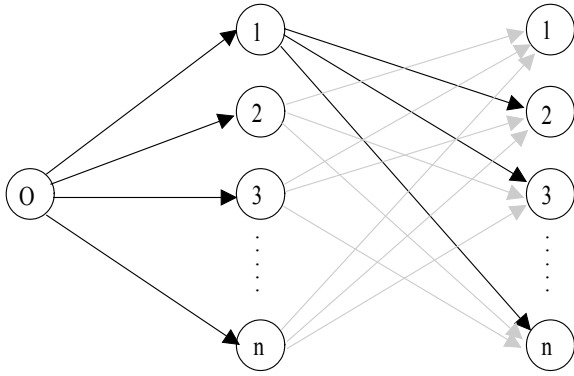


Figure 2. Link structure of parallel-machine scheduling problem for ACS

나타내도록 한다(<Figure 2> 참조). 페르몬은 <Figure 2>의 링크상에만 존재하며, 각 에이전트의 탐색과정에서는 할당되는 기계에 관계없이 동일하게 활용된다.

3.1 상태전이규칙(State Transition Rule)

각 에이전트는 임의의 기계의 최초운반을 위한 가상의 작업을 나타내는 노드로부터 출발하여 주어진 상태전이규칙에 따라 다음 노드(대상 작업)를 차례로 선택하여 모든 노드가 선택 되어질 때까지 탐색한다. 임의의 위치에서 다음 노드 선택을 위한 대상 링크 집합 CL 은 각 기계에 마지막으로 할당된(없으면 최초작업을 위한 가상의 작업) 노드들로부터 기 선택된 노드를 제외한 모든 노드들로의 링크들이 포함된다. 예를 들어 8 개의 작업을 2대의 기계로 작업하는 상황에서 첫 번째 기계에는 작업 1, 작업 2, 그리고 작업 3이 순차적으로, 두 번째 기계에는 작업 4와 작업 5가 순차적으로 기 할당되었다면, 다음 선택을 위한 대상 링크 집합 CL 에는 첫 번째 기계의 (3, 6), (3, 7), (3, 8)과 두 번째 기계의 (5, 6), (5, 7), (5, 8)이 포함된다. 대상 링크집합중 에이전트의 선택은 현 시점 t 에 각 링크에 분비되어 있는 페로몬의 정보 $\tau_{ij}(t)$ 와 휴리스틱 정보 η_{ij} 를 활용한다. ACS에서 휴리스틱 정보란 에이전트가 축적된 정보인 페로몬외에 다음 노드 선택을 위해 활용할 수 있는 즉시 정보를 의미하며 최단거리를 구하는 네트워크 문제에서는 해당 링크 거리의 역수를 사용한다. 일정계획 문제에서는 좋은 해를 구하기 위한 기준이 되는 즉시 정보로 여러 가지를 고려할 수 있으나 일반적으로 완료요구시점만 주어진 경우 완료요구시점 기준의 해가 좋은 결과를 주지만 작업투입가능시점이 존재하면 작업투입가능시점 기준의 해가 좋은 결과를 주는 사실에 기초하여, 본 연구에서는 휴리스틱 정보로 선택대상 작업의 가능 작업시작 시점의 역수를 이용한다. 가능 작업시작 시점이란 선택 대상작업을 실제로 시작할 수 있는 시점으로 대상작업의 작업투입가능시점과 선행작업의 완료시점을 비교하여 결정된다. 이에 따른 k 기계의 (i, j) 링크의 휴리스틱 정보 η_{ij} 는 식 (14)로 표현된다.

$$\eta_{ij} = \frac{1}{\max\{r_j, \varepsilon_i\}}, \quad \varepsilon_i = x_i + p_i + \left(\sum_{k \in SM} s_{\alpha k} \cdot y_{\alpha k} + \sum_{\substack{m \in SJ \\ m \neq i}} s_{mi} \cdot y_{mik} \right) \quad (14)$$

본 연구에서 제시하는 IACS는 상태전이규칙으로 페로몬과 휴리스틱 정보에 대한 최대값 선택과 확률적 선택, 그리고 랜덤 선택을 사용한다. 최대값 선택 확률 q_M 과 랜덤 선택 확률 q_R 이 주어지고($q_M + q_R \leq 1$) 페로몬과 휴리스틱 정보의 상대적인 중요도를 결정하는 파라미터 α , β 의 값이 주어졌다면, 에이전트는 0과 1사이의 난수(q)에 따라 서로 다른 선택기준을 적용하게 된다. 만약 $q \leq q_M$ 이면 식 (15)에 따라 최대 기준 값을 갖는 링크를 선택한다(최대값 선택).

$$\arg \max_{(i,j) \in CL} \{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta\} \quad (15)$$

만약 $q_M < q < (1 - q_R)$ 이면 식 (16)에 따라 확률값을 계산하여 룰렛휠(roulette wheel) 방식으로 링크를 선택한다(확률적 선택).

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{(q,r) \in CL} [\tau_{qr}(t)]^\alpha \cdot [\eta_{qr}]^\beta} \quad (16)$$

만약 $q \geq (1 - q_R)$ 이면 식 (17)에 따라 동일한 확률값을 계산하여 룰렛휠(roulette wheel) 방식으로 링크를 선택한다(랜덤 선택).

$$Q_{ij} = \frac{1}{n(CL)} \quad (17)$$

3.2 지역 업데이트 규칙(Local Updating Rule)

각 에이전트들은 탐색이 끝나고 난후 해값(z)으로 만들어진 페로몬을 방문한 각 링크에 분비(변경)하고, 실제 페로몬이 화학물질로써, 시간이 지남에 따라 증발하게 되는 것처럼 페로몬 증발계수(Pheromone Evaporation Parameter) ρ ($0 < \rho < 1$)를 사용하여 기존 페로몬은 감소시킨다. 지역 페로몬 업데이트는 식 (18)에 따른다.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta_{ij}^L, \quad (18)$$

$$\Delta_{ij}^L = \begin{cases} \frac{1}{z} & \text{링크 } (i, j) \text{를 에이전트가 방문} \\ 0 & \text{otherwise} \end{cases}$$

3.3 전역 업데이트 규칙(Global Updating Rule)

모든 에이전트들이 탐색과정을 마치게 되면 가장 좋은 해를

탐색한 에이전트의 해값(z^G)으로 식(19)에 따라 페로몬 증발 계수 $\omega(0 < \omega < 1)$ 를 사용하여 페로몬 양을 변경한다.

$$\tau_{ij}(t+1) = (1-\omega)\tau_{ij}(t) + \Delta_{ij}^G, \quad (19)$$

$$\Delta_{ij}^G = \begin{cases} \frac{1}{z^G} & \text{링크 } (i, j) \text{를 해당 에이전트가 방문} \\ 0 & \text{otherwise} \end{cases}$$

4. 알고리즘 검증

본 연구에서 제시한 IACS를 검증하기 위하여 여러 가지 실험 데이터를 생성하여 ILOG CPLEX에 의한 최적해와 비교하였다.

4.1 실험 데이터 생성

실험데이터는 하루(480분)를 계획기간으로 하여 각 작업의 투입가능시점 r 는 $[0, 360]$ 구간에서, 완료요구시점 c 는 $[r+120, 480]$ 구간에서 랜덤하게 생성 하였다. 또한, 각 작업의 작업시간 p_i 는 $[20, 40]$ 구간에서, 순서의존적인 준비시간은 s_{ij} 는 $[10, 20]$ 구간에서 (각 기계에서 최초 작업 일때는 $[5, 10]$ 구간에서) 랜덤하게 생성 하였다. 목적함수의 가중치 w_D, w_T, w_S 는 delay 시간, tardy 시간과 작업준비시간의 상대적인 중요도에 따라 상황에 맞게 설정된 값을 사용하지만 실험에서는 모두 1로 가정하였다.

4.2 파라미터 선택

제시된 IACS는 파라미터들(최대값 선택 확률 q_M , 랜덤 선택 확률 q_R , 페로몬과 휴리스틱 정보의 상대적인 중요도를 결정하는 α 와 β , 그리고 지역 페로몬 증발계수와 전역 페로몬 증발계수인 ρ 와 ω)의 값에 따라 그 성능에 많은 차이가 존재한다. 실험에 사용될 이들 파라미터의 값을 결정하기 위하여 6 가지 파라미터의 모든 조합을 조사하여 최적의 조합을 찾는것은 거의 불가능한 일이므로 인위적으로 선택된 조합들로 일련의 사전실험을 통해 파라미터 값($q_M=0.3, q_R=0.05, \alpha=0.6, \beta=0.4, \rho=0.05, \omega=0.1$)을 일차로 선정하였다. 최종 파라미터 선택을 위해 6가지 파라미터를 q_M, q_R, α 와 β , 그리고 ρ 와 ω 의 4가지로 구분하여 각 분류별로 포함되지 않은 파라미터들은 일차선정 값으로 고정시키고 해당 파라미터들의 일차선정 값 근방을 조사하였다. 실험은 10개의 작업을 1대의 기계에 할당하는 경우 2가지와 15개의 작업을 2대의 기계에 할당하는 경우 2가지, 총 4가지 실험 데이터를 4-1의 데이터 생성조건에 따라 생성하여 실행하였다.

생성된 실험데이터에 대해 각 분류별로 작업수 만큼의 에이전트로 반복(Iteration) 2000번을 최대로 하여 ILOG CPLEX에 의한 최적해에 최초로 도달하는 반복시점을 조사하였다. <Fig-

ure 3>, <Figure 4>, <Figure 5>, <Figure 6>는 각 실험을 10회씩 실행한 후 평균하여 각 최적해 도달 반복시점의 편차(GIN (Gap of Iteration Number) = (반복횟수 - 최소반복횟수)/최소반복횟수 $\times 100$)를 각 분류별 파라미터 값의 변화에 따라 비교 정리한 것이다.

실험결과 $q_M=0.3, q_R=0.05, \alpha=0.6, \beta=0.4, \rho=0.05, \omega=0.1$ 일때 알고리즘의 성능이 가장 좋음을 알 수 있다. 특히 <Figure 4>에서 보듯이 본 연구에서 기존의 ACS에 추가로 도입한 랜덤 선택 상태전이규칙의 사용은 알고리즘의 성능을 상당히 향상시킴을 알 수 있다.

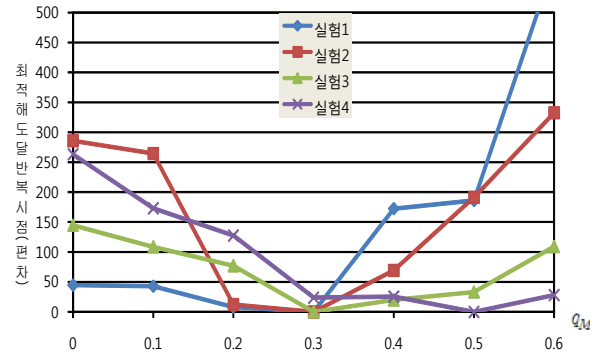


Figure 3. GIN with changes in q_M

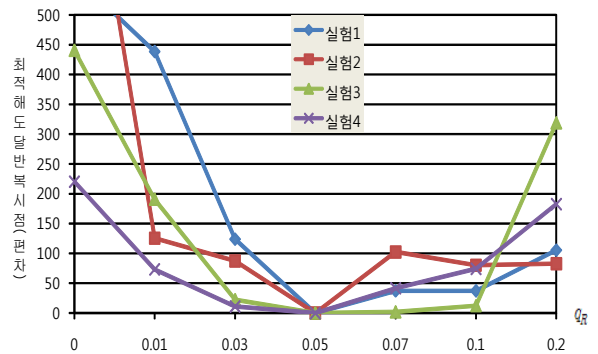


Figure 4. GIN with changes in q_R

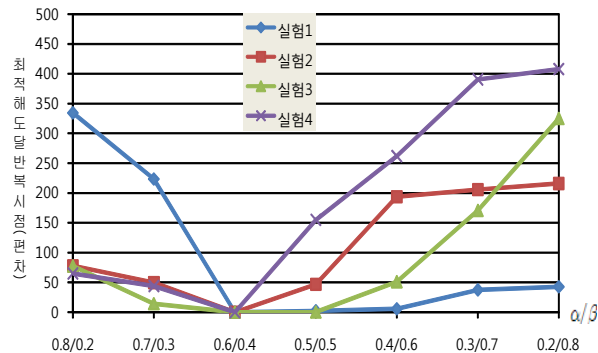


Figure 5. GIN with changes in α and β

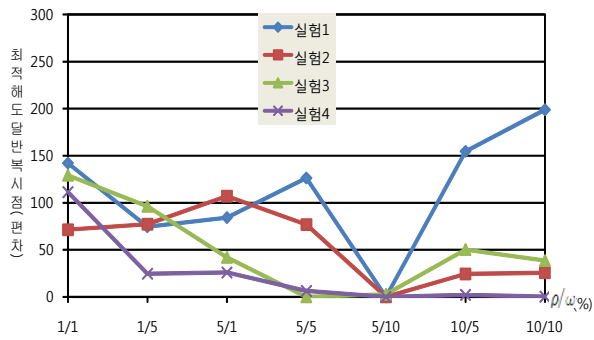


Figure 6. GIN with changes in ρ and ω

4.3 실험결과

IACS를 검증하기 위하여 본 연구에서는 10개의 작업을 1대의 기계에 할당하는 경우, 15개의 작업을 2대의 기계에 할당하는 경우, 그리고 20개의 작업을 3대의 기계에 할당하는 경우의 3가지 종류의 실험 데이터를 4-1의 생성조건에 따라 각기 10개씩 생성하였다. 생성된 각 실험데이터에 대해 IACS에 의한 해(앞에서 선택된 파라미터 값들 이용)를 ILOG CPLEX에 의한 최적해와 비교하였다. 각 실험데이터는 작업수 만큼의 에이전트로 반복(Iteration) 3000번을 최대로(반복 1000번 이상 최적해의 개선이 없으면 종료) 알고리즘을 수행하여 그 결과

Table 1. Experimental Results

실험데이터			최적해	ACS 해			IACS 해		
번호	작업수	기계수		평균	표준편차	RFD(%)	평균	표준편차	RFD(%)
1	10	1	451	516.6	26.7	14.55	451	0.0	0.00
2			722	727.4	11.3	0.75	722	0.0	0.00
3			694	696.2	6.6	0.32	694	0.0	0.00
4			325	325	0	0.00	325	0.0	0.00
5			635	658.5	17.7	3.70	635	0.0	0.00
6			1209	1218.7	10.8	0.80	1209	0.0	0.00
7			467	479.4	8.5	2.66	467	0.0	0.00
8			509	521.5	12.8	2.46	510.8	1.5	0.35
9			618	622.6	9.7	0.74	618	0.0	0.00
10			1161	1190.7	33.8	2.56	1161	0.0	0.00
11	15	2	307	320.8	5.1	4.50	307	0.0	0.00
12			437	479.8	14.4	9.79	440.3	4.1	0.76
13			287	302.3	8.7	5.33	288.8	0.6	0.63
14			315	330	0.0	4.76	330	0.0	4.76
15			390	418.8	17.4	7.38	390	0.0	0.00
16			220	231	0.0	5.00	220	0.0	0.00
17			647	746.8	31.3	15.43	661.8	17.2	2.29
18			375	396.1	5.3	5.63	375	0.0	0.00
19			344	362.7	12.3	5.44	344	0.0	0.00
20			332	354.6	2.8	6.81	332.5	0.5	0.15
21	20	3	281	310.7	4.1	10.57	289.6	1.3	3.06
22			374	393.2	4.9	5.13	375.1	1.6	0.29
23			260	280.9	8.2	8.04	263.5	2.3	1.35
24			587	640.6	17.5	9.13	589	2.0	0.34
25			328	391.4	7.4	19.33	334.8	4.2	2.07
26			311	320.5	0.9	3.05	311.1	0.3	0.03
27			287	306	2.0	6.62	294	0.0	2.44
28			317	338.7	6.9	6.85	317.1	0.3	0.03
29			248	254.7	5.5	2.70	248	0.0	0.00
30			317	370.8	14.3	16.97	318.8	1.2	0.57
평균					10.2	6.23		1.2	0.64

들을 <Table 1>에 정리하고, 랜덤 선택 상태전이규칙을 적용하지 않았을 경우인 기본적인 ACS에 의한 해와의 비교도 동시에 수행하였다. <Table 1>에는 각 실험데이터에 대해 ACS와 IACS를 10회씩 실행하여 그 해들의 평균과 표준편차를 정리하였고, 동시에 해 평균의 상대비율편차(Relative Percentage Deviation : RFD)를 계산하였다. RFD는 최적해에 대한 알고리즘에 의한 해의 편차의 비를 의미하며 식 (20)으로 표현된다.

$$RFD = \frac{\text{알고리즘해} - \text{최적해}}{\text{최적해}} \times 100 \quad (20)$$

RFD가 5%란 것은 알고리즘에 의한 해가 최적해에 비해 5% 초과한다는 것이므로, RFD의 값이 적을수록 알고리즘의 효과(Effectiveness)가 좋다는 것을 의미한다.

<Table 1>의 실험 결과 RFD를 보면 IACS인 경우 그 평균이 0.64%로 대단히 우수하다. 이는 ACS의 6.23%에 비해서도 상당히 우수함을 알 수 있다. 최근에 Behnamian *et al.* (2009)이 유사한 병렬기계 일정계획문제(작업투입시점 제약이 없음)에 대해 ACO(Ant Colony Optimization), SA(Simulated Annealing), 그리고 VNS(Variable Neighborhood Search)를 결합하여 HMH(ACO/SA /VNS), SAV(SA/VNS), ANV(ACO/VNS), VNS의 네가지 하이브리드 알고리즘을 제시하고 실험을 통하여 각 알고리즘의 평균 RFD를 계산하였다. 그 결과 HMH 1.97%, SAV 4.82%, ANV 5.98%, VNS 7.38%로 세가지 휴리스틱 모두를 결합한 HMH가 가장 좋은 결과를 보였다. 문제 상황이 조금 다르고 실험 데이터가 서로 달라 정확한 비교는 되지 않지만, IACS의 성능이 다른 휴리스틱 방법에 비해서도 뛰어난 것을 알 수 있다. <Table 1>의 실험 결과 표준편차 역시 IACS인 경우 평균 1.2로 ACS의 경우는 10.2보다 현저히 적음을 알 수 있다. 이는 IACS를 같은 실험 데이터로 여러번 실행해도 큰 변화 없이 매번 유사한 해를 찾는다는 것을 의미하며 이는 IACS가 해의 강건성(robustness)에 있어서도 대단히 우수함을 의미한다.

IACS의 랜덤 선택 상태전이규칙의 사용의 효과는 <Figure 7>, <Figure 8>, <Figure 9>, <Figure 10>을 보면 확실히 알 수 있다. <Figure 7>, <Figure 8>은 실험 데이터 17을 대상으로 각기 IACS와 ACS에 대해 반복횟수 1000번까지의 해의 수렴과

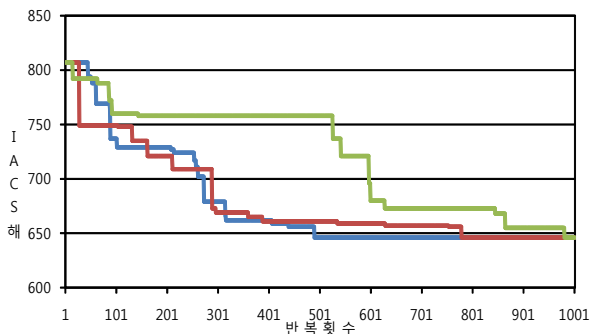


Figure 7. Converting process of IACS

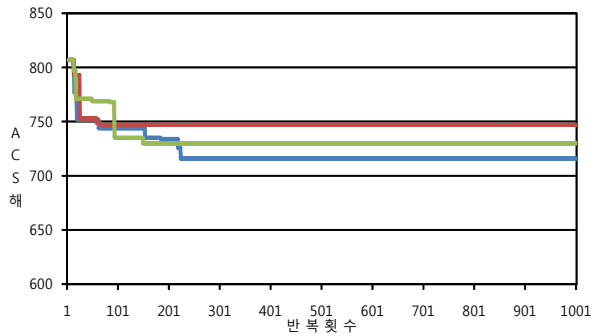


Figure 8. Converging process of ACS

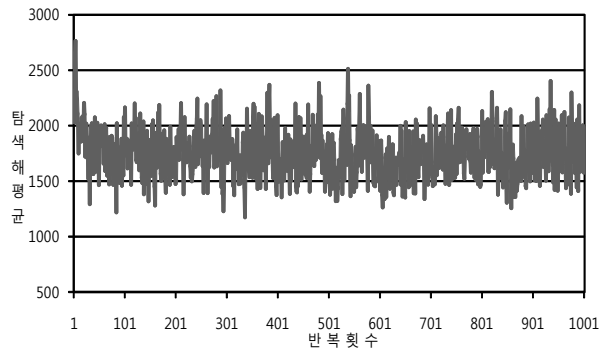


Figure 9. Mean of search solutions; IACS

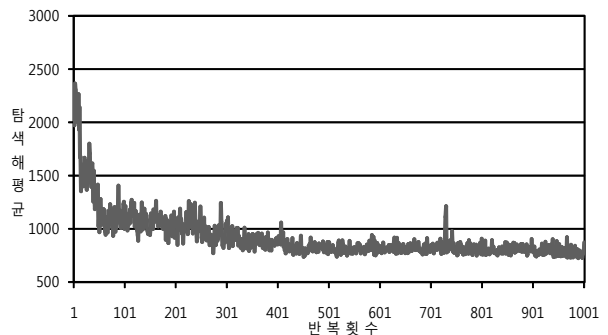


Figure 10. Mean of search solutions; ACS

정을 3회씩 수행하여 나타낸 것이고, <Figure 9>, <Figure 10>은 같은 실험 데이터를 대상으로 각기 IACS와 ACS에 대해 반복횟수별 모든 에이전트가 탐색하는 해의 평균을 나타낸 것이다. ACS의 경우 <Figure 10>에서 보는바와 같이 에이전트들이 탐색하는 해의 평균이 반복횟수가 증가함에 따라 점차 수렴하여 <Figure 8>과 같이 국부 최적해로의 수렴 현상이 발생함을 알 수 있다. 그러나 랜덤 선택 상태전이규칙을 도입한 IACS의 경우 <Figure 9>에서 보는바와 같이 에이전트들이 반복횟수가 증가하여도 지속적으로 다양한 해공간을 탐색함으로써, <Figure 7>과 같이 국부 최적해로 수렴하지 않고 최적해에 도달함을 알 수 있다.

작업투입시점과 완료요구시점을 동시에 가지며 순서의존

적인 준비시간이 존재하는 병렬기계 일정계획문제는 전형적인 NP-hard 문제이므로 최적화 모형을 통해 실시간에 우수한 해를 얻는 것이 불가능하나(작업의 수가 20개일 이상일 경우, ILOG CPLEX를 이용한 최적해의 탐색시간이 수 시간에서 수십 시간 이상이 걸린다) IACS가 해를 찾는 탐색시간은 1-2분이내일 정도로 적어 실제문제에 적용 시 효율적인 병렬기계 일정계획의 수립이 가능할 것이다.

5. 결론

본 연구에서는 작업투입시점과 완료요구시점을 동시에 가지며 순서의존적인 준비시간이 존재하는 병렬기계 일정계획문제를 다루었다. 두 가지 벌과 시간(delay 시간과 tardy 시간)과 작업준비시간의 가중합 최소화 목적을 갖는 최적화 모델을 제시하고, 보다 효율적으로 해를 도출하기 위해 개선 개미군집시스템을 활용한 메타 휴리스틱 알고리즘을 제시하였다. 제시된 알고리즘의 성능을 평가하기 위해 실험을 한 결과, 최적해에 근사한 해를 짧은 계산시간 내에 탐색하는 탁월한 효율성을 입증하였다. 특히 본 연구에서 기존의 ACS에 추가로 도입한 랜덤 선택 상태전이규칙의 사용은 유전알고리즘의 돌연변이와 같은 기능을 수행케 함으로써 개미군집시스템의 국부 최적해로의 수렴 현상을 보완하여 알고리즘의 성능을 상당히 향상시킴을 알 수 있었다.

본 연구에서 개발된 알고리즘을 활용하여 사용자가 편리하고 신속하게 실시간 일정계획을 수립할 수 있도록 자동 및 수동 생산일정계획을 위한 시스템을 개발하여 이를 생산 일정계획 업무에 활용하면 생산성 및 고가의 생산 설비 가동률을 극대화하는데 기여하고, 이를 통한 원가 절감 효과는 기업의 경쟁력 강화에 근간이 될 것이다.

참고문헌

Behnamian, J., Zandieh M., and Ghomi, F. (2009), Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm, *Expert Systems with Applications*, **36**, 9637-9644.

Bell, J. E. and McMullen, P. R. (2004), An colony optimization technique for the vehicle routing problem, *Advanced Engineering Informatics*, **18**, 41-48.

Bobrowski, P. M. and Kim, S. C. (1994), The impact of sequence dependent setup time on job shop scheduling performance, *International Journal of Production Research*, **32**, 1503-1520.

Cheng, C. and Mao, C. (2007), A modified ant colony system for solving the traveling salesman problem with time window, *Mathematical and Computer Modeling*, **46**, 1225-1235.

Dorigo, M., Birattari, M., and Stutzle, T. (2006), Ant Colony Optimization : Artificial Ants as a Computational Intelligence

Technique, *TR/IRIDIA/2006-023*, University Libre De Bruxelles, Belgium.

Dorigo, M. and Blum, C. (2005), Ant colony optimization theory : A survey, *Theoretical Computer Science*, **344**, 243-278.

Dorigo, M. and Di Caro, G. (1999), The Ant Colony Optimization Meta-heuristic, *New Ideas in Optimization*, McGraw-Hill.

Dorigo, M. and Gambardella, L. M. (1997), Ant Colony System : A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation I*, **1**, 53-66.

Dorigo, M., Maniezzo, V., and Colomi, A. (1996), The Ant System : Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems*, **26**, 29-41.

Flynn, B. B. (1987), The effects of setup time on output capacity in cellular manufacturing, *International Journal of Production Research*, **25**, 1761-1772.

Frederickson, G., Hecht, M. S., and Kim, C. E. (1978), Approximation algorithm for some routing problems, *SIAM Journal on Computing*, **7**, 178-193.

Gharehgozli, A. H., Tavakkoli-Moghaddam, R., and Zaerpour, N. (2009), A fuzzy-mixed-integer goal programming mode for a parallel-machine scheduling problem with sequence-dependent setup times and release dates, *Robotics and Computer-Integrated Manufacturing*, **25**, 853-859.

Garey, M. and Johnson, D. (1979), Computers and intractability : A guide to the theory of NP-completeness, New York : W. H. Freeman.

Hahn, C. K., Bragg, D. J., and Shin, D. W. (1989), Impact of the setup variable on capacity and inventory decisions, *Academic Management Review*, **13**, 91-103.

Mendes, A. S., Muller, F. M., Franca, P. M., and Moscato, P. (2002), Comparing metaheuristic approaches for parallel machine scheduling problems, *Production Planning and Control*, **13**, 143-154.

Nessah, F., Yalaoui, F., and Chu, C. (2005), New heuristics for identical parallel machine scheduling with Sequence-dependent setup times and dates, In Proceedings of the international conference on industrial engineering and systems management, *Marrakech, Morocco*, 32- 41.

Pinedo, M. (1995), Scheduling theory, algorithms, and systems, *Prentice-Hall*.

Shyu, S. J., Lin, B. M. T., and Lin, P. Y. (2004), Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time, *Computers and Industrial Engineering*, **47**, 181-193.

Solimanpur, M., Vart, P., and Shankar, R. (2004), Ant colony optimization algorithm to the inner-cell layout problem in cellular manufacturing, *European Journal of Operations Research*, **157**, 592-606.

Tahar, D. N., Yalaoui, F., Chu, C., and Amodeo, L. (2006), A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times, *International Journal of Production Economics*, **99**, 63-73.

Ying, K. and Liao, C. (2004), An ant colony system for permutation flow-shop sequencing, *Computers and Operations Research*, **31**, 791-801.