

# 컨베이어 분기점에서의 최적 인출 컨베이어 선택 문제

한 용 회\*

삼성전자 DS부문

## Optimal Conveyor Selection Problem on a Diverging Conveyor Junction Point

Yong-Hee Han<sup>†</sup>

Device Solution Division, Samsung Electronics

This research investigates the problem of minimizing setup costs in resequencing jobs having first-in, first-out(FIFO) constraints at conveyORIZED production or assembly systems. Sequence changing at conveyor junctions in these systems is limited due to FIFO restriction. We first define the general problem of resequencing jobs to workstations satisfying precedence relationships between jobs(Generalized Sequential Ordering Problem, GSOP). Then we limit our scope to FIFO precedence relationships which is the conveyor selection problem at a diverging junction(Diverging Sequential Ordering Problem, DSOP), modeling it as a 0-1 integer program. With the capacity constraint removed, we show that the problem can be modeled as an assignment problem. In addition, we proposed and evaluated the heuristic algorithm for the case where the capacity constraint cannot be removed. Finally, we discuss the case study which motivated this research and numerical results.

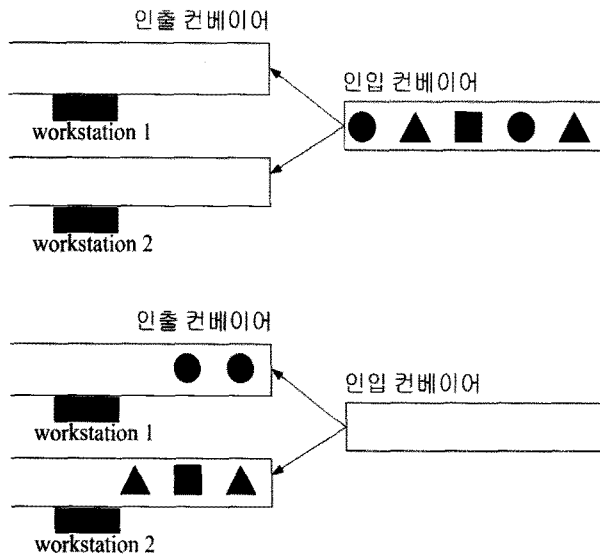
**Keywords :** Conveyor System, Setup Minimization, Resequencing, Sequential Ordering Problem

### 1. 서론

최근 IT 기술의 눈부신 발전 및 이를 이용한 공장 자동화 측면에서의 진보는 생산 유연성(manufacturing flexibility)을 과거에 비해 크게 향상시켰다. 현 시점에서 운용되고 있는 상당수의 대량 생산 시스템은 예전 방식의 transfer 라인처럼 보이지만 사실은 매우 유연하고 매우 다양한 종류의 제품(전자제품, 자동차, 기타 제품)을 하나의 라인에서 생산 가능하다. 생산 유연성이 중요한 요인 중 하나는 생산 유연성이 셋업/교체에 필요한 비용 및 다양한 소비자의 요구사항에 대응하는 데 걸리는 시간과 밀접하게 관련되어 있기 때문이다. 하지만, 가장 유연한 생산 시스템에 있어서도 셋업은 불가피하게 발생하며,

많은 경우에 있어서 이러한 셋업의 발생은 가공되는 물품의 가공 순서를 바꿈으로써(resequencing) 감소될 수 있다.

컨베이어(conveyor)는 현대 대량 생산 시스템에서 가장 자주 쓰이는 물품 이송 방법이다. 컨베이어는 상대적으로 간단한 제어를 통해 대량의 물품을 이송할 수 있으며, 컨베이어 자체가 생산 공정 사이의 버퍼(buffer) 역할을 수행한다. 컨베이어의 구조상 컨베이어의 각 동작 단위(segment)는 선입 선출(FIFO) 방식으로 작동하며, 컨베이어 동작 단위 내에서의 물품간 순서(sequence)는 고정되어 있다. 셋업의 발생 여부가 물품의 특정 속성(색상, 규격, 중량 등)에 의해 결정될 경우 - 즉, 직전에 가공된 물품의 특정 속성과 현재 가공되는 물품의 동일 속성이 상이할 경우에만 셋업이 발생할 경우 - 셋업에 직접적으로 영향을 미



<그림 1> 컨베이어 분기점을 이용한 셋업 감소 예제

치는 요소는 가공 기계로 물품을 운송하는 컨베이어 상에서의 물품의 순서이다. 컨베이어 상에서의 물품간 순서를 바꾸기 위해서는 일반적으로 bypass, transfer, merge, diverge, spur, 자동 창고(AS/RS), 오프라인(off-line) 버퍼(컨베이어 자체는 인라인(in-line) 버퍼의 일종이라고 볼 수 있음) 등의 전용 설비가 필요하다.

컨베이어 상에서의 물품간 순서에 따라 셋업이 발생하는 대표적인 현장 사례는 자동차 도장 공정에서의 도료 색상 변경이다. 도장 공정은 크게 전처리(pretreatment), 하도/전착(electro deposition), 중도(primer coat line), 상도(top coat line), 검사/수리(inspection and repair line) 공정으로 구분되는데, 하도 공정은 차체에 전착 도료를 부착시키는 과정이며, 중도 공정은 자동 도장기나 Air Spray Gun을 사용하여 도료를 부착하는 공정이고, 마지막으로 상도 공정은 자동차 차체의 색깔을 결정하는 공정으로 다양한 종류의 도료가 사용된다. 이 과정에서 중도 공정까지는 차체의 색깔에 상관없이 같은 도료를 사용하여 진행되는 데 반해 상도 공정에서는 고객/대리점에서 주문한 색상에 따라 사용되는 도료가 결정된다. 상도 공정의 경우 차량 도색 직전에 도장한 차량과 도장 대상 차량의 색상이 다를 경우 자동 도장기 내부의 잔류 도료를 신너(thinner)를 이용하여 세정한 후 새로운 도료를 투입하여 도장 작업이 이루어진다(문덕희 외[4]). 따라서 동일한 색상의 차종이 연속해서 상도 부스로 투입되지 않을 경우 잔류 도료의 세정(셋업) 비용이 발생하게 된다. 이러한 세정 비용을 줄이기 위해 생산 현장에서는 일반적으로 CRS (Color Rescheduling Storage, Selectivity Bank 혹은 Color Selection Lane이라고도 부름)-상도 부스 진입 전에 생산 순서를 동일 색상별로 그룹화해주는 별도 저장 공간을 설

치, 사용한다.

물품 간 순서 변경을 위해 CRS와 같은 전용 설비를 사용할 경우, 설비 구입 및 설치 관련 추가 비용이 발생하고 라인내 공간을 점유한다(라인내 공간 점유 문제는 차량과 같은 대규모의 물품 이송시 더욱 중요한 문제가 됨). 반면 <그림 1>에서 예시한 바와 같이 컨베이어 연결점(junction point)에서의 컨트롤 로직에 따라서도 연결점 전과 연결점 후의 물품간 순서 변경이 가능하다.

컨베이어 연결점을 이용한 물품간 순서 변경은 별도 설비 구입 및 설치가 불필요하고 별도 공간 점유가 없는 장점이 있다. 본 연구의 주제는 이러한 컨베이어 연결점들을 이용한 셋업 최소화 문제의 모델링 및 분석이다. 구체적으로, 컨베이어의 분기점(diverging point)에서 셋업의 최소화를 위해 물품을 어떤 컨베이어로 인출해야 하는지에 대한 최적화 문제를 수학적으로 모델링했다.

참고로 셋업 발생 빈도를 측정하는 여러 지표 정의가 가능하지만, 본 연구에서는 CRS의 효율을 평가하는데 일반적으로 사용되는 그룹화 효율(Grouping Ratio) 지표를 사용하기로 한다. 그룹화 효율은 통과 물품 수/셋업 횟수로 정의되며, 지표값과 셋업 비용과는 반비례관계가 성립한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 관련 연구 성과를 조사하고 제 3장에서는 컨베이어의 분기점에서의 의사 결정 문제를 정의하고 정수계획법으로 모델링하였다. 제 4장에서는 제 3장에서 논의된 모델에서 capacity가 제약 조건이 아닌 경우 capacity 제약식을 제외함으로써 해당 문제가 할당 문제(assignment problem)로 변환됨을 보였고, 제 5장에서는 capacity 제약식 제거가 불가능할 경우에 사용 가능한 휴리스틱 알고리즘을 제시하였다. 제 6장에서는 자동차 조립 공장의 도장 공정에의 적용 예제 및 시뮬레이션 모델 구축을 통한 모델 검증, 마지막으로 제 7장에서는 본 논문의 요약 및 향후 연구 과제를 논하였다.

## 2. 관련 논문 조사

만일 상도 부스가 한 개이고 CRS가 라인형일 경우 (즉, CRS의 구성이 한 개의 공통 입구와 한 개의 공통 출구를 가진 여러 개의 평행한 컨베이어로 구성되고 입구와 출구가 상이할 경우) CRS상의 인출 라인 선택 문제는 SOP (Sequential Ordering Problem)라는 보다 일반적인 문제(즉, 도장 공정상의 문제에 한정되지 않고, 하나의 컨베이어에서 인출되는 물품에 대한 최적 인입 컨베이어를 선택하는 문제)로 정의 가능하다. Escudero[14]에 의해 처음 정의된 SOP는 우선순위 제약식을 가진 비대칭적 Hamilto-

nian 경로 문제로 정의될 수 있다. SOP는 NP-hard이며 (Ascheuer[8]) 비대칭적 외관원 문제와 밀접히 연관되어 있다. Spieckermann et al.[27]은 branch and bound 알고리즘을 SOP에 적용한 사례를, Bonsma et al.[10]은 SOP에서의 FIFO 제약이 없는 경우의 문제의 복잡도(complexity)를 다루었다.

Morley and Schelberg[24], Morley[22], Morley and Ekberg [23]는 트럭 조립 공장에서 트럭을 상도 부스에 할당시 총 makespan과 총 색상 변경 횟수를 최소화하는 알고리즘을 다루었다. 이들 논문은 시장 기반 경매 알고리즘을 미국 General Motors사의 조립 공장에 적용한 결과 이전 대비 그룹화 효율이 평균적으로 100% 상승하였다고 보고하였다. Gambardella and Dorigo[16]는 개미 알고리즘을 SOP에 적용하였다. Campos et al.[11]는 [22~24]에서 제시한 시장 기반 경매 알고리즘과 개미 알고리즘 기반 threshold 알고리즘을 비교하였고, 유전자 알고리즘을 사용하여 이들 두 알고리즘상의 파라미터값을 계산하였다. Kittithreerapronchai and Anderson[19]는 시장 기반 알고리즘 및 개미 알고리즘을 시뮬레이션 하였으며, 각 알고리즘상의 일부 파라미터 값은 목적함수의 변화와 거의 무관하므로 제거 가능성을 밝혔다. Asassi[9]는 자동 창고를 이용한 동일 색상 차종을 그룹화에 대해 논하였다. 여기서 자동 창고는 일종의 random access 오프라인 버퍼의 역할을 하며, 도장 공정 이전에는 그룹화 효율을 올리는 역할 및 도장 공정 이후에는 자동 창고 이전의 차량 순서를 회복하는 역할을 수행한다. Myron[25]는 대규모의 동일 색상의 차량 블록의 형성 관련 연구를 논하였다. 구체적으로, 원하는 위치의 차량의 인입 및 인출이 자유로운 자동 창고를 이용시 간단한 블록 보호 규칙의 준수만으로도 셋업 개수를 상당히 줄일 수 있음을 시뮬레이션을 통해 보였다. 상기 언급된 논문인[9, 11, 16, 19], [22~25]는 모두 인공지능 기법을 사용하였는데, 이러한 인공지능 기법을 이용한 접근 방법은 현장에 적용하기 쉽고 예기치 못한 공장 비가동 등의 시스템 변동에 효율적으로 대응 가능한 장점이 있는 반면, 해의 품질에 대한 보증이 불가능한 약점을 가지고 있다. 최적화 모델링 접근 방법은 Choe et al.[12]에서 최적해의 상한치(upper bound)를 얻기 위해 처음 사용되었으며, 그룹화 효율을 높이면서 부하의 풀림 방지를 위해 자동 창고가 사용될 경우에 대해 time window 를 가진 외관원문제로 모델링되었다.

Lahmar and Benjaafar[20]는 오프라인 버퍼를 이용한 resequencing문제를 연구하였으며, N개의 버퍼 공간을 가진 경우에 대한 문제를 1개의 버퍼 공간을 가지는 N개의 문제로 분할(decomposition)하여 휴리스틱 알고리즘을 개발하였다. 또한 resequencing과 속성 할당 (feature assignment)

을 동시에 고려하는 문제 (resequencing and feature assignment problem, RFAP)도 같이 논하였다. Lim and Xu[21]는 [20]에서 제기된 RFAP 문제를 beam 탐색 휴리스틱을 통한 반복적 탐색을 통해 짧은 시간 내 최적해를 찾는 방법을 논하였다. Färber and Coves[15]는 resequencing문제를 포함하는 전반적인 sequencing문제에 대한 서베이 리포트를 작성하였다.

컨베이어 시스템에서의 연결점을 이용한 셋업 감소는 SMT(surface mount technology, 표면 실장 기술) 조립 라인 등의 다른 생산 라인에도 적용 가능하며, Jeong et al.[18]는 SMT 조립라인에서 bypass 연결점을 가진 경우의 셋업 감소 방법에 대해 다루었다.

유사한 셋업 최소화 문제는 철도의 차량 분류 기지(classification yard)에서도 발생한다 (Siddiquee[26], Yagar et al.[28], Daganzo[13]). 철도의 차량 분류 기지에서의 셋업 최소화 문제와 본 논문에서 다룬 셋업 최소화 문제와의 가장 큰 차이점은 전자에서는 기정의된 화차간 순서를 100% 준수하는 것이 필요하며, 열차 내 화차가 100% 정확한 순서로 정렬될 때까지 지속적으로 개별 화차는 정해진 라인을 순환(re-humping)하나, 후자에서는 셋업을 줄이기 위한 목적으로 정해진 라인을 순환하지 않는다. 차량 분류 기지에서의 셋업 최소화 문제에 대해서도 혼류 조립 라인(mixed assembly line) 관점에서 일부 연구가 이루어졌으나, 본 연구에서 서술한 환경과 관련된 연구는 없었다.

국내 관련 논문으로는 CRS를 설치하고 운영하기 위한 소프트웨어 개발에 대한 연구(박해규 외[5]), CRS Layout 및 운영 알고리즘을 변경 시 그룹화 비율의 변화를 시뮬레이션으로 검토한 연구 (문덕희 외[3]), CRS 운영 시 입고 및 출고 알고리즘에 대한 연구([3], 최원준 외[7]), CRS 설치시의 투자 및 경제성 검토에 대한 연구[3]가 있다. 특히, [7]은 3.77대 내외, [3]은 5~6대의 그룹화 효율 달성이 보고되었다. 참고로 [3]은 국내 D 자동차회사, 기타 연구는 모두 국내 H 자동차회사를 대상으로 수행되었다. 상기 언급된 국내 관련 논문에서는 모두 각 자동차회사에 설치된 CRS의 운영 시 greedy 휴리스틱 알고리즘을 사용한 결과만 논의되었고, 수학적 모델링이나 최적화 문제에 대한 논의는 없었다.

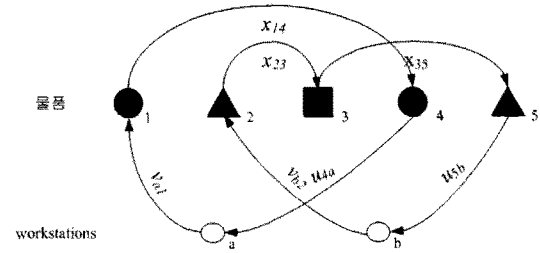
결론적으로, 다수 컨베이어에서 단일 컨베이어로 연결될 경우 셋업 최소화를 위한 인출 라인 선택 문제에 대한 최적화 문제에 대한 모델링 (SOP문제) 및 인공지능을 이용한 연구는 기존에 다수 이루어졌으나, 단일 컨베이어에서 다수 컨베이어로 분기될 경우 셋업 최소화를 위한 인입 라인 선택 문제에 대한 최적화 문제에 대한 모델링은 본 연구에서 최초로 다루어졌다(상세 내용은 하기 제 3장 참조).

### 3. GSOP와 DSOP 모델

순서가 있는(directed) 그래프  $G = (Y \cup Z, U \cup V \cup W \cup X)$ 에 대해 생각해보자. 여기서  $Y$ 는 물품 1, ...,  $N$ ,  $Z$ 는 물품을 가공하는 workstation 1, ...,  $Q$  ( $Y + Z = |Y \cup Z|$ )와 대응한다고 정의된다. 또한  $i, j \in Y$ 이고  $q, q' \in Z$ 일 때  $U, V, W, X$ 는 각각 arc  $(j, q), (q, i), (q', q), (i, j)$ 와 대응한다고 정의된다. arc  $(i, j)$ 와 연관된  $C_{ij} \in \mathcal{R}$ 은 물품  $i$ 의 가공 종료 후 물품  $j$ 의 가공 시작과 연관된 셋업 비용으로 정의된다. 참고로 셋업 비용은 workstation에서 연속으로 가공되는 두 물품의 속성(차량의 경우 색상 등)이 다를 경우에만 발생한다. arc  $(q, i)$ 와 연관된  $\overline{C}_q \in \mathcal{R}$ 은 workstation  $q$  자체와 연관된 셋업 비용(예 : workstation  $q$ 를 한 번이라도 사용 시 발생하는 비용)이며,  $U$ 와  $W$ 내의 arc와는 연관된 셋업 비용이 없다. 참고로 만약 arc  $(j, q)$ 와 연관된  $\overline{C}_q \in \mathcal{R}$ 를 workstation  $q$  자체와 연관된 셋업 비용으로 정의 가능하며, 이 경우  $V$ 와  $W$ 내의 arc와는 연관된 셋업 비용이 없도록 정의되어야 한다. 물품간 순서 제약은 별도의 acyclic digraph  $G' = (Y, P), P \subset \{(i, j) \mid i, j \in Y\}$ 로 정의되며, 물품  $i$ 가 모든 가능한 물품 순서(job sequence)에서 물품  $j$ 를 앞서는 경우 arc  $(i, j) \in P$ 이다. 참고로  $P$ 는 전이적 속성(transitive property)을 가진다. 즉, 만약 arc  $(i, j) \in P$ 이고 arc  $(j, k) \in P$ 이면, arc  $(i, k) \in P$ 가 성립한다.

상기 정의를 이용하여, GSOP(Generalized SOP)는 물품간 순서 제약  $P$ 를 만족하면서 총 셋업 비용(=  $C_{ij} + \overline{C}_q$ )을 최소화하는  $G$ 내의 각 workstation들을 위한 물품 순서를 찾는 문제라고 정의된다 (물품 순서에서 제외되는 물품은 없다고 가정함). 네트워크 문제로 재 정의하면, GSOP는 순서 제약  $P$  하에서 최소 비용으로  $G$ 내의 모든 node를 커버하는 (즉,  $S_1 \cup \dots \cup S_m = Y \cup Z, S_q \cap S_{q'} = \{\emptyset\}, 1 \leq q, q' \leq m \leq Q$ ) 상호 배타적인 directed cycles  $S_1, \dots, S_m$ 을 찾는 문제이다. 따라서 SOP는  $|Z| = 1$ 이고  $W = \{\emptyset\}$ 인 GSOP의 특수한 경우라고 정의될 수 있으며, 이 경우, SOP는 순서 제약  $P$ 를 만족하면서 총 비용을 최소화하는 Hamiltonian cycle을 찾는 문제와 동일하다. SOP는 또한 비대칭적 외판원 문제의 generalized case라고 정의될 수 있으며[7], 이 경우, 라인형 CRS상에서의 인출 라인 선택 문제는 SOP의 special case이다.

본 연구의 주제인 DSOP(Diverging SOP)는 순서 제약  $P' = \{(i, j) \mid i < j, (i, j) \in X\}$ 를 가지는 GSOP의 또 다른 special case로 정의된다. 참고로  $P'$ 는 선입 선출 제약이 모든 물품 사이에 적용되는 순서 제약으로 작용하는 경우에 대한 것이다. 단일 컨베이어에서 다수 컨베이어로 분기될 경우 셋업 최소화를 위한 인입 라인 선택 문제



<그림 2> DSOP 예제

가 DSOP로 모델링 가능하다. <그림 2>는 <그림 1>의 예제를 네트워크로 표현했다. 즉,  $Y = \{1, \dots, 5\}, Z = \{a, b\}, P' = \{(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)\}$ 인 경우의 DSOP 예제에 대한 한 가능해로서 a 컨베이어에는 물품 1 → 4, b 컨베이어에는 물품 2 → 3 → 5가 순서대로 인입되는 경우를 보여준다. 모델링을 위해 하기와 같이 변수 및 상수를 정의한다.

$$x_{ij} = \begin{cases} 1 & \text{물품 } j \text{가 물품 } i \text{가공 직후에가공되는 경우} \\ 0 & \text{기타경우} \end{cases}$$

$$u_{iq} = \begin{cases} 1 & \text{물품 } i \text{가 workstation } q \text{에서 최후에 가공되는 물품인 경우} \\ 0 & \text{기타경우} \end{cases}$$

$$v_{qi} = \begin{cases} 1 & \text{물품 } j \text{가 workstation } q \text{에서 최후에 가공되는 물품인 경우} \\ 0 & \text{기타경우} \end{cases}$$

$$w_{q'q} = \begin{cases} 1 & \text{workstation } q' \text{에는 물품이 할당되지 않고 workstation } q \text{에는 최소 1개 이상의 물품이 할당되는 경우} \\ 0 & \text{기타경우} \end{cases}$$

$Q$  = workstation 개수

$N$  = 물품 개수

$A$  = load balancing 제약

상기 정의된 변수 및 상수를 이용하여, DSOP는 하기 정수계획법 모델로 모델링된다.

$$\text{Minimize } \sum_{i=1}^{N-1} \sum_{j=i+1}^N C_{ij} x_{ij} + \sum_{j=1}^N \overline{C}_q v_{qi} \quad (1)$$

$$\sum_{j=i+1}^N x_{ij} + \sum_{q=1}^Q u_{iq} = 1 \quad \forall i \quad (2)$$

$$\sum_{i=1}^{j-1} x_{ij} + \sum_{q=1}^Q v_{qj} = 1 \quad \forall j \quad (3)$$

$$\sum_{i=1}^N u_{iq} + \sum_{q'=1}^Q w_{q'q} = 1 \quad \forall q \quad (4)$$

$$\sum_{j=1}^N v_{qj} + \sum_{q'=1}^Q w_{q'q} = 1 \quad \forall q \quad (5)$$

$$\sum_{j=0}^A x_{i+j, i+j+1} \leq A \quad \forall i \quad (6)$$

$$x_{ij}, u_{iq}, v_{qj}, w_{q'q} \in \{0, 1\} \quad \forall i, j, q, q' \quad (7)$$

제약식 (2)에 의해 각 물품은 후속으로 가공될 물품을 할당받거나( $\sum_{j=i+1}^N x_{ij} = 1$ ) 특정 workstation에서 최후에 가공되는 물품이 된다( $\sum_{q=1}^Q u_{iq} = 1$ ). 유사한 방식으로, 제약식 (3)에 의해 각 물품은 선행으로 가공될 물품을 할당받거나( $\sum_{i=1}^{j-1} x_{ij} = 1$ ) 특정 workstation에서 최초로 가공되는 물품이 된다( $\sum_{q=1}^Q v_{qj} = 1$ ). 순서 제약 P'에 의해 제약식 (2)와 (3)에서 index i는 index j보다 작게 유지되어야 한다. 제약식 (4)에 의해 각 workstation은 최초로 가공되는 물품을 하나 할당받거나( $\sum_{i=1}^N u_{iq} = 1$ ) 전혀 할당받지 못한다( $\sum_{q=1}^Q w_{q'q} = 1$ ). 유사하게, 제약식 (5)에 의해 각 workstation은 최후로 가공되는 물품을 하나 할당받거나( $\sum_{j=1}^N v_{qj} = 1$ ) 전혀 할당받지 못한다( $\sum_{q=1}^Q w_{q'q} = 1$ ). 참고로,  $w_{q'q}$ 의 정의에 의해,  $u_{jq} = w_{q'q} = 1$ 인 경우, 물품 j는 workstation q(q'가 아님)에서 최후에 가공되는 물품이 된다. 제약식 (4)와 (5)에서 최소 1개 이상 물품을 할당받은 workstation과 전혀 할당받지 못한 workstation이 구분되며, 이 정보는 해당 workstation의 setup cost와 연계되어 목적함수에서 고려되어진다( $\sum_{j=i+1}^N \bar{C}_q v_{qj}$ ). (6)은 line balancing 제약식으로, 인출 라인 선택시 연속적으로 A + 1번 이상 특정 workstation을 선택하지 못하게 하며, 특정 workstation으로 물품이 몰릴 경우 해당 workstation이 capacity 제약으로 인해 bottleneck이 되는 경우를 방지한다. 마지막으로, 모델의 의미상 모든 변수는 정수이어야 하므로 제약식 (7)이 추가되었다.

#### 4. Capacity가 제약 조건이 아닌 경우의 최적해 계산 방법

제 3장에서 제시된 정수계획법 모델의(complexity는 NP hard인 관계로 상대적으로 작은 크기의 데이터에 대한 최적해를 구하는 데에도 상당한 시간이 소요된다. 일반적

으로 DSOP와 연관된 현장에서의 의사 결정 문제(최적 인출 라인 결정)가 실시간 의사 결정 문제와 유사한 속성-현 물품에 대한 인출 라인 결정은 다음 물품이 컨베이어에서 인출되기 전까지 완료되어야 하며, 도장 공정의 경우 인출되는 차량 사이의 시간은 일반적으로 1분 내외임을 가지는 점을 감안할 때 이는 해당 알고리즘의 현장 적용을 곤란하게 만드는 심각한 문제가 된다. 현장 적용과 관련된 또 한 가지 문제점은 생산 현장에서의 컨베이어의 제어가 일반적으로 PLC(Programmable Logic Controller)에 의해 이루어진다는 점이다. PLC는 일반적으로 PC(Personal Computer)에 비해 연산 속도 및 메모리 용량에 있어 한계를 가진다. 따라서 정수계획법 문제에 대한 최적해를 구하는 알고리즘을 PLC에서 구현하기는 현실적으로 불가능하다. 결과적으로 PC에 기반을 둔 별도 시스템 구축이 불가피하고, 이는 전체 시스템 구축 비용의 증가로 나타나게 된다. 상기 두 가지 이유에 의해 최적해에 근사하는 양질의 해를 빠른 시간 내에 구할 수 있는 알고리즘 구현이 필요하다.

상기 정수계획법 모델에서 제약식 (6)은 workstation의 capacity가 한정되어 있을 경우 전체 라인의 load balancing을 위해 적용 필요하다. 하지만 각 workstation의 처리 속도가 충분히 빠르기 때문에 임의의 물품 순서가 각 workstation에 할당 가능할 경우, 제약식 (6)은 제거 가능하다. 만일 (6)이 제거될 경우, 제약식 (7)도 total unimodularity 성질을 이용하여 제거 가능하다. 제약식 (6) 및 (7)이 제거된 선형계획법 모델은 할당 문제가 되며 Hungarian 방법에 의해  $O(N^3)$ 의 복잡도로 최적해 획득이 가능하다. 또한 Hungarian 방법에 의해 구해진 해는 (6) 및 (7)이 포함된 원 문제에 대한 최저 한계치(low bound)로 사용이 가능하다.

#### 5. Capacity가 제약 조건인 경우의 Rule-Based 휴리스틱 알고리즘

제 4장에서 설명된 대로, DSOP와 연관된 현장에서의 의사 결정을 위해서는 빠른 시간 내 양질의 해의 획득이 필요하며, Capacity가 제약 조건이 아닌 경우에는 제 4장에 설명된 방법에 의해 빠른 시간 내 최적해 계산이 가능하다. 하지만 Capacity가 제약 조건(즉, 제약식 (6)이 binding constraint)인 경우에는 polynomial complexity를 가지는 최적해 획득 방법이 존재하지 않는 관계로 양질의 해를 빠른 시간 내 획득할 수 있는 휴리스틱 알고리즘 개발이 필요하며, 해당 알고리즘이 PLC에 구축되어야 하고 PLC 프로그래밍(ladder diagram)이 PC 프로그래밍 대비 제약이 많은 점을 감안한다면 규칙 기반(= rule-based) 휴리스

틱이 가장 적절한 해결책이라 판단된다. 또한 규칙 기반 휴리스틱은 타 알고리즘 대비 상대적으로 이해하기가 쉬워 현장 관계자들의 이해 및 협조를 얻기가 용이하며, 이는 규칙 기반 휴리스틱이 타 알고리즘 대비 제조 현장에서 선호되는 가장 큰 이유이기도 하다. 제안된 휴리스틱 알고리즘은 다음과 같다.

- $A_q$  = 인출 컨베이어  $q$ 가 최근 연속으로 선택된 횟수
- $L_q$  = 인출 컨베이어  $q$ 에 가장 마지막으로 인입된 물품
- $D_k$  =  $k$ 번째 인출 대기중인 물품 ( $D_0$  = 현재 인출 예정인 물품의 속성)
- $C(Z)$  = 물품  $Z$ 의 속성
- $E, F$  = 알고리즘에서 사용되는 임시 집합
- $h, G_q$  = 알고리즘에서 사용되는 임시 변수

- step 1 : set  $A_q \leftarrow 0$  for  $\forall q, E \leftarrow \{\emptyset\}, h \leftarrow 1$ .
- step 2 : for all  $k (k \leq q)$  having  $A_k < A$ ,  
if  $C(D_0) = C(L_q)$ , then add  $k$  to  $E$ .
- step 3 : if  $E = \{\emptyset\}$ , then goto step 5  
else goto step 4
- step 4 : for all  $q \in E$ ,  
choose  $q^*$  with the smallest  $A_q$  value.  
goto step 8
- step 5 : for all  $k \notin F$  having  $A_k < A$ ,  
if  $C(D_h) = C(L_q)$ , then  
{  
add  $k$  to  $F$   
set  $G_k \leftarrow 10^h$   
}  
}
- step 6 : if  $|F| < k$ , then  
{  
set  $h \leftarrow h+1$   
goto step 5  
}  
else goto step 7
- step 7 : for all  $q \in F$ ,  
choose  $q^*$  with the smallest  $G_q$  value.  
for the same  $G_q$  value, choose  $q^*$  with the  
smallest  $A_q$  value.
- step 8 : send  $D_0$  to conveyor  $q^*$   
set  $L_{q^*} \leftarrow C(D_0)$   
set  $D_k \leftarrow D_{k+1}$  for  $\forall k$   
if  $A_{q^*} > 0$ , then set  $A_{q^*} \leftarrow A_{q^*} + 1$   
else set  $A_{q^*} \leftarrow 1$   
set  $E \leftarrow \{\emptyset\}, F \leftarrow \{\emptyset\}$

set  $h \leftarrow 1, G_q \leftarrow 0$  for  $\forall q$   
goto step 2

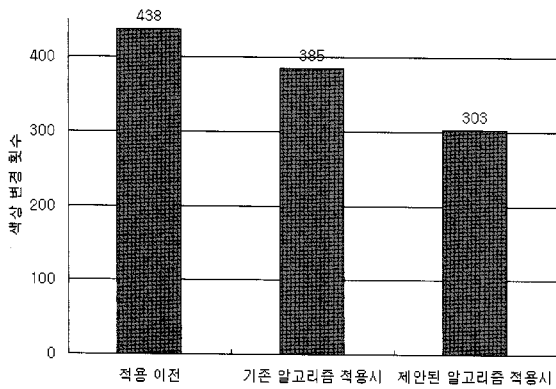
상기 알고리즘의 기본 개념은 인입 대기중인 차량의 속성(=  $C(D_h)$ )과 각 인출 컨베이어에 마지막으로 인입된 차량의 속성(=  $C(L_q)$ )을 비교하여 각 컨베이어의 적합도( $G_q$ )를 평가하는 것이다.  $C(D_0) = C(A_q)$ 인 경우 적합도 고려 없이 바로 컨베이어 선택이 가능하지만(step 4), 그렇지 않은 경우 인출 대기중인 각 물품별로 각 컨베이어의 적합도를 계산한다 ( $h$ 값에 비례한 가중치 부가). 예를 들어,  $C(D_1) = C(L_2), C(D_1) \neq C(L_3)$  경우, 컨베이어 2에서의 셋업 최소화 가능성을 최대화하기 위해  $D_0$  물품은 가급적 컨베이어 2로 인출되어서는 안 된다. 따라서 이 경우  $G_2 = 10, G_3 = 10^h \geq 10^2$ 가 할당된다. 최종적으로  $G_q$  값이 가장 높은 컨베이어가 물품에 대한 인출 컨베이어로 선택되므로(step 7), 컨베이어 2는 인출 컨베이어로 선택되지 않는다.

### 6. Case Study 및 수치 분석

차량 관련 데이터로는 데이터 수집 시스템상의 한계로 총 4,897개 차량 (82시간 동안의 생산량에 해당)에 대한 가공 순서, 색상, 컨베이어 연결점 별 입/출입 시간 데이터가 사용되었다. 이들 차량들의 색상별 분포를 업체 데이터베이스에 기록된 지난 1년 간의 평균 색상별 분포와 비교한 결과 동일 확률 분포를 가짐을 확인하였다. 기타 컨베이어 속도 및 각 workstation의 처리 속도 등의 데이터는 업체 데이터베이스의 정보를 사용하였다.

해당 공장에서(단일 컨베이어에서 다수 컨베이어로 분기되는 각 분기점에서) 사용되고 있는 기존 알고리즘의 상세 로직은 다음과 같다. 참고로 하기 알고리즘은 해당 공장의 컨베이어 설비 셋업 및 관리를 담당하고 있는 하청 업체에서 공장의 요구사항(1 : 가급적이면 동일 색상 차량이 연속해서 위치하도록 인출 컨베이어 선택, 2 : 1번 조건에 해당되는 컨베이어가 없는 경우 컨베이어간 load balancing 추구)을 반영하여 자체적으로 구현한 알고리즘이다.

- step 1 : set  $A_q \leftarrow 0$  for  $\forall q, E \leftarrow \{\emptyset\}$
- step 2 : for all  $k (k \leq q)$ ,  
if  $C(D_0) = C(L_q)$ , then add  $k$  to  $E$ .
- step 3 : if  $E = \{\emptyset\}$ , then goto step 5  
else goto step 4
- step 4 : for all  $q \in E$ ,  
choose  $q^*$  with smallest  $A_q$  value.



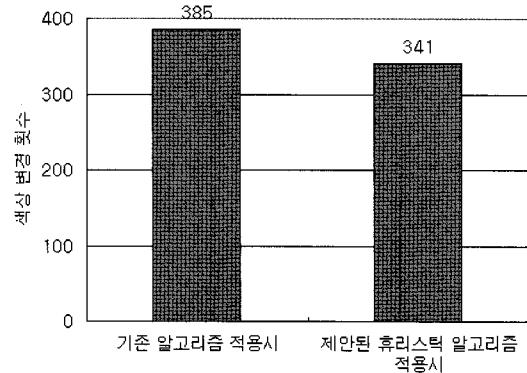
<그림 3> Capa 제약조건이 없을 경우의 알고리즘별 시뮬레이션 분석 결과

```

goto step 6.
step 5 : for all q ≤ k,
        choose q* with smallest Aq value.
step 6 : send D0 to conveyor q*
        set Lq* ← C(D0)
        set Dk ← Dk+1 for ∀ k
        if Aq* > 0, then set Aq* ← Aq* + 1
        else set Aq* ← 1
        set E ← {∅}
        goto step 2
    
```

상기 알고리즘에서 step 2는 각 인출 컨베이어에서 가장 마지막에 인입된 차량의 색상과 인출 대기중인 차량의 색상과 비교하여, 색상이 일치할 경우 현 컨베이어에서 대기 차량 인출 후 해당 인출 컨베이어로 인입한다. step 3은 두 개 이상 일치하는 인출 컨베이어 존재 시 차량이 가장 적은 인출 컨베이어를 선택한다.

해당 공장의 경우처럼 각 workstation의 처리 속도가 충분히 빠르기 때문에 임의의 물품 순서가 각 workstation에 할당 가능할 경우, load balancing 제약식 (6)의 제거를 통해 제 4장에서 제안된 Hungarian 방법을 사용하여 최적해 획득이 가능하며, 이 경우 기존에 공장에서 사용되고 있는 알고리즘 대비 본 논문 제 4장에서 제안한 알고리즘의 현장 적용시의 개선 효과를 비교하기 위해 별도 구축된 시뮬레이션 모델 및 실제 공장 데이터 (총 4,897개 차량, 9개 색상, 2개 도장 workstation 경우) 를 사용하여 <그림 3>와 같이 분석을 수행하였다. <그림 3>에서 ‘적용 이전’은 (아무 알고리즘도 적용되지 이전의) 최초 물품 순서에 변동이 없을 경우를 의미하며, ‘기존 알고리즘 적용 시’와 ‘제안된 알고리즘 적용 시’는 최초 물품 순서에 각 알고리즘이 적용된 경우를 의미한다. 분석 결과, <그림 3>에 보이는 바와 같이 기존 알고리즘 적용 시 그룹

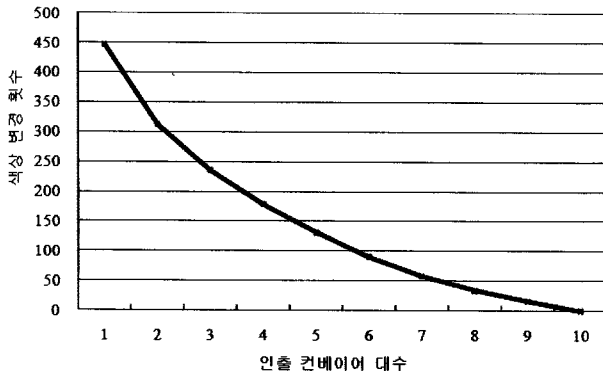


<그림 4> Capa 제약조건이 있는 경우 사용 가능한 휴리스틱 알고리즘의 시뮬레이션 분석 결과

화 효율이 알고리즘 적용 이전 대비 12%, 제 4장에서 제안된 알고리즘 적용 시 알고리즘 적용 이전 대비 31% 감소되는 것으로 나타났다. 따라서 제 4장에서 제안된 알고리즘 적용이 관련 비용 (프로그래밍 및 현장 테스트 등) 을 감안하더라도 충분한 비용 절감 효과가 있는 것을 확인하였다.

각각의 workstation의 처리 속도가 충분히 빠르지 않을 경우 load balancing 제약식 (6)의 제거가 불가능하며, 이 경우 빠른 해 획득을 위해서는 제 5장에서 제안된 휴리스틱 사용이 불가피하다. 이 경우 기존에 공장에서 사용되고 있는 알고리즘과 본 논문 제 5장에서 제안한 알고리즘을 비교하기 위해 <그림 3>과 동일한 시뮬레이션 모델 및 실제 공장 데이터 및 파라미터를 사용하여 <그림 4>과 같이 분석을 수행하였다. 참고로 기존 알고리즘 사용시 각 workstation의 처리 속도가 충분히 빠르지 않을 경우에도 load balancing에 문제가 없음이 시뮬레이션 모델을 통해 확인되었으며, 이는 최초 물품 순서에서의 그룹화 효율이 매우 낮은 관계로 대부분의 차량이 step 2를 건너뛰어 step 4에서 모든 컨베이어에 대한 load balancing이 고려되어 컨베이어 선택이 이루어지기 때문이다. 분석 결과, 제 5장에서 제안된 휴리스틱 알고리즘 적용시 그룹화 효율이 기존에 공장에서 사용되는 알고리즘 대비 평균 11% 감소되는 것으로 나타났다.

마지막으로, DSOP 문제의 구조에 대한 더 나은 이해를 위해 하기와 같은 분석을 수행하였다. 먼저, 랜덤하게 생성된 500대의 차량 순서(10개 색상)의 경우 인출 컨베이어의 대수 변화에 따른 색상 변경 횟수의 변화를 <그림 5>에서 분석하였다. <그림 5>에서 인출 컨베이어가 1대일 경우 컨베이어 연결점에서의 차량 순서 변화가 불가능하므로 색상 변경 횟수는 불변인 반면, 컨베이어가 10대 이상인 경우 색상 변경 횟수가 0이 됨을 알 수 있다. 일반적으로, 대상 차량들의 전체 색상 종류 수보다



<그림 5> 인출 컨베이어 대수 변화에 따른 색상 변경 횟수 변화

인출 컨베이어 대수가 같거나 클 경우 색상 변경 횟수는 0이 된다 (각 인출 컨베이어마다 전용 색상을 지정 운영할 경우 색상 변경이 전혀 없이 전 차량에 대한 도장 공정 진행이 가능하다). 마지막으로 파라미터값의 여러 조합 (색상 수, 색상별 확률 분포, 차량 수 등)에 대해서도 그래프의 모양은 <그림 5>와 유사함을 확인하였다.

### 7. 요약 및 향후 연구 과제

본 연구에서는 기존 연구된 SOP를 확장하여 물품 간 우선순위 제약조건 하에서 총 셋업 비용을 최소화하는 물품 순서를 찾는 문제로서 GSOP를 정의하였다. 그 다음 GSOP의 special case로서 물품 간 우선순위 제약 조건이 FIFO인 DSOP를 정의하고 정수계획법 문제로 모델링하였다. 또한 capacity 제약이 없는 경우의 DSOP의 최적해를 찾는 방법으로서 할당 문제로 변환 및 최적해 획득이 가능함을 보이고, capacity 제약이 있는 경우에 적용 가능한 휴리스틱 알고리즘을 제시 및 평가하였다. 마지막으로, 본 연구에서 제안된 알고리즘의 적용시 예상되는 그룹화 효율 향상 효과를 별도 개발된 시뮬레이션 모델을 통해 검증하였다.

향후 연구 과제로는 다수의 인입 컨베이어가 하나의 연결점을 통해 다수의 인출 컨베이어로 연결된 경우의 모델링 및 그룹화 효율 향상 알고리즘 개발, 오프라인 버퍼 존재시의 최적 알고리즘 개발, 두 가지 색상(two-tone color)의 도로 사용시의 알고리즘 변경 적용 등이 있다.

### 참고문헌

[1] 김경태, “공장 자동화 시스템의 컴퓨터 시뮬레이션 분석에 관한 연구”, 석사 학위 논문, 중앙대학교 : 10-27, 1992.

[2] 김원경, 김충규; “자동차 도장공정의 생산성 향상을 위한 시뮬레이션 분석”, 한국시뮬레이션학회 논문지, 8(2) : 1999.

[3] 문덕희, 송 성, 하재훈; “자동차 공장에서 CRS와 PBS의 전용레인 설치에 관한 시뮬레이션 연구”, 한국시뮬레이션학회 2004 춘계학술대회 논문집 : 77-82, 2004.

[4] 문덕희, 김하석, 송 성, 김경환; “자동차 도장공장의 Color Rescheduling Storage 설치를 위한 시뮬레이션 분석”, IE Interfaces, 16(2) : 211-221, 2003.

[5] 박혜규, 최원준, 신현오; “자동차 도장공장의 Color Selection 시스템의 자동화”, 산업공학, 9(2) : 19-37, 1996.

[6] 정은도; “도장실무 핸드북”, 일진사 : 11, 1990.

[7] 최원준 외; “도장부스가 2개인 자동차 조립 공장의 생산 순서 계획 작성”, 한국 경영과학회/한국 산업공학회 '97 춘계 공동학술대회 논문집 : 451-454, 1997.

[8] Ascheuer, N.; “Hamiltonian path problems in the on-line optimization of flexible manufacturing systems (Ph. D. Thesis),” Technology University of Berlin, 1995.

[9] Atassi, F. R.; “Implementation of block painting in Ford’s in-line vehicle sequencing environment (MS Thesis),” in System Design and Management Program Cambridge : MIT, 1996.

[10] Bonsma, P.; “Complexity results on restricted instances of a paint shot problem for words,” *Discrete Applied Mathematics*, 154 : 1335-1343, 2006.

[11] Campos, M., Bonabeau, E., Theraulaz, G., and Deneubourg, J.; “Dynamic scheduling and division of labor in social insects,” *Adaptive Behavior*, 8 : 83-92, 2001.

[12] Choe, K. I., Sharp, G. P., and Serfozo, R. S.; “Aisle-based order pick systems with batching, zoning, and sorting,” in *Progress in Material Handling Research*, 1993.

[13] Daganzo, C. F.; “Static blocking at railyards : Sorting implications and track requirements,” *Transportation Science*, 20 : 189-199, 1986.

[14] Escudero, L.; “An inexact algorithm for the sequential ordering problem,” *European Journal of Operational Research*, 37 : 236-253, 1988.

[15] Färber, G. and Coves, A.; “Overview on sequencing in mixed model flowshop production line with static and dynamic context,” Internal report IOC-DT-P-2005-7, Universitat Politècnica de Catalunya, 2005.

[16] Gambardella, L. and Dorigo, M.; “An ant colony system hybridized with a new local search for the sequential ordering problem,” *INFORMS Journal on Computing*, 12 : 237-255, 2000.

[17] Han, Y.; “Dynamic sequencing of jobs on conveyor systems



- for minimizing changeovers (Ph. D. Thesis),” in School of Industrial and Systems Engineering, Georgia Institute of Technology, 2004.
- [18] Jeong, M. K., Perry, M., and Zhou, C.; “Throughput Gain with Parallel Flow in Automated Flow Lines,” *IEEE Transactions on Automation Science and Engineering*, 2 : 84-86, 2005.
- [19] Kittithreerapronchai, O. and Anderson, C.; “Do ants paint trucks better than chickens? Market versus response thresholds for distributed dynamic scheduling,” in *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, Canberra, Australia, 2003.
- [20] Lahmar, M. and Benjaafar, S.; “Sequencing with limited flexibility,” *IIE Transactions*, 39 : 937-955, 2007.
- [21] Lim, A. and Xu, Z.; “Searching optimal resequencing and feature assignment on an automated assembly line,” *Journal of the Operational Research Society*, 60(3) : 361-371, 2009.
- [22] Morley, R.; “Painting trucks at General Motors : The effectiveness of a complexity-based approach,” The Ernst and Young Center for Business Innovation, Cambridge, 1996.
- [23] Morley, R. and Ekberg, G.; “Cases in chaos : Complexity-based approaches to manufacturing,” The Ernst and Young Center for Business Innovation, Cambridge 1998.
- [24] Morley, R. and Schelberg, C.; “An analysis of a plant-specific dynamic scheduler,” in *Final Report, Intelligent, Dynamic Scheduling for Manufacturing Systems* : 115-122, 1993.
- [25] Myron, D. L.; “Paint blocking in Ford’s in-line vehicle sequencing environment(MS Thesis),” in *Leaders for Manufacturing Program* Cambridge: MIT, 1996.
- [26] Siddiquee, M. W.; “Investigation of Sorting and Train Formation Schemes for a Railroad Hump Yard,” in *Traffic Flow and Transportation*, G. F. Newell, Ed. New York : Elsevier : 377-387, 1972.
- [27] Spieckermann, S., Gutenschwager, K., and Voss, S.; “A sequential ordering problem in automotive paint shops,” *International Journal of Production Research*, 42 : 1865-1878, 2004.
- [28] Yagar, S., Saccomanno, F. F., and Shi, Q.; “An efficient sequencing model for humping in a rail yard,” *Transportation Research*, 17A : 251-262, 1983.