

## A mean-absolute-deviation based method for optimizing skid sequence in shipyard subassembly

Kyung-Tae Lee\*, Yung-Keun Kwon\*

\*Student, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea

\*Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea

### [Abstract]

In this paper, we propose a method of optimizing the processing order of skids to minimize the span time in a conveyor environment of the shipbuilding subassembly process. The subassembly process consists of a series of fixed tasks where the required work time is varied according to the skid type. The loading order of skids on a conveyor which determines the span time should be properly optimized and the problem size exponentially increases with the number of skids. In this regard, we propose a novel method called UniDev by defining a measure of the mean-absolute-deviation about the time difference among simultaneously processed tasks and iteratively improving it. Through simulations with various numbers of skids and processes, it was observed that our proposed method can efficiently reduce the overall work time compared with the multi-start and the 2-OPT methods.

▶ **Key words:** Subassembly process, Conveyor environment, Skid, Order optimization, Mean-absolute-deviation

### [요 약]

본 논문은 조선소 소조립 공정의 스킨드 컨베이어 환경에서 전체 작업 시간을 최소화하기 위해 스킨드 처리 순서를 최적화하는 방법을 제안한다. 소조립 공정은 일련의 정해진 작업들로 구성되며 스킨드의 종류에 따라 요구되는 작업 시간이 다르다. 전체 작업 시간을 결정짓는 스킨드의 컨베이어 투입 순서는 적절하게 최적화되어야 하며 문제 크기는 스킨드 개수에 따라 지수적으로 증가한다. 이러한 점에서 UniDev라 불리는 새로운 방법을 제안하는데 동시간에 진행되는 작업 소요 시간의 차이에 관한 평균절대편차를 정의하고 이를 점진적으로 개선한다. 다양한 스킨드 개수와 공정의 개수에 대한 시뮬레이션 결과 제안된 방법이 Multi-Start 방법과 2-OPT 방법보다 전체 작업 소요 시간을 효과적으로 줄일 수 있음을 확인하였다.

▶ **주제어:** 소조립 공정, 컨베이어 환경, 스킨드, 순서 최적화, 평균절대편차

- 
- First Author: Kyung-Tae Lee, Corresponding Author: Yung-Keun Kwon
  - Kyung-Tae Lee (lmuffin@naver.com), Department of Electrical, Electronic and Computer Engineering, University of Ulsan
  - Yung-Keun Kwon (kwonyk@cicweb.ulsan.ac.kr), Department of Electrical, Electronic and Computer Engineering, University of Ulsan
  - Received: 2022. 10. 19, Revised: 2022. 12. 09, Accepted: 2022. 12. 14.

## I. Introduction

선박의 건조 과정은 크게 선박의 외부 골격을 만드는 선각 공정과 선박 내부의 각종 시설 및 설비를 장치하는 의장 공정, 그리고 도장 공정으로 구분된다. 이때, 선각 공정은 전처리 공정, 소조립 공정, 대조립 공정으로 순차적으로 진행된다. 전처리 공정에서는 소조립 공정에 사용하기 위한 형태로 강재를 절단 및 성형하는 작업을 진행한다. 소조립 공정에서는 블록 단위로 구성되는 대조립 공정을 위한 부품을 구성하는 작업을 진행한다. 마지막으로 대조립 공정에서는 소조립 공정에서 구성된 부품을 통해 블록을 구성하며, 각 블록은 의장 공정 및 도장 공정을 거쳐 도크로 운반된 후 탑재하게 된다. 이처럼 선박의 건조 과정은 설계에서부터 안벽 의장까지 일련의 공정들이 상호 영향을 미치는 과정으로 구성되어 있다. 그리고 일부 건조 과정 중 병목이 발생할 수 있는 부분에 대한 일정 계획의 최적화를 통해 전체적인 생산성 향상을 위한 다양한 연구가 진행되고 있다 [1].

선각 공정 중 소조립 공정은 일련의 공정이 순서대로 컨베이어 환경에서 진행되며, 이때 컨베이어 위에서는 스킵 단위로 작업이 진행된다. 각각의 스킵은 사전에 계획된 일정대로 컨베이어에 투입되는데, 스킵의 종류에 따라 각 공정별 작업 시간에 차이가 있기 때문에 병목 현상이 발생하게 된다. 이 문제를 해결하기 위해 조선소 내에서는 작업장의 형태 변형, 크기 확장 등 여러 물리적인 환경 개선을 통한 방법을 고려하지만, 그에 필요한 예산과 공간적 한계로 인해 뚜렷한 문제개선에 어려움이 있는 상황이다. 이에 따라 기존에 진행되었던 연구에서는 앞서 존재했던 작업장의 공간적 한계 등을 고려하여 추가적인 설비의 투자나 변경 없이 스킵 투입 순서를 중심으로 문제를 해결하는 방법들이 제안되었다.

본 논문은 소조립 공정의 스킵 투입 순서를 효율적으로 최적화하여 전체 작업 시간(Span Time)을 최소화하는 방법을 제안한다. 먼저, 본 연구의 필요성을 확인하기 위해 컨베이어 환경에서 존재하는 몇 가지 제약 사항을 제시하고, 제약 사항을 기반으로 구성된 컨베이어 환경에서 스킵의 투입 순서에 따라 전체 작업 시간이 달라짐을 보인다. 그리고 본 논문에서 제안하는 방법의 성능을 비교하기 위해 Multi-Start 방법과 2-OPT 방법을 구현하였고, 스킵의 수와 스킵의 필요 공정 수를 변화시켜 다양한 환경에서 비교 실험을 진행하였다. 마지막으로 실험 결과를 통해 본 논문에서 제안한 방법이 효율적으로 투입 순서를 최적화하며 실행 시간이 더 빨라질 수 있음을 확인하였다.

## II. Preliminaries

### 1. Related Works

조선 소조립 공정의 컨베이어 환경에서 스킵 투입 순서를 최적화하는 문제의 크기는 스킵 수에 지수적으로 증가한다. 이러한 문제의 해를 구하는 과정은 Brute-Force 방법을 통해 모든 경우의 수를 탐색하는 방법, Meta-heuristics[2] 방법을 통해 최대한 다양한 공간을 탐색하는 방법, 그리고 Heuristics을 통해 지역적인 공간에서 최적해를 탐색하는 방법이 있다. 이때 Brute-Force 방식은 전역 최적해를 보장하지만 실행 시간(Running Time)이 문제의 크기에 지수적으로 증가한다는 한계가 있다. 따라서 대규모의 실제 문제에서는 일반적으로 시뮬레이티드 어닐링[3], 유전 알고리즘[4][5] 등의 Meta-heuristics 방법 혹은 Heuristics 방법을 활용하게 된다.

선행 연구[6]에서는 생산 시스템에서 다단계로 구성되는 작업에 대해 두 종류의 Heuristics 방법을 제안하였으며 작업 수와 작업할 기계 대수를 바탕으로 실험 환경을 구성하였다. 선행 연구[7]에서는 유전 알고리즘을 활용하여 조선 건조 과정 중 블록 탑재 공정에 대한 일정계획 최적화 방법을 제안하였다. 선행 연구[8]에서는 마찬가지로 유전 알고리즘을 활용하여 판넬 블록 조립 공정에 대한 일정계획 최적화 방법을 제안하였다. 선행 연구[9]에서는 소조립 공정을 대상으로 하였지만 로봇 용접 공정에 한해서 진행되었으며 용접량 및 용접 순서를 유전 알고리즘을 통해 최적화하였다. 소조립 공정의 투입 순서를 대상으로 진행한 연구는 사례가 많지 않았다. 유전 알고리즘을 기반으로 제안된 방법[10]은 컨베이어에 투입하기 위한 스킵의 구성과 스킵의 투입 순서를 유전 정보로 활용하여 최적의 조합을 탐색하는 방법을 제안하였지만 스킵의 수는 27개 이하로 제한되었으며 공정 수는 5개로 고정하였다. 시뮬레이티드 어닐링을 기반으로 제안된 방법[11]은 작업자의 배치와 운용에 대한 몇 가지 시나리오를 가정하였으며 다양한 종류의 스킵에 대한 투입 순서를 탐색하였지만 스킵의 수는 12개, 공정 수는 6개로 고정하였다. 본 논문에서는 사전에 제시된 스킵의 구성을 유지하면서 작업자의 작업 속도가 모두 동일한 환경을 전제로 대규모의 스킵 컨베이어 문제에 적용 가능한 방법을 제안하였다.

### 2. Problem Formulation

조선 소조립 공정은 스킵 단위로 컨베이어에 투입되어 판계, 배재, 취부, 자동 용접, 수동 용접, 사상 등 일련

Skid	Span Time	
	Process X	Process Y
A	1	2
B	2	4
C	3	1
D	4	3

A. Skid Table

Case of Schedule	Skid	Schedule Chart	Total Span Time
Case 1	A	[Gantt chart for Case 1, Skid A]	14
	B	[Gantt chart for Case 1, Skid B]	
	C	[Gantt chart for Case 1, Skid C]	
	D	[Gantt chart for Case 1, Skid D]	
Case 2	B	[Gantt chart for Case 2, Skid B]	16
	A	[Gantt chart for Case 2, Skid A]	
	C	[Gantt chart for Case 2, Skid C]	
	D	[Gantt chart for Case 2, Skid D]	
Case 3	A	[Gantt chart for Case 3, Skid A]	13
	D	[Gantt chart for Case 3, Skid D]	
	B	[Gantt chart for Case 3, Skid B]	
	C	[Gantt chart for Case 3, Skid C]	

B. Schedule of Skid Conveyor

Fig. 1. An Example of Skid Processing on a Conveyor

의 정해진 공정을 통해 블록을 생산하는 과정을 뜻한다. 이때, 스키드는 전처리 공정에서 성형된 강재를 통해 구성된다. 본 논문에서 진행한 스키드 컨베이어 환경에 대한 제약 사항은 아래와 같다.

- 컨베이어에 투입되는 스키드의 각 공정은 선후 관계가 존재하기 때문에 반드시 앞의 공정이 마무리된 후 다음 공정을 진행할 수 있다.
- 스키드가 컨베이어 위에서 이동하기 위해서는 컨베이어 위에 존재하는 모든 스키드의 공정이 종료되어야 이동할 수 있다.
- 다양한 종류의 스키드가 존재하며, 스키드의 종류에 따라 각 공정별 작업 시간은 같지 않다.

2.1 Span Time

주어진 스키드 투입 순서의 효율성을 평가하기 위해서 스키드의 투입 순서에 따라 작업이 진행되는 과정에서 컨베이어 위에서 발생하는 작업 시간과 대기 시간을 고려한 전체 작업 시간을 계산해야 하며 이를 위해 필요한 변수를 먼저 아래와 같이 정의한다.

- $N$  : 스키드의 개수
- $M$  : 공정 개수 ( $M \leq N$ )
- $\Pi$  : 스키드 투입 순서, ( $\{1, 2, 3, \dots, N\}$ 의 순열)
- $S_i^\pi$  : 스키드 투입 순서  $\Pi$ 의  $i$ 번째 스키드 ( $i \in \{1, 2, 3, \dots, N\}$ )
- $T^{\pi_{i,j}}$  :  $S_i^\pi$ 의  $j$ 번째 공정 작업 시간, ( $j \in \{1, 2, 3, \dots, MRIGHT\}$ )

이제 스키드 투입 순서  $\Pi$ 에 대한 전체 작업 시간  $span(\Pi)$ 은 다음과 같이 정의한다.

$$span(\Pi) = \sum_{t=1}^{N+M-1} f(t) \quad (1)$$

where  $f(t) =$

$$\begin{cases} \text{MAX}(T_{1,t}^\pi, T_{2,t-1}^\pi, \dots, T_{t-1,2}^\pi, T_{t,1}^\pi) & (1 \leq t < M) \\ \text{MAX}(T_{t-M+1,M}^\pi, T_{t-M+2,M-1}^\pi, \dots, T_{t-1,2}^\pi, T_{t,1}^\pi) & (M \leq t \leq N) \\ \text{MAX}(T_{t-M+1,M}^\pi, T_{t-M+2,M-1}^\pi, \dots, T_{N-1,t-N+2}^\pi, T_{N,t-N+1}^\pi) & (N < t \leq N+M-1) \end{cases}$$

(Fig. 1)은 스키드 투입 순서에 따른 전체 작업 시간 차이를 보이는 예이다. (Fig. 1A)는 네 종류의 스키드 A, B, C, D에 대한 두 개의 공정 X, Y의 각 작업 시간을 나타내며 반드시 공정 X가 처리된 후 공정 Y가 처리되어야 한다. (Fig. 1B)는 (Fig. 1A)의 정보를 바탕으로 세 개의 작업 순서 Case 1, Case 2, Case 3을 구성했을 때 각 스키드의 투입에 따라 진행되는 상황에 대한 간트차트이며 각 작업 순서의 마지막에 전체 작업 시간을 표기하였다. Case 1에서는 스키드 B의 공정 Y와 스키드 C의 공정 X에서 1의 대기 시간, 스키드 C의 공정 Y와 스키드 D의 공정 X에서 3의 대기 시간이 발생했다. Case 2에서는 스키드 B의 공정 Y와 스키드 A의 공정 X에서 3의 대기 시간, 스키드 A의 공정 Y와 스키드 C의 공정 X에서 1의 대기 시간, 스키드 C의 공정 Y와 스키드 D의 공정 X에서 3의 대기 시간이 발생했다. 마지막으로 Case 3은 스키드 A의 공정 Y와 스키드 D의 공정 X에서 2의 대기 시간, 스키드 D의 공정 Y와 스키드 B의 공정 X에서 1의 대기 시간, 스키드 B의 공정 Y와 스키드 C의 공정 X에서 1의 대기 시간이 발생했다. 각 대기 시간의 합을 구하면 Case 1은 4시간, Case 2는 7시간, Case 3은 4시간으로 가장 많은 대기 시간이 발생한 Case 2에서 전체 작업 시간이 가장 길게 나타났다. 또한, 같은 대기 시간이 발생한 Case 1과 Case 3의 전체 작업 시간은 Case 1에서 전체 작업 시간이 조금 더 길게 나타났다. 이처럼 일반적으로 전체 작업 시간은 대기 시간 차이에 영향을 받으며, 대기 시간이 같은 경우에도 차이가 있을 수 있음을 알 수 있다.

```

Function UniDev( $\Pi$ )
  //  $S^{\pi_i}$  is the  $i^{\text{th}}$ -order skid in schedule  $\Pi$ 
  bestSolution  $\leftarrow \Pi$ 
  for  $e=1$  to epoch
    Randomly select  $t \in \{1, \dots, N+M-1\}$  with a
      probability proportional to
      
$$\frac{1}{B-A+1} \sum_{k=A}^B u(S^{\pi_{t-k+1,k}})$$

      where  $A = \max(t-N+1, 1)$  and
       $B = \min(t, M)$ 
    if  $t < M$ 
       $m \leftarrow t$ 
    else if  $M \leq t < N$ 
       $m \leftarrow M$ 
    else
       $m \leftarrow N + M - t$ 
    Randomly select  $i \in \{t-m+1, \dots, t\}$  with a
      probability proportional to  $u(S^{\pi_{i,t-i+1}})$ 
     $x \leftarrow i$ 
    Randomly select  $y \in \{1, \dots, N\}$  with a probability
      inversely proportional to  $e(S^{\pi_x}, S^{\pi_y})$ 
    Swap the order of  $S^{\pi_x}$  and  $S^{\pi_y}$  in  $\Pi$ 
    if span( $\Pi$ ) < span(bestSolution)
      bestSolution  $\leftarrow \Pi$ 
  return bestSolution

```

Fig. 2. Pseudo Code of UniDev

### III. The Proposed Method

#### 1. Mean Absolute Deviation

스키드 컨베이어는 다음 공정으로 넘어가기 위해서 컨베이어 위에 올려져 있는 모든 스키드의 공정이 종료되어야 한다. 따라서 컨베이어 위에서 발생하는 병목 현상은 컨베이어 위에 올려져 있는 스키드 사이의 작업 대기 시간의 차이로 발생한다고 할 수 있다. 본 논문에서는 컨베이어 위에서 진행되는 스키드 사이의 작업 시간 차이를 최소화하여 병목으로 낭비되는 시간을 줄여 전체 작업 시간을 최소화하는 방법을 제안한다. 이를 위해 스키드 투입 순서  $\pi$ 로부터 간트차트를 구성한 후, 컨베이어 위에 올려져 있는  $i$ 번째 스키드  $j$ 번째 공정  $S_{i,j}^{\pi}$ 에 관한 작업 시간의 절

대 편차를 수식 (3)과 같이 정의하였다.

•  $m(S_{i,j}^{\pi})$  :  $S_{i,j}^{\pi}$ 의  $j$ 번째 공정과 동시에 컨베이어 위에 처리 중인 모든 공정의 작업 시간 평균

•  $u(S_{i,j}^{\pi})$  :  $S_{i,j}^{\pi}$ 의  $j$ 번째 공정에서 발생하는 절대 편차

$$m(S_{i,j}^{\pi}) = \frac{1}{M} \sum_{k=1}^M T_{i+j-k,k} \quad (2)$$

$$u(S_{i,j}^{\pi}) = |T_{i,j}^{\pi} - m(S_{i,j}^{\pi})| \quad (3)$$

#### 2. UniDev (Uniform Deviation-Based Heuristic)

본 논문에서는 매 탐색마다 서로 다른 두 개의 스키드  $x, y$ 에 대한 투입 순서를 교체하는 UniDev (Uniform Deviation-Based) 휴리스틱 탐색 방법을 제안한다. 이때, 스키드  $x$ 는 전체 작업 스케줄 중 작업 시간의 절대 편차 정보를 기반으로 선택하고, 스키드  $y$ 는 스키드  $x$ 와 교체가 이루어졌을 때, 교체 후 스키드  $x$ 와 스키드  $y$ 의 편차 합 정보를 기반으로 선택한다. 이를 위해 스키드  $y$ 를 선택하는 과정에서 필요한 스키드  $x$ 와 스키드  $y$ 의 교체 후 편차 합  $e(S_x^{\pi}, S_y^{\pi})$ 를 아래와 같이 정의하였으며, 이를 이용한 UniDev 휴리스틱을 (Fig. 2)와 같이 구현하였다.

$$e(S_x^{\pi}, S_y^{\pi}) = \sum_{k=1}^M |T_{x,k}^{\pi} - m(S_{y,k}^{\pi})| + \sum_{k=1}^M |T_{y,k}^{\pi} - m(S_{x,k}^{\pi})| \quad (4)$$

(Fig. 2)에서 보듯이 UniDev는 초기 스키드 투입 순서  $\Pi$ 를 bestSolution으로 시작하여 epoch의 크기만큼 탐색을 진행한다. 먼저, 컨베이어 위에서 진행되는  $N+M-1$ 의 각 작업 시점에 대한 평균절대편차를 구한다. 작업 시점은 각 평균절대편차의 크기에 비례한 확률 분포로 하여 선택한다. 스키드  $x$ 는 선택된 작업 시점의 절대편차의 크기에 비례한 확률 분포로 하여 선택한다. 스키드  $x$ 와 편차합에 반비례하는 확률로 스키드  $y$ 를 선택한다.  $\Pi$ 는 선택된 두 스키드  $x, y$ 의 순서를 교체하여 업데이트 한다. 마지막으로, bestSolution는  $\Pi$ 와 전체 작업 시간을 비교하여 성능 개선이 있을 때 업데이트 한다. 작업 시간을 공유하는 인접한 스키드 사이의 관계로 인해 매 탐색 시 개선이 이루어지는 것은 아니다. 하지만 확률 분포를 통한 선택으로 다양한 해를 탐색할 수 있으며, 평균절대편차가 큰 구간의 교체를 진행함으로써 대기 시간 감소를 기대할 수 있다.

### IV. Experiments and Results

본 논문에서 제안한 UniDev 방법의 성능 비교를 위한 실험을 Table 1과 같이 네 종류의 방법을 구성하였다.

Table 1. Characteristics of Compared Methods

Features	Multi-Start	2-OPT	UniDev	UniDevHalf
Search Type	Random	Iterative Heuristic	Iterative Heuristic	Iterative Heuristic
Swap Element Selection	NA	Deterministic	Probabilistic	Random + Probabilistic
Search Area	Global	Local	Local	Local
Convergence Rate	Low	High	High	High

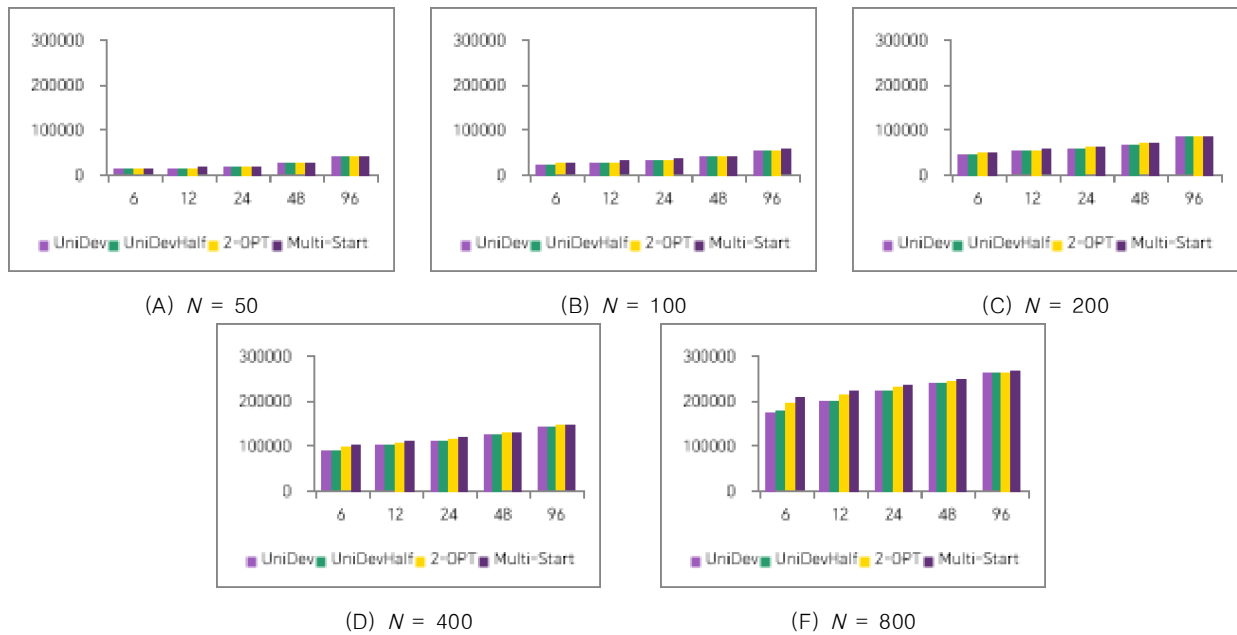


Fig. 3. Comparison Results of Span Time according to the Number of Skid

Multi-Start 방법은 다수의 임의 해를 탐색한 후 그 중 최적 해를 반환한다. 특정 순서 사이의 교체 과정이 없고 전역적으로 탐색하며 수렴도가 매우 낮은 특징이 있다. 2-OPT 방법은 전체 작업 시간의 개선이 없을 때까지 인접한 투입 순서끼리 교체를 반복한다. 교체 대상이 정해져있고 지역적으로 탐색하며 수렴도가 높은 특징이 있다. UniDevHalf 방법은 교체할 투입 순서를 임의로 결정하며 교체할 위치는 UniDev 방법과 동일하다. 앞의 2-OPT와 UniDev 방법 대비 임의의 투입 순서를 선택하기 때문에 상대적으로 수렴도가 낮다. 각 방법의 실험은 다양한 문제 크기에 대한 비교를 위해 스키드 개수( $N=50,100,200,400,800$ )와 공정 개수( $M=6,12,24,48,96$ )의 크기를 변화시켜 진행하였으며 각 방법의 최종해는 최대 30,000회의 탐색 과정 중 발견된 최적해를 의미한다. 총 30번 반복 실험을 통해 전체 작업 시간의 평균에 대한 결과 비교를 (Fig. 3)에 나타내었다. 실행 시간은 매 실험 중 최대 탐색 횟수까지 소요된 시간을 측정했다. 실행 시간 평균에 대한 결과 비교를 (Fig. 4)에 나타내었다(상세한 결과는 각각 Table 2과 Table 3 참조). 이때, Multi-Start 방법의 경우 임의로 구성된 투입 순서만을 사용하기 때문에 탐색 시간 결과에서 제외하였다.

(Fig. 3)에서 보듯이 본 연구에서 제안하는 UniDev와 UniDevHalf 방법은 Multi-start와 2-OPT보다 좋은 성능을 보여주었다. 또한 모든 방법들에 대해서 총 작업 시간은 스키드 개수  $N$ 과 공정 수  $M$ 에 비례하였다. 하지만 성능 향상의 크기는 스키드 개수  $N$ 에 비례한 반면 공정 수  $M$ 과는 반비례하는 경향을 보였다. 이는 공정 수  $M$ 이 커짐에 따라 컨베이어 위에서 작업 시간을 공유하는 스키드의 수가 증가하기 때문이다. 한편, (Fig. 4)에서 보듯이 탐색 시간 역시 스키드 개수  $N$ 과 공정 수  $M$ 에 비례하였다. 비교적 작은 크기의 문제에서는 제안된 방법이 2-OPT 방법 대비 더 많은 탐색 시간이 걸렸다(UniDev의 경우  $N \leq 400$ , UniDevHalf의 경우  $N \leq 200$ ). 반면 제안하는 방법이 문제의 크기에 따라 탐색 시간의 증가 속도가 2-OPT보다 작아서 스키드 개수  $N \geq 400$  일 때 UniDevHalf 방법이, 스키드 개수  $N=800$  일 때 UniDev 방법이 2-OPT 방법보다 빠르게 탐색하였다. 마지막으로 UniDevHalf 방법의 경우 전체 작업 시간에 관한 성능은 UniDev 방법에 비해 유사하거나 약간 뒤처지지만 교체할 스키드를 선택하는 과정이 생략되어 탐색 시간은 (Fig. 4)와 같이 거의 절반에 가까운 수준으로 감소하였다.

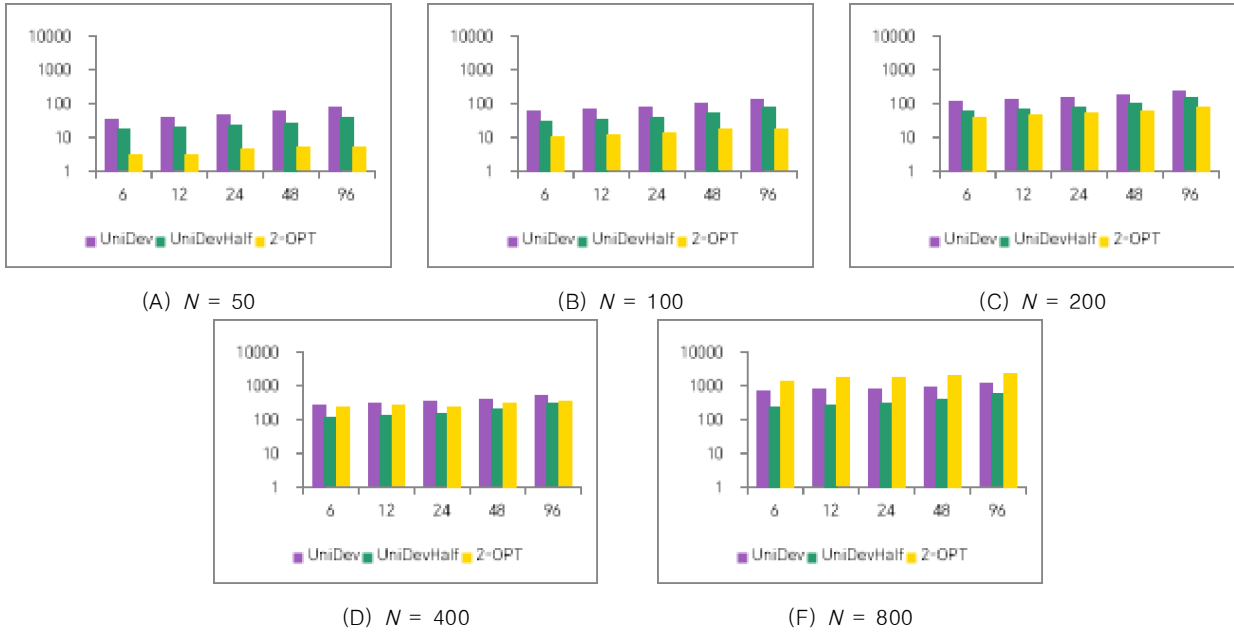


Fig. 4. Comparison Results of Running Time according to the Number of Skid

Table 2. Details of Span Time Results

( N , M )	$\mu ( \sigma )$			
	UniDev	UniDevHalf	2-OPT	Multi-Start
( 50 , 6 )	12731 (1274)	12741 (1289)	13014 (1153)	13327 (1101)
( 50 , 12 )	15479 (977)	15465 (963)	15709 (873)	15963 (750)
( 50 , 24 )	18990 (826)	18974 (823)	19329 (770)	19619 (741)
( 50 , 48 )	25994 (848)	26020 (842)	26440 (707)	26751 (658)
( 50 , 96 )	40615 (898)	40628 (902)	41029 (792)	41330 (705)
( 100 , 6 )	24427 (2566)	24431 (2564)	25452 (2310)	26393 (2187)
( 100 , 12 )	27639 (953)	27671 (971)	28806 (889)	29704 (791)
( 100 , 24 )	32515 (898)	32527 (896)	33538 (762)	34191 (701)
( 100 , 48 )	40144 (1052)	40161 (1029)	41141 (803)	41667 (672)
( 100 , 96 )	54203 (558)	54224 (578)	55304 (507)	55911 (491)
( 200 , 6 )	45706 (2711)	45725 (2729)	48917 (2444)	51239 (2294)
( 200 , 12 )	52543 (2064)	52522 (2052)	55376 (1690)	57205 (1589)
( 200 , 24 )	59649 (880)	59714 (970)	61882 (924)	63174 (870)
( 200 , 48 )	67962 (802)	67940 (800)	70018 (626)	71059 (581)
( 200 , 96 )	83168 (476)	83160 (505)	85194 (378)	86112 (336)
( 400 , 6 )	87783 (4510)	87937 (4460)	97254 (3748)	103295 (3491)
( 400 , 12 )	101550 (2467)	101609 (2440)	108581 (2278)	112847 (2009)
( 400 , 24 )	112580 (1498)	112601 (1471)	117385 (1359)	120258 (1215)
( 400 , 48 )	124367 (1228)	124387 (1228)	127953 (1101)	129917 (993)
( 400 , 96 )	141485 (657)	141523 (688)	144461 (608)	145983 (534)
( 800 , 6 )	174840 (4769)	175244 (4763)	194336 (4372)	206832 (4037)
( 800 , 12 )	200210 (3801)	200320 (3911)	214078 (3390)	223149 (2971)
( 800 , 24 )	222105 (2195)	222091 (2173)	230746 (2023)	236438 (1788)
( 800 , 48 )	239759 (1297)	239821 (1289)	245255 (1108)	248825 (995)
( 800 , 96 )	259606 (741)	259768 (799)	263703 (613)	266220 (599)

Table 3. Details of Running Time Results

( $N$ , $M$ )	$\mu$ ( $\sigma$ )		
	UniDev	UniDevHalf	2-OPT
( 50 , 6 )	36 (1)	17 (0)	3 (1)
( 50 , 12 )	40 (2)	19 (1)	3 (1)
( 50 , 24 )	47 (2)	22. (1)	4 (2)
( 50 , 48 )	59 (2)	28 (1)	5 (2)
( 50 , 96 )	83. (3)	39 (1)	5 (2)
( 100 , 6 )	63 (2)	30 (1)	10. (4)
( 100 , 12 )	69 (3)	34. (2)	12 (3)
( 100 , 24 )	79 (4)	40 (1)	13 (5)
( 100 , 48 )	99 (5)	52 (2)	17 (6)
( 100 , 96 )	136 (6)	74 (3)	17 (5)
( 200 , 6 )	120 (6)	58 (3)	41 (11)
( 200 , 12 )	130 (6)	64 (2)	43 (16)
( 200 , 24 )	147 (7)	77 (3)	50 (17)
( 200 , 48 )	180. (9)	100 (4)	62 (21)
( 200 , 96 )	244 (11)	144 (6)	78 (32)
( 400 , 6 )	281 (15)	113 (5)	217 (69)
( 400 , 12 )	299 (16)	127 (5)	250 (92)
( 400 , 24 )	330 (18)	152 (7)	247 (82)
( 400 , 48 )	390 (21)	198. (9)	313 (95)
( 400 , 96 )	509 (28)	286 (12)	333 (168)
( 800 , 6 )	727 (42)	236 (10)	1409 (404)
( 800 , 12 )	756 (43)	264 (11)	1695 (671)
( 800 , 24 )	814 (41)	314 (14)	1904 (716)
( 800 , 48 )	930 (47)	407 (17)	2059 (952)
( 800 , 96 )	1138 (61)	583 (26)	2400 (667)

## V. Conclusions

본 논문에서는 조선 소조립 공정에서 스키드 컨베이어 환경을 대상으로 전체 작업 시간을 최소화하기 위한 스키드 투입 순서 최적화 방법으로 UniDev 방법을 제안하였다. 제안된 방법의 성능을 확인하기 위해 다양한 크기의 문제에서 Multi-Start 방법 및 2-OPT 방법과 비교를 진행하였고, 문제의 크기가 커질수록 UniDev 방법이 전체 작업 시간을 줄이는데 효율적임을 확인하였다. 하지만 UniDev는 스키드 정보만을 바탕으로 최적화를 진행하기 때문에 우선시 되어야 할 스키드, 실제 작업자의 작업 효율, 작업 공간 등 현실적인 요소를 반영하는데 한계가 있다. 향후 연구에서는 현실 문제에서 고려될 수 있는 스키드의 우선 순위 및 작업 환경을 반영한 연구를 수행하고자 한다.

## ACKNOWLEDGEMENT

This work was supported by Institute of Information & Communication Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2020-0-00869, Development of 5G-based Shipbuilding & Marine Smart Communication Platform and Convergence Service)

## REFERENCES

- [1] J. G. Bark, M. G. Kim "Optimization of Quantity Allocation using Integer Linear Programming in Shipbuilding Industry" JSNAK, Vol.57, No.1, pp.45-51, Jan, 2020. DOI: 10.3744/SNAK.2020.57.1.045
- [2] I. H. Osman, J. P. Kelly "Meta-heuristics theory and applications" JORS, Vol.48, No.6, pp.657-657, Jan, 1997. DOI: 10.1057/palgrave.jors.2600781

- [3] J. S. Park, J. B. Im, S. H. Ji "Optimization of Satellite Structures by Simulated Annealing" KSME, A, Vol.29, No.2, pp.262-269, Feb, 2005. DOI: 10.3795/KSME-A.2005.29.2.262
- [4] H. C. Kim "Optimization of Bi-criteria Scheduling using Genetic Algorithms" JICS, Vol.6, No.6, pp.99-106, Dec, 2005.
- [5] S. G. Min, C. U. Moon, M. W. Lee, K. H. Chung, J. C. Park "A Genetic algorithm application for the load balancing of ship erection process" IE Interfaces, Vol.13, No.2, pp.225-233, Jun, 2000.
- [6] G. G. Jo, S. C. O, T. Y. Yang "Operations Scheduling for Multi-item, Small-sized Production" KIIE, Vol.11, No.2, pp.57-73, Dec, 1985.
- [7] J. W. Lee, H. J. Kim "Erection Process Planning & Scheduling using Genetic Algorithm" SNAK, Vol.32, No.1, pp.9-16, Feb, 1995.
- [8] H. R. Choi, K. R. Ryu, K. K. Cho, H. S. Lim, J. H. Hwang "A Scheduling System for Panel Block Assembly Shop in Shipbuilding using Genetic Algorithms" KIISS, Vol.2, No.2, pp.29-42, Dec, 1996.
- [9] J. Y. Park, J. J. Seo, H. J. Kang "Optimization of Robot Welding Process of Subassembly Using Genetic Algorithm in the Shipbuilding" JWJ, Vol.27, No.2, pp.57-62, Apr, 2009. DOI: 10.5781/KWJS.2009.27.2.057
- [10] H. C. Bae, K. C. Park, B. C. Cha, I. K. Moon "Scheduling of Shipyard Sub-assembly Process using Genetic Algorithms" IE Interfaces, Vol.20, No.1, pp.33-40, Mar, 2007.
- [11] I. H. Hwang, J. Noh, K. K. Lee, J. Shin "Short-term Scheduling Optimization for Subassembly Line in Ship Production Using Simulated Annealing" KSFS, Vol.19, No.1, pp.73-82, Mar, 2010.

## Authors



Kyung-Tae Lee received the B.S. and M.S. degrees in Computer Science and Engineering from Ulsan University, Korea, in 2018 and 2021 respectively. Lee is currently a Student in the Department of Computer Science,

Ulsan University. He is interested in genetic algorithm.



Yung-Keun Kwon received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Seoul University, Korea, in 1999, 2001 and 2006, respectively. Kwon is currently a Professor in the Department of

Computer Science, Ulsan University. He is interested in genetic algorithm, and optimization.