

교착 회피를 고려한 Job-Shop 일정의 최적화

정동준* · 이두용** · 임성진***

(2000년 1월 3일 접수)

Optimization of Job-Shop Schedule Considering Deadlock Avoidance

Dong Jun Chung, Doo Yong Lee and Seong Jin Yim

Key Words : Job-Shop Scheduling (Job-Shop 일정계획), Deadlock Avoidance Policy (교착 회피 정책), Digraph (유향그래프), Simulated Annealing (시물레이티드 어닐링)

Abstract

As recent production facilities are usually operated with unmanned material-handling system, the development of an efficient schedule with deadlock avoidance becomes a critical problem. Related researches on deadlock avoidance usually focus on real-time control of manufacturing system using deadlock avoidance policy. But little off-line optimization of deadlock-free schedule has been reported. This paper presents an optimization method for deadlock-free scheduling for Job-Shop system with no buffer. The deadlock-free schedule is acquired by the procedure that generates candidate lists of waiting operations, and applies a deadlock avoidance policy. To verify the proposed approach, simulation results are presented for minimizing makespan in three problem types. According to the simulation results the effect of each deadlock avoidance policy is dependent on the type of problem. When the proposed LOEM (Last Operation Exclusion Method) is employed, computing time for optimization as well as makespan is reduced.

1. 서론

전통적인 관점에서의 Job-Shop 일정계획은 여러 문헌에서 다루어져 왔지만^(1,2,13,14) 대부분 시스템 내의 각 기계에 무한대의 용량을 가진 버퍼가 있다고 가정하고, 제한된 용량의 버퍼로 인해 발생하게 되는 교착을 고려하지 않고 있다. 그러나 물류의 이동이 자동으로 이루어지는 무인 Job-Shop 시스템에서는 각 기계의 버퍼 용량이 제한되어 있기 때문에, 효율적인 자원의 이용과 생산성의 향상을 위하여 교착 (deadlock)이 발생하지 않는 일정을 구하는 것이 중요하다. 교착을 고려한 일정 계획에 대한 연구는 최근 자동화된 Job-Shop 시스템에 대한 관심이 증대된 이후에 시작되었는데, Job-Shop 시스템에서 교착을 피하면서 일정을 최

적화하는 것에 대한 연구는 Ramaswamy 와 Joshi⁽⁹⁾의 것을 들 수 있다. 그러나 교착을 고려하는 일정 계획 문제의 복잡성 때문에 실제 생산 시스템에 적용하는 데에는 한계가 있고, 이것을 극복하기 위해서는 교착 회피 알고리즘을 도입한 일정 계획 방법이 필요하다.

한편, 교착과 관련된 연구들은 이제까지 주로 교착 회피 정책을 제안하는 것과 이를 이용한 생산 시스템의 실시간 제어에 초점이 맞추어져 왔다. Viswanadham 등⁽⁴⁾은 GSPN (Generalized Stochastic Petri Net)을 이용한 교착방지 방법과 교착 회피 방법을 제시하였고, Banaszak 과 Krogh⁽³⁾은 PPN (Production Petri Net)을 정의하고 교착 회피 알고리즘 (DAA : Deadlock Avoidance Algorithm)을 제안하였다. 교착과 부품들의 작업 순서에 의해서 발생하는 사이클의 관계에 대해서는 Wysk 등⁽⁵⁾이 밝힌바 있고 그 주요 개념에 기반해서, Fanti 등⁽⁶⁾은 부품과 자원들간 상호작용상의 특징을 유향그래프 (digraph)로 모델링하여 교착 회피 정책을 제안하

* 硏究性 中 公 業 硏 究 所

** 회원, 한국과학기술원 기계공학과

E-mail : leedy@kaist.ac.kr

TEL : (042)869-3229 FAX : (042)869-3210

*** 한국과학기술원 기계공학과 박사과정

였다. Lawley⁽⁷⁾는 경로 유연성을 가지는 유연생산 시스템에서의 교착 회피 문제를 특성화하고 각 기계들이 하나의 버퍼를 가지는 경우와 여러 개의 버퍼를 가지는 경우에 대하여 교착 회피 방법을 제시하였고, Wu⁽⁸⁾는 교착 제어를 위해 필요한 동시적인 자원 경쟁과 제조 과정상의 특징을 CROPN (Colored Resource-Oriented Petri Net)으로 모델링하여 동적인 자원할당의 개념을 도입하였다. 이상에서 살펴본 교착과 관련한 연구들은 일정의 최적화를 생각하지 않기 때문에 Job-Shop 시스템에서는 일정을 최적화하는 방법을 부수적으로 요구하게 된다.

본 논문에서 다루고 있는 교착 배제 일정계획 (deadlock-free scheduling)과 관련된 연구들은 위에서 언급한 바와 같이 교착을 고려하지 않은 전통적인 일정계획에 대한 연구와 교착 회피 (deadlock avoidance)에 대한 것으로 나누어 볼 수 있다. 전자는 최적화 개념이 도입되지만 교착을 고려하지 않는 가정이 현실성이 없고, 후자는 실제 생산 시스템에 적용이 가능하지만 최적화가 불가능한 장 단점을 가지고 있다. 따라서 본 연구에서는 교착 회피 정책을 적용한 일정계획 방법을 제안하여 교착을 피함과 동시에 일정을 최적화하는 방법을 제안한다. 일정의 최적화를 위해서는 작업 리스트⁽¹³⁾를 이용한 일정계획 방법을, 교착을 피하기 위해서는 Fanti 등⁽⁶⁾이 제안한 교착 회피 정책을 사용하였는데, Fanti 등⁽⁶⁾의 연구에서 제안된 교착 회피 정책은 기존에 제안된 교착 회피에 대한 연구 중에서도 Job-Shop 시스템에 적용이 용이하고, 비교적 덜 제한적인 정책이므로 본 논문에서 적용하였다.

본 연구에서는 교착 회피 정책을 이용한 일정 계획에서 최적화가 가능함을 보이고 문제 유형에 따른 교착 회피 정책의 성능 차이를 비교한다. 일정을 표현하기 위해서 모든 작업에 각각 임의의 수 (random key⁽¹⁰⁾)를 할당한다. 이렇게 할당된 임의의 수에 의해 후보 리스트 (candidate list)를 구성해 나가면서 교착 회피 정책을 적용하여 작업들을 해당하는 기계에 할당하면 하나의 일정이 얻어진다. 따라서 일정계획은 최적의 비용함수를 얻어내는 일정을 만들도록 각 작업에 대한 임의의 수를 구하는 것이다. 이러한 임의의 수를 구하는 최적화 알고리즘으로는 무작위 탐색 방법 중의 하나인 시뮬레이티드 어닐링 (Simulated Annealing)을 사용하였고 최소화하여야 할 목적함수는 전체 생산시간으로 하였다.

2. 후보 리스트를 이용한

교착 배제 일정계획

Job-Shop 시스템의 일정계획 문제에서 버퍼를 무한대로 가정하고 각 부품의 순서 제한 조건과 기계 제한 조건을 고려하여 풀었을 때 구해지는 일정의 한 예를 Fig. 1 과 같이 간트 차트로 나타내었다. 간트 차트의 각 작업을 나타내는 블록에서 괄호 안의 숫자는 차례로 부품 번호와 작업 번호를 나타낸다.

Fig. 1 의 일정을 각 기계의 버퍼가 없다고 가정하고 검토해 보면 Fig. 1 에서 (1,1)과 (2,1)이 투입된 초기 시간에 이미 교착이 발생함을 알 수 있다. 즉 부품 1 의 두 번째 작업 (1,2)가 할당되기까지 부품 1 은 기계 1 에서 기다리고 있음에도 불구하고 (2,2)가 기계 1 로 투입되어야 한다. 이는 기계 1 에 버퍼가 존재하지 않으면 불가능하다. 따라서 작업 (1,1)과 (2,1)은 서로 교착 상태를 이루고 있다.

일반적으로 NP-complete 문제라고 알려져 있는 Job-Shop 일정계획 문제에서 이러한 교착 상태들까지 일일이 확인하며 일정계획을 한다는 것은 매우 어려운 일이다. 교착 배제 일정계획을 위해 본 논문에서 사용하는 방법은 작업 리스트⁽¹³⁾를 구성하고 각 작업을 할당할 때마다 교착 회피 정책에 의거해 할당의 가부를 결정하는 것이다. 본 논문에서 사용되는 일정계획 방법의 개요를 Table 1 과 같이 9 개의 작업이 수행되는 시스템에 대하여 간단히 설명하면 다음과 같다.

Table 1 에서 주어진 것과 같이 아홉 개의 각 작업이 0 에서 1 까지의 임의의 수 (random key⁽¹⁰⁾)들에 대응하는 작업 구성을 KAC (Key Assignment Configuration, 키할당구성)라 정의한다.¹ 여기서 현재 할당을 기다리고 있는 작업들로 후보 리스트 (candidate list)를 만드는데 처음엔 일단 각 부품들의 첫 번째 작업으로 이루어진다. 후보 리스트를 구성할 때에는 각 작업의 임의의 수 값에 따라 정

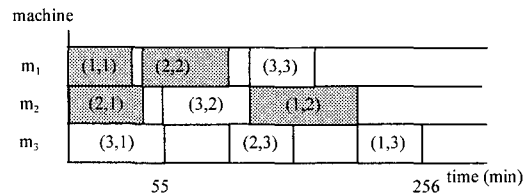


Fig. 1 A schedule with infinite buffer capacity

¹ 실제로 본 논문의 임의의 수 값은 알고리즘의 효율성을 위하여 자연수를 사용하였지만 일반적인 설명을 위해 예제에서는 실수 값의 임의의 수를 사용하였다.

Table 1 A KAC of an example Job-Shop system

| job number | operation number | random key |
|------------|------------------|------------|
| 1 | 1 | 0.81 |
| | 2 | 0.25 |
| | 3 | 0.47 |
| 2 | 1 | 0.31 |
| | 2 | 0.73 |
| | 3 | 0.51 |
| 3 | 1 | 0.98 |
| | 2 | 0.63 |
| | 3 | 0.14 |

렬해 준다.

<첫 번째 후보 리스트>

(3,1,0.98)(1,1,0.81)(2,1,0.31)

이 때 사용되는 교착 회피 정책이 첫 째, 시스템 내의 부품 수는 2 개 이하로 유지, 두 째, 투입되어 있는 부품은 자유롭게 이동 하는 것을 내용으로 한다고 가정하면, 교착 회피 정책에 의거하여 시간이 0 일 때 임의의 수 값이 큰 순서로 (3,1,0.98)과 (1,1,0.81)이 할당 가능하다. 이때 다음 후보 리스트는 다음과 같이 갱신된다.

<두 번째 후보 리스트>

(3,2,0.63) (2,1,0.31) (1,2,0.25)

<두 번째 후보 리스트>에서도 임의의 수 값에 의해 정렬되었다. 임의의 수는 작업이 할당되는 순서를 바꾸어 주는 역할을 하게 되고 이웃 해를 생성해 내는 데에 이용된다. 예를 들어 <첫 번째 후보 리스트>에서 교착 회피 정책을 적용할 때 작업 (2,1)의 임의의 수가 0.81 보다 크다면 작업 (1,1)보다 먼저 할당된다. 각 작업들의 교착 회피 정책에 의거한 할당은 어떤 특정 시간 (System Clock, 4 장 참조)에 결정되는데, <첫 번째 후보 리스트>에서 두 개의 작업이 할당되었을 때, 작업 (3,1)의 가공시간이 더 짧다면, <두 번째 후보 리스트>에서 할당 가능한 작업을 선택하는 시점은 작업 (3,1)이 끝나는 시점이 된다. 여기서 회피 정책에 의거하여 <두 번째 후보 리스트>에서는 작업 (3,2)가 할당되고 후보 리스트는 다음과 같이 바뀌게 된다.

<세 번째 후보 리스트>

(2,1,0.31) (1,2,0.25) (3,3,0.14)

이와 같이 시스템의 상황을 피드백 받아 후보 리스트를 통해 할당 가능한 작업을 선택하는 방법으로 각 System Clock 에 따른 모든 작업들의 할당 순서가 결정되면 유일한 일정이 생성된다. 시물레

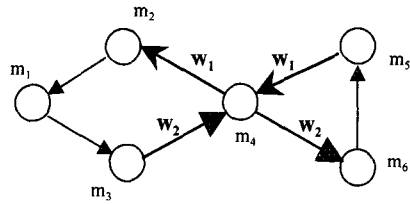


Fig. 2 A simple second level cycle

이티드 어닐링을 이용한 최적화에서는 임의의 수 값에 변화를 주어 KAC 를 바꾸어 주면 작업들의 할당 순서가 바뀌게 되고 그에 따른 일정도 다르게 구해지는데, 이렇게 바뀌어진 KAC 를 이웃해로 하여 탐색을 해 나간다.

3. 교착 회피 정책

(Deadlock Avoidance Policy)

일반적으로 생산 시스템에서의 교착은 순환대기상태 (circular wait condition)를 일컫는다.⁽³⁾ 어떤 기계들의 집합 M 에 속하는 모든 기계들에서 부품들의 집합 J 에 속하는 모든 부품들이 가공 중이고, 집합 J 에 속한 부품들의 다음 작업들이 가공되어야 할 기계들의 집합이 M 과 같다면 이러한 경우 M 의 기계들과 J 의 부품들은 순환대기상태를 이루게 되는데 이때 기계들과 부품의 작업 경로로 이루어지는 사이클을 일차 레벨 사이클이라 정의한다. Fanti 등⁽⁶⁾은 Job-Shop 생산 시스템을 유향 그래프 (digraph)를 이용해서 표현하였는데 어떤 시스템에 적어도 하나의 일차 레벨 사이클이 존재할 경우의 상태를 교착 상태 (deadlock state)라 정의하였고 이에 기반한 교착 회피 정책을 제시하였다.

또한 Job-Shop 시스템에서는 다음 단계에 교착을 일으킬 수 밖에 없는 특별한 상태가 존재하게 되는데 이를 이차 레벨 교착 (second level deadlock)이라 한다. 이차 레벨 교착은 이차 레벨 사이클 (second level cycle)에 의해 발생하게 되는데 이차 레벨 사이클은 두 개 이상의 일차 레벨 사이클을 새로운 노드로 구성하여 이루어진다. 두 개의 일차 레벨 사이클이 오직 하나만의 기계를 공통으로 갖고 그 공통의 기계를 통하면서 지나가는 작업 경로가 두 일차 레벨 사이클에 걸쳐 존재할 경우 두 일차 레벨 사이클 노드를 연결해 주는 아크를 정의하고, 이 아크의 연쇄로 발생하는 사이클이 형성될 때 이를 이차 레벨 사이클로 정의한다.⁽¹⁰⁾ 두 개의 일차 레벨 사이클로 이루어지는 간단한 이차 레벨 사이클의 예를 Fig. 2 에 나타내었다. 그

림에서 m 은 기계를 w 는 작업 경로를 나타낸다.

Fanti 등은 교착 회피를 위해 5 가지의 제한 정책 (Restriction Policy, RP)을 제안하였다. 모든 제한 정책은 시스템 내에 사이클이 생기지 못하도록 하는 것을 기본 개념으로 한다. 각 RP 에서의 제어 규칙은 2 가지로 나누는데 첫 번째는 어떤 상태에서 투입 가능한 부품들의 투입 (1-타입 이벤트)을 결정하는 규칙이고 두 번째는 이미 투입되어 있는 부품들의 다른 기계로의 이동 (2-타입 이벤트)을 결정하는 규칙이다.

RP1 은 전체 시스템 내의 부품의 수를 제한하는 규칙이고, 그 수를 시스템 내에 존재하는 사이클 중 가장 크기가 작은 사이클의 용량보다 1 만큼 작게 유지할 수 있을 때에 투입하는 정책이다. RP2 는 시스템 내의 부품 수를 제한하되 그 부품에 관련된 사이클의 집합을 확인하고 여기에 속하는 사이클들과 연관이 있는 부품들의 투입 상태만을 고려하여 부품의 투입을 제어하는 정책이다.² RP1 과 RP2 에서 2-타입 이벤트는 제한하지 않는다.

RP3 는 1-타입 이벤트와 2-타입 이벤트의 발생이 새로운 사이클을 형성하지 않는다면 이벤트를 발생시키는 제어 정책으로서 이차 레벨 교착이 일어나지 않는 시스템에서는 최소 제한적 (least restrictive)이다. 어떤 시스템의 교착을 피하기 위하여 교착 회피 정책을 적용할 경우, 실제로 교착을 일으키지 않는데도 배제되는 상태들이 존재하게 되는데, 여기서 “최소제한적” 이라 함은 그러한 상태들이 배제되지 않도록 함을 의미한다. 그런데 RP3 를 사용하기 위해서는 이차 레벨 사이클이 시스템 내에 존재하지 않아야 한다는 조건이 필요하다. 한 단계 예상 (look-ahead)만으로는 이차 레벨 사이클에서 발생하는 이차 레벨 교착을 피한다는 보장을 할 수 없기 때문이다. RP4 와 RP5 는 RP3 에 대해 1-타입 이벤트의 발생에 대해 추가적인 제한을 가하는 정책으로서 새로운 부품을 투입하더라도 시스템에서 이차 레벨 교착이 발생하지 않도록 하는 정책이다. 따라서 RP4 와 RP5 는 최소 제한적이지 못하다.

RP4 는 새롭게 투입하여야 할 부품이 사이클을 형성하지 않고, 또한 현재 시스템 내에 투입되어 있는 부품들의 수를 최소 규모의 이차 레벨 사이클의 용량보다 2 만큼 적게 유지할 수 있을 때에 투입하는 정책이다. RP5 는 RP4 와 같이 새로운 부품의 투입을 이차 레벨 사이클의 생성과 관련하여

제한하는 정책인데, RP2 가 RP1 에 대해 그러했듯이 투입하려는 부품에 관련된 이차 레벨 사이클의 집합을 확인하고 여기에 속하는 이차 레벨 사이클들과 연관이 있는 부품들의 투입 상태만을 고려하여 부품의 투입을 제어한다.

4. 교착 회피 정책을 적용한 일정 계획

4.1 시뮬레이티드 어닐링

시뮬레이티드 어닐링 (SA)은 조합 최적화 문제의 최적해를 찾기 위한 확률적인 탐색 방법이다. Kirkpatrick 등⁽¹¹⁾ 은 처음으로 최적화 문제를 풀기 위해 SA 를 제안하였고, 일정 계획 문제에 SA 를 적용한 연구들이 보고되었다.^(12,13,15)

본 논문에서는 KAC 의 최적 변경 (optimal permutation)을 찾기 위해 표준 SA 가 사용되었다. 표준 SA 는 두 개의 루프로 구성되어 있다 하나는 내부루프로써 탐색 루프이고 다른 하나는 외부루프로써 온도 제어 루프이다. SA 는 초기 구성 X_0 와 초기 온도 T_0 에서 시작한다. 탐색 루프에서는 온도를 고정시킨 상태에서, 교란 연산자 (perturbation operator)를 이용하여 현재의 구성 $X_{current}$ 로부터 새로운 구성 X_{new} 를 생성한다. 그리고 나서 새로운 구성의 비용함수를 계산하고 이전 해와의 차이를 구해내어 새로운 구성을 취할 것인지를 결정한다. 만약 비용함수가 감소하면 새로운 구성은 받아들여진다. 그렇지 않을 경우엔 새로운 구성은 $\exp[-\Delta C/T]$ 의 확률을 가지고 받아들여지는데 여기서 ΔC 는 구성들의 비용함수의 차이이고 T 는 특정 단계에서의 온도이다. 이 내부 루프는 준평형 (quasi-equilibrium)상태에 도달할 때까지 반복된다. 그리고 나서 외부 루프에서는 현재 k 번째 단계에서의 온도 $T(k)$ 를 다음 단계를 위해서 감소시킨다. 내부 루프는 새로운 온도 $T(k+1)$ 에서 또 다시 반복된다. 이러한 전체적인 과정은 종료 기준 (stopping criterion)을 만족할 때까지 반복된다.

SA 를 일정계획 문제에 적용하기 위해서는 구성들의 집합, 각 구성들의 비용함수, 초기 구성과 이웃 구조 (neighborhood structure)들이 정확히 정의되어야 한다. 본 논문에서는 해를 KAC 로, 비용함수는 전체 생산시간 (makespan)으로 하였다. 하나의 KAC 는 각 작업에 부여된 임의의 수에 의해 결정되고 각 부품들의 작업 순서 조건에 의해 순차적으로 갱신되는 후보 리스트를 구성하는 데에 사용된다. 후보 리스트로부터 교착 회피 정책을 만족하는 작업들을 추출해가며 일정계획을 하면

² 여기서 부품과 사이클이 관련되어 있다 함은 부품의 작업 경로와, 사이클에서 기계들간의 순환 경로가 하나 이상 일치함을 의미한다.

유일하게 일정이 결정되고 해당하는 비용함수가 얻어지게 된다. 각 해에 대한 이웃 구성은 교란 연산자에 의해 정의되고 이웃은 현재의 해에 교란 연산자를 적용하였을 경우 얻어지는 해들의 집합으로 정의한다.

본 연구에서는 현재 구성의 이웃을 구하기 위한 연산자로 두 가지가 사용되었다. 그 중 하나는 이동 연산자 (move operator)이고 다른 하나는 교환 연산자 (exchange operator)이다. 각 연산자는 임의의 수의 조작을 통하여 이루어진다.⁽¹⁰⁾ 이동 연산자는 KAC 에서 임의로 하나의 작업을 선택한 후에 그 작업의 임의의 수를 다른 값으로 바꾸어 줌으로써 KAC 로부터 이루어지는 전체적인 할당 순서를 바꾸어 주는 연산자이고, 교환 연산자는 KAC 에서 임의의 두 개의 작업을 뽑아내어 둘의 임의의 수를 바꾸어 줌으로써 KAC 로부터 이루어지는 전체적인 할당 순서를 바꾸어 주는 연산자이다. 그러나 항상 두 개의 연산자가 또 다른 일정을 생성해 내도록 KAC 를 변화 시키는 것이 아님을 고려해야 한다. 이러한 가능성을 최소화 하기 위하여 본 논문에서는 재번호매김 (renumbering)의 방법을 사용하였다. 사용된 임의의 수 값들은 모두 자연수로서 1 에서부터 할당해야 할 전체 작업의 개수만큼의 자연수로 이루어지게 되는데, 하나의 일정을 구하면서 각 작업이 간트 차트에 투입되는 순서에 따라 각 작업의 임의의 수 값을 1 부터 차례대로 다시 부여해 준다. 즉, k 번째에 간트 차트로 할당되는 작업을 $o[k]$ 라 한다면 이 작업의 교란되기 직전의 임의의 수 값은 k 가 된다. 이렇게 하면 교란 연산자를 적용하기 직전의 작업리스트의 임의의 수 구성은 재번호매김 이전의 임의의 수 구성과는 분명 다르겠지만 똑같은 일정을 의미하게 되고, 이어지는 교란에 의해 또 다른 일정을 생성할 가능성이 커지게 된다. 임의의 수 조작에 있어서의 일련의 과정을 Fig. 3 에 나타내었다.

4.2 교착 배제 일정의 생성

전통적인 Job-Shop 일정계획 문제는 n 개의 부품으로 이루어진 집합 $J = \{j_i\}_{i=1}^n$ 과 m 개의 기계로 이루어진 집합 $M = \{m_k\}_{k=1}^m$ 로 구성되어 있다. 각 부품들의 작업 수를 $NO[i]$ 라 한다면 각 부품들을 이루고 있는 작업들 $o_{i1}, o_{i2}, \dots, o_{i1}, \dots, o_{iNO[i]}$ 는 이미 정해져 있는 순서대로 진행된다. 따라서 할당해야 할 전체 작업의 개수는 다음과 같다.

$$L = \sum_{i=1}^n NO[i] \tag{1}$$

일정계획은 각 작업들의 해당 기계에 대한 시

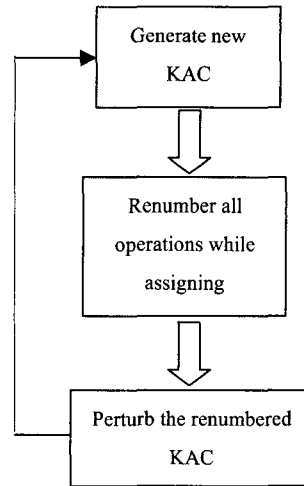


Fig. 3 Mutation of a KAC by manipulating random keys

작 시각 t_{ij} 을 결정하는 것이다. 각 작업 o_{ij} 이 작업시간 τ_{ij} 동안 진행된다고 할 때 전체 생산시간 C_{max} 의 최적값 C_{max}^* 를 구하는 것을 목적으로 하는 일정계획의 경우 C_{max}^* 는 다음과 같이 표현된다.

$$C_{max}^* = \min(C_{max}) = \min_{feasible\ schedules} (\max(t_{ij} + \tau_{ij}) : \forall j_i \in J, m_k \in M) \tag{2}$$

본 논문에서는 시스템 상태의 정보를 피드백 받고 교착 회피 정책을 고려하여 작업을 할당해 주기 위해서 System Clock 과 Job Clock, 전갱신과 후갱신의 개념을 도입하였다. System Clock α 는 일정계획에서 작업들을 할당해 나갈 때 어떤 부품의 투입이나 투입된 부품들의 이동이 결정되는 시점을 의미한다. 일정계획 중에 발생하는 모든 System Clock 은 초기시간이나 어떤 작업이 끝났을 때에 해당한다. Job Clock β 는 어떤 부품의 작업이 할당되었을 경우 해당하는 작업이 끝나는 시간을 그 System Clock 에서의 각 부품들의 Job Clock 이라 한다. $\alpha[x]$ 에서 작업 o_{ij} 이 할당된다면 새롭게 갱신되는 Job Clock 은 다음과 같다.

$$\beta_i = \alpha[x] + \tau_{ij} \tag{3}$$

어느 System Clock 에서 더 이상 할당 가능한 작업이 없으면 System Clock 을 갱신하여야 하는데 바로 이때 부품들의 Job Clock 이 사용되고 기존에 할당되어 있는 부품들의 Job Clock 중에서 가장 작은 값을 갖는 부품의 Job Clock 을 System Clock 으로 한다.

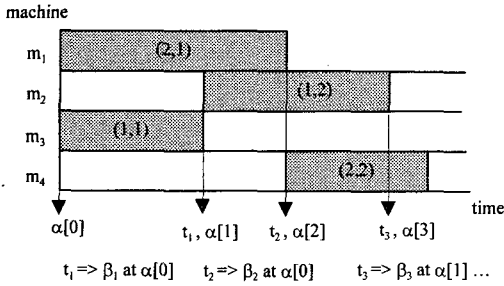


Fig. 4 System clock and job clock

$$\alpha[x+1] = \min \beta_i, \forall \beta_i > \alpha[x] \quad x = 0, 1, 2, \dots \quad (4)$$

System Clock 과 Job Clock 과의 관계를 Fig. 4 에 나타내었다.

전갱신 ($U_{pre}(o_{il})$) 은 작업 o_{il} 의 할당이 결정되었을 경우 작업을 할당함과 동시에 행해지는 시스템 상황의 갱신이다.

$$t_{il} = \alpha[x] \text{ 이면 } \alpha[x] \text{ 에서 } U_{pre}(o_{il}) \text{ 시행} \quad (5)$$

모든 작업은 부품 번호, 작업 번호, 해당 기계, 가공 시간, 작업 경로 타입(type) 등의 정보를 담고 있는데 이 정보들을 시스템 상황의 갱신과 System Clock 의 갱신에 이용한다. 전갱신에서는 할당된 작업이 첫 번째 작업일 경우 해당하는 1,2 차 사이클에 대해 부품의 수를 하나 증가시키고 이전 작업이 차지하고 있던 기계를 비운다. 전갱신 $U_{pre}(o_{il})$ 에는 식 (3)으로 나타낸 Job Clock 의 갱신과 함께 다음과 같은 내용이 포함된다.

$l=1$ 이면,

$$\bullet JN(q+1, c_k) = JN(q, c_k) + 1, \quad \forall c_k \in C[j_i] \quad (6)$$

$$\bullet JN(q+1, c^2_k) = JN(q, c^2_k) + 1, \quad \forall c^2_k \in C^2[j_i] \quad (7)$$

$$MS(m[o_{i,l-1}]) = 0, \quad MS(m[o_{il}]) = 1 \quad (8)$$

여기서 상태 $q+1$ 은 상태 q 에 이어지는 다음 상태를 나타낸다. $JN[q, c_k]$ 는 현재 상태 q 에서 투입되어 있는, 사이클 c_k 와 관련된 부품의 개수이고 $C[j_i]$ 는 부품 j_i 와 관련된 사이클들의 집합이다. 또한, $JN[q, c^2_k]$ 는 현재 상태 q 에서 투입되어 있는, 이차 레벨 사이클 c^2_k 와 관련된 부품의 개수이고 $C^2[j_i]$ 를 부품 j_i 와 관련된 이차 레벨 사이클들의 집합이다. $MS(m[o_{il}])$ 은 작업 o_{il} 이 수행되어야 하는 기계에 대한 상태를 나타내는 기계 상태 표시로서 0 일 경우 비어있는 것으로, 1 일 경우 채워져 있는 것으로 정의한다.

후갱신 ($U_{post}(o_{il})$) 은 System Clock 을 이동시킨 후, 그 시점에서 끝나는 작업 o_{il} 의 작업 정보를 이용한 시스템의 갱신이다. 그런데 같은 System Clock 에서 여러 작업이 동시에 끝났을 경우는 해

당하는 모든 작업들에 대해 후갱신을 시행하여야 한다. 따라서 본 논문의 알고리즘에서는 최소 Job Clock 과 같은 Job Clock 을 가지는 모든 작업에 대해 후갱신을 시행한다.

$$t_{il} + \tau_{il} = \alpha[x] \text{ 이면, } \alpha[x] \text{ 에서 } U_{post}(o_{il}) \text{ 시행} \quad (9)$$

후갱신에서는 다음 작업이 할당될 수 있도록 현재 작업이 끝났음을 알리며, 해당 작업이 마지막 작업일 경우에는 해당하는 1,2 차 사이클에 대해 부품의 수를 하나 감소시킨다. 후갱신 $U_{post}(o_{il})$ 에는 다음과 같은 내용이 포함된다.

$l = NO[i]$ 이면,

$$\bullet JN(q+1, c_k) = JN(q, c_k) - 1, \quad \forall c_k \in C[j_i] \quad (10)$$

$$\bullet JN(q+1, c^2_k) = JN(q, c^2_k) - 1, \quad \forall c^2_k \in C^2[j_i] \quad (11)$$

교착 회피 정책을 적용한 일정계획 알고리즘에서 목적함수의 계산은 Fig. 5 와 같은 순서로 진행된다.

4.3 교착 회피 정책의 적용

후보리스트를 이루고 있는 작업들의 집합을 O_{CL} 이라 하고, $B_{check}(o_{il})$ 은 블로킹을 검사하기 위한 함수라 하자. $B_{check}(o_{il})$ 은 어떤 작업 o_{il} 이 수행되어야 할 기계의 상태를 나타내는 함수로서 전갱신에서 갱신되어지는 $MS(m[o_{il}])$ 의 정보를 이용하여 검사한다. 0 일 경우 블로킹이고 1 일 경우 블로킹이 아니라고 정의한다면, 모든 작업들은 다음 조건을 만족할 경우에만 교착 회피 정책에 의해 작업 할당이 가능하다.

$$o_{il} \in O_{CL}, \quad B_{check}(o_{il}) = 0 \quad (12)$$

여기서 $B_{check}(o_{il})=1$ 일 경우 부품 j_i 가 블로킹에 걸렸음을 나타내기 위해 블로킹 상태 표시 $B(i)$ 값은 1 로 바뀌게 된다. 나중에 o_{il} 이 할당된다면 $B(i)$ 는 0 으로 바뀐다.

RP2 는 각 부품이 사이클들과 관련된 정도를 고려하여 투입을 결정하기 때문에 RP1 보다는 우수한 성능을 보장하고, 같은 이치로 RP5 또한 RP4 에 대하여 그러하다. 그렇지만 시스템의 특성에 따라 RP2 와 RP4 의 성능은 우열이 달라질 수 있다. 따라서 본 논문에서는 RP2, RP4, RP5 에 대한 결과만을 나타내어 비교하였다. Fanti 등⁽⁶⁾이 제안한 다섯 가지 교착 회피 정책 중 세 가지 정책을 적용한 방법과 본 논문에서 교착 배제 일정계획의 성능을 향상시키기 위해 제안한 최종작업 배제방법의 내용은 다음과 같다.

< RP2 >

Capacity(c_k)를 일차 레벨 사이클 c_k 의 크기(일차

레벨 사이클을 구성하고 있는 기계 수를 나타낸다고 할 때 o_{il} 에 대한 할당 규칙은 다음과 같다.

$l=1$ 이면,

$$JN(q, c_k) < [Capacity(c_k) - 1], \forall c_k \in C[j_i] \text{ 일}$$

경우 $t_{il} = \alpha[x]$ (13)

$l \neq 1$ 이면,

$$B(i) = 1 \text{ 이거나 } \beta_i = \alpha[x] \text{ 일 경우 } t_{il} = \alpha[x] \quad (14)$$

< RP4 >

$Capacity(c_k^2)$ 을 이차 레벨 사이클 c_k^2 의 크기(이차 레벨 사이클을 구성하고 있는 기계 수)를 나타낸다고 할 때 o_{il} 에 대한 할당 규칙은 다음과 같다.

$$l=1 \text{ 이면, } JN(q, s) < \min_k [Capacity(c_k^2) - 1], \forall c_k^2$$

이고 $D_{check}(o_{il}) = 0$ 일 경우 $t_{il} = \alpha[x]$ (15)

$l \neq 1$ 이면, $B(i) = 1$ 이거나 $D(i) = 1$ 이거나

$\beta_i = \alpha[x]$ 이고, $D_{check}(o_{il}) = 0$ 일 경우 $t_{il} = \alpha[x]$

(16)

여기서 $D_{check}(o_{il})$ 은 o_{il} 이 할당된다고 가정할 때 일차 레벨 교착이 발생하는지의 여부를 나타내는 함수이고 0 일 경우 교착이 발생하지 않는 것으로 정의한다. $D(i)$ 는 교착 상태 표시로서 이전에 부품 j_i 가 $D_{check}(o_{il})$ 에 의해 할당이 거부되었는지의 여부를 나타내는 표시로서 1 일 경우 $D_{check}(o_{il})$ 에 의해 할당이 거부된 적이 있다는 것을 나타낸다. 나중에 o_{il} 이 할당된다면 $D(i)$ 는 0 으로 바뀐다.

< RP5 >

어떤 부품의 첫번째 작업이 아닐 경우에는 <RP4>에서의 식 (16)이 똑같이 적용되고 첫번째 작업일 경우의 o_{il} 에 대한 할당 규칙은 다음과 같

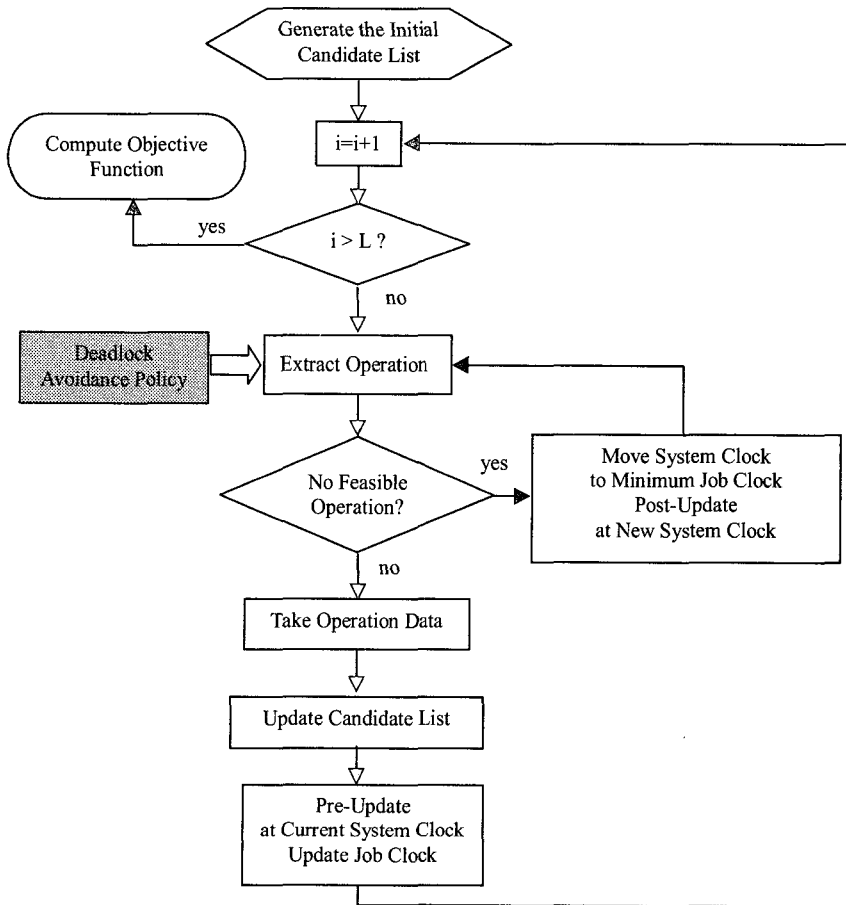


Fig. 5 Evaluation of schedule

다.

$l=1$ 이면,

$$JN(q, c^2_k) < [Capacity(c^2_k) - 2], \quad \forall c^2_k \in C^2[j_i]$$

이고 $D_{check}(o_{ii})=0$ 일 경우 $t_{ii} = \alpha[x]$

(17)

<최종작업 배제방법 (Last Operation Exclusion Method)>

최종작업 배제방법 (LOEM)은 어느 부품의 마지막 작업에 도달하였을 경우에 적용되는 시스템의 갱신 방법이다. 마지막 작업은 바로 다음 단계에 시스템을 빠져 나가기 때문에 그 어떤 사이클에도 기여할 수 없다. 따라서 어떤 부품의 마지막 작업이 시작될 경우 그 부품에 관련된 모든 사이클에서 부품 수를 하나씩 감소시켜도 무방하다.

즉, 어떤 상태 q 에서의 System Clock 이 $\alpha[x]$ 일 때, $t_{ii} = \alpha[x]$ 이고, $l = NO[i]$ 일 경우엔 전갱신에서 후갱신에서의 식 (10) 과 (11)을 적용한다. 본 논문에서 사용된 System Clock 에 따른 시스템의 갱신방법을 사용하면 LOEM 의 적용이 가능해지기 때문에 실험 결과를 따로 분리하여 구하였다.

5. 실험 결과

제안된 교착 회피 정책을 적용한 일정계획 방법의 타당성을 검증하기 위해서 3 가지 유형의 문제에 대해서 C 언어를 이용하여 프로그래밍하고 Pentium PC 500MHz 컴퓨터에서 실험하였다. 모든 작업들의 가공 시간을 20 에서 100 사이의 임의의 자연수로 하고, 시뮬레이티드 어닐링을 각 항목에 대해 50 회 반복 시행하고 전체 생산시간의 평균과 표준 편차를 비교해 보았다.

본 논문에서는 교란 회수에 따라 더 넓은 영역의 이웃 해가 생성되는 것을 확인하고 교란 회수를 변화 시키면서 각 경우에 대하여 원하는 수용 비율 0.9 를 만족시키는 초기 온도를 계산하여 어닐링 일정에서의 파라미터 값으로 사용하였다. 감쇠 함수 $T(k+1) = \gamma * T(k)$ 에 사용되는 감쇠 계수 (decay factor)는 Job-Shop 일정계획 문제에서 일반적으로 많이 사용되는 0.95 의 값을 일률적으로 적용한다. Epoch Length (L)는 동일 온도 단계에서 탐색 되는 이웃 해들의 수를 의미한다. L 을 크게 해주어 가능한 한 많은 이웃 해를 탐색하는 것이 바람직하지만, 이 때 계산 시간은 증가하게 된다. JN 을 부품의 수, MN 을 기계의 수라고 하였을 때 L 은 문제의 크기를 고려하여 $(JN * (JN-1) * MN) / 2 * 10$ 으로 정하였다. 종료 조건으로서 일정 온도에서 내부루프를 통해서 받아들여지는 새로운 해의 비

용함수가 연속으로 일정 회수만큼 반복되면 종료하는 것으로 하였고 본 논문에서는 50 회로 고정하였다.

비교적 작은 규모의 문제로 <9J-8M 문제>(9Job-8Machine)를, 보다 큰 규모의 문제로 <12J-11M 문제>(12Job-11Machine)를, 사이클이 복잡하게 얽혀 있는 문제로 <12J-6M 문제>(12Job-6Machine)를 예로 들어 교착 배제 일정계획을 적용하였다. <9J-8M 문제> 는 9 가지의 부품이 가공되어야 하는 문제인데 부품 j_1, j_2, j_3 는 7 개의 작업으로 구성되어 w_1 의 작업 경로를 따르고, 부품 j_4, j_5, j_6 은 4 개의 작업으로 구성되어 w_2 의 작업 경로를 따르며, 부품 j_7, j_8, j_9 는 5 개의 작업으로 구성되어 w_3 의 작업 경로를 따른다. <12J-11M 문제>는 12 가지의 부품이 가공되어야 하는 문제인데 부품 j_1, j_2, j_3 는 6 개의 작업으로 구성되어 w_1 의 작업 경로를, 부품 j_4, j_5, j_6 은 9 개의 작업으로 구성되어 w_2 의 작업 경로를 따르며, 부품 j_7, j_8, j_9 는 4 개의 작업으로 구성되어 w_3 의 작업 경로를 따르고, 부품 j_{10}, j_{11}, j_{12} 는 7 개의 작업으로 구성되어 w_4 의 작업 경로를 따른다. <12J-6M 문제> 는 12 가지의 부품이 가공되어야 하는 문제인데 부품 j_1, j_2, j_3, j_4 는 6 개의 작업으로 구성되어 w_1 의 작업 경로를, 부품 j_5, j_6, j_7, j_8 은 6 개의 작업으로 구성되어 w_2 의 작업 경로를 따르며, 부품 $j_9, j_{10}, j_{11}, j_{12}$ 는 6 개의 작업으로 구성되어 w_3 의 작업 경로를 따른다. 각 문제에서 각 부품들의 작업 경로와 관련된 사이클들을 정리하면 Table 2 와 같다. 여기서 각 문제 유형에서의 작업 경로, 기계, 1 차 레벨 사이클, 2 차 레벨 사이클이 표기상으로 중복되지만 각 문제 유형은 여타 문제 유형과는 독립적이다.

세 가지 유형의 문제에 대하여 교착회피 정책을 적용하여 일정을 최적화한 결과는 Table 3 과 같다. <12J-6M 문제>에서는 RP4 와 RP5 를 적용했을 때의 효과가 동일하였기 때문에 RP4 에 대해서만 나타내었다. 표에 나타난 수들은 무한 버퍼의 경우와 각 교착 회피 정책을 적용한 항목에 대하여 시행된 실험에서 얻어진 전체 생산시간의 평균으로서 시간의 단위는 분이다. 괄호 안의 수는 표준 편차를 나타내고 본 논문에서의 접근방법이 어느 정도의 신뢰성을 가지고 최적 일정을 찾는지 나타내고 있다. Table 3 에서의 NP (number of perturbation)는 시뮬레이티드 어닐링에서의 이동 연산자와 교환 연산자가 각각 적용된 회수를 나타낸다. 반복 실험에서의 최소값들은 왼쪽 항목란의 괄호 안에 표시하였다.

Table 2 Configuration of three problems

| | part route | machine sequence | related first level cycle | related second level cycle |
|-------------------|----------------|--|---|---|
| <9J-8M problem> | w ₁ | m ₁ - m ₇ - m ₂ - m ₈ - m ₃ - m ₄ - m ₈ | c ₁ , c ₂ , c ₃ | c ₁ ² |
| | w ₂ | m ₅ - m ₈ - m ₆ - m ₇ | c ₃ | |
| | w ₃ | m ₃ - m ₄ - m ₈ - m ₁ - m ₇ | c ₁ , c ₂ | c ₁ ² |
| <12J-11M problem> | w ₁ | m ₁₀ - m ₆ - m ₂ - m ₅ - m ₈ - m ₉ | c ₁ , c ₂ , c ₆ | c ₁ ² , c ₃ ² |
| | w ₂ | m ₉ - m ₅ - m ₆ - m ₂ - m ₇ - m ₁₀ - m ₁₁ - m ₇ - m ₄ | c ₁ , c ₂ , c ₃ , c ₄ , c ₆ | c ₁ ² , c ₂ ² , c ₃ ² |
| | w ₃ | m ₁₀ - m ₆ - m ₂ - m ₄ | c ₁ , c ₅ , c ₆ | c ₁ ² , c ₃ ² |
| | w ₄ | m ₁ - m ₂ - m ₄ - m ₃ - m ₂ - m ₇ - m ₁₀ | c ₃ , c ₄ , c ₅ , c ₆ | c ₂ ² , c ₃ ² |
| <12J-6M problem> | w ₁ | m ₁ - m ₂ - m ₃ - m ₄ - m ₅ - m ₆ | c ₁ , c ₂ , c ₃ , c ₄ , c ₅ , c ₆ , c ₇ , c ₈ , c ₉ | c ₁ ² , c ₂ ² , c ₃ ² , c ₄ ² |
| | w ₂ | m ₆ - m ₃ - m ₁ - m ₅ - m ₄ - m ₂ | c ₁ , c ₂ , c ₄ , c ₅ , c ₆ , c ₇ , c ₈ , c ₉ , c ₁₀ | c ₁ ² , c ₂ ² , c ₃ ² , c ₄ ² |
| | w ₃ | m ₄ - m ₁ - m ₂ - m ₃ - m ₅ - m ₆ | c ₁ , c ₂ , c ₃ , c ₄ , c ₅ , c ₇ , c ₈ , c ₉ , c ₁₀ | c ₁ ² , c ₂ ² , c ₃ ² , c ₄ ² |

Table 3 Simulation results

| | | NP=2 | 3 | 4 | 5 | 6 |
|-------------------|-----------------------|-----------------|-----------------|-----------------|-----------------|----------------|
| <9J-8M problem> | Infinite Buffer(747) | 748.84 (12.88) | 747 (0) | 751.64 (19.06) | 753.52 (18.35) | 757.52 (19.30) |
| | RP2 (1153) | 1153 (0) | 1153 (0) | 1153 (0) | 1153 (0) | 1153 (0) |
| | RP4 (1068) | 1068.7 (2.1) | 1068.28 (1.37) | 1068.14 (0.98) | 1068.42 (1.66) | 1068 (0) |
| | RP5 (836) | 836 (0) | 836 (0) | 836 (0) | 836 (0) | 836 (0) |
| <12J-11M problem> | Infinite Buffer (951) | 951.11 (0.74) | 952.94 (5.32) | 965.76 (11.23) | 967.80 (14.59) | - |
| | RP2 (1605) | 1605.48 (1.30) | 1605.18 (0.76) | 1605.4 (1.2) | 1605.39 (1.18) | 1605.10 (0.62) |
| | RP4 (1524) | 1533.56 (5.92) | 1532.91 (6.66) | 1530.54 (5.50) | 1528.64 (4.42) | 1526.77 (4.05) |
| | RP5 (1239) | 1247.94 (12.09) | 1243.2 (9.13) | 1241.74 (6.44) | 1240.24 (4.31) | 1239 (0) |
| <12J-6M problem> | Infinite Buffer (993) | 1004.22 (9.99) | 1023.16 (28.55) | 1033.44 (44.59) | 1051.83 (40.22) | - |
| | RP2 (2642) | 2642 (0) | 2642 (0) | 2642 (0) | 2642 (0) | 2642 (0) |
| | RP4 (2094) | 2101 (2.91) | 2101.08 (2.61) | 2098.98 (2.99) | 2097.37 (2.76) | 2096.38 (2.84) |

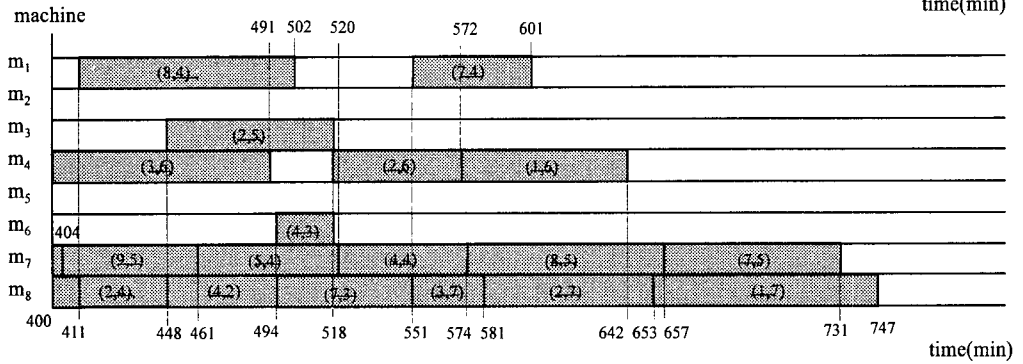
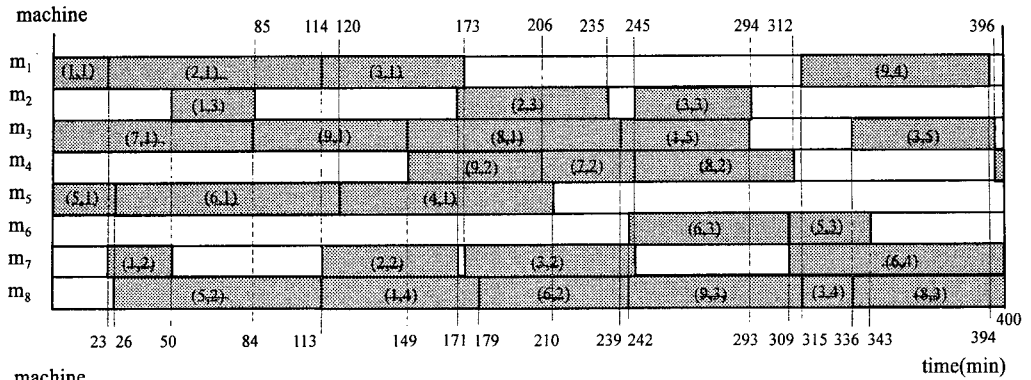
Table 4 Application of LOEM

| | | Large Last Processing Time | Ordinary Last Processing Time |
|-------------------|-----------------|----------------------------|-------------------------------|
| <9J-8M problem> | Ordinary Method | 1411 | 836 |
| | LOEM | 1411 | 794 |
| | Improvement | 0% | 5% |
| <12J-11M problem> | Ordinary Method | 1778 | 1239 |
| | LOEM | 1552 | 1192 |
| | Improvement | 13% | 4% |
| <12J-6M problem> | Ordinary Method | 3025 | 2094 |
| | LOEM | 2333 | 1855 |
| | Improvement | 23% | 11% |

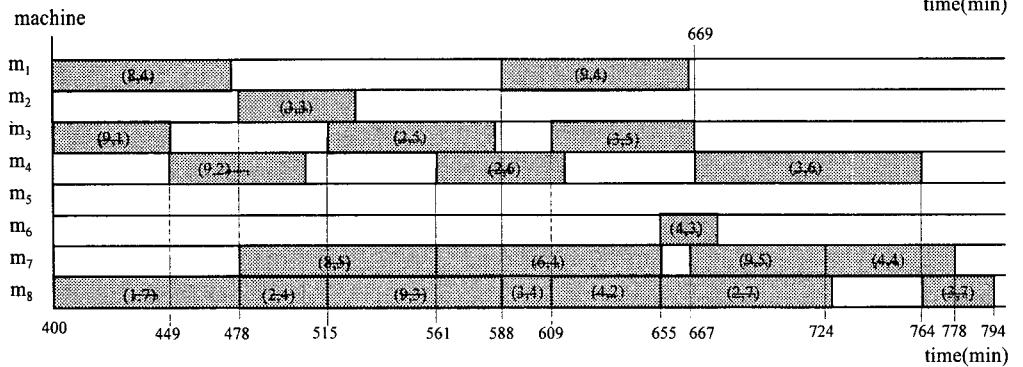
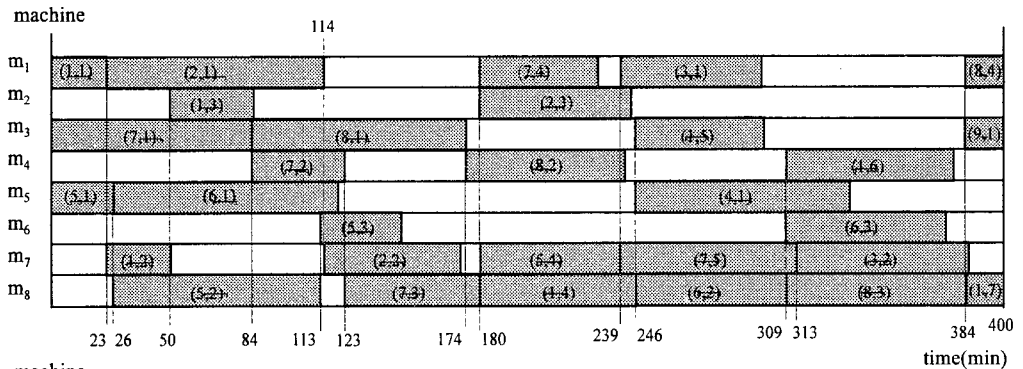
무한 버퍼 일정계획에서는 본 논문에서 사용한 방법으로도 Branch and Bound 알고리즘으로부터 보장되는 최적값과 같은 값을 찾을 수 있었고 교착 회피 정책을 적용한 실험에서도 각 정책의 효과를 반영하는 타당한 결과를 확인할 수 있다. 구해진 일정 중 <9J-8M>문제에 대해서 무한버퍼의 경우와 RP5, LOEM 을 같이 적용한 경우 두 가지 경우의 간트 차트를 Fig. 6 에 나타내었다.

전반적으로 버퍼를 무한하게 가정한 문제에서

는 교란 회수를 작게 하였을 경우 결과가 좋았고 교착 배제 일정계획에서는 시간은 많이 소비하게 되지만 교란 회수를 크게 하였을 경우 결과가 좋게 나오는 경향을 보였다. 이것은 교착 회피 정책을 적용함으로써 어떤 System Clock 에서 실제 할당 할 수 있는 작업들의 수가 제한되어 있기 때문에 키랄당구성에 교란을 가한다 하더라도 그 회수가 적으면 실제 일정은 크게 변화가 없음을 의미한다.



(a) Case of infinite buffer capacity



(b) Case of applying RP5 and LOEM

Fig. 6 Scheduling results of <9J-8M problem>

LOEM 을 RP5 와 함께 적용하였을 경우 전체 생산시간의 감소뿐만 아니라 최적화를 위한 계산 시간도 줄어드는 효과가 있었는데, 교란 회수가 6 회일 경우 <9J-8M 문제>에서는 평균 7.12 초에서 6.66 초로, <12J-11M 문제>에서는 평균 165 초에서 93 초로, <12J-6M 문제>에서는 평균 103 초에서 18 초로 LOEM 을 적용하지 않았을 때에 비하여 각각 계산 시간이 줄어들었다. LOEM 을 적용하면 최종작업이 진행되는 동안에도 하나의 부품을 새로 투입하여 가공할 수 있는 가능성이 있기 때문에 마지막 작업들의 가공시간이 큰 경우 그 적용에 있어 더욱 효과를 볼 수 있다. 세 가지 문제에서의 마지막 작업들의 가공시간을 모두 200 으로 바꾸어 준 새로운 문제에서 RP5 와 함께 LOEM 을 적용했을 때와 LOEM 을 적용하지 않았을 때 일정계획을 하고 전체 생산시간의 최저값을 구한 결과는 Table 4 와 같았다.

표에서 보듯이 <12J-11M 문제>와 <12J-6M 문제>에서는 LOEM 을 적용했을 때 전체 생산시간 감소율의 증가폭이 컸고 사이클이 얽혀있는 정도가 컸던 <12J-6M 문제>에서 더욱 두드러졌다. 그러나 <9J-8M 문제>에서는 예상과는 달리 전혀 향상이 없는데 이것은 시스템 내에서 가공될 수 있는 부품이 비교적 많아서, 마지막 작업에 들어가는 부품을 시스템에서 가공되는 부품 수에서 제외하고 새로운 부품을 투입해서 얻어지는 이익보다, 큰 가공시간으로 인해 뒤따르는 부품들에서 발생하는 블로킹에 의한 시간 손실이 더 크기 때문이다.

6. 결론

물류의 이동이 자동으로 이루어지는 Job-Shop 시스템에서는 교착을 피하면서 일정의 최적화가 이루어져야 한다. 이를 위해 본 연구에서는 기계들의 버퍼가 존재하지 않는 Job-Shop 시스템에서 교착 배제 (deadlock-free)를 보장하는 일정의 최적화에 대하여 다루었다. 최적화 방법으로는 각 작업에 임의의 수를 할당하여 키할당구성을 만들고 시뮬레이티드 어닐링을 이용하는 것을 제시하였고 교착 배제를 보장하는 일정을 얻는 방법으로는 후보 리스트를 생성한 후 System Clock 과 Job Clock 개념의 도입과 교착 회피 정책의 적용을 통해 오프 라인으로 일정계획하는 방법을 제안하였다. 또한 기존의 제안된 교착 회피 정책에 덧붙여 비용함수를 줄일 수 있는 시스템 갱신 방법인 최종작업 배제방법을 사용하고 실험 결과를 비교하였다.

본 논문에서는 3 가지 유형의 시스템에 대하여

기존의 교착 회피 정책과 제안된 최종작업 배제방법을 적용했을 경우의 결과에 대해 분석해 보았는데 최적화의 결과는 제안된 최적화 방법의 타당성을 보여주고 있고, 문제의 유형과 크기에 따라 시뮬레이티드 어닐링을 적용했을 때의 경향이 달라지는 것을 확인할 수 있었다. <9J-8M 문제>는 사이클이 복잡하게 얽혀있지 않은 시스템의 경우 RP5 와 같은 간단한 교착 회피 정책의 도입만으로도 교착을 피하면서 무한 버퍼 경우의 최적값에 근접할 수 있음을 보여 준다. 그러나 <12J-6M 문제>와 같이 사이클이 복잡하게 얽혀 있는 시스템에서는 단순한 교착 회피 정책만으로는 교착 배제 일정계획을 통해 큰 효과를 얻기 어렵다. 제안된 최종작업 배제방법의 적용은 전체 생산시간 뿐만 아니라 최저값을 구해내기까지의 계산 시간 또한 줄일 수 있고, 단순한 시스템보다 <12J-6M 문제>와 같이 사이클이 얽혀있는 정도가 높은 복잡한 시스템에서 더욱 효과적인 것으로 나타났다.

본 논문에서는 기존에 제안된 교착 회피 정책을 사용하여 최적화를 수행하였으나 사용된 회피 정책들은 3 장에서도 살펴 보았듯이 최소 제한적 (least restrictive)이지 않다. 특히 사이클이 복잡하게 얽혀있는 시스템의 경우엔 RP5 마저도 지극히 제한적일 수 있다. 앞으로 기존의 제한 정책보다 덜 제한적인 교착 회피 정책을 개발함과 함께 비용함수를 줄이기 위한 시스템 상태의 갱신 방법에 대한 연구가 요구된다.

참고문헌

- (1) 임성진, 이두용, 1997, "페트리네트를 이용한 유연생산시스템의 다중목표 스케줄링," 대한기계학회논문집(A), 제 21 권, 제 5 호, pp. 769~779.
- (2) Gere, W. S., 1966, "Heuristics in Job Shop Scheduling," *Management Science*, Vol. 13, No. 3, pp. 167~190.
- (3) Banaszak, Z. and Krogh, B., 1990, "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process flows," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 6, pp. 724~734.
- (4) Viswanadham, N., Narahari, Y. and Johnson, T. L., 1990, "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models," *IEEE Transactions on Robotics and Automations*, Vol. 6, No. 6, pp. 713~723.
- (5) Wysk, R. A., Yang, N. S. and Joshi, S., 1991, "Detection of Deadlocks in Flexible Manufacturing Cells," *IEEE Transactions on Robotics and Automations*, Vol. 7, No. 6, pp. 853~859.
- (6) Fanti, M. P., Maione, B., Mascolo, S. and Turchiano, B., 1991, "Event-Based Feedback Control for

- Deadlock Avoidance in Flexible Production Systems," *IEEE Transactions on Robotics and Automations*, Vol. 13, No. 3, pp. 347~363.
- (7) Lawley, M. A., 1999, "Deadlock Avoidance for Production Systems with Flexible Routing," *IEEE Transactions on Robotics and Automations*, Vol. 15, No. 3, pp. 497~509.
- (8) Wu, N., 1999, "Necessary and Sufficient Conditions for Deadlock-Free Operation in Flexible Manufacturing Systems Using a Colored Petri Net Model," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 29, No. 2, pp. 192~204.
- (9) Ramaswamy, S. E. and Joshi, S. B., 1996, "Deadlock-Free Schedules for Automated Manufacturing Workstations," *IEEE Transactions on Robotics and Automations*, Vol. 12, No. 3, pp. 391~400.
- (10) Bean, J. C., 1994, "Genetic Algorithms and Random Keys for Sequencing and Optimization," *ORSA Journal on Computing*, Vol. 6, No. 2, pp. 154~160.
- (11) Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, "Optimization by Simulated Annealing," *Science*, Vol. 220, pp. 671~680.
- (12) Hoitomt, D. J., Luh, P. B. and Pattipati, K. R., 1993, "A Practical Approach to Job-Shop Scheduling Problems," *IEEE Trans. on Robotics and Automation*, Vol. 9, No.1, pp.1~13.
- (13) Kim, J. U. and Kim, Y. D., 1996, "Simulated Annealing and Genetic Algorithms for Scheduling Products with Multi-Level Product Structure," *Computers Ops Res.*, Vol. 23, No. 9, pp. 857~868.
- (14) Jain, A. S. and Meeran, S., 1999, "Deterministic Job-Shop Scheduling : Past, Present and Future," *European Journal of Operational Research*, Vol. 113, pp. 390~434.
- (15) Van Laarhoven, P. J. M., Aarts, E. H. L. and Lenstra, J. K., 1992, "Job Shop Scheduling by Simulated Annealing," *Operations Research*, Vol. 40, No. 1, pp.113~125.