J. Soc. Korea Ind. Syst. Eng Vol. 37, No. 4 : 98-105, December 2014 ISSN : 2005-0461(print)
http://dx.doi.org/10.11627/jkise.2014.37.4.98 ISSN : 2287-7975(online)

A Study on the Assembly Line Design Tool with a Pre/Post Process

Byoung-Hoon Moon* · Seong-Hoon Choi**†

*Depart. of Technology Convergence, Sangmyung University
**Depart. of Management Engineering, Sangmyung University

전후 처리 기능을 포함하는 조립라인 설계 방법론

문병훈* · 최성훈**†

*상명대학교 기술융복합학과
**상명대학교 경영공학과

According to a simple survey on the current status of the assembly line design, it was found that trial and error methods on the basis of experiences have been used mainly in domestic manufacturing industries, even though there exist a lot of excellent line balancing studies. It seems that more practical researches should be carried out to develop user-oriented line balancing tools especially for small and medium-sized enterprises. This study presents a design of the line balancing tool which can support the line balancing tasks of nonspecialists. The proposed design tool is composed of three major modules: pre-process, line balancing, and post-process. In particular, pre-process and post-process are newly proposed to increase its ease of use. We applied the proposed design to a test problem and test result showed that our practical method may contribute to enhance the efficiency of production operations management.

Keywords : Line Balancing, Pre/Post Process, Redundancy Check Algorithm

1. 서 론

작업장 간의 균형을 맞추는 조립라인 밸런싱 문제(assembly line balancing problem, 이하 ALBP)는 중요한 의사결정 사항으로 반세기 이상 생산운영관리 분야의 연구자와 실무자의 관심을 끌어오고 있다[4, 5, 8, 13]. 조립라인은 많은 인력이 투입되는 프로세스이므로 생산성 향상을 위해서 최대한 효율이 높도록 설계해야 한다.

본 논문의 목적은 현장의 니즈를 반영하여 조립라인의 생산성 향상에 실질적으로 도움이 될 수 있는 라인 설계 방법론을 제안하는 것이다.

Received 10 November 2014; Finally Revised 26 November 2014; Accepted 26 November 2014
† Corresponding Author : shchoi@smu.ac.kr

먼저 조립라인 설계와 관련하여 현황과 니즈를 파악하기 위하여 본 연구자가 생산운영관리 관련 업무 담당자들을 대상으로 직접 간이설문조사(표본 크기 45)를 실시하였다. 라인 설계 방법을 묻는 질문에 대하여 대기업의 경우 81%, 중소기업의 경우 72%가 경험이나 벤치마킹에 의존한다고 답변하였다. 벤치마킹은 주로 기존에 설치되어 있는 라인을 참고로 하는 것으로 경험에 의존하는 방법의 일종이라고 볼 수 있다. ALBP와 관련하여 그동안 수많은 연구 결과가 발표되고 소프트웨어가 개발되었음에도 불구하고, 많은 기업들은 조립라인 설계에 있어서 여전히 경험에 의존하고 있는 것이 현실이다. 이는 라인 밸런싱 연구나 툴에 한계가 있기도 했지만, 조립라인 설계 방법론에 대한 연구 결과를 현장에 접목시키기 위한 노력이 부족했음을 의미하는 것이다.

생산 라인은 신설된 후에 반복적인 검증 활동을 통해 개선이 이루어지거나 현장의 상황에 따라 수시로 변경된다. 그러나 라인 변경에 따른 관리 절차가 있는지에 대한 설문에 대해서 대기업은 53%, 중소기업은 79%가 그렇지 않다고 대답하였다. 라인 변경관리 절차가 없는 이유에 대해서는 라인 담당자의 관심 부족과 절차 미비로 답변하고 있다. 대부분의 국내 기업에서 라인 변경에 대한 관리 절차가 없는 것으로 보아 기업들이 얼마나 기본에 대한 관심이 부족한 상태에서 업무를 수행하고 있는지를 알 수 있다.

조립라인 설계를 위하여 전용 툴을 사용하고 있는지에 대한 설문 결과는 대기업은 36%, 중소기업은 27%만이 사용하고 있다고 답변하였다. 중소기업의 경우 27%가 사용하고 있다고 답변을 하였으나, 사용하는 툴에 대한 답변이 없는 것으로 보아 적극적으로 활용되고 있다고 보기는 어려웠다. 그리고 조립라인 설계 툴의 선택에 있어서 가장 중요한 기준으로 사용자 용이성(55%)을 선택하였는데, 이를 통해 기존 툴들의 사용 편의성에 대한 고려가 부족했음을 알 수 있다.

따라서 기존의 ALBP 연구 결과와 사용 편의성을 고려하는 방법론을 접목시켜 현장에서 용이하게 사용할 수 있는 툴을 개발하면, 현장에서 좀 더 체계적으로 라인 설계 업무활동을 수행하는데 도움이 될 것으로 기대된다.

기존의 ALBP와 관련된 연구는 툴의 전체적인 구성보다는 주로 라인 밸런싱 기법을 위한 연구 위주로 수행되어 왔다. 일반적으로 ALBP에 대한 해법은 최적해 기법과 근사해 기법으로 구분할 수 있다. 전자에는 분지한계기법, 선형 및 정수계획법, 동적계획법, 목표계획법, 네트워크 이론 적용 기법 등이 있고, 후자에는 휴리스틱 기법으로 많은 방법들이 제안되어 왔는데 최근에는 유전자 알고리즘을 중심으로 한 메타 휴리스틱 방법이 주를 이루고 있다고 할 수 있다[4, 5, 11, 12, 13].

ALBP 해결 기능을 포함하고 있는 생산운영관리 관련 소프트웨어로는 IBM[8]의 PLM(Product Life cycle Management) 솔루션과 1999년 일본의 JASI가 개발하여 일본 기업 800여 개 회사에 공급한 TIME PRISM이 있다[1, 10]. 전자는 구성 모듈의 하나인 Process Engineer에서 라인 밸런싱 기능을 지원하고 있으나, IBM 제품의 경우 PLM 솔루션을 구입해야 라인 밸런싱 기능을 사용할 수 있으므로 구입비용 면에서 부담이 되는 중소기업에서는 활용하기는 힘든 제품이다. 한편 TIME PRISM은 현재 대기업을 중심으로 국내 기업에 다수 보급되어 있으며, 산업공학 관련 학과가 설치되어 대학에 보급되어 있는 것으로 파악되고 있다. TIME PRISM의 주요 기능에는 동작 및 시간연구, 개선활동, 레이팅 분석, 표준시간 설정, 비교분석 시뮬레이션, 동영상 편집/녹음, 공정편성 등을 포함하고 있는 것으로 알려져 있다. 하지만, 조립라인 설계 업무에서 활용할 수 있도록 중복이 없는 정확한 선후행도의 작성 지원 기능과 라인 밸런싱 기법의 활용, 라인 밸런싱 이후 변경 기능이 부족한 것으로 파악된다. 그 외에 FLO(Flexible Line Optimizer)라는 라인 밸런싱 툴이 있는데 전자제품 조립라인이 많은 대기업 L사에서 자체로 개발하여 사용하고 있는 것으로 파악되었다.

논문의 구성은 다음과 같다. 제 2장에서는 본 연구에서 제안하는 조립라인 설계 방법론에 대해 기술하기로 한다. 제 3장에서는 적용 예제를 제시한다. 마지막으로 제 4장에서 결론을 기술한다.

2. 조립라인 설계 방법론

여기서는 조립라인 설계에 대해 구체적으로 알아본 후, 본 연구에서 제안하는 조립라인 설계 방법의 주요 구성 모듈인 전처리 모듈, 라인 밸런싱 모듈, 그리고 후처리 모듈에 대해 기술하기로 한다. 먼저 앞으로 사용될 기호와 변수를 아래와 같이 정의하기로 한다.

기호 및 변수 정의 :

m : 조립라인의 요소작업(elementary operation) 개수

n : 이용 가능한 작업장(workstation) 수의 상한. 단, $n \le m$

$M = \{1, 2, \cdots, m\}$: 라인을 구성하는 작업장에 할당해야 할 요소작업의 집합

$N = \{1, 2, \cdots, n\}$: 라인을 구성하는 작업장의 집합

t_i : 요소작업 i의 작업시간($i \in M$)

$P = \{(i, j) | i$는 j의 직접 선행 요소작업이고 $i, j \in M\}$: 직접 선행 요소작업(제 2.2절의 [정의 1] 참조) 집합

$P_i = \{k | (k, i) \in P$, 즉 k는 i의 직접 선행 요소작업$\}$: 요소작업 i의 모든 직접 선행 요소작업 집합

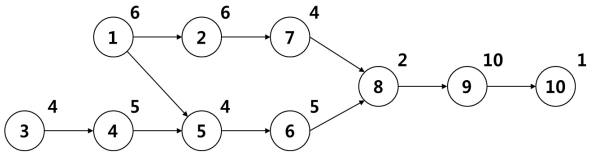
$Q_i = \{k | (i, k) \in P$, 즉 k는 i의 직접 후행작업$\}$: 요소작업 i의 모든 직접 후행 요소작업 집합

S_i : 작업장의 번호를 조립라인의 앞에서부터 순차적으로 부여할 때, 요소작업 i가 속하는 작업장의 번호

2.1 개요

ALBP의 해를 구하기 위해서는 우선 생산 제품을 구성하는 전체 작업을 m개의 요소작업으로 분할하고, 각 요소작업 i의 t_i를 설정한다. 그리고 요소작업들 사이에 존재하는 작업순서 제약을 파악한다. 흔히 선행 제약(precedence constraint)으로 불리며 선후행도로 표현된다[3, 7].

<Figure 1>에 선후행도가 예시되어 있다[4]. 그림에서 원 안의 값은 요소작업의 번호이고 원 우측상단의 값은 요소작업의 작업시간이다.



<Figure 1> Precedence Diagram Example

이렇게 데이터가 준비되면, 작업장 간의 작업시간이 균형을 이룰 수 있도록 m개의 요소작업을 n개의 작업장에 할당한다. 각각의 작업장에서 작업을 수행하는 데 있어서 최대로 허용되는 시간은 모든 작업장에 대해 동일하며, 이 허용시간을 흔히 생산주기 또는 사이클 타임으로 부른다. 목적함수는 n 또는 사이클 타임의 최소화이다.

본 연구에서 제안하는 조립라인 설계 방법론은 앞서 기술한 설문 결과에서 도출된 사용자의 요구 사항을 반영한 것이다. 즉, 사용 편의성을 높여 현장에 쉽게 사용할 수 있는 방법론 개발에 중점을 두었다. 먼저 조립라인 설계 업무에 대해 살펴보면, 아래와 같이 크게 3단계로 구성할 수 있다.

- **단계 1 : 전처리 모듈**
  - -요소작업 구성
  - -요소작업 시간 결정
  - -선후행도 작성
  - -중복 선행조건 체크
  - -선후행도 보완

- **단계 2 : 라인 밸런싱 모듈**
  - -ALBP 해 구하기

- **단계 3 : 후처리 모듈**
  - -해 수정하기

첫 번째 단계는 데이터를 확보하고 체크하는 과정으로 작업시간을 포함하여 제품 생산에 필요한 요소작업을 구성하고 선후행도를 작성한다. 선후행도를 정확하게 작성하기 위해서는 대상 제품에 대한 상세한 지식과 더불어 풍부한 작성 경험이 필요하다. 그러나 중소기업에서는 선후행도를 적절하게 구성할 수 있는 전문가를 보유하기 힘든 부분이 있어, 선후행도에 대한 검증 및 수정 기능을 툴 설계 범위에 포함시켰다.

두 번째 단계인 라인 밸런싱은 다른 툴에서도 활용하고 있는 COMSOAL[8]을 활용하였다. 추가로 특정 요소작업의 경우, 사전에 작업장이 할당된 경우도 있어 이 부분도 라인 밸런싱이 가능하도록 설계하였다.

마지막으로 세 번째 단계는 ALBP 해를 구하는 과정에서 반영하지 못한 다양한 제약사항들을 반영하여 해를 수정하는 과정이다. 그런데 특정 작업장에 할당되어 있는 요소작업의 위치를 변경할 경우, 다른 요소작업과의 선후행 조건을 확인하여 수정해야 한다. 이를 툴에 반영하여 조립라인 설계 담당자가 선후행 조건을 고민하지 않고 수정할 수 있도록 설계하였다. 이제 3단계 구성 모듈의 주요 내용에 대해서 기술하기로 한다.

2.2 전처리 모듈

첫 번째 단계인 전처리 모듈에서는 먼저 제품을 생산하는데 필요한 작업들을 요소작업으로 분할하고 각각의 요소작업에 대한 작업시간 데이터를 확보한 후 선후행도 작성이 이루어진다. 이때 선후행도에는 선행 요소작업을 중복하여 지정하여 오류를 포함하는 경우가 발생할 수 있다. 만일 중복된 선행 요소작업이 존재하면 이를 제거한 후 선후행도를 재구성해야 한다. 먼저 직접 선행 요소작업과 간접 선행 요소작업을 아래와 같이 정의한다. 그리고 중복 선행 요소작업을 정의하기로 한다.

[정의 1] 직접 선행 요소작업과 간접 선행 요소작업

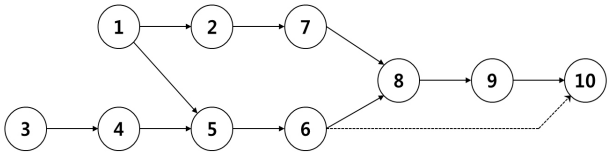
요소작업 i보다 먼저 작업이 완료되어야 하는 선행 요소 작업 중 요소작업 i에 직접적으로 연속되는 요소작업을 '직접 선행 요소작업'이라고 하고, 요소작업 i의 선행 요소 작업 중 간접적으로 연속되는 요소작업을 '간접 선행 요소작업'이라고 한다.

[정의 2] 중복 선행 요소작업

만일 어떤 요소작업 r이 요소작업 i의 직접 선행 요소작업이면서(즉, $r \in P_i$) i의 다른 직접 선행 요소작업의 간접 선행 요소작업이면, r을 i의 '중복 선행 요소작업'이라고 한다.

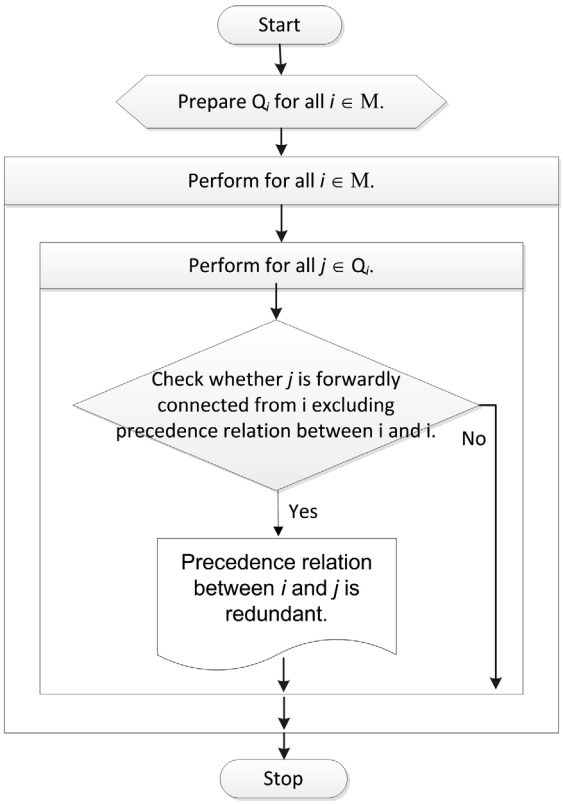
<Figure 2>를 예로 들면, 요소작업 6과 9는 요소작업 10의 직접 선행 요소작업이다(즉, $P_{10} = \{6, 9\}$). 그리고 집합 '$N - \{6, 9, 10\}$'에 속하는 모든 요소작업들은 10의 간접 선행 요소작업이다. 그런데 10의 직접 선행 요소작업인 6은 9의 간접 선행 요소작업이기도 하므로 10의 중복 선행 요소작업이다. 여기서 요소작업 6이 완료되고 나서 다른 요소작업 8과 9의 수행 후에 요소작업 10을 수행하는 것으로 정의되었으나, 요소작업 6의 수행 후에 다른 요소작업의 수행 없이 바로 요소작업 10의 수행으

로 재정의 되고 있어, 불필요한 정의이다. 중복 선행 요소작업의 간단한 판정 방법은 '어떤 요소작업 r이 요소작업 i의 직접 선행 요소작업이면서 동시에 간접 선행 요소작업이면 중복'이다.



<Figure 2> Redundancy in a Precedence Diagram

저자들이 아는 한, 요소작업 간의 선후행 조건의 중복에 대한 고려는 본 논문에서 최초로 제안하는 것이다. 중복 체크 기능은 위의 정의 1과 정의 2를 기반으로 하여 <Figure 3>와 같이 비교적 간단한 알고리즘으로 구현할 수 있다.



<Figure 3> A Simple Redundancy Check Algorithm

이제부터 ALB-Research Group[2]이 제공하고 있는 문제를 이용하여 중복 체크 기능의 효용성에 대해 알아보기로 한다. 참고로 ALB-Research Group[2]은 ALBP 연구와 실무 적용을 지원하기 위해 Scholl이 중심이 되어 운영하고 있는 비영리 웹사이트이다. ALBP와 관련된 다양한 자료를 제공하고 있는데, 특히 유명한 것은 데이터를 포함한 다양한 벤치마킹용 문제와 최적해이다. 그리고 Scholl과 Klein이 제안한 알고리즘인 SALOME에 대한 컴퓨터 프로그램을 제공하고 있다[5, 11]. SALOME는 양방향 분지한계 기법을 기본으로 하고 있으며 단일 작업장, 선행 작업, 목표 생산주기의 세 가지 기본제약 하에서의 ALBP(흔히, 단순 ALBP로 불림)에 대해 가장 우수한 알고리즘으로 알려져 있다[5, 11].

먼저 ALB-Research Group이 제공하고 있는 벤치마킹용 25개의 선후행도(참고로 이들 데이터의 요소작업 개수의 범위는 7~297개이고 요소 작업시간의 범위는 1~5,689로 매우 다양함)에 대해 <Figure 3>의 알고리즘을 적용하여 중복 여부를 체크하였다. 그 결과, 25개 중에서 한 개의 중복이 있는 선후행도가 1개, 그리고 2개의 중복이 있는 선후행도가 3개로 밝혀졌다. 위의 문제들은 ALBP 연구의 권위자들이 국제저명 학술지에 기고한 것들이다. 이를 통해 기존에는 중복 체크를 고려하지 않고 있음을 알 수 있다. 경험과 이론적인 지식이 충분하지 못한 중소기업의 라인 밸런싱 업무 중에는 더욱 많은 중복 현상이 발생되고 있을 것으로 추정된다.

중복이 존재할 경우, 기존 알고리즘의 성능에 어떤 영향을 주는지 알아보기 위해 ALB-Research Group이 제공하고 있는 25개 선후행도를 이용하는 다양한 벤치마킹용 문제들 중에서 요소작업의 크기별로 5가지를 선택하여 실험을 실시하여 <Table 1>의 결과를 얻었다. 각 선후행 데이터에 대해 두 가지의 목표 사이클 타임을 적용하고, 각각에 대해(한 요소작업이 중복 후행 관계를 가질 확률, 한 요소작업이 중복 후행 관계를 가지는 경우 후행 관계가 중복일 확률) 쌍을 3가지로 설정하여 무작위로 문제를 생성하였다. 즉, (2%, 4%), (4%, 8%), 그리고 (8%, 16%)로 3문제씩 적용하여 총 30문제를 구성하였다(표에서 실험 문제의 수가 27개인 이유는 무작위로 문제를 생성하는 과정에서 중복이 발생하지 않아 중복이 없는 원래 문제와 동일하게 되어서 해당 테스트 문제들을 제거했기 때문이다). 표에서 Number of Precedence Relations은 각 문제에 존재하는 총 선행 관계의 개수이고 Proportion of Redundancy는 그 중에서 중복인 선행 관계의 비율이다.

SALOME를 이용하여 <Table 1>의 테스트 문제들을 푼 결과, 모두 ALB-Research Group에서 제공하고 있는 최적해를 구하였으나 표에서 알 수 있듯이 계산시간이 평균 13.28% 증가하였다. 표의 Computational Time Difference는 "(중복 존재 문제 계산시간-중복 비존재 문제 계산시간)/중복 비존재 문제 계산시간×100%"로 계산된 값이다. 통계적으로 유의한 차가 있는지 t-검정을 실시한 결과, p-value가 0.526%로 유의수준 α = 1% 하에서 '중복 선행관계가 존재하는 문제의 계산시간이 더 크다'는 결과를

<Table 1> Test Results for Cases with Redundant Relations

Test Problem No.	Precedence Relation Data Name (Number of Tasks)	Given Cycle Time	Number of Precedence Relations	Proportion of Redundancy(%)	Computational Time Difference(%)
1	ARC83 (83)	3,985	117	3.42	100.00
2			151	25.17	40.00
3		6,883	114	0.88	50.00
4			117	3.42	16.67
5			145	22.07	50.00
6	ARC111 (111)	5,755	184	4.35	0.00
7			202	12.87	-10.00
8		11,570	178	1.12	11.40
9			211	16.59	-1.58
10	BARTHOL2 (148)	109	178	1.69	-4.96
11			184	4.89	-4.32
12			217	19.36	-4.14
13		129	178	1.69	-6.67
14			193	9.33	13.33
15			200	12.50	6.67
16	SCHOLL (297)	1,515	436	2.98	7.78
17			495	14.55	4.11
18			659	35.81	4.45
19		1,883	445	4.94	1.12
20			552	23.37	-0.13
21			890	52.47	-3.53
22	TONGE70 (70)	160	90	4.44	56.25
23			88	2.27	12.50
24			140	38.57	12.50
25		220	91	5.50	3.06
26			90	4.44	0.00
27			102	15.69	4.08
Average					13.28

얻을 수 있었다.

따라서 선후행도에서 불필요하게 중복되어 있는 선후행 관계를 찾아내어 삭제한 이후의 엄밀한 선후행도를 이용하여 ALBP의 해를 구하는 것이 효율적임을 알 수 있다.

2.3 라인 밸런싱 모듈

단계 2의 라인 밸런싱 모듈은 전처리 모듈에서 확보한 데이터를 이용하여 ALBP를 풀어서 각 요소작업을 할당할 작업장을 결정하는 것이다. 본 논문에서는 COMSOAL 기법을 기반으로 목표 사이클 타임이나 고정된 작업장 수를 고려하여 균형 효율이 가장 높은 조립라인 배치를 구성하고자 했다. 참고로 COMSOAL은 컴퓨터를 이용하여 무작위로 요소작업을 선정하여 조립라인을 형성하는 시뮬레이션 기법이다[8]. 본 연구에서는 작업장 수 최소화와 사이클 타임 최소화의 두 접근 방식을 반영하였다. 또한, 특정 요소작업이 사전에 특정 작업장에 할당되어 고정되어 있을 경우도 고려하여 툴을 구성하였다. COMSOAL의 구체적인 수행절차는 많이 알려져 있으므로 본 논문에서는 생략하기로 한다.

2.4 후처리 모듈 Impact

후처리 모듈은 단계 2에서 구한 ALBP의 해를 수정할 수 있는 기능을 제공한다. ALBP 해를 구하는 과정에서 반영하지 못했던 다양한 제약사항들을 반영하여 해를 수정할 수 있게 함으로써 해의 현실성을 높이는 과정이다.

특정 작업장에 할당되어 있는 요소작업을 다른 작업장으로 이동하고자 할 경우, 아래의 [Property 1]과 [Property 2]를 적용하면 요소작업의 선후행 조건을 위반하지 않으면서 이동 가능여부를 쉽게 확인할 수 있다.

[Property 1] S_i에 속해 있는 요소작업 i를 새로이 선행 작업장 S_i'(단, $S_i' < S_i$)로 이동하려고 할 때, '$S_i' < S_k$ 조건'을 만족하는 요소작업 k(단, $k \in P_i$)가 한 개 이상 존재하면, 선후조건 위반이 발생한다.

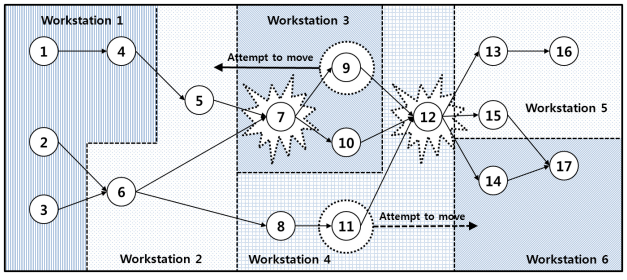
(증명) 선후조건 위반이 발생하지 않으려면, P_i에 속해 있는 모든 k(즉, $k \in P_i$)는 요소작업 i보다 먼저 완료되어야 하므로 $S_i' \geq S_k$이어야 한다. 따라서 $S_i' < S_k$인 요소작업 k가 한 개라도 존재하면 선후조건 위반이 발생한다. ■

[Property 2] S_i에 속해 있는 요소작업 i를 새로이 후행 작업장 S_i'(단, $S_i' > S_i$)로 이동하려고 할 때, $S_i' > S_k$를 만족하는 요소작업 k(단, $k \in Q_i$)가 한 개 이상 존재하면, 선후조건 위반이 발생한다.

(증명) Property 1과 유사하게 증명됨. ■

즉, 직접 선행 요소작업들보다 더 앞쪽(작은) 번호의 작업장으로 요소작업을 옮기려고 할 때, 선후조건 위반이 발생하거나, 또는 직접 후행작업들보다 더 뒤쪽(큰) 번호의 작업장으로 요소작업을 옮기려고 할 때, 선후조건 위반이 발생한다.

Whitehouse[14]가 제시한 ALBP를 이용하여 예로 들기로 한다. <Figure 4>에서 요소작업 9는 현재 작업장 3에 할당되어 있다(즉, $S_9 = 3$). 만일 작업장 3에서 선행 작업장 2로 이동하려고 할 때($S_9' = 2$), 선행 요소작업 7이 작업장 3($S_7 = 3$)에 배치되어 있기 때문에 선후조건 위반이 발생한다. 요소작업 9의 작업장 이동은 불가능하거나, 요소작업 7과 함께 이동해야 한다. 그러나 같이 이동할 경우 균형 효율에 대한 문제가 있으므로 다수의 요소작업을 이동하는 것은 용이하지 않다. 반대로 후행 작업장으로의 이동을 예로 들어보자. 요소작업 11의 후행 작업장 이동하고자 할 때(즉, $S_{11} = 4 \Rightarrow S_{11}' = 6$), 동일 작업장에 있는 후행 요소작업 12로 인해 요소작업 11의 작업장 이동은 불가능하다.

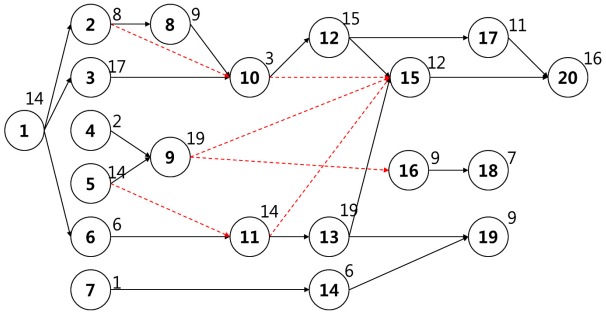


<Figure 4> Precedence Violation Example

3. 적용 예제

3.1 전처리 모듈 적용 예

전처리 모듈의 유효성을 확인하기 위해 <Figure 5>의 선후행도를 이용하여 전처리 모듈을 수행하지 않고 바로 라인 밸런싱 모듈을 수행한 결과와 전처리 작업 후의 라인 밸런싱 수행 결과를 비교하였다.



<Figure 5> Precedence Diagram of the Test Problem

선후행도는 임의로 20개의 요소작업으로 구성하였고, 작업시간은 분단위로 무작위로 구성하였다. 목표 생산주기는 32분으로 설정하였다. 참고로 이해를 돕기 위해 <Figure 5>에서 중복 선행 요소작업은 점선으로 표시하였다.

전처리 작업을 수행하지 않고, COMSOAL의 시뮬레이션 횟수를 1,000회로 설정하여, 라인 밸런싱 모듈을 수행한 결과가 <Table 2>에 제시되어 있다. 총 작업장 개수는 8 개이고, 이때 라인 균형 효율은 94.20%로 계산되었다.

<Table 2> Line Balancing Result Before Applying the Pre-Process Module

Workstations	Cycle time	Elementary operations			
1	28	5	1		
2	19	3	4		
3	26	9	6	7	
4	28	11	14	2	
5	28	13	16		
6	28	19	18	8	10
7	27	12	15		
8	27	17	20		

이제 중복 선행 요소작업의 존재 여부에 따른 라인 밸런싱 결과에 대한 차이를 알아보기 위하여, 전처리 작업을 수행한 후 라인 밸런싱 모듈을 적용하여 ALBP의 해를 구하기로 한다. 전처리 모듈을 수행한 결과, 요소작업 2, 5, 9, 10, 그리고 11이 중복 선행 요소작업으로 파악되었다.

중복 선행 요소작업을 제거한 후, 라인 밸런싱을 실시한 결과를 정리하면 <Table 3>과 같다. 총 작업장 개수는 7개이고, 이때 라인 균형 효율은 97.24%로 계산되었다. 전처리 작업을 수행함으로써 작업장 개수를 1개 줄이면서 균형 효율이 3.04% 향상된 해를 얻을 수 있었다.

<Table 3> Line Balancing Result After Applying the Pre-Process Module

Workstations	Cycle time	Elementary operations			
1	30	5	1	4	
2	31	6	3	2	
3	31	8	9	10	
4	30	12	11	7	
5	30	17	13		
6	31	14	16	19	18
7	28	15	20		

이는 중복 선행 요소작업을 가지고 있는 요소작업 10, 11, 15, 16의 작업장 할당시 유연성이 떨어져서 발생되는 결과라고 할 수 있다. 따라서 우수한 라인 밸런싱 해를 얻기 위해서는 좋은 알고리즘을 사용하는 것뿐만 아니라, 중복 선행 요소작업이 존재하지 않는 정확한 입력 데이터를 확보하는 것이 중요함을 알 수 있다.

3.2 후처리 모듈 적용 예

후처리 작업은 ALBP의 해를 구한 후에 현장의 상황이나 이슈를 고려하여 해로 제시된 작업장의 구성을 변경시키려고 할 때, 특정 작업장에 배치되어 있는 어떤 요소작업을 선행 작업장으로 이동하거나 후행 작업장으로 이동시켜 해를 변경하면 선후조건을 위반하는지를 체크하여 알려주는 기능이다.

먼저, 6번째 작업장에 배치되어 있는 요소작업 14를 선행 작업장 4로 이동을 시도하는 경우를 살펴보자. 요소작업 14의 선행 작업인 요소작업 7의 작업장이 4번째이므로 [Property 1]에 따라 선후조건 위반이 발생하지 않으므로 이동이 가능하나, 작업장 4의 사이클 타임이 34분으로 목표 생산주기에서 2분이 초과됨을 알 수 있다. 이 경우 새로운 해를 적용하려면, 목표 생산주기를 조정하거나 작업개선이나 자동화 등을 통하여 소속 요소작업들의 작업시간을 감소시켜야 할 것이다.

이제 요소작업 7을 가능하면 뒤로 이동하고자 하는 경우를 검토해보기로 한다. 작업장 5나 6으로 이동이 가능하나, 요소작업 7의 후행 작업인 요소작업 14의 경우 $S_{14}=6$이므로 $S_7^{'}=7$은 [Property 2]에 따라 선후행 조건 위반으로 작업장 7로의 이동은 불가능하다.

특정 작업장에 할당되어 있는 요소작업의 위치를 변경할 경우, 다른 요소작업과의 선후행 조건의 위반여부를 파악하기 위해 많은 경우에 대해 검토해야 할 것이다. 그러나 본 연구에서 제안하는 후처리 모듈을 적용하면 간단히 선후행 조건 위반여부를 판단할 수 있으므로 조립라인 설계 툴의 실용성을 높이는데 기여할 수 있을 것으로 예상된다.

4. 결 론

조립라인 설계 업무는 중소기업에서도 생산성 향상을 위해 중요하게 인식하고 있고, 이를 위해 많은 고민들을 하고 있는 중요한 업무이다. 중소기업이 시장에서 살아남기 위해서는 효율적인 생산운영관리를 통해 생산성을 향상시켜 제조원가를 절감해야 하며, 높은 라인 효율의 조립라인 설계와 운영은 이러한 생산운영관리의 시발점이 되는 업무이다.

그러나 본 연구를 통해 수행한 조립라인 설계 업무에 대한 간이설문 결과에서 많은 중소기업들이 기존의 연구결과를 수용하는 체계적인 설계 방법론보다는 주로 과거의 경험에 의존해오고 있으며, 이러한 이유로 기존의 연구들의 한계를 엿볼 수 있었다. 이에 본 논문에서는 기존 라인 밸런싱 연구 결과를 현장에 좀 더 적극적으로 활용하는데 기여할 수 있는 방법론을 제시하고 적용 예제를 통해 유효성을 보였다.

조립라인 설계 업무는 ALBP 문제를 푸는 단계에서 그치지 않고, 선후행도 검토 등의 전처리와 초기해의 조정 기능도 포함하여야 한다. 이를 위해 본 연구에서는 선후행도의 검증을 통해 전문 인력 확보가 힘든 중소기업에서도 손쉽게 선후행도를 구성할 수 있도록 툴을 설계하였고, 산출된 초기해를 쉽게 수정할 수 있도록 요소작업의 작업장 이동시, 이동 가능여부를 판단할 수 있는 후처리 기능을 제안하였다.

추후 연구과제로 본 논문에서 제안하고 있는 조립라인 설계 방법론을 VITAMAX[3, 6]와 같은 생산운영관리 소프트웨어의 제공 기능으로 포함시키는 것을 고려하고 있다. VITAMAX는 엑셀로 요소작업을 관리하고 별도의 데이터베이스를 보유하고 있으므로, 이 데이터베이스를 통해 입력 데이터를 받아, 본 연구의 모듈을 수행하고, 그 결과를 출력 데이터로 구성하여 VITAMAX에서 활용 가능하도록 하면 실용성을 높일 수 있을 것으로 예상된다.

그리고 좀 더 효율이 높은 라인 밸런싱 알고리즘의 도입을 추가 연구과제로 고려할 수 있을 것이다. 본 연구에서는 COMSOAL에 기반을 둔 라인 밸런싱 알고리즘을

적용하였는데, 단순 ALBP에 대해 성능이 가장 우수한 것으로 알려져 있는 SALOME를 기반으로 하는 라인 밸런싱 모듈을 개발하여 도입하는 것을 고려할 수 있을 것이다. 아울러 본 연구에서는 요소작업이 특정 작업장에 미리 할당되는 할당제약을 수용한 라인 밸런싱 모듈을 구현하였는데, '작업장 고정' 할당제약 이외에 '친밀요소작업군', '비친밀 요소작업군', '작업장 형태' 등의 다양한 할당제약[4, 5]을 다루는 라인 밸런싱 모듈을 추가할 수 있을 것이다.

Acknowledgement

This research was supported by the 2013 Research Grant from Sangmyung University.

References

[1] AB&S, Introduction to Time Prism, www.timeprism.co.kr, 2014.

[2] ALB-Research Group, http://www.assembly-line- balancing.de, 2014.

[3] Atworth Co. Ltd., VITAMAX Users' Manual, Seoul : Atworth Co. Ltd., 2010.

[4] Becker, C. and Scholl, A., A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 2006, Vol. 168, p 694-715.

[5] Becker, C. and Scholl, A., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 2006, Vol. 168, p 666-693.

[6] Choi, S.H. and Mo, C.W., A Visual Motion and Time Study Software, VITAMAX. *Journal of the Society of Korea Industrial and Systems Engineering*, 2010, Vol. 33, No. 2, p 33-38.

[7] Gurevsky, E., Battaia, O., and Dolgui, A., Stability measure for a generalized assembly line balancing problem. *Discrete Applied Mathematics*, 2013, Vol. 161, p 37-394.

[8] Hwang, H., Work Study (2nd ed), Seoul : Youngchi Publishers, 2005.

[9] IBM, *Digital Process Plan Solution*, PLM Solution Roadshow, 2009.

[10] JIET, http://www.jiet.co.jp, 2014.

[11] Scholl, A. and Klein, R., SALOME : A bidirectional branch and bound procedure for assembly line balancing. *INFORMS Journal on Computing*, 1997, Vol. 9, p 319-334.

[12] Song, W.S. and Kim, Y.G., Two-Sided Assembly Line Balancing with Preemptive Multiple Goals Using an Evolutionary Algorithm. *Korean Management Science Review*, 2009, Vol. 34, No. 2, p 101-111.

[13] Topaloglu, S., Salum, L., and Supciller, A.A., Rule-based modeling and constraint programming based solution of the assembly line balancing problem. *Expert Systems with Applications*, 2012, Vol. 39, p 3484-3493.

[14] Whitehouse, G.E., IIE Microsoftware : Production Control (IBM PC), Atlanta, GA : Institute of Industrial Engineers, 1983.