

주문생산방식을 따르는 조립시스템에서의 생산계획 및 일정계획을 위한 알고리즘

Algorithms for Production Planning and Scheduling in an Assembly System Operating on a Make-to-Order Basis

박문원* · 김영대**

Moon-Won Park* · Yeong-Dae Kim**

Abstract

This paper focuses on production planning and scheduling problems in an assembly system operating on a make-to-order basis. Due dates are considered as constraints in the problems, that is, tardiness is not allowed. Since the planning problem is a higher-level decision making than the scheduling problem, the scheduling problem is solved using a production plan obtained by solving the planning problem. We suggest heuristic procedures in which aggregated information is used when the production planning problem is solved while more detailed information is used when the scheduling problem is solved. Since a feasible schedule may not be obtained from a production plan, an iterative approach is employed in the two procedures to obtain a solution that is feasible for both the production planning and scheduling problems. Computational tests on randomly generated test problems are done to show the performance of these algorithms, and results are reported.

1. 서론

기존의 조립시스템은 대량생산을 위한 전용 조립라인이나 수작업에 의존하는 흐름공정의 형태가 대부분이었으나, 근래에 들어서는 소비자들의 다양한 소비 패턴과 기업들간의 경쟁 심화로 인해 다품종의 제품을 주문에 의해 생산하는 조립시스템이 널리 확산되고 있다. 본 연구에서는 이와 같이 다양한 제품들을 주문에 의해 생산하는 조립시스템에 대해 다룬다. 이러한 조립시스템은 몇

개의 작업장들로 구성되며, 각 작업장은 다시 하나 이상의 기계설비 혹은 한명 이상의 작업자들로 구성된다. 또한, 이 조립시스템에서는 외부고객이 사용할 완제품을 생산하는 경우도 있으며, 이 시스템의 상위 시스템에서 필요로 하는 부분품을 생산하는 경우도 있다. 이러한 경우, 각각의 주문에 관한 정보는 생산하여야 할 제품의 종류 및 수량, 그리고 주문의 납기로 구성된다.

여기서 다루는 조립시스템은 전자제품 생산공장 혹은 자동차공장 내에서 쉽게 찾아볼 수 있다. 자동차 녀다운

* 전북대학교 산업공학과

** 한국과학기술원 산업공학과

(knocked-down) 부품들을 포함하는 D자동차의 부품포장공장을 일례로 들면 다음과 같다. (여기서, 자동차 너다운 부품들이란 이들이 조립되어졌을 때 완전한 자동차가 될 수 있는 조립이 가능한 부품들을 뜻한다.) 너다운 부품들은 주로 외국의 현지 자동차 조립 공장들로 운반되어 조립되기 때문에 컨테이너에 꾸러지게 되고, 컨테이너 작업은 화물선의 선적일지(납기)에 맞추어 완료되어야 한다. 이들 너다운 부품들은 화학적·물리적 손상으로부터 보호하고 컨테이너에 담거나 꺼내기 용이하도록, 우선 플라스틱 혹은 방청비닐 주머니나 박스 등에 포장한다. 이렇게 포장된 부품들을 다시 골판지나 나무로 만든 케이스에 포장하고 최종적으로 컨테이너에 싣는다. 이 공장은 다양한 차종의 너다운 부품들을 포장할 수 있는 주머니 포장 작업장, 박스 포장 작업장, 케이스 포장 작업장 및 컨테이너 선적 작업장 등 4개의 작업장들로 구성되어 있으며, 각 작업장은 복수의 기계(장비) 혹은 복수의 작업자들이 병렬로 작업하는 병렬기계공정(parallel machine shop)의 형태를 하고있다.

이 조립시스템이 주문생산방식 하에서 운용될 경우 납기를 지키는 것은 매우 중요한 관리목표가 된다. 예를 들어, 주문의 선적일정이 확정되어있거나 이 시스템에서 생산하는 제품들을 부품으로 사용하는 상위 시스템의 생산계획이 이미 수립되어있으면 납기 지연은 허용될 수 없으며, 납기 지연이 이론상으로 허용될 수 있다고 하더라도 납기 지연비용 자체가 재고유지비용 등 다른 비용들에 비해 매우 큰 경우가 대부분이다. 따라서, 본 연구에서는 주문의 납기는 반드시 지켜져야 한다고 가정한다. 즉, 조립시스템의 생산계획 및 일정계획은 납기계약 하에서 수립된다. 하지만 납기를 지키기 위해 서둘러서 미리 제품을 생산한다면 재고비용이 증가하게 된다. 따라서 본 연구에서는 납기계약 하에서 재고비용을 최소화 하는 것을 목적으로 하는 생산계획 및 일정계획 문제에 대해 다룬다.

이러한 특성들을 가지는 조립시스템에서의 생산계획 및 일정계획 문제는 다음과 같이 정의할 수 있다. 생산계획 문제란 주문의 납기를 어기지 않는 한도 내에서 재고비용을 최소화하기 위하여 각 품목들의 작업장에서의 생산기간(production period)을 결정하는 문제이며, 일정계획 문제는 주어진 생산계획을 바탕으로 각 작업장에 존재하

는 몇 대의 기계들 중 어떤 기계에 얼마만큼의 작업을 할당할 것이며 각 기계에서 생산하여야 할 작업들의 작업순서와 작업시간을 결정하는 문제이다.

납기를 지키는 생산계획을 생성하기 위한 기존의 연구들은 대부분 자원을 납기일부터 거꾸로 할당해 나가는 후진적 기법(backward approach)을 이용한다. 이 후진적 기법을 사용하면 주문들이 납기에 맞추어 완료되거나 납기에 가깝게 완료되므로 재고비용의 발생을 줄일 수 있다는 장점이 있다[4]. 대표적인 후진적 기법으로는 자재소요계획(Material Requirements Planning)이 있는데, 이 기법은 장비의 용량적 제약을 정확히 고려하지 않고 조달기간(lead time)을 주어진 값으로 가정하는 등의 문제점들 때문에 실현 불가능한 생산계획을 생성할 수도 있다[10]. 따라서, 이 생산계획이 실현 가능해지도록 수정하는 용량소요계획(CRP; Capacity Requirements Planning)이 함께 사용되곤 한다[17]. 하지만 용량소요계획은 전진적 기법(forward approach)을 주로 사용하기 때문에 자재소요계획과 용량소요계획은 서로 쉽게 연결되기 힘들다고 알려져 있다[16].

자재소요계획 외에도 후진적 기법을 적용하여 생산계획이나 일정계획을 생성하는 연구결과는 다수 존재한다. Kim[7,8]은 선후 공정 관계가 있는 작업들의 개별공정(job shop) 일정계획 문제를 위한 후진적 기법을 제시하였다. Lalsare and Sen[10]은 작업순서를 후진적 기법을 적용한 우선순위 규칙에 의하여 결정하였다. 최근에는 Park and Kim[12]이 복수의 기계들로 구성된 작업장들로 이루어진 조립시스템에서의 생산계획을 위해 후진적 기법을 기본으로 하는 발견적 기법을 제시하였다.

본 연구에서 다루는 일반적인 조립시스템에서의 생산계획 및 일정계획 문제를 다룬 기존연구는 거의 찾기 힘들다. 하지만, 하나의 작업장이 한대의 기계로만 구성된 조립시스템에서의 일정계획 문제에 관한 연구는 다소 존재한다. Lee 등[11]은 3대의 기계가 2단계로 구성된 조립시스템에서 총 작업소요시간을 최소화하는 문제가 NP-complete임을 증명하고 이 문제를 분지한계법(branch and bound method)을 이용하여 풀었다. Potts 등[15]은 이 문제를 기계가 3대 이상일 때에 대해 확장연구를 수행하였다. 한편, Park and Kim[14]은 기계가 3대 이상인 조립시스템에서 납기를 만족시키면서 재고 비용을 최소화하는

일정계획을 생성할 수 있는 분지한계법을 제시하였다.

조립시스템을 포함하여 대부분의 생산시스템에서는 생산계획 및 일정계획에 관한 의사결정이 순차적으로 이루어진다. 즉, 생산계획이 일정계획보다 상위의 의사결정 사항이기 때문에 전체 시스템의 생산계획 문제를 해결한 후 그 결과를 바탕으로 작업장들의 일정계획을 수립한다. 그러나, 이러한 순차적 방법에는 몇 가지 문제점이 있다. 첫 번째 문제점은 생산계획 문제를 풀어서 생성한 생산계획 자체가 실현 불가능할 수 있다는 것이다. 그 이유는 생산계획을 수립할 때에는 일정계획 단계에서 쓰이는 상세한 정보가 아닌 개략적인 정보를 사용하기 때문이다. 이 경우에는 생성된 생산계획을 만족시킬 수 있는 일정계획 자체가 존재하지 않을 수 있다. 또 다른 문제점은 생성된 생산계획이 사실은 실현 가능하지만 너무 여유가 없어서 이 생산계획으로부터 실현 가능한 일정계획을 흔히 사용되는 일정계획 기법으로는 찾지 못할 수 있다는 것이다.

이러한 문제점들을 피하기 위하여 가장 쉽게 생각할 수 있는 방법이 생산계획 수립 시 작업 시간에 일정 비율의 여유시간을 포함시키는 방법이다. 즉, 생성한 생산계획으로부터 실현 가능한 일정계획을 찾을 수 없다면 작업시간을 조금 증가시켜 전체 시스템의 생산계획 문제를 풀고 그 생산계획으로부터 (원래의 작업시간을 이용하여) 일정계획을 작성해본다. 만일 그렇게 하여도 실현 가능한 일정계획을 찾을 수 없다면 실현 가능한 일정계획을 찾을 수 있을 때까지 작업시간을 조금씩 증가시키면서 이 방법을 반복한다. 그런데, 이러한 순차적반복적 방법을 사용하면 너무 느슨한 생산계획이 생성되기 때문에 재고비용의 증가를 초래할 수 있다.

본 연구에서는 이러한 순차적·반복적 방법의 문제점들을 해결할 수 있는 방안을 제시한다. 이 방안은 기존에 독립적으로 개발된 생산계획 및 일정계획 기법들을 간단한 방법을 통해 하나로 결합시킨다. 이 방안에서는 순차적·반복적 방법들과는 달리 하나의 작업장에 대해서 생산계획을 생성하여 그 결과를 이용하여 일정계획을 작성한 후 실현 가능하다면 다음 작업장에 대해 생산계획과 일정계획을 작성해나가는 절차를 반복하여 전체 작업장들의 생산계획과 일정계획을 수립한다. 만일 생성된 특정 작업장의 생산계획으로 실현 가능한 일정계획을 작성할 수 없다면, 이 작업장에서 생산하는 품목들 중 일부 품목

의 생산기간을 조정하여 일정계획 문제를 푼다. 이러한 조정은 실현 가능한 일정계획을 생성할 때까지 반복된다. 또한, 주어진 생산계획이 너무 느슨해서 필요 이상의 재고 비용을 초래할 경우, 좀 더 비용을 줄일 수 있는 생산계획이 되도록 결과를 수정하기도 한다.

본 논문은 다음과 같이 구성되어 있다. 우선 2절에서는 이 연구에서 다루는 시스템에 대해 전체적인 설명을 하였으며, 다루고자 하는 생산계획 및 일정계획 문제를 명확히 정의하고 이 문제들에서 사용된 몇 가지 가정 및 전제 조건들을 언급하였다. 3절에서는 생산계획 및 일정계획 사이에서 발생할 수 있는 문제들을 해결하기 위한 방안을 제시하고, 4절에서는 이 방안의 성능을 평가하기 위하여 수행한 계산실험 결과를 제시하였다. 마지막 절에서는, 최종 결론과 추후 연구과제에 대해 언급하였다.

2. 대상 조립시스템에서의 생산계획 및 일정계획 문제

2.1 조립시스템 개요

본 연구에서 고려하는 조립시스템은 주문에 의해 생산을 한다. 각각의 주문은 완제품의 종류, 수량 및 납기로 구별된다. 완제품들의 납기는 주어져 있으며 각각 다를 수 있다. 하나의 완제품을 만들기 위해서는 다양한 품목들이 조립되거나 가공되어야 한다. 여기서 품목이란 특정 작업장에서 일련의 공정을 거쳐서 완성되는 부분품, 반제품 또는 완제품을 일컫는다. 본 연구에서는 각 품목의 생산은 하나의 작업장에서만 이루어진다고 가정한다. 동일한 완제품에 필요한 품목들 사이에는 선후공정관계가 존재한다. 즉, 하나의 상위 품목의 생산을 위해서는 몇 개의 하위 품목들이 필요하며, 하위 품목들의 생산은 상위 품목의 생산시작시점까지 완료되어야 한다. 따라서, 본 연구에서는 완제품을 제외한 품목들의 납기는 그 품목의 상위 품목의 생산시작시점으로 설정한다. 납기 전에 완성된 품목은 납기까지 재고 재고로 대기한 후 다음 작업장에 투입된다. 완제품의 경우는 납기일까지 완제품 저장창고에 저장된 후 고객에게 인도된다.

본 연구에서 고려하는 조립시스템은 그림 1과 같이 몇 개의 작업장들로 구성되며 하나의 작업장에는 몇 대의 기계들 혹은 몇 명의 작업자들이 병렬로 존재한다. 각 작업

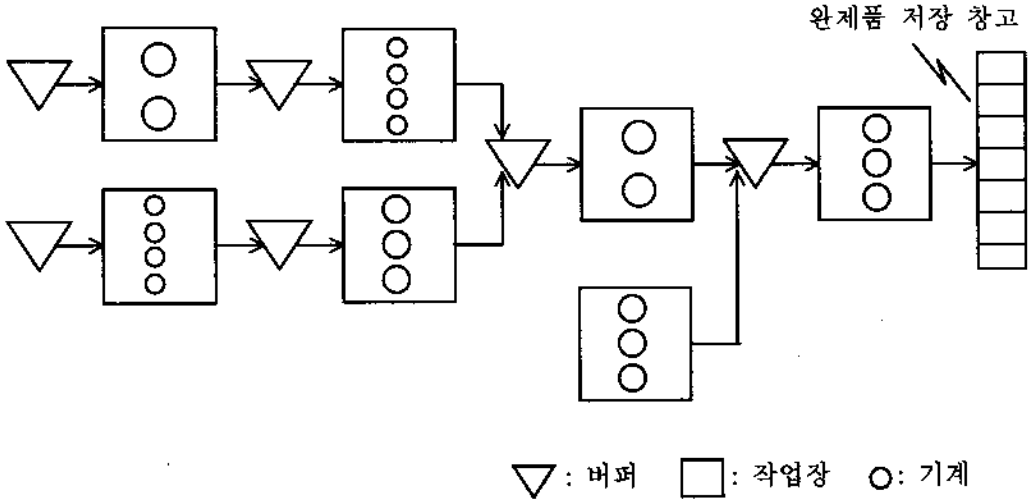


그림 1. 조립 시스템의 일례

장의 가용시간은 생산계획 기간에 포함되는 모든 단위기간동안 일정하다고 가정한다. 작업장들 간의 물류는 최하위 품목을 만드는 최하위 작업장(most upstream workstation)들부터 최상위 품목(완제품)을 만드는 최상위 작업장(most downstream workstation)으로 흐른다. 이 논문에서는 편의상 상위 품목을 생산하는 작업장이 더 작은 작업장 번호를 가지도록 작업장 번호를 할당한다. 예를 들어 K 개의 작업장이 조립시스템에 존재한다면, 작업장 1이 최상위 작업장이 되며, 작업장 K 가 최하위 작업장이 된다. 즉 작업장 $k+1$ 에서 작업장 k 로의 물류는 있을 수 있으나, 작업장 k 에서 작업장 $k+1$ 로의 물류는 존재하지 않는다.

2.2 생산계획 문제

일반적으로 생산계획 문제에서는 생산계획 기간을 1주나 1개월 등으로 생각하는 경우가 많으며, 이 기간은 1교대 혹은 1일 등의 단위기간으로 이루어져 있다. 따라서, 생산계획 문제란 주어진 주문들을 만족시키기 위하여, 각 단위기간 당 각 작업장에서 생산하여야 할 품목들의 양을 결정하는 문제이다. 이를 다른 관점에서 보면, 각 품목의 생산시작시점과 생산완료시점을 결정하는 문제이다. 본 연구에서의 생산계획 문제에서는 각 품목의 로트 크기는 그 품목에 해당하는 완제품의 주문량을 모두 생산하는데 필요한 그 품목의 생산량과 동일하다고 가정한다.

(하지만 일정계획 문제에서는 이 가정을 사용하지 않는다.) 또한, 최상위 품목(완제품)들은 최상위 작업장에서 생산된다고 가정한다.

이 문제를 보다 명확히 정의하기 위하여 수리적으로 모형화 하면 다음과 같은 혼합 정수계획이 된다. (수리모형의 제시에 앞서 이 모형에서 사용한 기호를 먼저 정의하도록 한다.)

- i 품목 ($i = 1, 2, \dots, I$)
- k 작업장 ($k = 1, 2, \dots, K$)
- t 단위기간 ($t = 1, 2, \dots, T$)
- $\phi(i)$ 품목 i 의 상위품목
- I_k 작업장 k 에서 생산되는 품목들의 집합 (I_k : 최상위 품목들의 집합)
- p_i 품목 i 를 생산하는데 필요한 작업부하 (실제 작업시간을 해당 작업장의 단위기간 당 가용시간으로 나눈 값)
- u_i 품목 i 의 납기 ($i \in I_1$)
- h_i 단위기간 당 품목 i 의 재고유지비용
- x_{it} 단위기간 t 에 품목 i 를 생산하기위해 할당된 작업부하
- y_{it} $x_{it} > 0$ 이면 1' 아니면 0의 값을 가지는 0-1 변수
- s_i 품목 i 의 생산시작시점

f_i 품목 i 의 생산완료시점

$$\begin{aligned}
 \text{(P) Minimize } & C_h(s) = \sum_{i \in I_1} h_i(s_{\phi(i)-s_i}) + \sum_{i \in I_2} h_i(u_i - s_i) & (1) \\
 \text{subject to } & \sum_{i \in I_1} x_{it} = \bar{p}_t & \text{for all } t & (2) \\
 & \sum_{i \in I_2} x_{it} \leq 1 & \text{for all } k, t & (3) \\
 & f_i \leq u_i & \text{for all } i \in I_1 & (4) \\
 & f_i \leq s_{\phi(i)} & \text{for all } i \in I_1 & (5) \\
 & y_{it} \geq x_{it} & \text{for all } i \text{ and } t & (6) \\
 & s_i \leq t y_{it} + T(1 - y_{it}) & \text{for all } i \text{ and } t & (7) \\
 & f_i \geq (t+1) y_{it} & \text{for all } i \text{ and } t & (8) \\
 & 0 \leq x_{it} \leq 1 & \text{for all } i \text{ and } t & (9) \\
 & y_{it} \in \{0, 1\} & \text{for all } i \text{ and } t & (10) \\
 & s_i, f_i \in \{1, 2, 3, \dots\} & \text{for all } i & (11)
 \end{aligned}$$

식 (1)은 최소화하고자 하는 목적함수로서 각 작업장에서 발생하는 재고유지비용의 합을 나타낸다. 식 (2)는 각 품목이 생산계획기간 내에 모두 생산되어야 한다는 제약 식이며, 식 (3)은 각 작업장의 단위기간 당 작업용량의 제약을 나타낸다. 식 (4)는 완제품의 납기가 반드시 지켜질 수 있도록 하기 위한 제약식이며 식 (5)는 품목들 간에는 선후공정관계를 나타낸다. 식 (7)과 (8)은 각각 품목들의 생산시작시점과 생산완료시점을 표현하기 위한 제약식이다. 이 식들은 식 (6)과 더불어 $t < s_i$ 이거나 $t \geq f_i$ 일 때 x_{it} 의 값이 0이 되도록 해준다.

이 생산계획 문제에는 과거에 이미 생산계획이 이루어졌으나 아직 생산이 완료되지 않은 주문이 포함되어 있을 수 있다. 이러한 주문들은 다음과 같은 방법으로 고려할 수 있다. 첫번째 방법은 이 주문들이 과거에 작성된 생산계획에 따라 생산될 수 있도록, 수리모형에서 그 주문에 해당하는 결정 변수들의 값을 고정시키고 문제를 푸는 방법이다. 다른 방법으로는 과거에 작성된 생산계획은 무시하고, 이 주문들과 새로 추가되는 주문들을 함께 고려하여 새로운 생산계획을 작성하는 방법이 있다. 후자의 방법은 rolling horizon으로 생산계획을 수립하는 방법들에서 흔히 사용되며, 본 연구에서도 이 방법을 따르도록 한다.

2.3 일정계획 문제

일정계획 문제란 생성된 생산계획이 실현되도록 각 작

업장에서 조립 혹은 가공하여야 할 작업들을 기계에 할당하고 각 기계들에서의 작업순서와 작업시간을 결정하는 문제이다. 따라서, 이 문제는 해당 작업장의 생산계획이 주어진 후에 풀리게 된다. 하나의 품목은 생산 수량에 따라 몇 개의 배치(batch)들로 나뉘게 되고, 하나의 배치는 한 기계에서 처리된다. 이 연구에서는 각각의 배치를 작업(job)이라 부른다. 여기서 각 작업은 작업시작 가능시간(ready time)과 납기(due time) 사이에 처리되어야 한다. 어떤 작업의 작업시작 가능시간이란 그 작업에 필요한 원자재 및 부품이 모두 도착한 시점을 뜻하는데, 이 시점은 생산계획 문제에서 그 품목의 생산에 필요한 원자재와 하위 부품들이 모두 도착하는 ‘생산시작시점’에 대응한다. 작업의 납기는 상위 작업의 생산이 시작되기 전까지 완료되어야 하므로, 그 품목의 상위 품목의 생산 시작 시점에서 이동 시간을 뺀 값이 된다. 단, 하나의 품목이 몇 개의 배치 즉 작업들로 나뉘어 생산될 수 있는데, 이 때 그 작업들의 작업시작 가능시간과 납기는 모두 동일하다. 모든 작업이 각 작업의 작업시작 가능시간과 납기 사이에 수행되면 작업들간의 선후공정관계는 만족된다. 결국 조립시스템에서의 일정계획은 각 작업장에 대해 독립적으로 생성된 일정계획들의 집합으로 표현될 수 있다.

각 작업장이 복수의 기계들이 병렬로 구성되어 있으므로, 각 작업장에서의 일정계획 문제는 작업들의 작업시작 가능시간과 납기가 주어진 병렬 기계 공정(parallel-machine shop)에서의 일정계획 문제가 된다. 이 일정계획의 목적함수는 앞에서 언급한 바와 같이 작업들의 재고유지비용의 합을 최소화 하는 것이다. 각 작업의 재고유지비용은 그 작업이 작업장에 투입되어 재공 재고로서 머무는 시간, 완제품에 대한 작업의 경우는 완성되어 납기까지 대기하는 시간에 비례한다.

3. 조립시스템에서의 생산계획 및 일정계획 방안

3.1 생산계획을 위한 알고리즘

본 연구에서 고려하는 생산계획 문제의 목적함수는 재고비용을 최소화하는 것이므로, 각 품목들의 생산을 가능한 한 늦게 시작할수록 더 좋은 생산계획을 얻을 수 있

다. 따라서, 각 주문을 납기에 맞추어 생산하거나 납기에 가깝게 생산할 수 있도록 고안된 후진적 기법을 사용하는 것이 바람직하다. 본 연구에서도 Park and Kim[12]의 *network-based algorithm* (NBA)라 불리는 후진적 기법을 적용하여 이 문제를 위한 생산계획을 생성한다.

NBA는 작업장 1(최상위 작업장)부터 작업장 K (최하위 작업장)까지의 생산계획을 차례로 생성한다. 먼저 작업장 1의 초기 생산계획을 간단한 휴리스틱 알고리즘을 이용하여 생성한 후, 이 작업장에서 생산하여야 할 품목들의 생산이 납기, 선후공정관계 및 용량적 제약을 만족하는 범위 내에서 최대한 늦게 시작하도록 이 생산계획을 수정한다. 즉, 다른 품목들의 생산계획을 변화시키지 않고 생산시작시점을 늦출 수 있는 품목들을 찾아서 이 품목들의 생산시작시점을 증가시킨다. 이 때, 재고비용이 가장 많이 감소될 것으로 기대되는 품목들부터 먼저 생산시작시점을 증가시킨다. NBA는 작업장 1의 생산계획이 더 이상 개선될 수 없을 때까지 이 절차를 반복한 후 작업장 2, 작업장 3, ..., 작업장 K 순으로 이를 반복하여 전체 생산계획을 생성한다. 이 알고리즘에서는 각 작업장의 생산계획을 네트워크로 표현하며, 최대 흐름 네트워크 문제(maximum flow network problem)를 풀어서 그 결과를 이용하여 해를 개선한다.

3.2 일정계획을 위한 알고리즘

생산계획 결과 각 품목들의 생산시작시점과 생산완료시점이 결정되면 이를 바탕으로 일정계획을 생성한다. 앞에서도 언급한 바와 같이 각 작업장에서의 일정계획 문제는 작업시작 가능시각과 납기를 고려하는 병렬 기계 공정에서의 일정계획 문제가 되는데, 이에 대한 기존 연구는 그리 많지 않다[1,2,3,13]. 본 연구에서는 이 문제를 해결하기 위해서 Park and Kim[13]이 제안한 탐색 알고리즘들 중 simulated annealing (SA)을 이용하도록 한다. 이 기법은 근래에 들어 각광을 받고 있으며, 해결하기 힘든 조합최적화문제(combinatorial optimization problem)들에 적용되어 좋은 결과를 주는 것으로 알려져 있다[5,6,9].

3.3 실현 가능한 생산계획과 일정계획의 생성을 위한 방안

3.1 및 3.2 절에서 제시한 방법들은 각각 생산계획 문

제와 일정계획 문제를 해결하기 위한 독립적인 알고리즘들이다. 따라서, NBA를 이용하여 생산계획 문제를 풀고 그 결과를 입력자료로 하여 SA를 이용하여 일정계획 문제를 해결하면 전체적인 생산계획 및 일정계획을 얻을 수 있다. 그러나, 이러한 절차로 생산계획 및 일정계획을 작성할 경우, 실현 불가능한(infeasible) 계획이 생성되거나 생산계획이 너무 여유가 없어서 실현 가능한 일정계획을 그 생산계획으로부터 찾기 힘들 수도 있다. 그 이유는 생산계획 문제는 일정계획 생성 시 사용하는 상세한 사항들을 모두 고려하지 않고 축약된 정보를 이용하기 때문이다. 일례로 어떤 작업장이 n 대의 기계 혹은 n 명의 작업자들로 구성되어 있다면 생산계획 단계에선 마치 n 배의 능력을 가진 한대의 기계 혹은 한명의 작업자로 구성된 것처럼 고려한다. 따라서, 각 작업장에서의 품목의 작업처리시간(processing time) 및 작업장에서의 대기시간(queuing time)을 정확히 추정하기 힘들다. 또한, 실제 생산에서는 각 품목별 주문량은 몇 개의 배치로 나누어지지만 생산계획에서는 주문량 전체를 최소 작업단위로 고려하기도 한다.

이러한 문제점들을 극복하기 위해서 근시안적이지만 간단하게 사용할 수 있는 방법은 생산시스템의 가용능력을 조금씩 줄여가면서 실현 가능한 계획을 얻을 때까지 생산계획 문제와 일정계획 문제를 반복적으로 푸는 방법이다. 즉, 생산계획 문제의 식 (2)를 $\Sigma x_k = p(1 + \rho)$, for all i 로 수정하거나 식 (3)을 $\Sigma x_k \leq 1 - \rho$, for all k, i 로 수정하여 생산계획 문제를 풀고 그 결과를 이용하여 일정계획 문제를 푸는 것이다. 이 때, 여유율 ρ 를 0부터 시작하여 조금씩 증가시켜 나간다. 이 방법을 정리해보면 다음과 같다.

방안 0

단계 0: ρ 값을 0으로 설정한다.

단계 1: 생산계획 문제의 수리모형에서 식 (2)를 $\Sigma x_k = p(1 + \rho)$, for all i 로 수정하고 NBA를 이용하여 생산계획 문제를 푼다.

단계 2: 단계 1에서 생성된 생산계획 결과를 입력자료로 하여 각 작업장에서의 일정계획 문제를 생성하고 SA를 적용하여 이 문제들을 푼다.

단계 3: 모든 작업장에 대해 실현 가능한 일정계획이

수립되면 종료한다. 그렇지 않으면 ρ 값을 0.1 만큼 증가시킨 후 단계 1로 간다.

방안 0과 같이 독립된 생산계획 및 일정계획 알고리즘을 순차적·반복적으로 적용하는 방법론은 생산/일정계획을 위한 상업용 패키지들과 시스템에 맞게 직접 구현된 소프트웨어들에서 쉽게 찾아볼 수 있다. 그런데, 방안 0을 이용하면 여유율이 증가할수록 단위기간 당 작업장에 걸리는 부하가 줄어들기 때문에, 단계 1에서 생성된 생산계획 결과를 입력자료로 하여 일정계획 문제를 풀 경우 실현 가능한 일정계획을 찾을 수 있는 가능성이 높아지지만, 시스템의 능력을 실제 능력보다 적게 추정하여 생산계획을 생성하기 때문에 결국 생산을 서두르게 되고 따라서 재고 비용의 증가를 초래할 가능성도 높아지게 된다.

본 연구에서는 이러한 순차적·반복적 방법의 비효율성을 개선하기 위하여 2가지 방안을 제시한다. 이 방안에서는 생산계획과 일정계획을 독립적으로 작성하지 않고, 최상위 작업장부터 최하위 작업장까지의 생산계획 및 일정계획을 순차적으로 생성하면서 확정해나간다. 즉, 현재 고려 중인 작업장의 생산계획을 생성한 후 이를 바탕으로 일정계획을 작성하여 그 계획이 실현 가능하면 그 작업장에서의 생산계획과 일정계획을 확정하고 다음 작업장으로 넘어간다. 만일 현재 고려 중인 작업장의 생산계획으로부터 실현 가능한 일정계획을 생성할 수 없다면, 적기에 생산될 수 없는 품목들 중 재고비용이 가장 싼 품목의 생산시작시점을 1 단위기간 만큼 앞당기는 '생산계획의 수정'을 행한 후 이를 바탕으로 다시 일정계획을 생성한다. 생산계획의 수정과정은 실현 가능한 일정계획을 수립할 수 있을 때까지 반복되어진다. 본 연구에서 새로이 제안하는 두 방안의 차이점은, 고려 중인 작업장의 생산계획이 이를 바탕으로 수립된 일정계획에 의해 실현 가능하다고 판명되었을 때, 첫번째 방안에서는 생산계획을 바로 확정하는 반면, 두번째 방안에서는 일정계획 정보를 이용해 생산계획을 개선할 수 있다면 개선한다는 것이다.

각 방안들의 실행 절차를 위해 사용한 기호와 실행 절차에 대한 설명은 다음과 같다.

생산계획을 위한 입력자료

- K 작업장 수
- I_k 작업장 k 에서 생산되어지는 품목들의 집합 (I_k : 완제품 집합)
- $\phi(i)$ 품목 i 의 상위 품목 번호 ($i \in I_k$)
- p_i 품목 i 에 필요한 작업 부하량
- u_i 품목 i 의 납기 ($i \in I_1$)
- h_i 품목 i 의 단위기간 당 재고 유지 비용
- q 1 단위기간의 길이 (시간)

생산계획의 결정변수

- s_i 품목 i 의 생산시작시점
- f_i 품목 i 의 생산완료시점

일정계획을 위한 입력자료

- r_i 품목 i 의 작업시작 가능시각 ($r_i = q(s_{i-1})$)
- d_i 품목 i 의 납기 ($i \in I_1$ 이면 $d_i = q(u_{i-1})$, $i \in I_k$ 이면 $d_i = q(s_{\phi(i)-1}$)
- A_i 품목 i 를 구성하는 작업들의 집합
- t_j 작업 j 의 처리 시간

일정계획의 결정변수

- c_j 작업 j 의 완료시점
- l_i 품목 i 의 투입시간 ($l_i = \min\{c_j - t_j, j \in A_i\}$)

2.3 절에 밝힌 바와 같이, 일정계획 문제에서 각 품목은 생산하여야 할 수량에 따라 몇 개의 작업들로 나뉘어 처리된다. 따라서, 하나의 품목에서 나누어진 작업들은 동일한 작업 시작 가능 시각과 동일한 납기를 갖는다. 즉, 작업 $j(j \in A_i)$ 의 작업 시작 가능 시각과 납기는 각각 r_i 와 d_i 가 된다. 한편, r_i 와 d_i 는 생산계획의 결정 변수인 s_i 에 의해 결정된다. 그런데, 생산계획 문제에서는 이산 시간인 단위기간이 사용되고 일정계획 문제에서는 연속 시간이 사용되기 때문에 이산 시간을 연속 시간으로 변환하여야 한다. 예를 들어, 단위기간이 8시간인 경우, 생산계획 문제에서의 1, 2, 3, ..., i 번째 단위기간의 시작시점은 일정계획 문제에서 0, 8, 16, ..., $8(i-1)$ 시가 된다. 따라서, r_i 값은 $q(s_{i-1})$ 로 설정한다. 같은 이유로, d_i 값은 품목 i 가 완제품이면 $q(u_{i-1})$ 로 완제품이 아니면 $q(s_{\phi(i)-1})$ 로 설정한다.

방안 1

단계 0: k 값을 1로 설정한다.

단계 1: 작업장 k 부터 K 까지의 생산계획을 NBA를 적용하여 생성한다. $k \leftarrow k+1$ 로 둔다.

단계 2: $k > K$ 이면 단계 3으로 가고 아니면 종료한다.

단계 3: 품목 $i \in I_k$ 에 대해 r_i 값과 d_i 값을 앞의 기호 설명에서의 정의대로 설정한다. 작업장 k 의 일정계획을 SA를 이용하여 생성한다. 이 때, 만일 SA가 실현 가능한 해를 발견하면 그 순간 SA의 실행을 멈춘다.

단계 4: 작업장 k 의 생산계획 및 일정계획이 실현 가능하면, 즉 모든 i 에 대하여 " $r_i + t_i \leq c_i \leq d_i, j \in A_i$ "를 만족하면, $k \leftarrow k+1$ 로 하고 단계 2로 간다. 실현 불가능하면 단계 5으로 간다.

단계 5: 작업장 k 의 일정계획 결과, 일정계획 제약을 어기는 작업들, 즉 " $c_i - t_i < r_i$ 혹은 $c_i > d_i, j \in A_i$ "를 만족하는 작업들(j)을 찾는다. 이 작업들을 포함하는 품목들 중 재고 유지 비용이 가장 싼 품목, 즉 품목 i 의 생산 시작 시점 $s_{i, k}$ 을 $s_{i, k-1}$ 로 수정한다.

단계 6: r_i 값을 $q(s_{i, k-1})$ 로 설정하고 작업장 k 의 일정계획을 생성한다. 이 일정계획이 실현 불가능하면 단계 5로 간다. 그렇지 않다면 $k \leftarrow k+1$ 로 하고 단계 1로 간다.

방안 2

단계 0: k 값을 1로 설정한다.

단계 1: $k \leq K$ 이면 작업장 k 의 생산계획을 NBA를 적용하여 생성하고 단계 2로 간다. $k > K$ 이면 종료한다.

단계 2: 품목 $i \in I_k$ 에 대해 r_i 값과 d_i 값을 설정한다. 작업장 k 의 일정계획을 SA 기법을 이용하여 생성한다. 이 때, SA가 실현 가능한 해를 발견하였다고 하더라도 더 나은 일정계획을 찾기 위해서 SA 자체의 stopping condition이 만족될 때까지 탐색을 계속한다.

단계 3: 작업장 k 의 일정계획이 실현 가능하면, 일정계획 결과를 이용하여 생산계획 결과인 품목들 $i \in I_k$ 의 생산 시작 시점 s_i 값들을 $[1/q]$ 로 설정

정, k 를 $k+1$ 로 하고 단계 1로 간다. (단, $[a]$ 는 a 보다 크지 않은 가장 큰 정수 값이다.) 작업장 k 의 일정계획이 실현 불가능하면, 단계 4로 간다.

단계 4: 작업장 k 의 일정계획 결과, 일정계획 제약을 어기는 작업들, 즉 " $c_i - t_i < r_i$ 혹은 $c_i > d_i, j \in A_i$ "를 만족하는 작업들을 찾는다. 이 작업들을 포함하는 품목들 중 재고 유지 비용이 가장 싼 품목, 즉 품목 i 의 생산 시작 시점 $s_{i, k}$ 을 $s_{i, k-1}$ 로 수정하고 단계 2로 간다.

방안 1에서는 현재 고려 중인 작업장의 일정계획을 생성하는 단계에서는 이 작업장의 생산계획 뿐 아니라 하위 작업장들의 생산계획이 미리 생성되어 있다. 따라서, 이 작업장의 생산계획이 수정되면 그 결과는 하위 작업장들의 생산계획에 까지 영향을 미치게 되므로 하위 작업장들의 생산계획도 다시 수정되어야 하는 단점이 있지만, 실현 가능한 일정계획이 찾았을 경우 SA가 더 나은 일정계획을 찾기 위해 탐색을 계속 하지는 않기 때문에 방안 2에 비해 수행 시간이 짧다는 장점이 있다.

4. 계산 실험

앞 장에서 제시한 3가지 방안들의 성능을 평가하기 위해서 본 연구에서는 계산 실험을 수행하였다. 이를 위하여, 조립시스템에 존재하는 작업장들의 수로 4가지 (3, 4, 5 및 6), 주문의 수로 3가지 (3, 5 및 7)와 납기의 분산 정도로 3가지(작음, 보통, 큼) 경우를 고려하여 총 36(4×3×3)개의 실험 문제들을 생성하였다. 이 때, 시스템에 존재하는 작업장들의 수가 다르면 시스템의 구성 역시 달라지게 되는데, 각 작업장 수에 따라 사용한 조립시스템의 형태는 그림 2에 도시하였다. 또한, 각 조립시스템의 형태 별로 하나의 주문을 완성하기 위해 필요한 품목들의 선후 공정 관계는 그림 3에 예시된 구조들 중 임의로 하나를 선택하였다. 그 외의 실험 문제를 위한 입력 자료들은 다음의 방법으로 생성하였다.

- 1) 생산계획에서 1 단위기간은 12시간으로 한다.
- 2) 각 작업장의 기계 수는 3, 4 혹은 5대 중 임의로 하

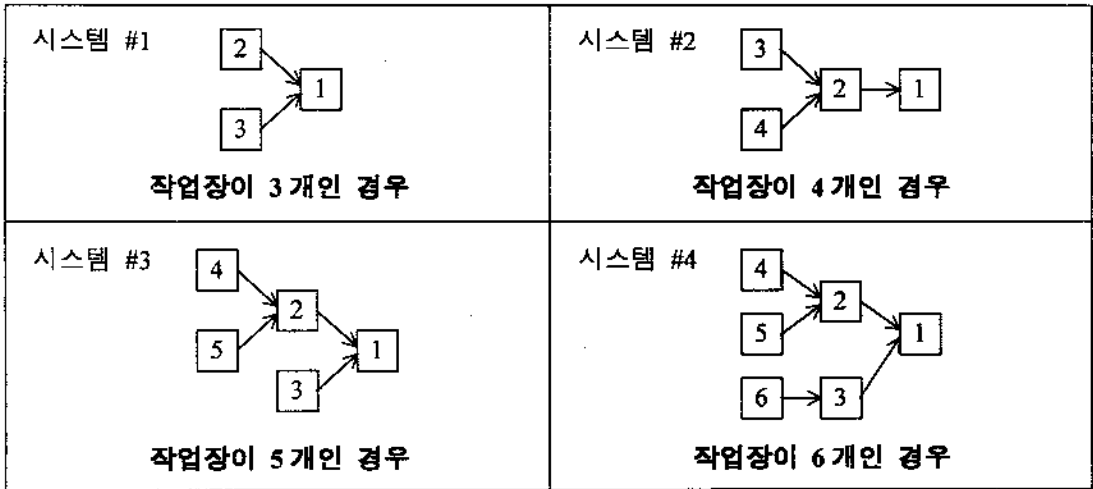


그림 2. 실험 문제에서 사용된 조립 시스템들 (사각형은 작업장을 나타냄)

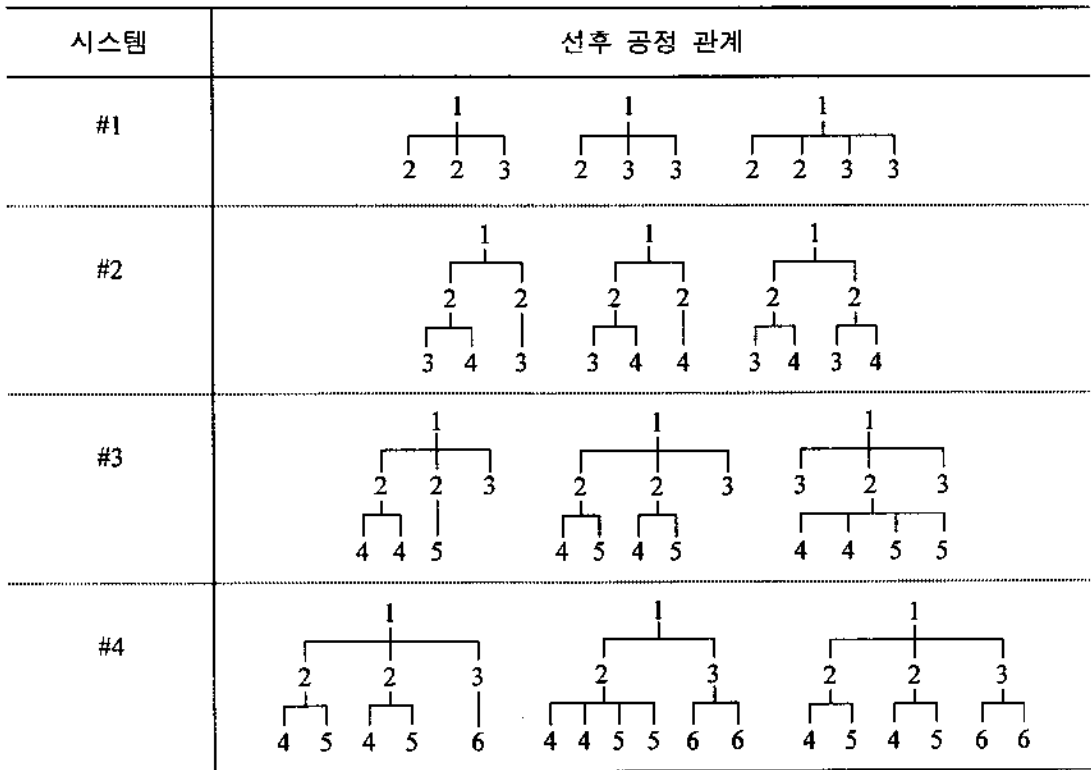


그림 3. 실험 문제에서 사용된 선후 공정 관계 (숫자들은 해당 품목들이 생산되어지는 작업장을 나타냄)

나를 선택한다.

3) 품목의 재고 유지 비용은 그 품목의 하위 품목들의

재고 유지 비용의 합보다는 크게 설정한다. 즉 h_i 는

모든 j 에 대하여 $\sum_{j \in A(i)} h_j + u$ 로 하는데, 이 때, $A(i)$

표 1. 각 방안들에 의해 생성된 생산계획/일정계획의 재고 비용

실험 문제			생산계획/일정계획 생성 방법		
작업장 수	주문 수	납기 분산 정도	방안 0	방안 1	방안 2
3	3	작음	41.4 (0.0)	41.4	41.4
		보통	40.3 (0.0)	40.3	40.3
		큼	42.7 (0.0)	42.7	42.7
	5	작음	70.5 (0.1)	67.1	64.6
		보통	66.7 (0.2)	61.2	61.2
		큼	64.2 (0.1)	62.4	61.2
	7	작음	134.0 (0.2)	120.7	119.3
		보통	125.0 (0.2)	115.7	114.0
		큼	109.6 (0.1)	105.7	105.7
4	3	작음	73.0 (0.0)	73.0	71.4
		보통	73.0 (0.0)	73.0	71.4
		큼	73.0 (0.0)	73.0	73.0
	5	작음	126.5 (0.1)	120.8	120.8
		보통	114.5 (0.1)	113.2	110.5
		큼	117.4 (0.0)	117.4	117.4
	7	작음	261.8 (0.2)	236.5	236.5
		보통	220.9 (0.1)	218.1	218.1
		큼	175.9 (0.0)	175.9	175.9
5	3	작음	86.7 (0.0)	86.7	86.7
		보통	86.7 (0.1)	77.5	77.5
		큼	86.7 (0.0)	86.7	83.1
	5	작음	158.1 (0.1)	150.6	150.6
		보통	147.4 (0.0)	147.4	145.7
		큼	160.3 (0.2)	145.7	145.7
	7	작음	321.7 (0.2)	295.1	288.4
		보통	316.4 (0.2)	285.0	249.9
		큼	297.6 (0.3)	257.6	253.0
7	3	작음	113.5 (0.0)	113.5	113.5
		보통	114.7 (0.0)	114.7	114.7
		큼	115.9 (0.1)	113.5	113.5
	5	작음	188.5 (0.1)	177.8	170.3
		보통	178.6 (0.0)	178.6	178.6
		큼	223.3 (0.2)	199.4	178.6
	7	작음	347.5 (0.1)	334.1	323.5
		보통	307.0 (0.0)	307.0	307.0
		큼	355.9 (0.2)	319.9	307.0

괄호 안의 숫자는 여유율을 나타냄

는 품목 i 의 하위 품목들의 집합을 나타내며 u 는 1과 2 사이의 균등 분포(uniform distribution)로부터 발생시킨 값을 사용한다.

- 4) 각 품목에 필요한 작업량은 $m_i \cdot v$ 로 하는데, 이 때, m_i 는 그 품목이 생산되어지는 작업장의 기계 수를 나타내며 v 는 1/12과 3 사이의 균등 분포로부터 발생시킨 값을 사용한다.
- 5) 각 주문의 납기는 d_{LB} 와 d_{UB} 사이의 이산형 균등 분포(discrete uniform distribution)로부터 발생시킨다. 이 때, d_{LB} 는 모든 품목들에 필요한 작업량들의 합에 가장 가까운 정수 값으로 설정하며, d_{UB} 는 납기의 분산 정도에 따라서 각각 $d_{LB} + 2$, $d_{LB} + 4$ 및 $d_{LB} + 6$ 값들 중 임의로 하나를 선택한다.
- 6) 각 품목들은 $[m_i/2+2]$, $[m_i/2+3]$ 혹은 $[m_i/2+4]$ 개 중 임의로 선택된 수의 작업들로 나뉘어진다. 여기서, m_i 는 그 품목이 생산되어지는 작업장의 기계 수를 나타내며, $[a]$ 는 a 를 반올림한 값을 뜻한다.
- 7) 각 작업의 처리 시간은 해당 품목의 작업량을 시간으로 환산한 값에 그 품목이 포함하는 작업의 수로 나눈 값으로 설정한다.

각 방안에 대한 실험 결과는 표 1과 같다. 이 표에서 알 수 있듯이, 방안 0을 사용할 경우 최고 30%의 여유율을 두어야 하는 경우도 발생하였다. 또한, 본 연구에서 새로이 제안한 방안(방안 1과 2)이 방안 0보다 언제나 좋은 결과를 주며, 방안 2가 방안 1보다 우수한 것을 알 수 있다. 비용 측면에서 볼 경우, 방안 2를 사용하면 방안 0을 사용할 때에 비해 평균 6.2%의 비용 절감 효과를 기대할 수 있으며, 방안 1의 경우는 평균 4.3%의 비용 절감 효과를 기대할 수 있다.

방안 0이 0의 여유율로 실현 가능한 생산계획 및 일정계획을 생성할 수 있다면, 결국 방안 0과 방안 1이 동일한 문제를 푼 결과가 되기 때문에 동일한 결과를 준다. 하지만, 방안 0을 적용한 결과 0보다 큰 여유율이 필요하다면, 이는 생산계획 시 모든 품목들의 작업시간을 증가시킨 것이 되고, 따라서 특정 품목에 대해서만 필요한 최소의 작업부하를 증가시키는 효과를 가지는 방안 1이 더 좋은 결과를 주게 된다.

한편, 방안 2에는 방안 1에 일정계획 결과를 이용하여

생산계획을 향상시키는 절차가 추가되어 있기 때문에 방안 1보다 같거나 더 나은 성능을 보이게 된다. 하지만, 방안 1에서 SA가 실현 가능한 일정계획을 하나라도 찾으면 탐색을 멈추기 때문에, 더 나은 일정계획을 찾기 위하여 탐색을 계속하는 방안 2보다 상대적으로 더 짧은 시간이 소요된다. 비록 표에 제시하지는 않았지만, 본 계산 실험에서는 방안 1의 연산 시간이 방안 2의 연산 시간의 30% 정도에 불과하였다.

표 1에 제시한 값들은 각 방안들이 생성한 실현 가능한 생산계획의 목적함수 값이다. 이 값은 (생산계획에서는 이산 시간을 사용하는 반면 일정계획은 연속 시간을 사용하기 때문에) 일정계획 문제에서의 목적함수 값과는 다소 차이가 있다. 비록 표에서는 제시하지 않았지만 일정계획의 목적함수 값을 비교해보면, 모든 실험 문제들에 대해 방안 2가 방안 1보다 언제나 더 나은 결과를 제공하였다. 그 이유는, 방안 1은 실현 가능한 일정계획을 발견하면 그 순간 SA의 실행을 멈추지만 방안 2는 더 나은 일정계획을 발견하기 위하여 탐색을 계속하기 때문이다.

각 방안들의 실행 시간은 일정계획 알고리즘인 SA 기법의 실행시간에 의해 좌우되었으며, SA 기법의 실행 시간은 이 기법에 포함되어 있는 다양한 모수(제어 변수; parameter)들의 값을 어떻게 설정하느냐에 따라 크게 변할 수 있다. Pascal 언어로 프로그래밍한 후 Pentium 프로세서가 장착된 개인용 컴퓨터에서 실행한 본 연구에서의 계산 실험에서는, 생산계획을 위해 사용한 시간이 어떤 문제에 대해서도 2분 이내였으나, 일정계획을 위해서는 문제에 따라 5분에서 2시간 30분까지의 시간이 필요하였다. 그런데, 본 연구에서 다루는 생산계획 및 일정계획 문제들은 앞서도 언급했듯이 자주 풀어야 하는 문제가 아니기 때문에, 제시한 방안들은 실험 시간에 제약을 크게 받지 않는다고 할 수 있다.

5. 결론

본 연구에서는 주문 생산 방식에 의해 운영되는 조립 시스템에서 실현 가능한 생산계획 및 일정계획을 생성하기 위한 방법을 제시하였다. 지금까지는 주로 방안 0과 같은 독립적인 알고리즘들을 이용한 순차적 반복적인 해결 기법들이 사용되어왔다. 이러한 방법은 재고비용을 과

다하게 발생시키는 경향이 있으므로, 본 연구에서는 조립 시스템의 특성을 고려하여 생산계획과 일정계획을 상위 작업장부터 하위 작업장까지 차례로 확정하여 전체 시스템의 생산계획과 일정계획을 달성하는 2가지 방안을 제시하였다. 이 방안들에서는 한 작업장의 생산계획 결과로부터 실현 가능한 일정계획을 생성할 수 없을 경우, 재고 비용이 되도록 작게 증가하도록 그 작업장의 생산계획을 수정한다. 또한, 수립된 일정계획 결과를 이용하여 주어진 생산계획을 개선하기도 한다. 계산 실험을 통하여 제시된 방안들이 기존에 사용되던 방법, 즉 생산계획과 일정계획을 독립적으로 생성하는 방법보다 우수하다는 것을 보였다.

본 연구에서 제안한 방안들은 조립시스템 내의 각 작업장들이 병렬 기계들로 구성되어 있을 경우에 사용할 수 있다. 하지만, 간단한 수정을 통하여 작업장들이 단일 설비 공정(single-machine shop), 흐름 공정(flow shop) 혹은 개별 공정(job shop)일 경우에 대해서도 이용될 수 있다. 즉, 제시한 방안이 포함된 일정계획 알고리즘은 병렬 기계 공정을 위한 것이지만, 이를 각 공정에 적합한 일정계획 알고리즘으로 바꾸어주면 다양한 형태의 조립시스템에 대해서도 사용할 수 있을 것이다.

참고문헌

- [1] Cheng, T.C.E., "A Heuristic for Common Due-date Assignment and Job Scheduling on Parallel Machines", *Journal of the Operational Research Society*, Vol.40, NO.12, pp.1129-1135, 1989.
- [2] De, P., Ghosh, J.B., and Well, C.E., "Due-date Assignment and Early/Tardy Scheduling on Identical Parallel Machines", *Naval Research Logistics*, Vol.41, NO.1, pp.17-32, 1994.
- [3] Hall, N.G., "Single- and multiple-processor models for minimizing completion time variance", *Naval Research Logistics Quarterly*, Vol.33, pp.49-54, 1986.
- [4] Hastings, N.A.J. and Yeh, C.-H., "Job Oriented Production Scheduling", *European Journal of Operational Research*, Vol.47, NO.1, pp.35-48, 1990.
- [5] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C., "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning", *Operations Research*, Vol.37, NO.6, pp.865-892, 1989.
- [6] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C., "Optimization by Simulated Annealing: An experimental Evaluation; Part II, Graph Coloring and Number Partitioning", *Operations Research*, Vol. 39, NO.3, pp.378-406, 1991.
- [7] Kim, Y.-D., "On the Superiority of a Backward Approach in List Scheduling Algorithms for Multi-Machine Makespan Problems", *International Journal of Production Research*, Vol.25, NO.12, pp.1751-1759, 1987.
- [8] Kim, Y.-D., "A Backward Approach in List Scheduling Algorithms for Multi-Machine Tardiness Problems", *Computers and Operations Research*, Vol.22, NO.3, pp.307-319, 1995.
- [9] Kim, Y.-D., Lim, H.-G., and Park, M.-W., "Search Heuristics for a Flowshop Scheduling Problem in a Printed Circuit Board Assembly Process", *European Journal of Operational Research*, Vol.91, pp.124-143, 1996.
- [10] Lalsare, P. and Sen, S., "Evaluating Backward Scheduling and Sequencing Rules for an Assembly Shop Environment", *Production and Inventory Management Journal*, Vol.36, NO.4, pp.71-78, 1995.
- [11] Lee, C.Y., Cheng, T.C.E., and Lin, B.M.T., "Minimizing the Makespan in the 3-Machine Assembly Type Flowshop Scheduling Problem", *Management Science*, Vol.39, pp.616-625, 1993.
- [12] Park, M.-W. and Kim, Y.-D., "A Heuristic Algorithm for a Production Scheduling Problem in an Assembly System", Technical Report #97-15, Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Yusong-gu, Daejeon 305-701, Korea, 1997.
- [13] Park, M.-W. and Kim, Y.-D., "Search Heuristics for a Parallel Machine Scheduling Problem with Ready

- Times and Due Dates", *Computers and Industrial Engineering*, Vol.24, Nos.3-4, pp.793-796, 1997.
- [14] Park, M-W. and Kim, Y-D., "A Branch and Bound Algorithm for a Production Scheduling Problem in an Assembly System under Due Date Constraints", Technical Report #97-23, Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Yusong-gu, Daejeon 305-701, Korea, 1997.
- [15] Potts, C. N., SevastJanov, S.V., Strusevich, V.A., Van Wassenhove, L.N. and Zwaneveld, C.M., "The Two-Stage Assembly Scheduling Problem: Complexity and Approximation", *Operations Research*, Vol. 43, pp.346-355, 1995.
- [16] Sharma, K., "Adding Intelligence to MRP Systems", *APICS- The Performance Advantage*, Vol.3, NO.3, pp.53-58, 1993.
- [17] Wortman, J.C., Euwe, M.J., Taal, M., and Wiers, V. C.S., "A Review of Capacity Planning Techniques within Standard Software Packages", *Production Planning and Control*, Vol.7, NO.2, pp.117-128, 1996.
-
- 97년 10월 최초 접수, 98년 8월 최종 수정