Journal of the Society of Korea Industrial and Systems Engineering
Vol. 33, No. 4, pp.45－57, December 2010.

부품 생산과 조립으로 구성된 2단계 조립 일정계획의 Flowtime 최소화 연구

이익선* · 윤상흠** · 하귀룡*** · 전재호****†

*동아대학교 경영학과
**영남대학교 경영학부
***포항공과대학교
****충주대학교 행정정보학과

Heuristic Algorithms for Minimizing Flowtime in the 2-Stage Assembly Flowshop Scheduling

Ik Sun Lee* · Sang Hum Yoon** · Gui Ryong Ha*** · Jaeho Juhn****†

*Department of Business Administration, Dong-A University
**School of Business, Yeungnam University
***POSTECH
****Department of Public Management Information System, Chungju National University

This paper considers a 2-stage assembly flowshop scheduling problem where each job is completed by assembling multiple components. The problem has the objective measure of minimizing total completion time. The problem is shown to be NP-complete in the strong sense. Thus, we derive some solution properties and propose three heuristic algorithms. Also, a mixed-integer programming model is developed and used to generate a lower bound for evaluating the performance of proposed heuristics. Numerical experiments demonstrate that the proposed heuristics are superior over those of previous research.

Keywords : Assembly-type Flowshop, Flowtime, Heuristic Algorithm

1. 서 론

제 2단계 조립 일정계획 문제(2-Stage Assembly Flowshop Scheduling Problem; 이후에서는 2AFS로 지칭)는 한 개의 완제품을 생산하기 위해 필요한 다수의 부품들을 준비하는 제 1단계와, 준비된 부품들을 조립하는 제 2단계가 직렬형태로 연결된 흐름라인 공정(Flowshop)에서 특정한 기준(Objective Measure)을 최적화하는 일정계획(Schedule)을 도출하는 것을 목적으로 한다[1, 2, 7, 10, 11, 13, 14, 15, 17]. 일반적으로, 제 1단계에서 이루어지는 소요부품들의 준비는 문제상황에 따라서 (1) 자체생산, (2) 외주조달(Outsourcing) 혹은 (3) 자체생산과 외주조달의 혼합된 형태로 이루어지는 것을 고려할 수 있는 반면에, 제 2단계에서 이루어지는 부품들의 조립은

논문접수일 : 2010년 08월 30일 논문수정일 : 2010년 11월 20일 게재확정일 : 2010년 12월 01일
† 교신저자 jhjuhn@cjnu.ac.kr
※ 이 논문은 2010년도 충주대학교 교내 학술연구비의 지원을 받아 수행한 연구임.

자체적으로 수행하는 상황만을 고려한다. 또, 현재까지의 2AFS 연구에서는 주로 최대작업완료시간(Maximum Completion Time : Makespan)과 총작업완료시간(Total Completion Time)을 주로 목적함수로 사용하고 있다. 일반적으로 시스템의 단위시간당 생산량(Throughput) 최대화를 추구하는 경우에는 최대작업완료시간 최소화가 목적함수로써 적합한 반면에, 생산시스템 내부의 재공재고(Work-in-process Inventory) 최소화가 필요한 경우에는 총작업완료시간 최소화를 목적함수로 상정하는 것이 적절한 것으로 알려져 있다[3].

현실에서 자주 발견되는 2AFS의 적용사례로는 소방차를 포함한 특수목적차량(Special Purpose Vehicle)이나 산업용 로봇 혹은 기계의 조립생산[1, 2, 10, 15], 주문형 컴퓨터 생산[13], 그리고 유연생산셀(Flexible Manufactu-ring Cell)[14] 등을 들 수 있다. 이와 같이 주로 고객 주문에 적합하게 소요부품을 준비한 후, 이들을 조립하여 완제품을 생산하는 형태의 제조업에서 2AFS가 빈번하게 나타나고 있으므로, 현실적인 관점에서 연구의 중요성이 매우 높다고 할 수 있다.

2AFS 일정계획 문제는 Lee et al.[10]의 소방차 조립생산 일정계획 연구에서 최초로 제시되었다. 동 연구에서 Lee et al.[10]은 두 종류의 부품(혹은 모듈)을 자체적으로 생산한 후, 이들을 조립하여 완제품을 생산하는 문제를 다루었으며, 이때 고려한 목적함수는 최대작업완료시간(Maximum Completion Time)의 최소화이었다. Lee et al.[10]의 연구 이후, 동일하거나 유사한 문제상황에 대해 다수의 후속연구가 진행된 바 있다. Lin et al.[11]은 Lee et al.[10]과 동일한 문제상황에서 제 2단계에 위치한 조립기계가 일괄조립(Batch Assembly)을 수행하는 상황을 상정하여 해법을 제시하였다. Sun et al.[14]은 Lee et al.[10]과 완전히 동일한 문제상황에 대해 보다 효과적인 발견적 해법들(Heuristic Algorithms)을 다수 제시하였다. 또, 전재호[1], 윤상흠, 전재호[2], 그리고 Sung and Juhn [15]은 두가지 형태의 부품을 필요로 하는 2AFS 상황에서 한 종류의 부품은 자체적으로 생산하는 반면, 나머지 종류의 부품은 외주로 조달(Outsourcing)하는 상황을 제시한 후, 문제의 복잡성을 규명하고, 발견적해법을 제시하였다. 이상의 선행연구들은 모두 두 개의 부품을 조립하는 상황을 고려한 반면에, Potts et al.[13]은 제 1단계에서 준비해야 하는 부품의 종류를 일반적인 다수개로 확장한 문제에 대해 발견적해법과 동 해법의 최악오차한계(Worst-case Error Bound)를 제시하였다. Potts et al.[13]에 의해 일반화된 2AFS 문제에 대해, Kovalyov et al. [9]은 Batch Size결정을 추가한 문제에 대해 연구를 수행하였으며, Hariri and Potts[7]는 상당히 큰 규모의 문제까지 최적해를 제공하는 분지한계해법(Branch and Bound Algorithm)을 제시하였다. Koulamas and Kyparisis[8]는 Potts et al.[13]의 문제를 제 3단계 일정계획문제로 확장시킨 후 발견적 해법을 제안하였다.

이상에서 기술한 연구들에서는 최대작업완료시간 최소화를 목적함수로 사용한 반면에, Tozkapan et al.[17]이 유일하게 Potts et al.[13]이 제시한 일반화된 다수 종류 부품 문제상황에 대해 가중총작업완료시간(Weighted Total Completion Time) 최소화를 목적함수식으로 고려하여, 다수의 발견적 해법들과 분지한계법을 제시하였다. 가중총작업완료시간은 총작업완료시간보다 일반화된 목적함수이어서 Tozkapan et al.[17]은 이 연구에서 제시한 해법들을 가중총작업완료시간 최소화문제는 물론 가중치를 고려하지 않은(모든 작업의 가중치를 동일하게 설정한) 총작업완료시간 최소화문제 풀이에도 적용하였다. 기존 2AFS 문제연구 모두가 최대작업완료시간을 최소화를 고려한 반면에, Tozkapan et al.[17]의 연구는 가중/비가중 총작업완료시간 최소화를 한꺼번에 최초로 수행했다는 점에서 큰 의미를 부여할 수 있다. 그러나, 보다 일반적인 목적함수를 상정한 Tozkapan et al.[17]의 해법들은 가중치가 없는 문제에 대해서도, 특별한 수정없이 사용됨으로써 성능이 떨어지는 단점을 노정하였다. 실제로, 일정계획연구에서는 이러한 단점을 회피하기 위해 상대적으로 특수상황(Special Case)인 문제를 먼저 심도있게 연구한 후, 이를 바탕으로 일반상황으로 이행해 가는 접근이 일반적이다.

본 연구에서는 이상에서 언급한 두 가지 상황(① 2AFS 연구에서 총작업완료시간 최소화 연구가 거의 이루어지지 않았음, ② Tozkapan et al.[17]은 가중총작업완료시간 최소화 문제를 상정하여 개발한 해법들로 비가중총작업완료시간 최소화 문제도 그대로 풀이함으로써 불가피하게 해법의 성능저하가 나타남)에 착안하여, 총작업완료시간 최소화를 목적함수로 설정한 후, 동 문제의 발견적 해법 성능향상에 촛점을 맞추었다. 이를 위해, 우선 동 문제의 해법이 갖추어야 할 성질(Solution Property)들을 모색한 후, 이에 기반하여 발견적 해법을 제안하였고, 성능평가를 위해 기존 연구인 Tozkapan et al.[17]의 해법들과 수치실험 비교평가를 수행하였다. 본 연구 결과는 일차적으로 총작업완료시간 최소화 문제에 대한 향상된 성능의 해법을 제안했다는 점과, 부가적으로 Tozkapan et al.[17]의 연구와 더불어 가중총작업완료시간 최소화 해법 개선연구의 토대를 제공한다는 점에서 기여점을 찾을 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 연구대상 문제를 명확히 정의하고 해법의 토대가 되는 몇 가지 성질을 규명한 후 본 문제를 혼합정수계획모형으로 정식화하였다. 제 3장에서는 발견적 해법 개발에 활용

되는 우월성질(Dominant Property)을 규명하고, 이를 기초로 세 개의 발견적해법을 제시하였다. 제 4장에서는 본 연구에서 제안한 발견적해법들과 Tozkapan et al.[17] 해법들의 성능을 수치실험을 통해 비교평가하였고, 마지막 제 5장에서 결론과 추후 연구에 대해 논의하였다.

2. 문제의 정의와 분석

작업시작 시점에 처리해야할 n개의 작업이 모두 대기중이고, 이 n개의 작업 각각은 모두 단계 1에서 m가지 종류의 상이한 부품을 생산한 후(단계 1에서 모든 종류의 부품이 생산완결된 이후에), 단계 2에서 조립하여 완성된다. 단계 1에서 각각의 상이한 부품을 전담하여 생산하는 기계(Dedicated Machine; 편의상 기계로 지칭하지만 작업을 수행하는 인력일수도 있음)가 m대 존재하고 단계 2에서 이들을 조립하는 기계가 1대 존재한다. 제 1단계에서 각각의 부품을 생산하는 기계는 $M_1, M_2, ..., M_m$으로 표기하고, 제 2단계에서 조립을 전담하는 기계는 M_{m+1}로 표기한다. 각 작업의 완료시간은 조립기계 M_{m+1}에서 조립이 완료되는 시점으로 측정된다. 작업 j가 조립기계 M_{m+1}에서 조립을 완결하는 시점은 C_j로 표기한다. 본 연구에서는 n개 작업 각각의 완료시간의 총합을 최소화하는 작업순서를 결정하는 것을 목적으로 한다. 따라서, 목적함수는 다음 식 (1)과 같이 표기된다.

$$\text{Minimize} \sum_{j=1}^{n} C_j \tag{1}$$

Graham et al.[6]에 의해 개발된 일정계획문제 구분표기에 따르면 본 문제는 $2AFS\|\Sigma C_j$와 같이 표기할 수 있다. 본 논문에서는 일정계획문제에서 일반적으로 채택되고 있는 다음과 같은 가정을 전제로 하고 있다[12].

① 모든 작업들은 출발시점에서 처리를 시작할 수 있도록 대기상태에 있다.
② 각 기계에서의 처리시간은 사전에 확정적으로 알려져 있다.
③ 각 기계에서 하나의 작업에 대한 부품생산(부품생산기계)이나 조립(조립기계)이 시작되면 해당 작업에 대한 처리가 모두 끝날 때까지 중간에 중단하지 않는다.
④ 각 기계는 한번에 하나의 작업에 대한 부품생산이나 조립이 가능하다.
⑤ 작업준비시간(Setup time)은 고려하지 않는다.
⑥ 단계1에서 생산된 부품을 단계 2의 조립기계로 운반하는 시간은 무시할 수 있을 만큼 작다고 가정한다.

본 논문에서 추가적으로 사용되는 기호는 다음과 같다.

- j : 작업 $j(j\in\{1,\cdots,n\})$를 표시하는 첨자
- $[j]$: 순서가 결정된 임의 일정계획에서 j번째 처리되는 작업을 표시하는 첨자
- J_j : 작업 j를 의미
- $J_{[j]}$: 임의 일정계획에서 j번째 처리되는 작업을 의미
- $a_{j,u}$: 1단계 부품생산기계 $u(u\in\{1,\cdots,m\})$에서 작업 j에 사용되는 부품생산시간
- b_j : 2단계 조립기계에서 작업 j에 소요되는 모든 부품을 조립하는 시간
- π : 전체작업중 처리순서가 확정된 작업들로 구성된 부분작업순열
- T_k : 부분작업순열 π의 가장 뒤에 처리되는 작업의 기계 k에서의 처리완료시점, $k\in\{1,\cdots,m+1\}$
- $C_{[j,u]}$: 임의 일정계획에서 j번째 작업이 기계k에서의 종결시점, 여기서 $j\in\{1,\cdots,n\}$, $u\in\{1,\cdots,m,m+1\}$.

이상의 기호를 사용하면 식 (1)은 다음과 같이 재표기된다.

$$\text{Minimize} \sum_{j=1}^{n} C_{[j,m+1]} \tag{2}$$

일차적으로 본 문제의 난이도에 대해 다음의 (정리 1)에서 규명하였다.

정리 **1** : $2AFS\|\Sigma C_j$는 강한 NP-Hard 문제이다.

증명 : 본 연구에서 고려하고 있는 문제 $2AFS\|\Sigma C_j$에서 부품생산기계가 1대인 경우만을 고려하면 $F2\|\Sigma C_j$ 문제로 단순화된다. 그런데, 이 단순화된 문제가 이미 강한 NP-Hard 문제임이 규명되어 있다[4]. 따라서 본 문제는 강한 NP- Hard이다.

(정리 1)은 $2AFS\|\Sigma C_j$ 문제의 최적해를 다항시간(Polynomial Time)내에 찾는 것이 거의 불가능함을 의미한다[5]. 따라서, 본 문제에 대해 최적해에 가까운 근사해를 짧은 시간내에 제공하는 발견적해법(Heuristic Algorithm)을 개발하는 것이 현실적인 관점에서 중요한 연구과제가 된다.

정리 2 : 단계 1의 부품생산기계 $M_1, \cdots, M_m$에 유휴시간이 존재하지 않는 일정계획중에서 $2AFS\|\Sigma C_j$의 최적해를 찾을 수 있다.

증명 : 자명하므로 증명은 생략한다.

정리 3 : 모든 기계에서 작업의 처리순서가 동일한 순열일정계획(Permutation Schedule)중에서 $2AFS\|\Sigma C_j$의 최적해를 찾을 수 있다.

증명 : 모든 비순열 일정계획(Non-Permutation Schedule)을 "작업쌍간 교환(Pairwise Interchange)과정"을 통해 순열일정계획해로 변환함으로써 해가 동일하거나 개선됨을 쉽게 보일 수 있다. 유사한 증명이 기존연구(예를 들어, 전재호[1] 혹은 Potts et al.[13])들에 자주 나타나고 있으므로 자세한 증명과정은 생략한다.

위 (정리 2)와 (정리 3)은 탐색해야 하는 해공간(Solution Space)를 줄여주는 역할을 수행한다. 즉, (정리 2)에 의해 비지연일정계획(Nondelay Schedule)들만 탐색하면 충분하게 되고[3], (정리 3)에 의해서, 탐색 대상해의 수가 $(n!)^{m+1}$에서 $n!$개로 대폭 감소하게 된다.

본 문제를 혼합정수계획모형으로 작성하기 위해 제약조건을 차례로 정리하면 다음과 같다. 일차적으로 단계 1의 부품생산이 완결되어야만 단계 2의 조립작업이 가능하므로 다음의 부등식이 성립한다.

$$C_{[j,m+1]} \geq C_{[j,u]} + b_{[j]}, \quad \forall j \in \{1, \cdots, n\}, u \in \{1, \cdots, m\} \quad (3)$$

(정리 2)에 의해 비지연일정계획만으로 고려대상이 축소되었으므로, 다음 등식이 성립한다.

$$C_{[j,u]} = \sum_{i=1}^{j} a_{[i,u]}, \quad \forall j \in \{1, \cdots, n\}, u \in \{1, \cdots, m\} \quad (4)$$

두 번째로 단계 2의 조립기계 자체도 j번째 작업처리가 직전 $(j-1)$번째 작업을 완결한 이후에 가능하므로 다음 부등식이 성립한다.

$$C_{[j,m+1]} \geq C_{[j-1,m+1]} + b_{[j]}, \quad \forall j \in \{2, \cdots, n\} \quad (5)$$

여기서, 여유변수(Slack Variable) $I_j \geq 0 (j \in \{2, \cdots, n\})$와 $W_{ju} \geq 0 (j \in \{2, ..., n\}, \ u \in \{1, \cdots, m\})$를 도입하여 부등식 (3)과 식 (5)를 등식으로 변환하면 다음과 같다.

$$C_{[j,m+1]} = C_{[j-1,m+1]} + b_{[j]} + I_j, \quad \forall j \in \{2, \cdots, n\} \quad (6)$$

$$C_{[j,m+1]} = C_{[j,u]} + b_{[j]} + W_{ju}, \quad \forall j \in \{1, \cdots, n\}, u \in \{1, \cdots, m\} \quad (7)$$

여유변수 I_j는 $(j-1)$번째 처리되는 작업과 j번째 처리되는 작업사이에서 조립기계 M_{m+1}에서 발생하는 유휴시간(Idle Time)을 나타낸다. 또, 여유변수 W_{ju}는 j번째 처리되는 작업에 대해, 각 부품기계 $M_u (u \in \{1, \cdots, m\})$에서 작업이 종결된 후, 조립기계 M_{m+1}에서 조립이 가능해질 때까지 기다리는 시간(Waiting Time)을 각각 나타내고 있다. 여기서, 식 (6)과 식 (7)을 변수 $a_{[j,u]}$, $b_{[j]}$, I_j(혹은 W_{ju})를 사용해서 다음과 같이 표현할 수 있다. 여기서, $C_{[0,m+1]} = \max_{u \in \{1, \cdots, m\}}(a_{[1,u]})$를 사용하였다.

$$C_{[j,m+1]} = \max_{u \in \{1, \cdots, m\}}(a_{[1,u]}) + \sum_{i=1}^{j} b_{[i]} + \sum_{i=2}^{j} I_i, \quad \forall j \in \{2, \cdots, n\} \quad (8)$$

$$C_{[j,m+1]} = \sum_{i=1}^{j} a_{[i,u]} + b_{[j]} + W_{ju}, \quad \forall j \in \{1, \cdots, n\}, u \in \{1, \cdots, m\} \quad (9)$$

위 식 (8)과 식 (9)가 동일한 값을 나타내고 있고, 여유변수 W_{ju}가 비음(Non-negative)이라는 사실로부터 제약식 (8)과 식 (9)(특정 작업에 소요되는 부품이 모두 완성되어야 해당 작업의 조립을 시작할 수 있다는 조건과, 모든 기계는 한번에 하나의 처리만 가능하다는 제약)는 다음과 같은 하나의 부등식으로 표현될 수 있다. 여기서, 표현의 편의를 위해 $\delta_j = \max_{u \in \{1, \cdots, m\}}(a_{ju})$를 사용하였다.

$$\delta_{[1]} + \sum_{i=1}^{j-1} b_{[i]} - \sum_{i=1}^{j} a_{[i,u]} + \sum_{i=2}^{j} I_i \geq 0, \quad \forall j \in \{1, \cdots, n\}, u \in \{1, \cdots, m\} \quad (10)$$

그런데, 식 (10)에서 $j=1$인 경우를 살펴보면 다음과 같이 그 자체로 항상 비음조건이 성립하는 무의미한 제약이 된다.

$$\delta_{[1]} - a_{[1,u]} = \max_{u \in \{1, \cdots, m\}}(a_{[i,u]}) - a_{[1,u]} \geq 0$$

따라서, 제약식 (10)은 다음과 같이 수정된다.

$$\delta_{[1]} + \sum_{i=1}^{j-1} b_{[i]} - \sum_{i=1}^{j} a_{[i,u]} + \sum_{i=2}^{j} I_i \geq 0, \quad \forall j \in \{2, \cdots, n\}, u \in \{1, \cdots, m\} \quad (11)$$

다음으로 식 (2)의 목적함수식을 식 (8)을 이용해 다시표기하면 다음과 같다.

$$\sum_{j=1}^{n} C_{[j,m+1]} = \sum_{j=1}^{n} \left(\delta_{[1]} + \sum_{i=1}^{j} b_{[i]} + \sum_{i=2}^{j} I_i \right) = n\delta_{[1]} + \sum_{j=1}^{n} \left(\sum_{i=1}^{j} b_{[i]} + \sum_{i=2}^{j} I_i \right) \quad (12)$$

이제 변수 $x_{ij}(i=1,\cdots,n;\ j=1,\cdots,n)$를 도입한 후, 식 (11), 식 (12) 및 일정계획의 기본성질을 포함한 실질적인 혼합이진정수선형계획으로 모형화하면 다음 식 (13)에서 식 (18)에 나타난 바와 같다. 여기서, x_{ij}는 작업 i가 전체 작업중 j번째 위치에서 처리된다면 1, 그렇지 않으면 0값을 갖는 이진정수형 변수이다.

식 (12)로부터 다음에 나오는 혼합정수계획모형의 목적식인 식 (13)으로의 유도과정은 다음과 같다. 우선, 설명의 편의를 위해 식 (12)의 두 번째 항을 전개해서 다시 써보면 다음과 같다.

$$\text{식 (12)} = n\delta_{[1]} + \sum_{j=1}^{n}\sum_{i=1}^{j} b_{[i]} + \sum_{j=1}^{n}\sum_{i=2}^{j} I_i. \qquad (12\text{-}1)$$

이해의 편의를 위해 위식의 항들을 하나씩 변환한다. 첫번째 항 $n\delta_{[1]}$에서 $\delta_{[1]}$은 전술한 정의에 따라 "첫번째 처리되는 작업의 부품준비시간중에서 최대값"을 의미한다. 그러나, 작업처리 순서가 결정되지 않은 상태이므로, x_{i1}(작업 i가 최초에 처리되면 1을 갖고, 그렇지 않으면 0을 갖는 이진변수)을 사용해 $\delta_{[1]}$을 표기하면 다음과 같다. 여기서, 당연한 사실이지만 작업 1부터 n까지 모든 작업중 하나의 작업만이 최초에 처리될 수 있음이 반영된다. 즉,

$$\delta_{[1]} = \delta_1 x_{11} + \delta_2 x_{21} + \cdots + \delta_i x_{i1} + \cdots + \delta_n x_{n1} = \sum_{i=1}^{n} \delta_i x_{i1}$$

따라서, $n\delta_{[1]} = n\sum_{i=1}^{n}\delta_i x_{i1} = \sum_{i=1}^{n} n\delta_i x_{i1}$이 성립한다. 다음으로, 식 (12-1)의 두 번째 항을 전개하면 다음과 같다.

$$\begin{aligned}\sum_{j=1}^{n}\sum_{i=1}^{j} b_{[i]} &= b_{[1]} + \\ &\quad b_{[1]} + b_{[2]} + \\ &\quad b_{[1]} + b_{[2]} + b_{[3]} + \\ &\quad \cdots\cdots \\ &\quad b_{[1]} + b_{[2]} + b_{[3]} + \cdots + b_{[n]} \\ &= nb_{[1]} + (n-1)b_{[2]} + \cdots + (n-j+1)b_{[j]} + \cdots + b_{[n]}.\end{aligned}$$

여기서, 위식의 $b_{[1]}$도 이전의 $\delta_{[1]}$과 같은 이유로 다음과 같이 표기할 수 있다. 즉,

$$b_{[1]} = \sum_{i=1}^{n} b_i x_{i1}.$$

이 사실을 위식의 모든 $b_{[j]}$값들에 적용해보면, 다음과 같이 표시할 수 있다.

$$\begin{aligned}&nb_{[1]} + (n-1)b_{[2]} + \cdots + (n-j+1)b_{[j]} + \cdots + b_{[n]} \\ &= n\sum_{i=1}^{n} b_i x_{i1} + (n-1)\sum_{i=1}^{n} b_i x_{i2} + \cdots \\ &\qquad + (n-j+1)\sum_{i=1}^{n} b_i x_{ij} + \cdots + \sum_{i=1}^{n} b_i x_{i\in} \\ &= \sum_{j=1}^{n}\sum_{i=1}^{n}(n-j+1)b_i x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(n-j+1)b_i x_{ij}.\end{aligned}$$

마지막으로, 식 (12-1)의 세번째 항을 전개하면 다음과 같이 나타난다.

$$\begin{aligned}\sum_{j=1}^{n}\sum_{i=2}^{j} I_i &= I_2 + \\ &\quad I_2 + I_3 + \\ &\quad I_2 + I_3 + I_4 + \\ &\quad \cdots\cdots \\ &\quad I_2 + I_3 + I_4 + \cdots + I_n \\ &= (n-1)I_2 + (n-2)I_3 + \cdots + (n-j+1)I_j + \cdots + I_n \\ &= \sum_{j=2}^{n}(n-j+1)I_j.\end{aligned}$$

이상의 과정을 통해 식 (12)가 다음 식 (13)으로 변환된다.

Minimize

$$\sum_{i=1}^{n} n\delta_i x_{i1} + \sum_{i=1}^{n}\sum_{j=1}^{n}(n-j+1)b_i x_{ij} + \sum_{j=2}^{n}(n-j+1)I_j \qquad (13)$$

Subject to

$$\sum_{i=1}^{n}\delta_i x_{i1} + \sum_{i=1}^{n}\sum_{j=1}^{k-1} b_i x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{k} a_{iu} x_{ij} + \sum_{j=2}^{k} I_j \geq 0, \quad \forall k\in\{2,\cdots,n\},\ u\in\{1,\ldots,m\} \qquad (14)$$

$$\sum_{i=1}^{n} x_{ij} \leq 1,\ \forall j\in\{1,\cdots,n\} \qquad (15)$$

$$\sum_{j=1}^{n} x_{ij} \geq 1,\ \forall i\in\{1,\cdots,n\} \qquad (16)$$

$$x_{ij}\in\{0,1\},\ \forall i\in\{1,\cdots,n\},\ j\in\{1,\cdots,n\} \qquad (17)$$

$$I_j \geq 0,\ \forall j\in\{2,\cdots,n\} \qquad (18)$$

일반적으로 규모가 큰 혼합정수계획모형의 최적해를 적정시간내에 구하는 효율적인 해법은 존재하지 않는다[16]. 본 연구문제의 난이도가 NP-Hard이므로 작업수가 커지면 최적해를 구하기 어렵다는 점은 이미 기술한 바 있다. 반면에 상기 혼합정수계획모형에서 정수제약을 완화하면 일반적인 선형계획모형이 되므로 완화된 모형(LP Relaxation Model; 완화선형계획모형)자체의 최적해는 효율적으로 빠르게 구할 수 있고, 이 완화선형계획

모형의 최적해는 원래 혼합정수계획모형의 하한값이 된다. 특히 이후의 절에서 수치실험 결과를 통해 나타난 바와 같이 완화선형계획모형을 사용한 하한값의 성능이 매우 우수하다. 따라서, 이후의 절에서 제안하는 발견적해법의 성능평가에 본 하한값이 활용된다.

3. 발견적 해법

본 절에서는 모든 규모의 $2AFS||\Sigma C_j$ 문제에 대해 짧은 시간내에 효율적으로 근사해를 제공하는 발견적 해법(Heuristic Algorithm)을 제시한다.

본 연구 문제보다 일반적인 형태인 $2AFS||\Sigma w_jC_j$ 문제에 대해 Tozkapan et al.[17]이 다음에서 기술한 바와 같이 발견적 해법들을 제시한 바 있다. Tozkapan et al.[17]은 일정계획연구의 일반적인 과정(단순한 목적함수를 대상으로 한 충분한 연구에서 출발하여 보다 복잡한 목적함수 연구로의 이행; 여기서는 ΣC_j에서 Σw_jC_j로 이행)을 따르지 않고, 최초 연구부터 가중총작업완료시간 최소화를 목적함수로 채택하였다. 만일, $2AFS||\Sigma w_jC_j$ 문제에 대해 Tozkapan et al.[17]이 제시한 발견적 해법들이 충분히 효율적이라면, 본 연구는 불필요하겠지만, 현재 제시된 발견적 해법들은 이후의 성능비교에서 나타난 바와 같이 상당한 개선의 여지가 존재한다. 이러한 비효율성은 목적함수의 범위를 일반화시킴에 상당부분 기인하는 것으로 판단되므로, 본 연구에서는 일반적인 일정계획 연구과정에 따라, 연구범위를 가중치가 1인 비가중총작업완료시간 최소화로 좁혀서 발견적 해법의 효율성을 제고하고자 하였다. 본 연구의 이와 같은 접근은 (1) $2AFS||\Sigma C_j$ 문제 자체의 효율적인 발견적 해법 제시와, 이를 토대로 (2) $2AFS||\Sigma w_jC_j$ 문제 연구의 내실화 제고 등의 기여점을 갖는다.

아래에서 보는 바와 같이, Tozkapan et al.[17]은 총 $m+4$의 매우 단순한 해법들을 사용하고 있다(아래의 해법들은 가중총작업완료시간 최소화에 적합하도록 가중치를 고려하는 해법들로 제시되었으나, Tozkapan et al.[17]에서 비가중총작업완료 최소화 문제풀이에서 가중치를 1로 설정하여 사용되었으므로, 가중치가 1인 경우로 재정리하였다). 단순함 자체는 이 해법의 큰 장점이라고 할 수 있으나, 낮은 효과성은 개선의 필요성을 안고 있는 약점이다. 또, 해법의 개수가 $m+4$개로 기계수에 정비례하고 있다는 점도 현실적인 적용에 장애가 되는 단점으로 지적할 수 있다. 따라서, 해법의 갯수를 줄이고, 해법의 단순함이 크게 훼손되지 않으면서 동시에 보다 양호한 성능을 갖는 해법의 개발이 필요하다. 구체적으로, 이 해법들은 아래에서 정리된 바와 같이, 크게 Group I과 Group II로 구분되며, Group I에서 $m+1$개의 발견적 해를 구하고, Group II에서 3개의 발견적 해를 추가로 구한다.

[Tozkapan et al.[17]의 발견적 해법 Group I]

- 발견적 해법 j, 여기서 $j\in\{1,\cdots,m,m+1\}$:
  - 기계 j의 작업시간만을 기준으로 SPT(Shortest Processing Time)순으로 작업을 처리한다.

[Tozkapan et al.[17]의 발견적 해법 Group II]

- 발견적 해법 $m+2$:
  - 단계 **1**(작업순서 결정을 위한 지표값 계산)
    모든 작업들에 대해서 부품준비기계와 조립기계에서 소요되는 시간중 최소 소요시간 계산. 즉,
    $MinPT_i=\min\{\min_{k\in\{1,\cdots,m\}}a_{ik},\ b_i\},\ i\in\{1,\cdots,n\}$,
  - 단계 **2**(작업 순서 결정)
    단계 1에서 계산한 $MinPT_i$ 값이 작은 작업부터 큰 작업 순으로 작업순서를 결정한다.

- 발견적 해법 $m+3$:
  - 단계 **1**(작업순서 결정을 위한 지표값 계산)
    모든 작업들에 대해서 부품준비기계와 조립기계에서 소요되는 시간중 최소 소요시간 계산. 즉,
    $AvePT_i=\frac{1}{(m+1)}\left\{\sum_{k=1}^{m}a_{ik}+b_i\right\},\ i\in\{1,\cdots,n\}$,
  - 단계 **2**(작업 순서 결정)
    단계 1에서 계산한 $AvePT_i$ 값이 작은 작업부터 큰 작업 순으로 작업순서를 결정한다.

- 발견적 해법 $m+4$:
  - 단계 **1**(작업순서 결정을 위한 지표값 계산)
    모든 작업들에 대해서 부품준비기계와 조립기계에서 소요되는 시간중 최소 소요시간 계산. 즉,
    $MaxPT_i=\max\{\max_{k\in\{1,\cdots,m\}}a_{ik},b_i\},\ i\in\{1,\cdots,n\}$,
  - 단계 **2**(작업 순서 결정)
    단계 1에서 계산한 $MaxPT_i$ 값이 작은 작업부터 큰 작업 순으로 작업순서를 결정한다.

본 연구에서 보다 효과적인 발견적 해법을 제시하기 위해, 2AFS에서 처리될 임의의 두 후보작업중 먼저 처리할 작업을 선택하는 토대가 되는 우월성질(Dominance Property)들을 다음 (정리 4), (따름정리 1), (따름정리 2)에서 정리하였다.

정리 4 : 작업처리 순서가 결정되지 않은 임의의 두 작업 i와 j가 다음 두 조건을 만족한다면, 작업 i를 작

업 j보다 먼저 처리하는 것이 그렇지 않은 경우보다 총작업완료시간이 작거나 같다.

$$\max\{\Delta+b_j, \max(\Delta_i, \theta)+b_i+b_j\} \le \max\{\Delta+b_i, \max(\Delta_j, \theta)+b_i+b_j\} \quad (C1)$$

$$\max(\Delta_i,\theta)+b_i+\max\{\Delta+b_j, \max(\Delta_i, \theta)+b_i+b_j\} \le \max(\Delta_j,\theta)+b_j+\max\{\Delta+b_i, \max(\Delta_j,\theta)+b_i+b_j\} \quad (C2)$$

여기서,

$$\Delta=\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}+a_{jk})-\max_{k\in\{1,\cdots,m\}}(T_k)$$
$$\Delta_i=\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik})-\max_{k\in\{1,\cdots,m\}}(T_k)$$
$$\Delta_j=\max_{k\in\{1,\cdots,m\}}(T_k+a_{jk})-\max_{k\in\{1,\cdots,m\}}(T_k)$$
$$\theta=T_{m+1}-\max_{k\in\{1,\cdots,m\}}(T_k).$$

증명 : 전체 작업들로 구성된 작업순열 $S_1=(\pi,i,j,\sigma)$, $S_2=(\pi,j,i,\sigma)$를 고려하자. 여기서, π는 현재 작업처리 순서가 결정된 부분작업순열을 나타내고, σ는 작업처리 순서 미결정 작업들중 작업 i,j를 제외한 작업들로 구성된 임의의 작업순열을 나타낸다. 편의상 이후에서 작업순열 S에서 작업 i의 완결시간을 $C_i(S)$라고 표기한다. 그러면, 본 정리의 증명은 (1) 작업순열 S_1에서의 두 작업 i와 j의 작업완료시간의 합이 작업순열 S_2에서의 그것보다 작거나 같음과 (2) 작업순열 S_1에서의 작업 j의 완결시간이 작업순열 S_2에서의 작업 i의 완결시간보다 작거나 같음을 보임으로써 충분히 이루어진다. 일차적으로 작업순열 S_1에 대해서 다음이 성립한다.

$$C_i(S_1)=\max\{\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}), T_{m+1}\}+b_i$$
$$C_j(S_1)=\max\{\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}+a_{jk}), C_i(S_1)\}+b_j$$
$$=\max\begin{bmatrix}\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}+a_{jk}),\\ \max\{\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}), T_{m+1}\}+b_i\end{bmatrix}+b_j$$
$$=\max\begin{bmatrix}\max_{k\in\{1,\cdots,m\}}(T_k+a_{ik}+a_{jk}),\\ \max_{k\in\{1,\cdots,m\}}(T_k+a_{ik})+b_i,\\ T_{m+1}+b_i\end{bmatrix}+b_j$$
$$=\max\{\Delta, \Delta_i+b_i, \theta+b_i\}+\max_{k\in\{1,\cdots,m\}}(T_k)+b_j.$$

마찬가지로 작업순열 S_2에 대해서 다음 식이 성립한다.

$$C_j(S_2)=\max\{\max_{k\in\{1,\cdots,m\}}(T_k+a_{jk}), T_{m+1}\}+b_j,$$
$$C_i(S_2)=\max\{\Delta, \Delta_j+b_j, \theta+b_j\}+\max_{k\in\{1,\cdots,m\}}(T_k)+b_i.$$

본 정리의 조건 (C1), (C2)가 성립한다는 전제하에서 이상의 등식들로부터, 다음의 두 부등식이 성립함을 알 수 있다.

$$C_j(S_1) \le C_i(S_2) \quad (1)$$
$$C_j(S_1)+C_i(S_1) \le C_i(S_2)+C_j(S_2) \quad (2)$$

따라서 본 정리가 증명된다.

상기 (정리 4)는 이후에 제시할 발견적 해법에 실질적으로 활용될 (따름정리 1, 2)를 유도하기 위한 복합적 정리이다. 아래에서 (정리 4)를 두가지 부분적인 경우($\max(\Delta_i, \Delta_j) \le \theta$인 경우와 $\min(\Delta_i, \Delta_j) \ge \theta$인 경우)로 분할하여, (따름정리 1, 2)를 유도하였다. 따라서, 설명과 이해의 편의를 위해, 상기 (정리 4)의 의미를 바로 논의하는 대신에, (따름정리 1, 2)의 현실적 의미를 아래에서 각각 설명하였다. 이들 (따름정리 1, 2)는 이후에 제안되는 발견적 해법에서 실질적으로 활용된다.

(따름정리 1)이 (정리 4)로부터 유도되는 과정은 다음과 같다. 즉, (정리 4)에 대해 $\max(\Delta_i, \Delta_j) \le \theta$인 첫 번째 부분경우를 고려하면 조건식 (C1)과 (C2)는 다음과 같이 변형된다.

$$\max(\Delta+b_j, \theta+b_i+b_j) \le \max(\Delta+b_i, \theta+b_i+b_j) \quad (C1)$$
$$\theta+b_i \le \theta+b_j \quad (C2)$$

여기서, θ는 상수이므로 (C2)는 다시 $b_i \le b_j$와 같이 변형되어 다음과 같이 정리된다.

따름정리 1. 작업처리 순서가 결정되지 않은 임의의 두 작업 i와 j에 대해 $\max(\Delta_i, \Delta_j) \le \theta$가 성립하면서 동시에 다음의 두 조건이 성립한다면, 작업 i를 작업 j보다 먼저 처리하는 것이 그렇지 않은 경우보다 총작업완료시간이 작거나 같다.

$$\max(\Delta+b_j, \theta+b_i+b_j) \le \max(\Delta+b_i, \theta+b_i+b_j) \quad (C1)$$
$$b_i \le b_j \quad (C2)$$

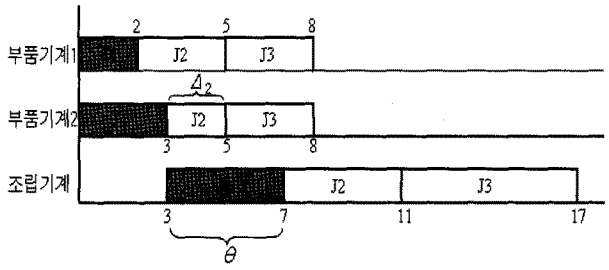
(따름정리 1)의 현실적 의미를 고찰하기 위해 다음과 같이 작업수 3개, 부품수 2개인 단순한 예제를 고려해 보자. 이때, 작업 1은 이미 최초위치에 처리하는 것으로 고정되어 있고, 작업 2와 3중에서 두 번째 위치에 처리할 작업을 선택하려는 상황이라고 가정한다.

<표 1> 단순 예제

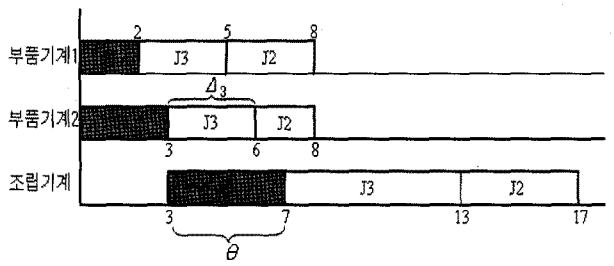
작업 k	부품1 준비시간 a_{1k}	부품2 준비시간 a_{2k}	조립시간 b_k
1	2	3	4
2	3	2	4
3	3	3	6

이러한 상황에서 작업 2와 작업 3간의 처리순서를 달리하여 표시한 간트(Gantt) 도표가 <그림 1>에 나타나 있다. <그림 1>에서 본 예제는 $\max(\Delta_2, \Delta_3) \le \theta$조건을 만족하고 있음을 알 수 있는데, 이는 이미 처리순

서가 결정된 작업의 조립완결시점 이전에 다음에 처리 가능한 후보작업의 부품을 준비할 수 있다는 의미이다. 결과적으로 이러한 후보작업이 여러 개 있는 경우에는 이 후보작업들의 조립시간이 짧은 작업을 먼저 처리하는 것이 목적함수값을 최소화하게 된다($1||\Sigma C_j$ 문제의 경우 최적해는 처리시간이 작은 작업을 먼저 처리하는 것[12]이라는 사실로부터 이 논의는 자명함). 전술한 (따름정리 1)의 (C2)는 정확하게 이러한 사실을 반영하고 있으며, (따름정리 1)의 대표적인 함의가 된다. (C1)은 문제사례가 아주 특이한 경우에 (C2)조건을 만족해도 결과적으로 선후관계가 잘못되는 경우를 방지하는 보조적인 조건이다. 실제로 본 단순사례는 (따름정리 1)의 두 조건을 만족하는 작업 2를 작업 3에 선행해서 처리할 때의 목적함수 값이 35로, 반대로 처리했을 때의 37보다 작음을 확인할 수 있다.



(a) 작업 2를 작업 3보다 선행해서 처리하는 경우



(b) 작업 3을 작업 2보다 선행해서 처리하는 경우

* 편의상 작업 i를 Ji로 표기하였음.

<그림 1> 단순예제의 처리순서별 간트도표

(따름정리 2)도 (따름정리 1)과 유사하게, (정리 4)에서 $\min(\Delta_i, \Delta_j) \geq \theta$가 만족하는 부분 경우이므로, (정리 4)의 각 조건이 $\min(\Delta_i, \Delta_j) \geq \theta$에 맞게 변형되어 아래와 같이 유도되었다.

따름정리 **2** : 작업처리 순서가 결정되지 않은 임의의 두 작업 i와 j에 대해 $\min(\Delta_i, \Delta_j) \geq \theta$가 성립하면서 동시에 다음의 두 조건이 성립한다면, 작업 i를 작업 j보다 먼저 처리하는 것이 그렇지 않은 경우보다 총작업완료시간이 작거나 같다.

$$\max(\Delta + b_j, \Delta_i + b_i + b_j) \leq \max(\Delta + b_i, \Delta_j + b_i + b_j) \quad (C1)$$

$$\Delta_i + \max(\Delta, \Delta_i + b_i) \leq \Delta_j + \max(\Delta, \Delta_j + b_j) \quad (C2)$$

(따름정리 2)는 $\min(\Delta_i, \Delta_j) \geq \theta$인 경우를 대상으로 하고 있는데, 이의 의미는 아직 순서가 결정되지 않아 순서결정 후보에 속한 작업들이 모두, 이미 순서가 결정된 직전 작업들이 완결되는 시점 이후에 부품준비가 가능할 정도로 부품준비시간이 많이 소요됨을 의미한다. 이러한 경우에는 (따름정리 1)의 경우와 달리, 조립작업시간만으로 선후관계를 판정할 수 없고, 개념적으로 "부품준비시간"과 "조립시간"을 함께 고려해야 하는데, (따름정리 2)는 이러한 의미를 담고 있으면서, 구체적으로 필요한 조건을 수식화 한 것이다.

그러나, (따름정리 1, 2)에서 제시하는 조건들을 완벽히 충족하는 문제 사례는 자주 발생하지 않는다는 한계점이 존재한다. 따라서, 아래에서 제시하는 발견적 해법들에서는 (따름정리 1, 2)의 조건중 일부만을 충족하는 작업을 선택하고 있다.

[발견적 해법 1(H1)]

단계 **0**(초기화)

- $u = 1,\ T_k = 0(\forall k \in \{1, \cdots, m+1\}),\ \pi^c = \{1, \cdots, n\}$

단계 **1**(최초처리 작업 찾기)

- 작업 x를 최초 처리작업으로 선택한다.
  여기서, $x = \text{argmin}_{j \in \pi^c}\{\max_{k \in \{1, \cdots, m\}}(a_{jk}) + b_j\}$

단계 **2**(주요값 수정)

- $T_k = T_k + a_{xk},\ \forall k \in \{1, \cdots, m\}$
- $T_{m+1} = \max\{\max_{k \in \{1, \cdots, m\}}(T_k), T_{m+1}\} + b_x$
- $\pi^c = \pi^c - \{x\}$
- $u = u + 1$

단계 **3**(최초 처리작업 이외의 처리작업 x 찾기)

- 작업집합 $\Omega = \{j | j \in \pi^c, d_j \leq T_{m+1}\}$ 를 찾는다.
  여기서 $d_j = \max_{k \in \{1, \cdots, m\}, j \in \pi^c}(T_k + a_{jk})$
- $\Omega \neq \varnothing$ 이면 $x = \text{argmin}_{j \in \Omega}(b_j)$
  (따름정리 1의 조건 2 적용)
- $\Omega = \varnothing$ 이면 $x = \text{argmin}_{j \in \pi^c}(d_j)$
  (따름정리 2의 조건 2 완화적용)

단계 **4**(u번째 처리작업으로 설정 및 종료여부 결정)

- 작업 x를 u번째 처리작업으로 설정
- $u = n$이면 종료하고, 그렇지 않으면 단계 2로 이동

[발견적 해법 2(H2)]

단계 **0**, 단계 **2**~단계 **4**. H1과 동일

단계 **1**(최초처리 작업 찾기)

- $x = \text{argmin}_{j \in \pi^c}\{\max_{k \in \{1, \cdots, m\}}(a_{jk})\}$인 작업 x를 최초 처리작업으로 선택한다.

[발견적 해법 3(H3)]
단계 0, 단계 2~단계 4. H1과 동일
단계 1(최초처리 작업 찾기)

- $x = \arg\min_{j \in \pi^c}\left(\sum_{k=1}^{m} a_{jk}/m\right)$인 작업 x를 최초 처리작업으로 선택한다.

이상에서 제시한 발견적 해법들은 최초처리작업 선택에서만 차이가 있고, 나머지 단계는 모두 동일하다. 본 연구의 목적함수인 총작업완료시간 최소화는 최초작업의 파급효과가 매우 크기 때문에(최초작업의 작업시간이 p이면 전체 목적함수값이 np만큼 증가하므로), 상기 해법들도 이 최초작업의 선택에서만 차이를 보이고 있다. H1, H2, H3는 공통적으로 단일기계에서의 총작업 완료시간 최소화 문제에서 사용하는 SPT 법칙(Short Processing Time Rule; 최소처리시간을 갖는 작업을 먼저 처리)을 기반으로 하고 있다. 또한, 세 방법들간에는 개념적으로(개별 부품준비시간+조립시간), (개별 기계의 처리시간), 그리고 (부품준비시간들의 평균값)이 가장 작은 작업을 각각 최초 작업으로 선택한다는 차이점이 있다. 세 방법에서 공통적으로 사용되는 단계 2~4는 (한 작업의 부품준비와 다른 작업완성용 부품의 조립)은 동시에 가능하다는 문제의 특성을 활용하여, (1) 현재 처리해야 할 작업의 부품준비시간내에 이전작업의 부품조립이 가능한 경우라면, 부품조립시간이 가장 짧은 작업을 선택하고, (2) 그렇지 못한 경우라면 부품준비시간이 가장 짧은 작업을 선택하도록 하는 과정으로 구성되어 있다.

4. 계산 실험

본 절에서는 임의로 생성된 수치값을 사용해 계산실험을 수행함으로써, 전절에서 제시한 해법들의 성능을 검토하였다. 하한값 계산에 사용된 선형계획완화모형은 Cplex와 C언어로, 그리고 탐색적 해법들은 C언어로 프로그래밍되어 Pentium IV PC에서 실행되었다. 일반적으로 해법의 성능분석은 최악오차한계(Worst-case Error Bound)를 분석적으로 증명하거나, 혹은 본 연구에서와 같이 수치실험을 통해 그 효과성(Effectiveness)를 검토하는 방식이 사용된다. 분석적으로 최악오차한계를 제시하는 것이 수학적으로 보다 완벽한 방법이지만, 고려대상 문제나 제시된 해법의 복잡도가 높아질수록 동접근법을 적용하는 것에는 한계가 있다. 따라서, 최악오차한계 제시가 곤란한 경우 본 연구에서와 같이 수치데이터를 사용한 계산실험 결과를 사용해 해법의 효과성을 검토하는 것이 일반적이다.

본 절에서 계산실험에 사용될 수치값들은 Tozkapan et al.[17]에서와 동일한 방법으로 생성한다. 이는 본 연구와 Tozkapan et al.[17]의 발견적 해법들간 성능비교를 주목적으로 하기 때문이다. 각 작업의 부품생산 및 조립시간은 모두 이산균등분포 $U[l,u]$로부터 생성한다. 이때, 단계 1(부품생산단계)과 단계 2(조립단계)의 상대적 처리시간의 크기차이에 따라 문제유형이 <표 2>과 같이 세가지로 구분된다.

<표 2> 수치실험 문제유형별 상/하한값

문제유형	단계 1		단계 2	
	l	u	l	u
A	1	100	1	100
B	1	80	20	100
C	20	100	1	80

이하에서는 일차적으로 최적해 대비 선형계획완화모형 하한값의 상대오차를 살펴봄으로써 하한값의 성능을 고찰하고, 다음으로 발견적 해법의 성능을 2단계로 나누어 살펴본다. 즉, 문제규모가 작은 경우에는 Tozkapan et al.[17]에 제시된 바와 같이, 간단한 분지한계법(Branch and Bound Algorithm)을 적용하여 최적해를 구할 수 있으므로 최적해 대비 발견적 해의 상대오차를 고찰하는 방법으로 제안한 발견적해법의 성능을 비교분석한다. 다음으로 문제규모가 일정수준 이상인 경우에는 최적해를 적정시간내에 도출할 수 없으므로 하한값에 대비한 발견적해의 상대오차를 검토한다.

선형계획완화모형을 이용한 하한값의 성능고찰을 위한 실험에서는 작업수(n)를 2종류(10, 15), 기계수(m; 편의상 부품생산기계와 조립기계의 합을 m으로 표기)도 2종류(5, 10)에 대해 <표 2>에 표시한 3가지 문제유형(A, B, C)의 조합들에 대해 각각 30개씩 총 360개의 임의생성된 수치자료를 사용하였다. 이와 같이 계산실험된 결과는 <표 3>에 요약되어 있다.

<표 3>에서 하한값의 성능평가를 위한 척도는 최적해에 대비한 최소, 평균 그리고 최대 상대오차를 사용하였다. 상대오차(RE; Relative Error)는 다음 식을 사용해 백분율 값으로 계산되었다.

$$RE(\%) = 100 \times (Opt - LB)/Opt$$

위 식에서 LB는 선형계획완화모형을 사용해 계산한 최적총작업완료시간의 하한값을 의미하고, Opt는 분지한계법을 적용해 계산된 최적총작업완료시간값을 나타낸다.

<표 3> 선형계획완화모형을 사용한 하한값의 성능

문제유형	최적해 대비 하한값의 상대오차(%)				
	n	m	최소	평균	최대
A	10	5	0.31	3.52	7.76
		10	0.50	4.20	6.67
	15	5	0.50	3.23	5.90
		10	1.24	4.05	7.26
B	10	5	0.26	2.11	5.50
		10	0.79	3.36	6.95
	15	5	0.00	1.59	3.77
		10	0.04	2.12	5.87
C	10	5	0.62	2.44	4.16
		10	0.18	2.69	4.45
	15	5	0.65	1.68	2.88
		10	1.17	2.30	3.71

<표 3>에 나타난 상대오차값들로부터 문제유형이나 작업수 및 기계수에 따라 상대오차의 규모에 차이는 존재하지만 평균적으로 5% 이내의 값을 보이므로 상당히 양호한 하한값이라고 판단된다. 일반적으로 작업수가 일정수준 이상의 값을 갖는 문제에 대해서는 최적해를 구하기 어려우므로 하한값대비 상대오차를 계산하는데, 이때 최적해와 하한값의 근사정도에 따라 하한값 대비 상대오차값이 큰 영향을 받게 되므로 하한값의 성능이 중요하다.

참고적으로 <표 2-하한값의 성능> 계산에 사용된 혼합정수계획모형의 선형계획완화해법(하한값 계산)과 분지한계법(최적해 계산)의 계산시간은 <표 4>에 정리하였다. <표 4>에 나타난 바와 같이, 하한값 계산시간은 작업수 n과 기계수 m이 증가하면 미세하게 커지는 경향이 있으나, 문제유형별로 큰 차이 없이 0.06초 이내의 짧은 시간이 소요되고 있음을 알 수 있다. 반면에, 분지한계법을 적용한 최적해 계산시간은 작업수와 기계수가 많아짐에 따라 상당히 빠르게 증가하고 있음을 볼 수 있다. 실제로, 작업수를 20개로 설정하여 실험하게 되면, 30개 임의생성 사례중에서 평균적으로 6~7개의 문제가 최적해 도출에 3,600초 이상의 시간이 소요되는 것으로 나타남으로써, 현재의 분지한계법을 작업수 20개 이상의 사례에 적용하는 것은 부적절한 것으로 판단되었다. 이러한 실험결과는 현재의 분지한계법에 보다 강력한 우월성질(Dominance Property; Fathoming Rule)들이 추가될 필요가 있음을 의미하는 것이지만, 본 연구의 촛점 (효율적인 발견적 해법 모색)에서는 벗어나므로 추후의 연구과제로 남겨둔다.

<표 4> 하한값과 최적해 계산시간

문제유형	n	m	하한값 계산시간		최적해 계산시간	
			평균	최대	평균	최대
A	10	5	0.01	0.11	0.02	0.16
		10	0.02	0.03	0.08	0.94
	15	5	0.02	0.03	4.34	139.40
		10	0.05	0.06	41.15	523.13
B	10	5	0.01	0.02	0.01	0.15
		10	0.01	0.02	0.03	0.47
	15	5	0.02	0.03	5.39	121.75
		10	0.04	0.05	43.99	511.09
C	10	5	0.01	0.02	0.06	0.62
		10	0.02	0.02	0.12	0.94
	15	5	0.02	0.05	7.08	134.37
		10	0.04	0.06	58.26	583.13

다음으로 작업수가 작은 문제(작업수 $n \in \{10, 15\}$)에 대해 Tozkapan et al.[17]에서와 마찬가지로 기계수(m)를 5, 10대로 하여 수치실험을 수행하였다. 실험에 사용된 수치데이터는 직전단계에서 하한값의 성능실험에 사용된 360개의 임의생성 수치값을 그대로 사용하였다. 또, 성능비교에 사용된 발견적해는 Tozkapan et al.[17]에 제시된 $(m+4)$개의 발견적해법들이 산출한 목적함수값 중 최소값(최소화문제이므로)과 본 연구에서 제안한 세 개의 발견적해법들(H1, H2, H3)이 제시하는 목적함수값중 최소값을 사용하여 최적목적함수값 대비 상대오차를 계산하여 비교하였다. 상대오차는 다음 식을 사용해 백분율 값으로 계산되었다.

$$RE(\%) = 100 \times (Heu - Opt)/Opt$$

식에서 Heu는 두 연구에서 제안한 발견적 해법들을 각각 모두 사용해서 계산한 목적함수값을 의미한다. 두 연구 모두 문제별로 항상 우월하게 최소값을 제시하는 발견적해법이 존재하지 않으므로 제안된 모든 발견적 해법을 사용한 후, 도출된 목적함수값중 최소값을 선택하여 비교에 사용하였다. 실험결과는 <표 5>에 요약되어 있다. 실험에 사용된 모든 발견적 해법들(Tozkapan et al.[17]과 본 연구에서 제안된 모든 발견적 해법들)의 계산시간은 모두 극히 짧게(0.000000초 이하) 나타나고 있는 관계로 표에는 표기하지 않았다.

<표 5>에 나타난 바와 같이 Tozkapan et al.[17]에서 제안한 발견적해법 대비 본 연구의 발견적해법이 평균적으로 4% 이상 최적해 대비 상대오차를 감소시킴으로써 보다 우월한 성능을 나타내고 있다고 판단된다. 특히

본 연구는 3개의 발견적해법을 사용하는 반면 Tozkapan et al.[17]은 그보다 두배 이상인 $(m+4)$개의 발견적 해법을 사용하고 있음을 감안한다면 본 연구의 발견적 해법이 상당히 효과적임을 알 수 있다.

<표 5> 최적 목적함수값 대비 발견적해법 성능비교

문제유형	n	m	발견적 해법의 최적 목적함수값 대비 상대오차(%) : 작은규모 문제					
			Tozkapan et. al.[17]			본 연구		
			최소	평균	최대	최소	평균	최대
A	10	5	0.00	5.67	18.41	0.00	0.92	3.28
		10	1.95	5.06	9.27	0.00	0.96	2.98
	15	5	1.15	8.75	16.92	0.28	1.85	4.24
		10	4.25	8.30	17.45	0.15	1.85	5.61
B	10	5	0.00	5.17	14.94	0.00	1.13	3.52
		10	0.25	6.15	11.24	0.00	0.91	2.70
	15	5	1.16	6.70	12.54	0.00	0.97	3.70
		10	2.95	7.87	13.71	0.00	1.04	2.95
C	10	5	1.44	3.29	6.87	0.00	0.70	3.49
		10	0.71	3.19	5.51	0.00	0.45	1.40
	15	5	1.78	4.18	8.11	0.17	0.79	1.80
		10	2.17	4.63	8.20	0.11	0.73	1.49
평균			1.48	5.75	11.93	0.06	1.03	3.10

주) 모든 발견적 해법의 계산시간이 0.000000초 이하로 나타남으로써 발견적 해법들의 계산시간을 별도로 표기하지 않음.

본 연구에서 제안한 발견적 해법들간의 상대적인 성능을 비교하기 위해, 문제유형별-작업수별-기계수별로 임의생성된 30개씩의 사례들에 대해 최소값을 산출한 횟수를 다음 <표 6>에 정리하였다. 발견적 해법들이 최소값을 동시에 산출하는 경우가 빈번히 발생하므로, 한 행의 합이 실험사례의 갯수인 30을 초과하는 현상이 일반적으로 나타난다. Tozkapan et al.에서 제시된 발견적 해법의 최소해 산출횟수는 별도로 정리하지 않았다. 본 연구의 해법이 우월한 성능을 나타내고 있다는 사실이 위의 <표 5>에 충분히 나타나고 있기 때문이다.

<표 6>에 나타난 바와 같이, 발견적 해법의 성능 순위가 평균적으로는 H2, H3, H1순이라고 볼 수 있다.전술한 바와 같이, 본 연구의 발견적 해법들은 최초작업 선택방법만을 달리하고 있고, 이후의 처리작업 선택은 동일하다. 또, 최초작업 선택방법도 공통적으로 SPT 법칙에 기반하고 있으나, SPT에 사용되는 처리시간의 설정방식에서 상이한 방식을 사용하고 있다. 실험결과로부터, 최초 처리작업의 선택이 발견적 해법의 성능에 큰 영향을 미치고 있음을 알 수 있다. 이는 총작업완료시간 최소화가 같은 일반적인 특징에 기인하고 있음을 알 수 있다. 또한, 최초 처리작업을 선택할 때, 모든 기계별 처리시간을 개별적으로 고려하는 H2가 평균적으로 가장 높은 성과를 나타내고 있음을 알 수 있다. H1과 H3는 두 가지 이상의 처리시간값에 기반하여 최초 처리작업을 선택한다는 공통점을 가지고 있고, 이로 인해 H2보다 열등한 성과를 내고 있는 것으로 해석할 수 있다. 또, (개별 부품준비시간+조립시간)값을 사용하는 H1보다는 (모든 부품준비 시간의 평균값)을 활용한 H3의 성능이 평균적으로 우월한 것으로 나타나는데, 이는 부품준비 시간만을 고려하는 것이 부품준비 시간과 조립시간을 함께 고려하는 것보다 최초작업선택에 적절하다는 것을 보여준다. 이는 한 작업의 부품준비와 다른 작업의 부품조립이 동시에 진행될 수 있다는 본 문제의 특성에 기인하는 것으로 해석된다. 그러나, 문제유형-기계수-작업수의 모든 조합별, 모든 사례에 대해 항상 우월한 결과를 제공하는 발견적 해법은 존재하지 않는다는 것을 실험을 통해 확인하였으므로, 이들의 보완적 사용이 필요하다고 하겠다.

<표 6> 제안된 발견적 해법들간 성능비교

문제유형	n	m	임의생성된 30개 사례중에서 최소 ΣC_i값 발견횟수		
			H1	H2	H3
A	10	5	16	26	22
		10	10	17	19
	15	5	13	24	16
		10	13	20	10
B	10	5	11	27	20
		10	10	23	15
	15	5	15	22	22
		10	12	22	17
C	10	5	10	25	20
		10	9	25	12
	15	5	16	24	18
		10	9	23	14
평균			12.0	23.2	17.1

마지막으로 규모가 큰 문제(작업수 $n\in\{20, 30, 40, 50\}$)에 대해 발견적 해법의 성능비교를 수행하였다. 본 수치실험에서도 기계수는 5대와 10대로만 한정하였고, 문제유형도 <표 2>에 제시된 3가지를 사용하여 각 문제조합에 대해 30개씩 총 720개의 수치데이터를 임의생성하여 사용하였다. 그러나, 동 수치데이터에 대해서는

일정한 시간내에 최적목적함수값을 계산하기 어려우므로, 전술한 선형계획완화모형 하한값을 상대오차계산의 기준값으로 사용하는 다음 식을 이용하였다.

$$RE(\%) = 100 \times (Heu - LB)/LB$$

이와 같이 각 문제조합별로 계산된 하한값 대비 상대오차의 최소, 평균, 최대값이 <표 7>에 요약되어 있다. <표 7>의 상대오차 계산에 사용된 발견적 해는 모든 발견적 해법들(Tozkapan et al.[17]과 본 연구에서 제안한 모든 발견적 해법)에서 공통적으로 매우 짧은 시간내(0.000000초 이내)에 계산되므로 계산시간을 별도로 나타내지 않았다. 또, <표 7>의 상대오차 계산에는 최적목적함수값 대신 선형계획완화모형의 해(하한값)를 사용하였기 때문에 전반적으로 상대오차값이 크게 나타남을 볼 수 있다. 이는 하한값이 일반적으로 최적목적함수값 이하이므로 당연한 귀결이다. 그러나, <표 7>에 나타난 바와 같이 Tozkapan et al.[17]의 해법을 사용한 상대오차값보다 본 연구 해법이 산출한 상대오차값의 크기가 평균적으로 6% 이상 감소함으로써, 큰 규모 문제에 대해서도 본 연구 해법의 우월성이 일관되게 유지됨을 알 수 있다. 뿐만 아니라, 두 연구의 최대상대오차값을 비교해보면 본 연구의 결과가 Tozkapan et al.[17]의 해법을 사용한 결과에 비해 평균 10% 정도 향상됨으로써, 최악상황의 오차값이 상당히 개선되었음을 알 수 있다. 이는 본 연구의 발견적해법이 기존 연구의 해법에 비해 상당히 안정적인 결과를 산출하고 있음을 의미하는 것으로써, 실무적 측면에서 본 연구의 발견적 해법이 매우 효과적으로 적용될 수 있음을 의미한다.

아래 <표 8>에는 <표 7>의 상대오차 산출에 적용된 (1) 하한값 계산시간과, (2) 본 연구에서 제안한 발견적 해법별 최소값 산출회수가 정리되어 있다. 하한값 산출에 사용된 선형계획완화기법의 계산시간은 작업수와 기계수가 증가함에 따라서 일정하게 증가하지만, 최대 6초 내외의 비교적 짧은 시간에 계산이 이루어짐을 볼 수 있

<표 7> 하한값 대비 발견적해법 성능비교

문제 유형	n	m	발견적 해법의 하한값 대비 상대오차(%) : 큰 규모 문제					
			Tozkapan et al.[17]			본 연구		
			최소	평균	최대	최소	평균	최대
A	20	5	5.77	13.33	21.50	1.31	5.08	10.24
		10	6.55	13.00	21.88	3.28	6.01	9.29
	30	5	6.62	13.47	22.28	2.05	5.18	9.11
		10	8.15	13.47	18.38	3.62	5.72	8.14
	40	5	8.88	11.89	18.95	3.00	4.58	6.89
		10	8.57	12.99	18.88	3.66	5.42	7.95
	50	5	6.86	12.37	20.11	2.54	4.20	6.29
		10	9.05	14.01	21.65	3.77	5.38	7.38
B	20	5	3.73	9.55	15.23	0.26	2.20	5.07
		10	4.33	10.30	16.95	0.18	3.07	7.66
	30	5	3.19	8.44	12.98	0.06	1.47	2.96
		10	4.68	10.33	17.08	0.46	2.33	4.48
	40	5	3.37	8.29	15.44	0.09	1.23	3.31
		10	5.18	10.44	17.43	0.45	2.46	5.09
	50	5	2.59	8.44	14.34	0.00	1.14	2.65
		10	4.53	9.04	13.82	0.57	1.86	4.28
C	20	5	3.61	7.11	12.93	0.97	2.52	5.08
		10	5.26	7.26	10.36	1.58	2.94	4.08
	30	5	3.92	6.57	10.54	0.66	2.16	3.84
		10	5.11	7.43	11.10	1.04	2.44	3.78
	40	5	3.51	6.29	8.68	1.42	2.22	3.25
		10	5.29	7.46	11.17	1.65	2.52	3.26
	50	5	4.14	6.03	8.44	0.80	2.22	3.35
		10	4.73	6.97	10.66	1.63	2.32	3.11
평균			5.32	9.77	15.45	1.46	3.20	5.44

주) 모든 발견적 해법의 계산시간이 0.000000초 이하로 나타남으로써 발견적 해법들의 계산시간을 별도로 표기하지 않음.

<표 8> 큰 규모 문제에서의 하한값 계산시간과 제안된 발견적 해법별 최소값 산출횟수

문제 유형	n	m	큰 규모 문제의 하한값 계산시간		큰 규모 문제의 해법별 최소값 산출횟수		
			평균	최대	H1	H2	H3
A	20	5	0.05	0.20	11	23	20
		10	0.11	0.17	10	18	14
	30	5	0.16	0.19	14	16	13
		10	0.46	0.55	11	22	17
	40	5	0.44	0.55	11	18	22
		10	1.53	2.41	9	16	16
	50	5	1.06	1.31	17	15	14
		10	4.41	6.05	15	13	10
B	20	5	0.04	0.05	14	23	18
		10	0.10	0.13	9	21	16
	30	5	0.13	0.17	15	24	17
		10	0.39	0.50	12	21	16
	40	5	0.35	0.47	10	25	24
		10	1.11	1.80	12	23	12
	50	5	0.76	0.89	16	23	18
		10	3.09	5.38	4	24	16
C	20	5	0.04	0.05	21	19	17
		10	0.10	0.13	15	21	15
	30	5	0.14	0.16	10	24	21
		10	0.40	0.53	13	21	15
	40	5	0.37	0.45	12	16	22
		10	1.32	1.83	13	15	14
	50	5	0.85	1.00	14	19	15
		10	3.79	5.25	12	16	15
평균					12.5	19.8	16.5

다. 물론, 하한값은 발견적해법의 성능평가를 위해서 일회적으로 산출되므로 계산시간 그 자체는 큰 의미를 갖지 않는다고 할 수 있다. 또, 본 연구에서 제안한 세가지 발견적 해법들이 큰 규모 문제에 대해 최소 ΣC_i 값을 산출하는 횟수로부터 평균적으로는 H2, H3, H1의 순으로 성능이 우월함을 알 수 있다. 그러나, 표에서 볼 수 있듯이, 문제유형-작업수-기계수 조합별로 항상 우월한 발견적 해법이 존재하지 않음도 알 수 있고, 이러한 결과들은 작은 규모 문제의 경우와 일관되게 해석될 수 있다.

5. 결 론

여러 종류의 부품을 생산한 후, 조립하여 완제품을 생산하는 2단계 조립흐름라인에서 총작업완료시간을 최소화하는 일정계획문제에 대해 연구하였다. 본 문제에 대한 우월성질들을 규명하고, 이에 기초하여 기존연구에서 제시된 발견적해법에 대비하여 평균적으로 4%에서 6%의 상대오차를 개선한 발견적해법을 제안하였다.

본 연구에서 대상으로 하고 있는 문제와 같이 NP-Hard 난이도를 갖는 문제의 경우, 일정수준 이하의 소규모 사례(Instance)에 대해서는 최적해를 구할 수 있지만, 일반적으로는 적정시간내에 최적해를 구하지 못한다. 따라서, 일정수준 이상 규모의 문제에 대해 효과적으로 근사해(Near-Optimal Solution)를 제공하는 발견적해법의 개발이 이론적 측면에서는 물론 실무적 차원에서 절실히 요청된다. 본 연구 결과는 이러한 관점에서 기여점을 찾을 수 있겠다.

그러나, 본 연구에서는 소규모 문제에 대해 최적해를 찾는 분지한계법을 개선하기 위한 연구를 다루지 않았고, 기존연구의 분지한계법을 그대로 활용하였다는 한계점을 가지고 있다. 추후 연구에서 다양한 해 조합들 간에 존재하는 우월성을 규명함으로써 최적해 자체를 찾는 분지한계법의 성능을 개선할 필요가 있다.

참고문헌

[1] 전재호; "부품외주를 고려한 Flowshop 일정계획문제 연구", 산업경영시스템학회지, 29(4) : 34-42, 2006.

[2] 윤상흠, 전재호; "부품외주를 고려한 Flowshop 일정계획 해법 개선", 산업경영시스템학회지, 31(2) : 80-93, 2008.

[3] Baker, K. R.; Introduction to Sequencing and Scheduling, John Wiley and Sons, 1974.

[4] Du, J. and Leung, J. Y. T.; "Minimizing mean flow time in two-machine open shops and flow shops," *Journal of Algorithms*, 14(1) : 24-44, 1993.

[5] Garey, M. R. and Johnson, D. S.; Computers and Intractability-A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[6] Graham, R. L., Johnson, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G.; "Optimization and approximation in deterministic sequencing and scheduling : A survey," *Annals of Discrete Mathematics*, 5 : 287-326, 1979.

[7] Hariri, A. M. A. and Potts, C. N.; "A branch and bound algorithm for the two-stage assembly scheduling problem," *European Journal of Operational Research*, 103 : 547-556, 1997.

[8] Koulamas, C. and Kyparisis, G. J.; "The three-stage assembly flowshop scheduling problem," *Computers and Operations Research*, 28 : 689-704, 2001.

[9] Kovalyov, M. Y., Potts, C. N., and Strusevich, V. A.; "Batching decisions for assembly production systems," *European Journal of Operational Research*, 157 : 620-642, 2004.

[10] 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*, 39 : 616-625, 1993.

[11] Lin, B. M. T., Cheng, T. C. E., and Chou, A. S. C.; "Scheduling in an assembly-type production chain with batch transfer," *OMEGA : The International Journal of Management Science*, 35(2) : 143-151, 2007.

[12] Pinedo, M.; Scheduling Theory, Algorithms, and Systems, Prentice-Hall, New Jersey, 2002.

[13] Potts, C. N., Sevast'janov, S. V., Strusevich, V. A., Wassenhove, L. N. V., and Zwaneveld, C. M.; "The two-stage assembly scheduling problem : Complexity and approximation," *Operations Research*, 43 : 346-355, 1995.

[14] Sun, X., Morizawa, K., and Nagasawa, H.; "Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling," *European Journal of Operational Research*, 146 : 498-516, 2003.

[15] Sung, C. S. and Juhn, J.; "Makespan minimization for a 2-stage assembly scheduling subject to component available time constraint," *International Journal of Production Economics*, 119 : 392-401, 2009.

[16] Taha, A. H.; Operations Research-An Introduction, Prentice Hall, New Jersey, 1997.

[17] Tozkapan, A., Kirca, O., and Chung, C. S.; "A branch and bound algorithm to minimize the total weighted flowtime for the two-stage assembly scheduling problem," *Computers and Operations Research*, 30 : 309-320, 2003.