

혼합모델조립라인의 생산성제고를 위한 작업순서결정*

최종열**

A Study on Sequencing of Mixed Model Assembly Line for Increasing Productivity*

Jongyeol Choi**

ABSTRACT

Mixed Model Assembly Lines(MMALs) are increasingly used to produce differentiated products on a single assembly line without work-in-process storage. Usually, a typical MMAL consists of a number of (1) stations doing exactly the same operation on every job, (2) stations involving operations with different choices, and (3) stations offering operations that are not performed on every job, or that are performed on every job but with many options.

For stations of the first type there is no sequencing problem at all. However, for the second type a set-up cost is incurred each time the operation switches from one choice to another. At the third type of stations, different models, requiring different amounts and choices of assembly work, creates an uneven flow of work along the line and variations in the work load at these stations. When a subsequence of jobs requires more work load than the station can handle, it is necessary to help the operations at the station or to complete the work elsewhere. Therefore, a schedule which minimize the sum of set-up cost and utility work cost is desired.

So this study has developed Fixed Random Ordering Rule(FROR), Fixed Ascending Ordering Rule(FAOR), Fixed Descending Ordering Rule, and Extended NHR(ENHR). ENHR is to choose optimal color ordering of each batch with NHR, and to decide job sequence of the batch with it, too. As the result of experiments, ENHR was the best heuristic algorithm.

NHR is a new heuristic rule in which only the minimum addition of violations from both partial sequence and unassigned sequence at every branch could be considered. And this is a heuristic sequencing rule for the third type of stations at MMAL. This study developed one more heuristic algorithm to test the performance of NHR, which is named as Practical Heuristic Rule(PHR).

* 이 논문은 1994년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

** 부산대학교 상업정보교육과 부교수

1. 서 론

오늘날 대량생산시스템은 고객의 욕구가 다양화됨에 따라 제품 다양화와 생산성 증대라는 두 가지의 상충되는 목표를 달성하기 위해 시장의 수요 변화에 신속하게 대처할 수 있는 신속성이 높은 다품종 소량생산을 지향하는 흐름으로 나아가고 있다. 제품 특성상 대량생산 시스템의 적용이 불가피한 자동차 산업, 가전제품 산업 등에서도 조립부품들을 다양화하고 이들 간의 호환성을 증대시킴으로써 고객의 욕구를 충족시킬 수 있는 다양한 제품생산이 가능한 모듈러 생산을 행하고 있으며, 각 작업장에서는 여러 가지 유형의 조립작업을 수행하고 있다. 그리고 생산성 증대를 위하여 각 공정간의 재공품 재고를 없애고 필요한 부품을 필요한 시기에 공급받는 JIT시스템을 실시하고 있다. 따라서 오늘날 대량생산시스템은 다양한 고객욕구를 충족시키고, 재공품 재고없이 단일조립라인에서 다양한 제품들을 생산하는 혼합모델조립(Mixed Model Assembly Line ; MMAL)을 많이 사용하고 있다.

MMAL은 그때그때 필요한 모델을 생산하기 위해 조립라인에서 여러 가지 모델들을 혼합생산한다. 따라서 MMAL은 신속성을 제고시키며 단일모델 조립라인의 효율성과 거의 같은 효율성을 달성할 수 있으며 작업의 연속적 흐름을 가능케 해주고 모델들의 일괄처리나 라인 변동을 제거해준다. 그러므로 오늘날과 같이 다양한 고객욕구를 충족시키기 위해 적절한 조립라인의 유형이 MMAL임을 알 수 있다.

한편 MMAL에서는 모델들이 달라지면 조립라인의 작업내용과 작업량이 달라지게 되며 라인상의 작업흐름이 원활하지 못하게 되고 각 작업장의 작업부하량은 변화한다. 따라서 MMAL의 효율적인 운영을 위해서는 첫째, 장비와 노동력의

적절한 할당, 둘째, 컨베이어 속도, 산출간격(Launching Interval)과 작업장 길이의 최적화, 마지막으로 라인에 투입되는 여러 가지 일감들의 작업순서결정 등이 해결되어야 할 것이다. 그러나 처음 두 가지는 MMAL의 실제와 밸런싱에 관계되는 문제이며 본 연구에서는 세번째 문제, 즉 라인에 투입되는 여러 가지 일감들의 작업순서결정 문제의 해결에 초점을 맞추고자 한다.

일반적으로 MMAL은 모든 일감에 대해 동일한 작업을 행하는 작업장, 작업시간이 동일한 여러 가지 사양 중 하나를 선택할 수 있는 작업장, 그리고 작업시간이 상이한 작업들을 수행하는 작업장으로 구성되어 있다. 첫째 유형의 작업장에서는 모든 일감에 대해 동일한 작업이 행해지므로 작업순서 변화가 아무런 영향을 미치지 않으므로 작업순서결정은 전혀 문제가 되지 않는다. 그러나 두번째 유형의 작업장에서는 작업순서의 변화에 따라 작업시간은 같지만 사양에 따라 작업내용이 달라지게 되어 준비비용(setup cost)이 발생하게 된다. 세번째 유형의 작업장에서는 필요로 하는 작업유형의 종류와 양에 따라 생산의 원활한 흐름에 변화를 초래하며, 만약 조립라인의 작업순서에 따라 작업장의 처리능력보다 많은 부하가 발생할 경우에는 후속 작업장의 원활한 작업을 위하여 해당 작업장의 작업자들을 지원하는 보완작업(utility work)을 행하여야 하며 이로 인한 비용이 발생하게 된다.

따라서 일반적인 MMAL(Generalized Mixed Model Assembly Line, GMMAL)에서는 준비비용과 보완작업비용의 합을 최소화하는 일정계획이 요구된다. 이에 관한 연구들이 수행되어져 오고 있으며, 특히 TSP(Traveling Salesman Problem)개념을 이용한 비선형 이진 혼합정수계획모델인 수리모델을 들 수 있다. 그러나 이 모델은 너무 복잡하여 현실문제를 적용할 경우 계산

이 불가능하며, 두 가지 비용의 합을 최소화하는 연구 즉 일반적 MMAL의 작업순서결정에 관한 연구는 아직 많이 진척되지 못하고 있는 실정이다.

그러므로 본 연구에서는 MMAL의 보완작업비용과 준비비용을 효율적으로 통제할 수 있는 작업순서결정을 위한 휴리스틱 규칙을 개발하고자 한다. 특히 본 연구에서는 작업장에서 행해지는 작업유형은 기본작업과 여러 가지 선택작업들이 있을 수 있으므로 각 작업장에서 기본작업과 두 가지의 선택작업이 행해지는 경우에 초점을 맞추어, 다작업장의 작업순서를 결정하기 위한 휴리스틱 규칙을 개발하고자 한다.

본 연구는 우선 단일작업장에서 준비비용을 고려하지 않는 경우의 최적작업순서 규칙을 설명한다. 그리고 모델의 단순화를 위해 보완작업의 대응척도로서 과부화량(violation) 개념을 설명한다.

다작업장의 경우 과부하 최소화모델을 이용해서 최적해를 구하는 것은 매우 많은 시간과 노력이 필요할 것이므로, 분지한계법(Branch & Bound Method)을 이용한 휴리스틱 규칙으로써 만족해를 도출하고자 한다. 그리고 이 휴리스틱 규칙의 유용성을 분석한다. 이를 위해 Random Data Set을 대상으로 시뮬레이션 실험을 통해 새로운 휴리스틱 규칙(NHR)과 실제 일반적으로 많이 이용되고 있는 휴리스틱 규칙(PHR)의 우위를 비교 분석한다.

제 4 장에서는 준비비용이 발생하는 두번째 유형의 작업장까지도 고려하는 일반적 MMAL의 작업순서결정을 위한 휴리스틱 규칙을 고찰한다. 그리하여 고정랜덤순서규칙(FROR), 고정오름차순규칙(FAOR), 고정내림차순규칙(FDOR), 그리고 확장NHR규칙(ENHR)의 4가지 휴리스틱 규칙을 개발하여 이를 비교 분석한다.

앞에서와 같이 연구를 수행하기 위해 본 연구는 두 가지 연구방법을 병행한다.

우선 기존의 연구들을 분석하기 위해 문헌조사를 행하며, 본 연구에서 개발한 휴리스틱 규칙의 유용성을 분석하기 위해 시뮬레이션을 통한 실험적 연구를 행한다.

2. 단일작업장에서의 MMAL 작업 순서 결정

2.1. 연구동향

Dannenbring[7], Graves[14], Park 등[17]은 정태적 유동작업장의 작업순서결정문제를 위한 휴리스틱 규칙들을 연구하고 평가했다. 표준유동작업장 작업순서결정문제는 재공품 재고를 허용하고 일감의 비동기적 이동을 가정하는 측면과 작업완수시간을 최소화하는 것을 목적으로 하고 있는 것이 MMAL과 상이하다.

Reddi와 Ramamoorthy는 MMAL의 특징인 완충재고를 갖지 않는 유동작업장 작업순서결정 문제를 다루었다[18]. 그러나 그들의 목적은 일정 계획의 총완수시간을 최소화하는 것이었다. 그들은 이 문제를 TSP(Traveling Salesman Problem)로 수식화하고, TSP 해법 절차를 효율적으로 만들기 위해 지배기준을 사용할 것을 제안했다. 그러나 그들은 이 혼합계산법의 계산상의 특성을 광범위하게 연구하지 않았다. MMAL에서는 납기일과 완수일은 일감들이 라인을 들어오는 순서에 의존한다. 이것은 NP-Complete 문제로 고려되는 표준유동작업장의 작업순서결정 문제보다 MMAL을 훨씬 더 복잡하게 한다[12]. 따라서 그들은 노동력 비효율성의 최소화와 완수시간 최소화 사이의 관계를 고려하지 않았다.

Thomopoulos는 자동차 산업에 대한 사례연구

에서 개발된 작업순서결정 문제를 다들 특정절차들을 논의했다[19]. 그는 여러 가지 비효율성(작업자 유희시간, 보안작업, 작업혼잡과 작업부족 등)에 대한 총별점을 최소화하는데 기초를 둔 방대한 계산기법을 시험했다.

Dar-El과 Cother는 조립라인의 길이를 최소화하고 작업자의 유희시간이 0이 되도록 하기 위하여 MMAL 작업순서결정을 위한 휴리스틱 규칙을 개발했다[8]. 이후에 Dar-El과 Cucuy는 정수 계획법을 사용하여 동일한 문제를 해결하기 위한 최적 계산해법을 개발했다[9]. Dar-El은 현장에서 쉽게 볼 수 있는 여러가지 MMAL유형들을 설명했다[10]. 그리고 그는 MMAL시스템들의 작업순서결정을 위해 설비길이의 최소화과 사이클타임의 최소화라는 두 가지 설계기준을 제시했다. 그리고 이러한 목표들을 달성하기 위하여 몇 가지 설계전략을 제시했다. 그러나 모든 접근법들을 새로운 MMAL의 설계와 계획을 위해 사용되었다. 따라서 조립라인에서 생산량 변화가 발생하고 현재의 작업장 길이가 제약으로 주어질 때 그의 계산법으로는 요구되는 생산을 달성할 수 있는 해를 도출할 수 없을 것이다.

Okamura와 Yamashina는 작업자가 어떤 작업장 입구에서 출구로의 이동거리를 최소화하는 작업순서를 발견하기 위한 휴리스틱 규칙을 제시했다[16]. 그러나 그들은 작업장 길이와 같이 작업장에 따라 상이한 여러 가지 사항들을 고려하지 않았다.

Monden은 JIT시스템을 적용을 위해 한 자동차회사의 MMAL 작업순서결정문제를 다루었다[15]. 즉, 각 부품에 대해 시간당 사용량을 가능한 한 일정하게 유지하는 것이었다. 그는 각 부품의 이용속도가 달라지는 것을 최소화하기 위하여 문제를 설정했다. 이 문제의 해는 반드시 생산능력의 최적이용을 가져다 주지는 않는다.

Burns와 Daganzo는 작업준비비용과 공간제약의 위반 갯수 사이의 트레이드 오프를 평가하기 위한 간단한 분석원칙들을 도출했다[5]. 그들은 노무의 비효율성이 무시할 만하다고 가정함으로써 문제를 이완시켰으며 준비비용들을 초래하는 작업들의 준비비용들에 기초를 둔 매우 간단한 순위결정 해법(rank ordering algorithm)을 기술했다. 그들은 또한 각 작업자의 생산능력사이의 관계를 연구했다. 하지만 이러한 가정은 비현실적이다. 왜냐하면 각 작업장은 컨베이어에 의해 서로 연결되어 있기 때문이다.

Bolat는 자동차 산업의 일반적인 MMAL문제를 다루었다[4]. 그는 기본작업과 선택작업을 행하는 작업장에서의 최적의사결정과 이를 이용한 휴리스틱 규칙을 제시하고 대응척도를 고안하여 문제를 단순화시키고 다작업장에서 계산상의 효율성을 증대시키는 휴리스틱 규칙을 제시했다. 그리고 도장부문과 같이 준비비용이 발생하는 부문도 포함시켜 혼합모델조립라인의 전반적 문제(Generalized Mixed Model Assembly Line, GMMAL)를 다루었다. 그러나 그는 두 가지의 작업유형 즉, 기본작업과 단 하나의 선택작업만을 고려했으며, 작업장간의 상호의존성을 고려하지 않았다.

Yano와 Rachamadugu는 MMAL을 구성하고 있는 작업장들의 중요도를 측정하여 작업순서결정을 시도하였으며[21], 또한 그들은 단일작업장에서 작업의 과부하를 최소화하는 작업순서규칙을 개발하고, 이를 토대로 다작업장의 작업순서결정을 위한 휴리스틱 규칙을 연구했다[22]. 그러나 그들은 두 가지 작업유형만을 다루는데 그쳤다.

한편 본 연구자[2]는 작업장에서 행해지는 작업유형은 기본작업과 여러 가지 선택작업이 있을 수 있으므로 각 작업장에서 기본작업과 두 가지의 선택작업이 행해지는 경우에 초점을 맞추어

MMAL의 세번째 작업장유형에 있어서 준비비용은 고려하지 않고 보완작업비용을 효율적으로 통제할 수 있는 다작업장의 작업순서결정을 위한 휴리스틱 규칙의 개발에 관한 연구를 수행했었다. 그러나 문제의 복잡성으로 말미암아 MMAL 작업순서결정시 항상 발생하는 준비비용을 고려하지 못했다.

본 연구에서는 각 작업장에서 기본작업과 두가지의 선택작업이 행해지는 경우에 초점을 맞추어, 보완비용과 준비비용의 합으로 구성되는 총비용을 고려한 다작업장의 작업순서를 결정하기 위한 휴리스틱 규칙을 개발하고자 한다.

2.2. 단일작업장에서의 MMAL 작업순서결정 모형

우선 다작업장의 작업순서 결정을 위한 휴리스틱 규칙을 개발하기 위한 기초작업으로 우선 단일작업장에서 야기될 수 있는 모든 상황을 고려한 최적 작업순서규칙을 연구했다.

본 연구에서는 우선 MMAL작업유형 중 세번째 작업이 이루어지는 단일작업장을 대상으로하며 선택작업 유형(P)은 1, 2 두가지가 있다고 가정하고, 준비비용이 존재하지 않는 경우를 가정했다. 그리고 기본작업과 선택작업 1, 2의 작업소요시간이 b_j , o_j^1 , o_j^2 이며 작업장의 길이가 L_j 인 작업장 j 를 가정했으며, 다음의 네 가지 조건이 모두 충족되는 상황을 가정했다.

- ① $o_j^2 > o_j^1 > b_j$, ② $b_j < 1$ ③ $o_j^1 > 1$ ④ $o_j^1 < L_j$

j 작업장의 작업자 수는 조정할 수 있으며 작업장의 길이는 위의 가정들이 성립하도록 설계되어 있다. 그리고 작업자가 특정 일감의 작업을 마치고 후속 일감이 위치한 곳으로 이동하는데 소요되는 시간은 0으로 가정한다.

t 시점에 작업장에 일감이 없을 경우 작업자들

은 작업장 입구에서 그 시점 이후에 들어오는 일감을 기다리게 되며, 따라서 후속 작업은 작업장 입구에서부터 행해진다. 이 경우 작업장이 t 시점에 재개된다(regenerated)고 한다.

만약 가동 후 첫번째 일감이 기본작업을 요구한다면 첫번째 작업의 소요시간이 다음 일감의 도착시간보다 짧으므로 ($b_j < 1$) 작업자는 두번째 일감이 도착하기 전에 그것을 완수하고 작업장 입구에서 다음 작업을 기다리게 될 것이다. 반면에 선택작업을 요구한다면 두번째 일감은 작업장에 도착한 지점으로부터 $o_j^1 - 1$ 만큼 이동한 위치에서 작업에 착수하게 될 것이다. 그러나 $o_j^1 \leq L_j$ 의 가정에 의하여 재개시점에 작업장에 들어 오는 어떤 유형의 일감들에 대해서도 보완작업은 발생하지 않는다. 그러나 여러 개의 선택작업을 요구하는 일감들이 연속해서 투입될 경우에는 각 일감의 작업시작 위치는 작업장 입구 즉, 재개시점으로부터 점점 멀어지게 되고 결국 L_j 를 초과하게 되어 보완작업이 요구될 것이다. 즉 재개시간 간격이 길어지면 보완작업량이 많아진다.

그러나 재개시간 간격이 너무 짧게 되면 유휴시간이 발생하여 작업자들은 작업완료 후 작업장 입구에서 다음 일감을 기다리게 될 것이다. 따라서 항상 일감이 존재하도록 하기 위해서는 작업자를 입구에서 멀리 떨어지도록 해야 한다. 그렇게 함으로써 다음 재개시점에서의 도착을 지연시킬 수 있다.

재개시점이 전체적으로 작업순서의 결과에 미치는 영향을 요약하면 다음과 같다. 재개시점 도착간격이 짧으면 재개회수가 많아져서 결과적으로 불필요한 유휴시간을 초래한다. 그리고 각 주기당 기대치보다 긴 평균 허용 공간을 초래하여 작업장의 이용율이 낮아지고 노동력의 효율성을 저하시키게 된다. 후기 주기에서는 선택작업을 요하는 일감들이 많이 남기 때문에 보완작업이 요

구될 것이다. 그 반면에 재개시점 도착간격이 길어지면 초기 주기에서 허용공간이 좁아지며 불필요한 보완작업을 야기시킨다. 그리고 이 경우에는 후기 주기에서는 유휴시간이 많이 발생한다.

그러므로 작업순서결정에는 보완작업의 최소화를 위한 시스템 재개빈도의 증대와 유휴시간의 최소화를 위한 시스템 재개빈도의 감소라는 두 가지 상충되는 목표가 존재함을 알 수 있다. 또한 각 주기의 공간과 노동력 이용을 재개주기의 회수와 시간 사이에는 부의 관계가 존재한다. 따라서 작업장의 이용율과 재개시간 간격이 어느 주기에서나 최적이 되도록 일감들의 순서를 결정해야 할 것이다. 다음 절에서는 이 목적을 달성할 작업 순서유형을 제시하고자 한다.

2.3. 단일작업장의 최적작업순서결정

2.3.1. 작업순서규칙 1

j 작업장의 길이가 L_j , 선택작업 1, 2의 작업 소요시간이 각각 a_1^j , a_2^j , 기본작업의 소요시간은 b_j 라 가정한다. 그리고 컨베이어의 속도는 1로서 항상 일정하다고 가정한다. 이 경우 작업자는 보완작업 (utility work)을 최소화하고 유휴시간을 최소화하기 위하여 우선 선택작업을 요구하는 일감들을 L_j 범위 내에서 최대한으로 많이 처리하고 난 뒤 기본작업을 요구하는 일감들을 처리해야 한다. 이 령게하면 작업자는 j작업장의 입구에서 출발하여 출구 쪽으로 끝까지 이동했다가 다시 입구로 되돌아오게 될 것이다.

선택 작업 1, 2를 요구하는 일감들을 k_1^j 개, k_2^j 개씩 처리하면 작업자가 j 작업장 입구로부터 이동하는 거리는 $k_1^j(a_1^j-1)+k_2^j(a_2^j-1)+1$ 이 되며, 이 길이가 L_j 와 정확히 일치된다면 다음 식이 성립될 것이다.

$$k_1^j(a_1^j-1)+k_2^j(a_2^j-1)=L_j-1 \quad (2.1)$$

이 식에서 미지수는 k_1^j , k_2^j 이다. 한편 j작업장에서 행해져야 할 작업일감 중 선택작업 1을 요하는 작업량(H_1^j)과 선택작업 2를 요하는 작업량(H_2^j)이 자료로써 주어지므로, (2.1)식을 충족시키는 정수 k_1^j 과 정수 k_2^j 의 조합 중 k_1^j/k_2^j 의 값이 H_1^j/H_2^j 의 비율과 같은 조합이 존재한다고 가정하고 이 비율과 같은 조합을 택한다.

한편 작업자는 기본작업을 요하는 후속일감을 완수한 후에는 작업장을 정리하고 작업장 입구로 $1-b_j$ 만큼씩 이동할 것이다. 그리하여 m_j 개의 일감을 완수하고 나면 처음 기본작업을 요하는 일감의 작업을 시작한 위치로부터 $m_j(1-b_j)$ 만큼 입구 쪽으로 이동할 것이다. 따라서 필요한 공간은 $m_j(1-b_j)$ 가 된다. 반면에 (2.1)식이 성립되면 최초의 기본작업을 수행할 때의 작업장에서의 작업자의 위치는 입구로부터 $k_1^j(a_1^j-1)+k_2^j(a_2^j-1)+1-1$ 만큼 떨어지게 되어 이 거리는 L_j-1 과 같다. 따라서 다음 식을 충족시키는 정수해 m_j 가 존재하면 일정계획에서 재개시점 이전에 작업장의 최대 이용과 유휴시간의 최소화가 달성될 수 있다.

$$m_j=(L_j-1)/(1-b_j) \quad (2.2)$$

이러한 사실을 이용하여 작업순서를 결정하는 다음과 같은 작업순서규칙을 생각할 수 있다.

〈작업순서규칙 1〉

(2.1)식과 (2.2)식을 충족시키는 k_2^j , k_1^j 과 m_j 가 주어지고 순서가 정해져야 할 기본작업을 요하는 충분한 일감들이 항상 존재할 경우에는, 다음과

같은 두 가지 단계를 거쳐 작업순서를 결정할 수 있다.

제 1 단계 선택작업 2, 1을 요하는 작업들로 처음 $k_2^2+k_1^1$ 개의 슬롯들을 채우고 기본작업을 요하는 일감들로 다음 m_1 개의 슬롯들을 채운다.

제 2 단계 매 $k_2^2+k_1^1+m_1$ 슬롯들에 대해서 1단계를 반복한다.

l_j 를 각 재개주기 동안 작업이 수행되는 일감 수 즉, $l_j=k_2^2+k_1^1+m_1$, G_j 를 반복주기 수, r_j 를 최후에 남는 부분주기의 슬롯 크기라 정의한다. 따라서 여기서 r_j 는 G_j 회 반복할당 후 남는 슬롯의 크기가 될 것이다. 한편 r_j 는 l_j 보다 작음을 알 수 있다. 이를 수식으로 표현하면 다음과 같다.

$$G_j = \frac{N}{l_j}, \quad r_j = N - G_j \cdot l_j \quad (2.3)$$

만약 총일감 수가 $k_2^2+k_1^1+m_1$ 의 정수배이고 각 선택작업량이 각기 k_2^2, k_1^1 의 동일 정수배이면 위의 규칙에 의거하여 할당할 때 보완작업도 발생하지 않고, 최적할당이 됨을 알 수 있다.

작업순서규칙 1은 각 선택작업량이 보완작업을 발생시키지 않고 할당할 수 있는 최적 선택작업량과 일치할 경우에만 적용될 수 있다. 이제 그렇지 않은 경우 즉, 어떠한 H_j^1, H_j^2 값에 대해서도 최적해를 제공해 주는 작업순서규칙을 고찰해 보기로 한다.

이를 위해서는 작업량이 $k_2^2+k_1^1+m_1$ 의 정수배가 아니고 각 선택작업 일감 수도 k_2^2, k_1^1 의 동일 정수배를 충족시키지 않는 상황에서 최적작업순서규칙을 고찰하기 위해, 우선 여러가지 상황에서

보완작업을 발생시키지 않는 최적 선택작업량 H_j^1, H_j^2 의 계산이 선행되어야 할 것이다[2, pp. 19-25.].

2.3.2. 작업순서규칙 2

작업순서규칙 1이 적용될 수 없는 경우 즉, 어떠한 H_j^1, H_j^2 에 대해서도 최적해를 제공해 주는 작업순서규칙 2를 고찰해 본다.

(2.1)식과 (2.2)식을 충족시키는 k_2^1, k_1^2 가 존재한다면, 다음과 같은 네 가지 단계를 거쳐 작업순서를 결정할 수 있을 것이다.

제 1 단계 각 l_j 중 처음 $k_2^2+k_1^1$ 슬롯들을 k_2^2 개의 두번째 선택작업 일감 (O_2)과 k_1^1 첫번째 선택작업 일감 (O_1)로 채우고 다음 m_1 슬롯들은 뛰어넘는다.

제 2 단계 가능하면 오래동안 「1 단계」를 반복한다. 이 과정 중 두 가지 선택작업 일감이 모두 할당될 경우에는 「제 4 단계」로 가고, 그렇지 않은 경우에는 「제 3단계」로 간다.

제 3 단계 최적 선택작업량 결정과정의「제 2 단계」에서 보았듯이 「작업순서규칙 1」에 의거하여 선택작업 일감 O_2 와 O_1 을 반복할당해 나가는 과정에서 G_j 회 할당하고도 O_2 와 O_1 이 동시에 남는 경우와, 반복할당 과정 중 두 가지 일감 중 어느 하나가 먼저 고갈되는 경우가 발생하였으며, 전자 즉 (case A)일 경우에는 A 단계, 후자 즉 (case B)일 경우에는 B단계로 간다.

제 4 단계 빈 슬롯에 기본작업 일감을 할당하고, 할당작업을 끝낸다.

A. (case A)일 경우

r_j 의 크기에 따라 (A.1), (A.2), (A.3) 중 한 과정을 거친다.

(A.1) $r_j \leq k_j^z$

(선택작업 일감 유형이 하나뿐일 경우에는 보완작업을 최소화하기 위하여 해당 선택작업 일감 k_j 개를 작업하고 난 후 기본작업 일감 m_j 개의 작업을 수행하면 될 것이다. 선택작업 일감유형이 O_1 일 경우에는 k_j 는 k_j^1 가 될 것이며, 선택작업 일감 유형이 O_2 일 경우에는 k_j 는 k_j^2 가 될 것이며 다음과 같은 수식으로 나타낼 수 있다. $k_j^1 = (L_j - 1) / (o_j^1 - 1)$, $k_j^2 = (L_j - 1) / (o_j^2 - 1)$)

G_j+1 주기의 첫 슬롯부터 O_2 를 최대한 할당 후 O_1 을 반복할당한다. 이 과정에서 두 가지 선택작업 일감이 모두 할당될 경우에는 「제 4 단계」로 간다. O_2 할당 과정에서 슬롯이 부족하면 (A.1.1)로 가고, O_1 할당 과정에서 슬롯이 부족하면 단계 (A.1.2)로 간다.

(A.1.1) G_j 주기의 빈 슬롯들의 끝에서부터 O_2 의 잔여분을 최대한 할당 후 O_1 을 반복할당한다. 이때 G_j 주기의 빈 슬롯에 두 가지 선택작업 일감이 모두 할당될 경우에는 「제 4 단계」로 가고, 슬롯이 부족할 경우에는 잔여분을 이용 가능한 슬롯에 균일 할당한 후 「제 4 단계」로 간다.

(A.1.2) G_j 주기의 빈 슬롯들의 끝에서부터 O_1 의 잔여분을 할당한다. 이때 G_j 주기의 빈 슬롯에 O_1 의 잔여분이 모두 할당될 경우에는 「제 4 단계」로 가고, 슬롯이 부족할 경우에는 O_1 의 잔여분을 이용 가능한 슬롯에 균일 할당한 후 「제 4 단계」로 간다.

(A.2) $k_j^z < r_j < k_j^1 + k_j^2$

이때 $RO_j^2 < k_j^z$ 이면 (A.2.1)으로 가고, $RO_j^2 > k_j^z$ 일 경우에는 (A.2.2)로 간다.

(O_2, O_1 를 k_j^1, k_j^2 개씩 반복할당시 두 가지 선택일감들이 같은 반복회수에서 고갈되거나 G_j 회 할당 후에도 남는 경우이며, 이때 G_j 회 할당 후에도 남은 O_2, O_1 의 개수는 다음과 같다. ($RO_j^1 = H_j^1 - G_j \cdot k_j^1$, $RO_j^2 = H_j^2 - G_j \cdot k_j^2$))

(A.2.1) G_j+1 주기의 첫 슬롯부터 O_2 를 최대한 할당한 후 O_1 을 반복할당한다. 두 가지 선택작업 일감이 모두 할당될 경우에는 제 4 단계로 가고, O_2 할당 과정에서 슬롯이 부족할 경우에는 (A.1.1)로 가고, O_1 할당 과정에서 슬롯이 부족할 경우에는 (A.1.2)로 간다.

(A.2.2) G_j+1 주기의 첫 슬롯부터 k_j^z 개의 O_2 를 최대한 할당한다. 그리고 나서 이 주기의 마지막 슬롯부터 O_2 의 잔여분을 최대한 할당한 후 O_1 의 잔여분을 할당한다. 이때 G_j+1 주기의 빈 슬롯에 두 가지 선택작업 일감이 모두 할당될 경우에는 제 4 단계로 가고, O_2 의 할당 과정에서 슬롯이 부족할 경우에는 (A.1.1)로 가고, O_1 의 할당 과정에서 슬롯이 부족할 경우에는 (A.1.2)로 간다.

(A.3) $r_j \geq k_j^1 + k_j^2$

$RO_j^1 \geq k_j^1$, $RO_j^2 \geq k_j^2$ 이면 (A.3.1)로 가고, 그 이외의 경우에는 (A.3.2)로 간다.

(A.3.1) G_j+1 주기의 첫 슬롯부터 O_2, O_1 을 k_j^2, k_j^1 개씩 할당하고, G_j+1 주기의 마지막 슬롯부터 O_2 의 잔여분을 최대한 할당한 후, O_1 의 잔여분을 할당한다. 이 과정에서 두 가지 선택작업 일감이 모두

할당될 경우에는 제 4 단계로 가고, 슬롯이 부족하면 이용 가능한 빈 슬롯에 미할당한 선택작업 일감들을 할당하고 제 4 단계로 간다.

(A.3.2) $RO_j^2 \leq k_j^2$ 이면 (A.3.2.1)로 가고, $RO_j^2 > k_j^2$ 이면 (A.3.2.2)로 간다.

(A.3.2.1) $r_j \leq RO_j^2 + k_j^1$ ($k_j^1 = [(L_j - 1) - RO_j^1 (o_j^1 - 1)] / [o_j^1 - 1]$) 이면 (A.3.2.1.1)로 가고, $r_j > RO_j^2 + k_j^1$ 이면 (A.3.2.1.2)로 간다.

(A.3.2.1.1) $G_j + 1$ 주기의 첫 슬롯부터 O_2 를 최대 할당한 후 O_1 을 할당한다. 이 과정에서 두 가지 선택작업 일감이 모두 할당될 경우에는 제 4 단계로 가고, 슬롯이 부족할 경우에는 1.2)로 간다.

(A.3.2.1.2) $G_j + 1$ 주기의 첫 슬롯부터 O_2 를 최대 할당한 후 O_1 을 k_j^1 범위 내에서 최대한으로 할당한다. 이 과정에서 두 가지 선택작업 일감이 모두 할당될 경우에는 제 4 단계로 가고, 슬롯이 부족할 경우에는 $G_j + 1$ 주기의 빈 슬롯들의 끝에서부터 O_1 의 잔여분을 반복할당한다. 이때 이 주기의 빈 슬롯에 O_1 의 잔여분이 모두 할당될 경우에는 제 4 단계로 가고 슬롯이 부족할 경우에는 잔여분을 이용 가능한 슬롯에 균일 할당한다.

(A.3.2.2) $G_j + 1$ 주기의 첫 슬롯부터 O_2 를 k_j^2 개 할당한 후, 이 주기의 마지막 슬롯부터 미할당된 O_2 를 최대 할당한 후, O_1 을 반복할당한다. 이때 두 가지 선택작업 일감이 모두 할당될 경우에는 제 4 단계로 가고, 슬롯

이 부족할 경우에는 잔여분을 이용 가능한 슬롯에 균일 할당한다.

B. (case B)의 경우에 대해서도 최적작업순서 규칙을 구할 수 있다[2, pp. 29 - 32].

작업순서규칙 2의 이해를 돕기 위해서 간단한 예제를 제시한다.

(예제) $L_j = 5$, $o_j^1 = 1.2$, $o_j^2 = 0.8$ 이고, 이며, $k_j^1 = 4$, $k_j^2 = 4$, $m_j = 20$ 이며, 작업장에서 처리해야 할 총 일감수(N)가 500개이고, $H_j^1 = 100$, $H_j^2 = 90$ 이라고 가정하자.

이 경우 작업순서 규칙2에 따라 아래와 같이 일감을 할당하기 위한 준비과정이 수행될 것이다.

$$I_j = 4 + 4 + 20 = 28 \quad G_j = [500 \div 28] = 17$$

$$r_j = 500 - 17 \times 28 = 24$$

$$RO_j^2 = 90 - 57 \times 4 = 22 \quad RO_j^1 = 100 - 17 \times 4 = 32$$

$$I_j^1 = 20, \quad I_j^2 = 5$$

작업순서 규칙2에 따라 할당을 하면 우선 17번째 주기까지 첫 슬롯부터 8번째 슬롯까지 O_2 와 O_1 을 각각 4개씩 배정한 후 20개씩의 슬롯을 각 주기의 뛰어 넘는 과정을 반복한다. 이후 위의 계산에서 보았듯이 O_2 가 22개, O_1 이 32개 남았으므로, (Case A)에 해당되며, $r_j > k_j^1 + k_j^2$ 이므로 (A.3)에 해당된다. 그리고 $RO_j^1 > k_j^1$, $RO_j^2 > k_j^2$ 이고, $RO_j^2 > k_j^2$ 이므로 (A.3.2.2)에 해당된다.

따라서 남아있는 $RO_j^2 = 22$ 중 5개를 $G_j + 1$ (18번째) 주기의 처음부터 5번째 슬롯에 할당하고 난 후, 미할당된 17개의 O_2 를 18번째 주기의 뒤에서부터 할당하고, 남아있는 18번째 주기의 빈슬롯 2개 ($r_j - RO_j^2 = 24 - 22 = 2$)에는 O_1 을 배정한다. 나머지 30개의 O_1 은 첫 번째 주기부터 17번째 주기에 걸쳐 균일하게 배정한다. 즉 13번째 주기까지는 빈 슬롯에 O_1 을 2개씩 추가 할당하고, 14번

제 주기부터 17번째 주기까지는 빈 슬롯에 O_1 을 1개씩 추가할당한다. 이제 제 4단계로 가서 비어 있는 슬롯들에 기본작업일감, B를 할당함으로써 할당작업을 끝낸다.

2.3.3 작업순서규칙 3

앞에서 개발된 작업순서규칙 2는 (2.1)식과 (2.2)식을 충족시키는 정수 k_j^1 , k_j^2 와 m_j 가 존재한다고 엄격히 가정하고 있다. 그러나 실제 상황은 그렇지 않으므로 (case 1)을 중심으로 다음과 같이 생각해 볼 수 있다. 우선 k_j^2 를 보완작업이 필요없는 k_j^1 의 최대 값, 즉 다음 식을 충족시키는 최대 정수로 정의함으로써 가정을 완화시킬 수 있다.

$$\sum_{p=1}^2 k_j^p(o_j^p-1) \leq L_j-1 \quad (2.4)$$

물론 이때에도 이러한 조건을 충족시키는 k_j^1 , k_j^2 의 조합 중 H_j^1 과 H_j^2 의 비율에 가장 가까운 k_j^1 , k_j^2 를 선택한다.

한편, k_j^1 , k_j^2 개의 일감 완수 후에는 작업자는 다음 작업을 수행하러 1만큼 입구 쪽으로 돌아간다. 따라서 $\sum_{p=1}^2 k_j^p(o_j^p-1)$ 만큼 떨어진 곳에서 작업을 시작하게 된다. 만약 기본작업 일감 m_j 개를 작업하고 나면, 작업자는 작업장 입구 쪽으로 $m_j(1-b_j)$ 만큼 이동하게 될 것이다. 따라서 유휴시간 없이 모든 일감들을 완수한 후 재개시점에 도달하기 위해서는 다음 식이 충족되어야 할 것이다.

$$\sum_{p=1}^2 k_j^p(o_j^p-1) = m_j(1-b_j) \quad (2.5)$$

이 식을 충족시키는 값은 정수가 아닐 수도 있다. 그러나 m_j 는 (2.6)식을 충족시키는 가장 작은 정수나 (2.7)식을 충족시키는 가장 큰 정수로 정의할 수 있을 것이다.

$$m_j(1-b_j) \leq L_j-1 \quad (2.6)$$

$$m_j(1-b_j) \leq \sum_{p=1}^2 k_j^p(o_j^p-1) \quad (2.7)$$

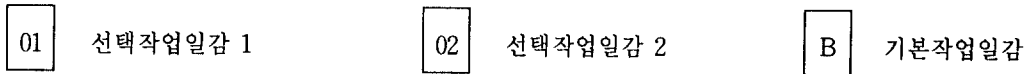
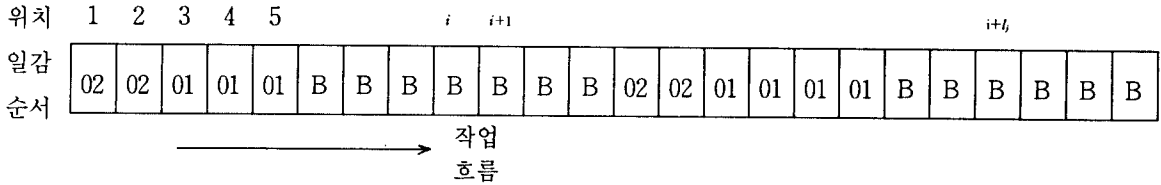
m_j 의 값이 전자일 때는 유휴시간이 발생하나 유휴시간 이후에는 시스템이 재개됨을 알 수 있다. 반면에 m_j 의 값이 후자일 때는 유휴시간은 발생하지 않지만 이 시스템은 재개 점에 도달할 수 없다. 따라서 L_j 개의 일감들을 수행한 후에는 가용공간이 줄어들게 되어 보완작업을 발생시킬 가능성이 있다. 본 연구에서는 보완작업을 야기시키지 않는 m_j 의 최소값인 전자의 정의를 따르기로 한다.

새로이 구해진 k_j^1 , k_j^2 , m_j 값을 작업순서규칙 2에 적용하는 작업순서규칙 3을 도출할 수 있다. 작업순서규칙 3에 따르면 $H_j^1 < H_j^1$ 이고 $H_j^2 < H_j^2$ 일 경우에는 여전히 보완작업이 최소화된다. 그렇지 않은 경우에는 최적성이 보장되지 않는다. 따라서 작업순서규칙 3은 휴리스틱 규칙임을 알 수 있다. 한편 MMAL 단일작업장에서 작업순서규칙 2에 따르면 최적작업순서를 결정할 수 있음을 알 수 있었다[2, pp. 34 - 39].

2.4. 성과측정을 위한 대응척도의 도입

2.4.1. (1) 과부하량의 개념

슬롯



〈그림 2.1〉 j작업장의 최적 작업순서

〈그림 2.1〉은 k_j^2 , k_j^1 와 m_j 값이 각각 2, 3, 7인 j 작업장의 최적 작업순서를 보여주고 있다. l_j 개의 연속된 일감들 중 $k_j^2+k_j^1$ 개의 선택작업 일감들과 m_j 개의 기본작업 일감들이 포함되어 있음을 알 수 있다. 만약 슬롯 $i+l_j$ 의 일감이 O_p 이면 a^p+b_j 만큼 총보완작업량이 발생할 것이다. 한편 l_j 개의 슬롯으로 구성되는 하위순서에는 $(i+1, \dots, i+l_j)$, $(i+2, \dots, i+l_j+1)$, \dots , $(i+l_j-1, \dots, i+2l_j)$ 에는 (k_j^p+1) 개의 O_p 와 k_j^q 개의 O_q 가 포함되어 있음을 알 수 있다. j작업장의 윈도우를 j작업장의 연속된 슬롯들의 집합으로 정의하고, 임의의 t윈도우에 포함된 O_1 , O_2 의 개수를 t윈도우의 부하량이라 정의하고, 이를 WL_t^p 로 나타낸다. WL_t^p ($p=1, 2$)의 수학적 표현은 다음과 같다.

$$\sum_{i \in R_j^p} X_{i1}^p + \dots + X_{i-1}^p + X_{ii}^p \quad t=1, \dots, l_j-1 \tag{2.8}$$

$$\sum_{i \in R_j^p} X_{i-l_j+1}^p + \dots + X_{i-1}^p + X_{ii}^p \quad t=l_j, \dots, N \tag{2.9}$$

$$\sum_{i \in R_j^p} X_{i-l_j+1}^p + \dots + X_{iN-1}^p + X_{iN}^p \quad t=N+1, \dots, N+m_j-1 \tag{2.10}$$

R_j^p : 작업장 j에서 선택작업 p를 요하는 일감들의 지수집합.

$X_{ik}^p \Rightarrow 1$ (일감 i가 슬롯 k에 할당되었을 때)
 0 (일감 i가 슬롯 k에 할당되지 않았을 때)

본 연구에서는 O_1 , O_2 로 인한 부하량 계산시 O_2 를 기준으로 하여 고찰하므로 O_1 의 추가로 인한 부하 WL_t^1 을 β_j^1 ($=\frac{O_j^1 - b_j}{O_j^2 - b_j}$)의 비율을 이용하여 조정함으로써 O_1 , O_2 로 인한 부하량을 다음과 같이 계산할 수 있다.

$$WL_t^p = \beta_j^1 \cdot WL_t^1 + WL_t^2 \tag{2.11}$$

끝으로 t 윈도우에서 j 작업장의 과부하량을 V_t^j 라 하고 이를 허용된 선택작업량의 초과분이라 하자.

$$V_i = \text{Max}(0, WL_i - k_i^z) \quad (2.12)$$

여기서 $t_j \leq \text{Min}(k_j^1, k_j^2)$ 이거나 $t_j \geq N + l_j - \text{Min}(k_j^1, k_j^2)$ 일 경우에는 윈도우의 길이가 짧기 때문에 작업순서규칙 2에 따른 할당이 행해질 경우에는 과부하가 발생하지 않는다. 따라서 이러한 부분 윈도우들은 고려대상에서 제외할 수 있다. 한편 위의 식은 다음과 같이 나타낼 수 있다.

$$WL_i - V_i + E_i = k_i^z \quad (2.13)$$

E_i 는 t 윈도우에 과부하가 없을 경우 그 값이 양인 여유변수이다. (2.1)식과 (2.2)식을 충족시키는 k_j^1, k_j^2 와 m_j 가 존재하고 $H_j^1 > H_j^z, H_j^2 > H_j^z$ 이면 작업순서규칙 2에 의해 구해지는 최적해는 끝부근의 소규모 윈도우들의 효과가 무시될 수 있을 때 거의 $[\beta_j^1(H_j^1 - H_j^z) + (H_j^2 - H_j^z)]l_j$ 단위의 과부하량을 지니게 된다.

한편 $H_j^1 < H_j^z, H_j^2 < H_j^z$ 이면 작업순서규칙 2에 의한 총과부하량은 0이다. 이 경우 작업순서규칙 2에 따라 일감을 할당하면, 각 윈도우의 선택작업량은 임계치 이하이다. 따라서 정의에 의하면 각 윈도우의 과부하량은 0이다.

작업순서규칙 2에 의해 생성된 작업순서의 총과부하량을 계산할 수 있다. 이때에도 앞에서와 마찬가지로 3가지 경우를 고려할 수 있다[2, pp. 45-62, pp. 110-126].

2.4.2. 대응척도의 도입

적절한 가중치를 지닌 대응척도가 총보완작업량과 거의 같은 기능을 행할 수 있다. 따라서 작업장별로 총과부하량을 총보완작업량으로 전환시키기 위해 가중치가 필요하다. 앞 절에서 우리는 작업순서규칙 2에 의한 작업순서에서 재개순서

유형을 여기는 일감은 대략 l_j 만큼의 과부하량과 $o_j^z - b_j$ 만큼의 보완작업량을 야기시킴을 보았다. 따라서 j 작업장에 있어서 보완작업량을 대응척도인 과부하량으로 표현하기 위하여 과부하량에 적용해야 할 가중치 θ_j 는 다음과 같이 나타낼 수 있다.

$$\theta_j = \frac{o_j^z - b_j}{l_j}$$

따라서 모든 작업장에서 발생하는 과부하량의 가중 합은 다음과 같이 나타낼 수 있으며, 이는 총보완작업량에 관한 하나의 대응척도로 생각할 수 있다.

$$\sum_{j=1}^M \theta_j \sum_{i=\text{Min}(k_j^1, k_j^2)+1}^{N+l_j-\text{Min}(k_j^1, k_j^2)-1} V_i$$

작업순서결정규칙 2에 의해 얻어지는 해들에서 두 가지 척도간의 관계는 다음과 같다. 즉, (case A)일 때 $r_j = k_j^z$ 이거나 $r_j = k_j^z + k_j^z$ 이고 과부하량이 $\theta_j = (o_j^z - b_j) / l_j$ 에 의해 가중처리될 경우 작업순서규칙 2에 의한 작업순서에 대해서 총보완작업량은 총가중과부하량과 같다. (case B)와 (case C)일 때는 $r_j = k_j^z$ 일 경우 θ_j 에 의해 가중처리된 총보완작업량과 같다[2, pp. 127 - 128].

따라서 특정 경우에는 과부하량이 보완작업량을 그대로 반영하고 있음을 알 수 있다. 그 밖의 경우에도 과부하량 계산시 발생 가능한 모든 경우를 고려하였으므로 보완작업량의 대응척도로 과부하량을 이용함에 문제가 발생하지 않을 것으로 생각되며, k_j^1 와 m_j 에 대한 가정을 완화하여 다 작업장을 고려할 때도 유용할 것으로 기대된다.

3. 다작업장에서의 MMAL 작업 순서결정

일반적으로 MMAL은 모든 일감에 대해 동일한 작업을 행하는 작업장, 작업시간이 동일한 여러 가지 사양 중 하나를 선택할 수 있는 작업장, 그리고 작업시간이 상이한 작업들을 수행하는 작업장 등 세 가지 유형으로 구성되어 있다. 3 장에서는 MMAL 작업장의 세 번째 유형 중, 작업장의 수가 M개이고 작업준비비가 발생하지 않는 MMAL의 작업순서결정문제의 수리모델과 이 모델의 휴리스틱 규칙들을 제시하고, 각 해법의 효율성을 검증하고자 한다.

3.1. 다작업장의 작업순서결정을 위한 수리모델 구축

TSP를 이용한 MMAL을 위한 일반수리모델(MPF)에서는 가능한 생산능력을 효율적으로 이용하고자 하는 데 그 목표를 두고 있다. 많은 연구자들이 이러한 유형의 목적함수를 사용했지만 아직까지 효율적이라고 검증된 해법은 존재하지 않는다. Okamura와 Yamashina의 모델은 MPF와 유사한 특성을 지니고 있다[16]. 이들 연구에서는 작업내의 허용된 공간을 효율적으로 이용하고자 했다. 그러나 그들의 모델은 최대 보완작업량을 최소화하고 있다. 이 목표는 매우 나쁜 평균성과를 초래한다. 최대보완작업량 최소화 목표보다는 평균 보완작업 소요량 최소화 목표가 바람직하다.

Coffman 등[6]과 Bird[3]는 과부하량과 생산능력의 효율적 이용간의 관계를 설정하지 않았다. 그들은 단순히 k_j 와 m_j 가 주어진 것으로 가정했다. 따라서 생산능력의 이용을 측면에서 그들의 해를 해석할 방법이 없다.

본 연구의 2 장에서는 각 작업장의 작업유형이 3개일 때 생산능력의 효율적 이용을 위해서는 보완작업량을 최소화하여함을 알 수 있었으며, 보완작업량에 대한 대응책으로서 과부하량의 개념을 도입했다. 따라서 TSP를 이용한 MMAL을 위한 일반수리모델(MPF)을 과부하량을 최소화하는 수리모델로 변형할 수 있다.

TSP를 이용한 MMAL을 위한 일반수리모델(MPF)을 과부하량을 최소화하는 수리모델(MMPF)로 수정하면 다음과 같다.

$$\text{Min. } \sum_{j=1}^M \theta_j \sum_{i=1}^{N+l_j-\text{Min}(k_j^1, k_j^2)-1} V_{ij} \quad (3.1)$$

$$\text{s.t. } \beta_j^1 \sum_{i \in R_j^1} \sum_{h=1}^{l_i} X_{ih}^1 + \sum_{i \in R_j^2} \sum_{h=1}^{l_i} X_{ih}^2 + V_{i-\text{Min}(k_j^1, k_j^2)_j}$$

$$+ E_{i-\text{Min}(k_j^1, k_j^2)_j} = k_j^{2'} \quad (j=1, \dots, M$$

$$t = \text{Min}(k_j^1, k_j^2) + 1, \dots, l_j - 1) \quad (3.2)$$

$$\beta_j^1 \sum_{i \in R_j^1} \sum_{h=1}^{l_i} X_{i-l+h}^1 + \sum_{i \in R_j^2} \sum_{h=1}^{l_i} X_{i-l+h}^2 +$$

$$V_{i-\text{Min}(k_j^1, k_j^2)_j} + E_{i-\text{Min}(k_j^1, k_j^2)_j} = k_j^{2'}$$

$$(j=1, \dots, M \quad t = l_j, \dots, N) \quad (3.3)$$

$$\beta_j^1 \sum_{i \in R_j^1} \sum_{h=1}^{N+l_j-t} X_{i-l+h}^1 + \sum_{i \in R_j^2} \sum_{h=1}^{N+l_j-t} X_{i-l+h}^2 +$$

$$V_{i-\text{Min}(k_j^1, k_j^2)_j} + E_{i-\text{Min}(k_j^1, k_j^2)_j} = k_j^{2'}$$

$$(j=1, \dots, M \quad t = N+1, \dots,$$

$$N+l_j-\text{Min}(k_j^1, k_j^2)-1) \quad (3.4)$$

$$\sum_{h=1}^N X_{ih}^2 = 1 \quad i=1, \dots, N \quad (3.5)$$

$$\sum_{h=1}^N X_{ih}^2 = 1 \quad h=1, \dots, N \quad (3.6)$$

$$X_{ih}^2 = 0, 1$$

$$V_{ij}^2, E_{ij}^2 \geq 0$$

$$\text{단, } \begin{cases} X_{ih}^1 = 0, 1 (O_1 \text{이 배정될 때만 } 1 \text{이됨}) \\ X_{ih}^2 = 0, 1 (O_2 \text{가 배정될 때만 } 1 \text{이됨}) \end{cases}$$

MMPF는 이진혼합정수계획 문제이다. MM-PF에서 이진변수의 수는 $2N^2$ 개이고 연속변수의 수는 $2(N+l_j-1)$ 개이다. 한편 윈도우의 일감 수가 $\min(k_j^1, k_j^2)+1$ 보다 작으면 과부하량은 발생하지 않는다. MMPF에서 총제약식의 수는 $(3N+l_j-1)$ 개이고 이중에서 (3.2), (3.3), (3.4)식, 즉 $(N+l_j-1)$ 개의 식은 과부하량과 관련된 것이다. 그리고 나머지 (3.5), (3.6)식, 즉 $2N$ 개는 할당제약식이다. 따라서 MMPF는 MPF보다 복잡도가 낮다.

한편 선택작업유형이 한가지인 MMAL에 관한 Bolat의 수리모델에서는, 이진변수가 N_2 개, 연속변수가 $2(N+m_j-k_j-1)$ 개, 정수제약 조건식은 $(3N+m_j-k_j-1)$ 개이다. 그는 선택작업 유형이 한가지인 이 수리모델에서 분지한계법을 적용하여 최적해를 산출하고자 하였으나 대규모 문제에서는 비효율적이라는 실험결과를 얻을 수 있었다 [4].

따라서 선택작업의 수가 두 개인 경우의 MMPF는 Bolat의 연구에서 설정한 수리모델보다 복잡하므로 최적해 산출이 대단히 어렵다. 그리하여 본 연구에서는 다작업장 혼합모델조립 라인의 작업순서결정을 위해 휴리스틱 규칙을 개발하고자 한다.

3.2. 휴리스틱 규칙 개발

3.2.1. 분지한계법을 이용한 휴리스틱 규칙

다작업장에서의 작업순서결정을 위하여 분지한계법(branch and bound method)을 적용할 수 있는데, 분지한계법의 적용 원리는 다음과 같다. 분지한계절차를 이용하여 어떤 단계에서 N 개의 일감 중 이미 작업순서가 결정되어 할당된 일감의 집합(partial sequence라 한다)을 분지한계 트

리(branch and bound tree)의 마디(node)라 하면, 어떤 마디에서 다음에 할당 가능한 일감의 수는 $N-N'$ 개 (N' : 어떤 마디에서 이미 할당된 일감의 수, 즉 $N-N'$ 은 남아 있는 일감의 수)가 되어 이 마디에서 분지되는 후속 마디의 수도 $N-N'$ 개가 된다.

n 개의 일감을 작업장에 할당하는 순서를 결정하는 문제는 $n!$ 개의 가능한 순서집합이 발생한다. 따라서 분지한계트리에서 마디(node)의 수(배정완료작업에 대한 경우의 수: subsequence의 수)는 $n+n(n-1)+\dots+n!$ 개가 된다. 따라서 대규모 MMAL문제의 최적해를 도출한다는 것은 컴퓨터 저장량이나 계산시간 측면에서 현실적으로 거의 불가능하다.

그러나 분지한계법에서는 깊이우선탐색(depth-first search)과 선형계획문제로의 이완이나 라그랑지 이완에 의하여 각 마디에서 이완문제의 해를 MMPF의 하한으로 고려함으로써 탐색할 마디의 수를 감소시킬 수 있다. 선택작업이 하나인 MMAL의 문제에 관한 Bolat의 연구에 의하면 한계전략으로 LP이완을 사용하는 경우에는 대규모 문제에서 최적해 산출이 어려운 것으로 밝혀진 바 있다[4]. 따라서 본 연구에서는 각 마디에서 하한을 산출하는 방법으로 제 2장에서 제시한 방법을 적용하고자 한다.

배정완료작업에 관련된 과부하량은 윈도우를 조사하여 헤아릴 수 있다. 분지한계법에서 배정완료작업의 모든 경우에 대하여 이런 정보를 저장한다면 최선의 경우에 있어서도 전체 일감의 순서를 결정하기까지는 너무나 많은 경우에 대한 과부하량을 저장해야 한다. 따라서 컴퓨터 저장용량의 문제가 발생하게 된다. 한 작업장에 관련된 작업순서결정은 다른 작업장들에 관련된 의사결정에 영향을 미치지 않고 배정완료작업은 각 작업장에서 재개시점에서 끝난다고 가정한다면, 각

작업장에 대해 남은 일감들의 순서를 독립적으로 결정할 수 있다. 따라서 남은 일감에 대해서 각 작업장별로 작업순서규칙 2를 적용하여 최적 총부하량을 발견할 수 있다. 이 접근법은 사전에 순서가 결정된 일감들의 최종 효과와 작업장간의 상호관계를 고려하고 있지 않으므로 배정완료작업과 미배정작업에 대한 작업순서규칙 2에 따른 작업순서의 과부하량 합이 최적 작업순서의 과부하량 수에 대한 하한이 될 것이다.

본 연구에서의 휴리스틱 분지한계법은 트리의 넓이우선탐색(breadth first search)을 적용하여 배정완료작업 즉, 후보 마디들(partial sequence) 중에서 과부하량의 추정치가 최소인 마디만을 선택하여 순서가 결정되지 않은 일감을 할당해 가는 방법이다.

분지한계 과정에서 생성되는 배정완료 작업의 과부하량 추정치는 이미 순서가 결정된 일감에 의해 각 작업장에서 발생하는 과부하량의 합과 나머지 일감에 의해 추가로 발생할 과부하량 하한치의 합이다. 한편 각 배정완료작업의 과부하량 하한치는 2장에서 설명한 작업순서규칙 2에 따른 과부하량 계산방법에 의해 산출된다.

본 연구의 휴리스틱 분지한계법은 FORTRAN으로 프로그래밍 되었고 난수에 의해 산출된 문제를 대상으로 그 효율성을 검증하고 있다.

본 연구에서 개발한 휴리스틱 규칙(New Heuristic Rule, NHR)의 해 산출절차는 다음과 같다.

NHR의 해산출 절차

단계 0: 초기화 및 입력자료

입력자료: $n, m, L_j, o_j^1, o_j^2, b_j$,

$RO(j, i)=0$: basic

1: option 1

2: option 2

초기화: $m_j \leftarrow (L_j - 1) / (1 - b_j)$

List = {1, ..., n}

queue = 0

level = 1

$n_0 = n$

단계 1: 보완작업을 발생시키지 않는 최대 선택 작업 일감량 (H_j^1, H_j^2)의 계산

$n_0 \leftarrow n - \text{level}$

단계 2: 후보 partial sequence에서 마지막으로 할당된 일감 다음에 할당할 일감의 선택

2-1: 할당되지 아니한 일감(즉, List에 있는 임의의 일감), i 를 선택하고 그 일감이 할당되었다고 가정한 경우, List - { i }의 일감을 대상으로 H_j^1, H_j^2 를 계산한다.

2-2: partial sequence에 일감, i 가 할당된 경우의 subsequence에서 과부하량을 계산하고 미할당된 일감, 즉 List - { i } 일감들에 의한 과부하량의 하한치를 계산한다.
 $npv \leftarrow$ subsequence에 의한 과부하량
 $tnvlt \leftarrow$ MMAL의 작업순서규칙에 의한 미배정된 일감의 과부하량 하한치

$isum \leftarrow npv + tnvlt$

2-3: List에 있는 모든 i 에 대하여 2-1과 2-2의 과정을 반복하여 $isum$ 이 최소인 일감 i^* 를 선택한다.

단계 3: partial sequence에 후보일감을 할당

queue \leftarrow queue + { i^* }

List = List - { i^* }

level \leftarrow level + 1

level = n 이면 단계 4로 간다. 아니면 단계 1로 간다.

단계 4: 결과의 출력 및 종료

3.2.2. 실무중심 휴리스틱 규칙

실제 현실에서 많이 이용되는 다음과 같은 휴리스틱 규칙(Practical Heuristic Rule, PRH)을 생각해 볼 수 있다.

단계 1: 각 작업장별 k^1, k^2, m_j 를 계산

단계 2: k_j^1, k_j^2, m_j 의 산술평균값 k^1, k^2, m 을 계산하여 이 세 값의 합을 l 로 둠

단계 3: 전체 작업장에 대하여 각 일감별 O_1, O_2 , 기본작업일감(B)의 비율을 계산하여 다음과 같은 규칙에 따라 일감을 l 개씩 반복배정

- 3-1: O_2 비율이 가장 높은 일감을 k^2 개 배정한다.
- 3-2: O_1 비율이 가장 높은 일감을 k^1 개 배정한다.
- 3-3: B 비율이 가장 높은 일감을 m 개 배정한다.
- 3-4: 미배정 일감이 존재하지 않으면 단계 3을 종료한다.

단계 4: 결과출력 및 종료

3.3. 제 기법의 평가를 위한 실험설계

본 연구는 두 가지 휴리스틱 규칙의 성능을 검

증하기 위하여 두 가지 실험자료 <표 3.1>과 <표 3.2>를 이용하여 실험을 수행했다.

작업장들의 길이와 작업시간은 식 (2. 1)과 식 (2. 2)에 의한 정수 k_j^1, k_j^2 와 m_j 가 존재하도록 설정되었다. 그리고 대상일감들의 수를 변화시켜 보았다. 각 작업장의 선택작업 수를 $k_j^1/l_j, k_j^2/l_j$ 의 비율로 배정했으며, 각 작업장에 있어서 각 일감의 선택작업유형은 난수를 발생시켜 결정했다. 각 작업장의 $k_j^1+k_j^2$ 가 m_j 보다 작도록 설정했다. 따라서 각 문제들에 있어서 l_j 개의 작업 중 평균 $k_j^1+k_j^2$ 개의 선택작업이 요구되었으며 작업당 평균 서비스시간은 거의 1 단위시간이었다.

그리고 작업장수의 변화에 따른 과부하량의 변화를 살펴보기 위해 작업장 수가 5개, 10개인 두 가지 경우를 고려했으며, 작업장 크기의 효과를 보다 명확히 알아보기 위해 작업장 수가 10개인 경우는 전자와 동일한 5개의 작업장이 추가되는 상황을 가정하여 실험자료를 구축했다.

한편 실제 현장에서는 각 작업장마다 선택작업 유형의 개수가 상이할 수 있지만 본 연구에서는 새로 개발한 휴리스틱 규칙의 효율성을 분석해 보기 위해 편의상 선택작업유형이 모두 세 가지로 이루어진 실험데이터를 이용하였다.

<표 3.1> 작업장 수가 5개인 실험자료

작업장번호	작업장길이	작업자 수	작업시간			k_j^1	k_j^2	b_j
			o_j^1	o_j^2	b_j			
1	2.0	1	1.20	1.80	0.75	1	1	4
2	3.0	1	1.30	1.70	0.66	2	2	6
3	3.0	1	1.4	1.8	0.66	3	1	6
4	4.0	1	1.20	1.80	0.70	3	3	10
5	7.0	1	1.50	2.00	0.60	4	4	15

〈표 3.2〉 작업장수가 10개인 실험자료

작업장번호	작업장길이	작업자 수	작업시간			k_j^1	k_j^2	b_j
			o_j^1	o_j^2	b_j			
1	2.0	1	1.20	1.80	0.75	1	1	4
2	3.0	1	1.30	1.70	0.66	2	2	6
3	3.0	1	1.40	1.80	0.66	3	1	6
4	4.0	1	1.20	1.80	0.70	3	3	10
5	7.0	1	1.50	2.00	0.60	4	4	15
6	2.0	1	1.20	1.80	0.75	1	1	4
7	3.0	1	1.30	1.70	0.66	2	2	6
8	3.0	1	1.40	1.80	0.66	3	1	6
9	4.0	1	1.20	1.80	0.70	3	3	10
10	7.0	1	1.50	2.00	0.60	4	4	15

또한 모집단에 대한 표본의 대표성을 높이기 위해 층화추출법을 이용하여 각 작업장의 작업일감수를 200개, 400개, 600개, 800개, 그리고 1000개로 설정하였다. 그리고, 작업장 수와 작업일감수가 동일한 경우에 있어서 난수를 변경하여 각기 7개씩의 데이터를 생성하였다. 그리하여 작업장 수가 5개, 10개인 경우에 대해 각기 35개씩의 표본자료를 이용하여 실험을 수행하였다.

3.4. 제 기법의 비교 분석

본 연구에서는 실험수행을 위하여 486PC (33MHz)를 이용하였다. 그리하여 작업장의 수가 5개, 10개인 경우에 있어서 반복실험을 통해 NHR(New Heuristic Rule) 과 PHR(Practical Heuristic Rule)의 성과를 비교 분석해 보았다. 통계분석 패키지로는 SAS /PC를 이용했다.

NHR과 PHR을 이용하여 각 일감수 별 7개 표본에 대한 실험결과 얻어진 평균 과부하량과 이

를 해당일감수로 나누어서 구한 일감수 단위당 평균 과부하량과 각 표본을 1회 실험하는데 소요된 CPU 시간(단위:초)을 〈표 3.3〉에 나타내었다. 이 표를 통해 우리는 과부하량 평균치와 소요 CPU시간의 두 가지 측면에서 두 기법을 비교 분석해 볼 수 있다.

〈표 3.3〉 NHR과 PHR의 과부하량 및 CPU시간

작업장수	기법 일감수	N H R			P H R		
		과부하량	평균	CPU시간	과부하량	평균	CPU시간
5	200	233	1.17	13.06	494	2.47	0.07
	400	489	1.22	54.29	1041	2.60	0.14
	600	717	1.20	122.78	1553	2.59	1.33
	800	994	1.24	218.63	2029	2.54	0.41
	1000	1270	1.27	341.50	2676	2.68	0.59
10	200	573	2.87	25.52	954	4.77	0.11
	400	1193	2.98	105.79	1997	4.99	0.24
	600	1551	2.59	240.66	2881	4.80	0.45
	800	2230	2.79	430.39	3862	4.83	0.72
	1000	2867	2.87	671.01	4936	4.94	0.99

우선 평균 과부하량을 비교해보면 작업장 수가 5일 경우에는 NHR의 평균 과부하량이 PHR의 그것보다 훨씬 작으며, 대부분 절반에도 못 미침을 알 수 있다. 그리고 작업장 수가 10일 경우에도 NHR의 평균 과부하량이 PHR의 값보다 훨씬 작음을 알 수 있다.

이 결과는 동일한 표본자료에 대해 두 가지 기

법을 적용하여 결과를 얻었으므로 쌍별(Paired) T-test를 통해 두 가지 기법의 성과를 비교분석할 수 있다. 그 결과 작업장 수가 5개, 10개인 경우 모두 매우 유의적인 차이를 나타내고 있음을 <표 3.4>에서 알 수 있다. 따라서 평균 과부하량을 일 고려할 때 NHR이 PHR보다 우수함을 알 수 있다.

<표 3.4> 쌍별 T-test 결과 요약표

작업장수	표본수	과부하량 차이평균	표준편차	T	$prob > t $
5	35	814.11	71.0137	11.4642	0.0001
10	35	1371.71	110.4640	11.5919	0.0001

한편 해를 구하는데 소요된 CPU시간은 분지한 계법을 적용한 NHR이 PHR에 비해 상대적으로 훨씬 많이 소요됨을 알 수 있다. 그러나 NHR의 소요 CPU시간의 절대치가 아주 큰 값이 아니고, 또 보다 빠른 PC기종을 사용하면 CPU 소요시간은 큰 문제가 되지 않으리라 생각할 수 있다. 그러므로 전체적으로 고려해 볼 때 NHR이 PHR보다 우수하다고 할 수 있을 것이다.

4. 준비비용을 고려한 일반MMAL의 작업순서결정

4.1. 일반 MMAL의 특성 및 연구동향

앞에서 MMAL 작업장 유형을 크게 3가지로 구성됨을 살펴보았다. 앞 장까지는 문제의 복잡성으로 인해 우선 두 번째 작업장에 초점을 맞추어 보완작업비용을 감소시키는 MMAL의 작업순서결정을 고찰했다. 그러나 대부분의 MMAL에서는 준비비용도 야기되므로 이를 고려함이 당연하

다할 수 있다.

준비비용을 야기시키는 작업유형의 대표적인 예로써 자동차산업의 도색작업을 들 수 있다. 그런데 특수경우를 제외하고 도색작업장에 있어서 발생하는 준비비용은 대부분 도색의 변화로 말미암은 작업전환을 위해 사용한 일부 도색재료의 폐기 등으로 발생하므로 작업색상의 순서에 따라서는 크게 달라지지 않음을 생각할 수 있다. 따라서 본 연구에서는 후속 색상의 종류에 무관하게 색상이 바뀔 때마다 항상 일정하게 준비비용이 발생한다고 가정했다.

일반 MMAL문제에 관한 연구는 흔하지 않다. Burns와 Daganzo가 이 문제를 다루었으나, 그들은 기본작업과 선택작업을 수행하는 작업장에서의 보완작업을 모형화하기 보다는 준비작업 유형으로 작업들 사이의 작업전환을 고찰했다[5]. 그들은 단순한 순위발주규칙(ranking ordering algorithm)을 제시하였으며, 일감들을 작업비용에 따라 구분 지었으며, 그룹의 수가 늘어날수록 한계절약분이 줄어든다는 것을 지적했다. Boat은

분지한계법을 이용한 단일작업장에서의 최적작업 순서규칙을 확장하여 색상선택에 이용했다[4]. 그리하여 그의 휴리스틱이 일반적인 휴리스틱규칙보다 나음을 제시했다. 그러나 앞에서도 언급했듯이 그는 선택작업의 유형을 1개로 국한하였다.

본 연구에서는 Bolat의 연구를 확장하여 각 작업장에서의 작업유형이 기본작업과 선택작업 1, 2, 즉 세 가지 작업유형이 행해지는 경우를 다루고자 한다. 그리하여 총 보완작업 비용을 최소화 가 달성될 수 있는 휴리스틱 규칙을 개발코자 한다. 일반 MMAL 모형은 앞에서 다룬 세 번째 유형의 작업장에 초점을 맞춘 MMAL모형에 도색 작업장과 같은 두 번째 유형의 작업장을 추가한 것과 같다. 따라서 우선 문제를 단순화하기 위하여 배취크기는 일정하다고 가정한 후 색상선택규칙에 따라 성과를 비교하여 우수한 휴리스틱 규칙을 도출하면 될 것이다.

최적 배취크기를 결정하는 것은 매우 어렵다. 왜냐하면 그것은 자료에 따라 달라지기 때문이다. 만약 어떤 일감에 대해서든지 각 색상이 거의 같게 요구되고 색상이 다른 옵션들과 독립적이라면 최적해 중의 하나는 거의 동일한 배취크기로 나타날 것으로 기대할 수 있다. 그리하여 본 연구에서는 배취크기를 b^* (b star)로 일정하게 두고 분석을 수행한다.

4.2. 일반 MMAL 작업순서결정을 위한 휴리스틱 규칙 개발

색상선택을 고려한 MMAL 작업순서결정규칙으로서 일반적으로 랜덤고정순서규칙, 고정내림차순규칙, 고정오름차순규칙 등을 생각할 수 있으며, 앞에서 고려한 NHR을 수행하여 색상을 선택하는 확장NHR규칙(ENHR)을 생각해 볼 수 있다.

4.2.1. 랜덤고정순서규칙 (Random Sequencing Rule, RSR)

우선 색상의 순서를 임의로 결정하고, 결정된 색상의 임의적 순서에 따라 배취크기 만큼씩 일감을 배정하는 규칙이다. 여기서는 색상순서의 결정 후 해당 색상의 일감을 배취크기 만큼씩 배정할 때 임의적으로 배정하는 방법(Random Sequencing Rule, RFR)과, 앞에서 개발된 NHR을 적용하는 방법(Fixed Random Ordering Rule, FROR)의 두가지를 생각해 볼 수 있다.

4.2.2. 고정내림차순규칙 (Fixed Descending Ordering Rule, FDOR)

각 색상별 일감 수를 계산하여 일감 수가 많은 색상부터 일감 수가 적은 색상으로 색상순서를 결정하여 배취크기 만큼씩 일감을 배정하는 방식이다. 물론 동일 색상의 일감 배정시 RSR이나 NHR을 사용할 수 있을 것이다. 앞서 NHR이 우수한 것으로 판단되므로 NHR을 적용하여 동일 색상 일감들 간의 배정순서를 결정한다. 이 방법은 색상의 반복주기 중 초기에 보다 큰 유연성을 가져다 줄 수 있다.

4.2.3. 고정오름차순규칙 (Fixed Ascending Ordering Rule, FAOR)

이것은 앞의 규칙과는 정반대로 일감 수가 적은 색상부터 일감 수가 많은 색상으로 색상순서를 결정하여 배취크기 만큼씩 일감을 배정하는 규칙이며 색상 반복주기의 끝 부분에 보다 큰 유연성을 가져올 수 있다.

4.2.4. 확장NHR규칙 (Extended New Heuristic Rule, ENHR)

앞의 세 가지 규칙은 배정 일감이 남을 때까지 고정된 색상순서를 유지하는 방식이었다. 그러나 이와 달리 각 배취크기씩 배정할 때마다 NHR 개념을 적용하여 가장 적은 과부하량을 발생시키는 색상을 찾아내어 색상순서를 결정하는 기법을 생각해 볼 수 있다. 이 경우 이용가능한 각 색상의 일감들에 대해 해당순서에 위치할 경우 발생할 총보완작업비용을 최소화하는 특정색상의 b^* 개의 일감을 찾아내고 나머지 일감들에 대해서는 총가중과부하량의 하한치를 결정한다. 그리고 나서 각 일감별 보완작업의 증분치와 가중과부하량의 합을 구하여 이를 최소화하는 색상을 택하여, 이 색상에 해당되는 일감들을 배정한다.

본 연구의 휴리스틱 분지한계법은 FORTRAN으로 프로그래밍 되었고 난수에 의해 산출된 문제를 대상으로 그 효율성을 검증하고 있다.

ENHR의 해 산출 절차

단계 0: 초기화 및 자료입력

단계 1: 일감별 색상배정

단계 2: 후속 배취의 색상결정 해당 일감순서 결정

2-1: 각 색상별로 남은 일감에 대해 배취크기 또는 남은 일감 수가 배취크기보다 작을 경우에는 그 전체를 대상으로 NHR을 적용하여 각 색상별 과부하량을 최소화하는 작업순서와 그에 해당되는 최소과부하량을 구하고 미배정한 일감들의 과부하량의 이론적 하한치를 구해 합산한다.

2-2: 2-1에서 구한 각 색상별 최소과부하량의 합 중에서 가장 작은 값을 지닌 색상을 찾아 이를 후속배취의 색상으로 결정한

다.

2-3: 후속배취의 작업순서는 색상순서 결정시 이미 구해진 것을 그대로 사용하여 일감순서를 누적해간다.

2-4: 미배정 일감이 남지 않을 때까지 2-1부터 2-3까지의 과정을 반복 수행한다.

단계 3: 누적된 전체 일감들의 작업순서에 따른 결과를 계산

단계 4: 결과 출력 및 종료

4.3. 제 규칙의 비교 분석

본 연구에서는 일반 MMAL 작업순서결정을 위한 여러 가지 휴리스틱 규칙의 성과를 비교 분석하기 위해 '3.3. 제기법의 평가를 위한 실험설계'에서 이용한 <표 3.1>과 같은 5개의 작업장으로 이루어진 랜덤데이터를 사용했다. 각 작업장에서 수행되는 일감 수가 200, 400, 600, 800, 그리고 1000개인 경우를 고려했다. 그리고 각 일감 수마다 7개의 상이한 Seed Number를 부여해 각기 다른 선택작업유형을 임의로 발생시켜 보다 다양한 상황에서 각 기법의 성과를 분석코자했다. 그리고 색상의 수는 일단 5가지로 한정했으며 각 일감의 색상은 난수를 이용하여 임의로 결정했다. 실험수행을 위해서 486PC(33MHz)를 이용하였으며 통계분석을 위해서 SAS/PC 통계패키지를 사용했다. 그리하여 여러 가지 규칙을 사용했을 때의 분석결과를 <표 4.1>과 같이 요약할 수 있다.

이 표에서 '과부하량'은 일감 수별로 7개의 표본을 이용하여 ENHR, FROR, FAOR, 및 FDOR을 적용했을 때의 평균 과부하량을 나타내고 있으며, '평균'은 단위 일감당 과부하량을, 그리고 ENHR의 경우 'CPU 시간'은 1회 실험시 소요되는 평균 CPU 시간을 나타내고 있다. 또한

RFR의 결과는 ENHR의 수행에 소요된 CPU시간(단위 :초)동안 RFR 기법을 적용하여 반복실험 했을 때의 ‘반복실험 회수’와 각 실험의 과부하량 중 ‘최소과부하량’과 그 때의 단위 일감당 과부하량을 요약하고 있다.

〈표 4.1〉 규칙별 분석결과 요약표

기법 일감수	ENHR			RFR			FROR		FAOR		FDOR	
	과부하량	평균	CPU시간	반복 실험 회수	최 소 과 부하량	평균	과부하량	평균	과부하량	평균	과부하량	평균
200	426	2.13	9.89	230	481	2.41	459	2.30	480	2.40	474	2.37
400	904	2.26	51.29	572	995	2.49	966	2.42	974	2.44	916	2.29
600	1207	2.01	118.47	826	1528	2.55	1295	2.16	1340	2.23	1328	2.21
800	1610	2.01	220.27	1152	2093	2.62	1711	2.14	1794	2.24	1727	2.16
1000	2034	2.03	356.20	1434	2656	2.66	2243	2.24	2309	2.31	2174	2.17

본 논문에서는 실험설계시 동일표본자료에 대해 5가지 할당기법을 적용하였으므로, 난괴설계(Randomized Block Design)에 따른 분산분석을

통해 5가지 경우의 과부하량 차이에 관한 통계적 검증을 수행하여 〈표 4.2〉와 같은 결과를 얻을 수 있었다.

〈표 4.2〉 난괴설계법에 따른 분산분석결과

분산의 원천	분 산	자유도	평균자승합	F값	P값
집단간	70429788.80	34	2071464.38		
집단내		140			
할당방법	1880228.17	4	470057.04	43.03	0.0001
잔 차	1485655.03	136	10923.93		
전 체	73795672.00	174			

위의 표에서 볼 수 있듯이 할당방법에 따른 과부하량의 차이는 매우 유의적이라고 할 수 있다. 따라서 각 방법간의 차이를 분석하기 위해 피셔(Fisher)의 최소유의차(LSD, Least Significant Difference), 던컨(Duncan)의 다중범위검증(Multiple Range Test), 셰페(Sheffe)의 다중비교법(Multiple Comparison Method), 튜키(T-

ukey)의 방법 등을 적용하여 〈표 4. 3〉과 같은 결과를 도출했다.

〈표 4.3〉 난피설계법에 따른 분산분석결과($\alpha=0.05$)

할당기법	과부하량 평균	다중분류 기법			
		피셔 기법	던컨 기법	쉐페 기법	튜키 기법
RFR	1550.43	A	A	A	A
FAOR	1379.46	B	B	B	B
FROR	1334.80	C B	C B	B	B
FDOR	1323.86	C	C	B	B
ENHR	1236.46	D	D	C	C

위의 표에서 알 수 있듯이 튜키와 쉐페의 다중분류 기법에 따르면 ENHR과 FDOR, FROR, FAOR과 RFR의 세 집단으로 분류할 수 있으며, 보다 엄격한 다중분류 기법인 피셔와 던컨의 분류기법에 의하면 5가지 할당방법이 모두 상이함을 알 수 있다. 그러나, 후자에 따르더라도 전자에 의해 한 집단으로 분류된 FAOR, FROR, FDOR 중 FAOR과 FROR, 그리고 FROR과 FDOR의 결과들 사이에는 큰 차이가 발생하지 않음을 알 수 있다.

위의 분석을 통해 다음과 같은 결론에 도달할 수 있다. 즉, ENHR이 가장 우수하며, 그 다음으로 우수한 FAOR과 FDOR 및 FROR은 그 우열을 가리기 힘들며, RFR의 결과가 가장 좋지 않음을 알 수 있다.

일반 MMAL 작업순서결정에서는 보완작업비용과 준비작업비용의 합인 총비용을 최소화하는 의사결정을 행하기 위해 배척크기를 변화시켜 준비회수에 따른 준비비용과 그때 발생하는 보완작업비용의 합을 구해야 할 것이나, 본 연구에서는 배척크기의 변화에 따른 분석을 수행하지 못하고 배척크기가 항상 일정한 경우에 대한 실험을 수행하였다. 그러나 본 연구에서 개발한 ENHR이 여타 휴리스틱 규칙에 비해 우수하므로 앞으로 일반 MMAL 작업순서결정을 위해서는 이 규칙

을 적용하면 MMAL의 생산성을 제고시킬 수 있을 것으로 기대된다.

5. 결 론

MMAL의 작업순서결정시에는 작업준비비용과 보완작업비용을 동시에 고려해야 한다. 그러나 두 가지 비용의 합을 최소화하는 수리모델은 너무 복잡하여 현실문제를 적용할 경우 계산이 불가능하며, 따라서 두 가지 비용의 합을 최소화하는 일반적 MMAL의 작업순서결정에 관한 연구는 아직 많이 진척되지 못하고 있는 실정이다. 그러므로 본 연구에서는 일반적 MMAL의 작업순서결정을 위한 휴리스틱 규칙의 개발에 초점을 맞추었으며, 각 작업장에서 기본작업과 한 가지의 선택작업이 행해지는 경우의 작업순서결정을 고찰한 대부분의 선행연구와는 달리 각 작업장에서 기본작업과 두 가지의 선택작업이 행해지는 다작업장의 작업순서결정 문제를 연구했다.

그리하여 분지한계법으로 각 가지에서 배정 완료작업들의 과부하량과 미배정작업들의 과부하량의 합이 최소인 경우만을 고려하는 휴리스틱 규칙(NHR)을 이용하여 일반 MMAL 작업순서결정규칙을 개발하고자 하였다. 그리고 NHR을 일관성에서 많이 적용하는 휴리스틱 규칙(PHR)

과 비교 분석하여 NHR의 우수성을 검증하였다.

그리고 일반 MMAL의 작업순서결정을 위한 휴리스틱 규칙으로 고정랜덤순서규칙(FROR), 고정오름차순규칙(FAOR), 고정내림차순규칙(FDOR), 그리고 확장NHR규칙(ENHR)의 4가지 휴리스틱 규칙을 개발하였다. 그리고 이를 랜덤 데이터를 통해 실험분석하여 그 성과를 비교 검토하였다. 그 결과 임의적으로 고정색상순서를 결정하는 FROR, 동일색상 일감 수가 적은 색상부터 먼저 고려하는 FAOR, 동일색상 일감 수가 많은 색상부터 먼저 고려하는 FDOR보다 각 배취 크기씩 작업 시마다 NHR 개념을 적용하여 전체적으로 가장 적은 과부하량을 발생시키는 색상을 찾아내어 색상순서를 결정하는 ENHR이 항상 우수함을 알 수 있었다.

그러므로 앞으로 MMAL작업장에서는 본 연구에서 개발된 휴리스틱 규칙을 적용함으로써 생산성을 향상시킬 수 있을 것으로 기대되며, 한 작업장에서 여러 가지 작업을 수행하는 FMS에도 본 연구를 확장·적용할 수 있을 것으로 기대된다.

그러나 MMAL의 각 작업장에서 행해지는 작업유형의 수가 여러 가지임에도 불구하고, 본 연구에서는 각 작업장에서 기본작업과 두 가지의 선택작업이 행해지는 다작업장의 작업순서결정 문제를 연구했다. 그리고 MMAL의 각 작업장의 작업유형 사이에는 상관관계가 존재할 수 있다. 즉 한 작업장에서 선택작업이 요구되는 일감은 특정 후속작업장들에서 요구되는 작업유형이 일정한 가능성을 배제할 수 없다. 그러나 본 연구에서는 문제를 단순화하기 위해 MEAL의 각 작업장의 작업유형이 독립이라고 가정했다.

따라서 앞으로 일반적인 MEAL의 실제상황에 바로 적용될 수 있는 보다 나은 작업순서규칙을 개발하기 위해서는 위에서 언급한 선택작업유형 수, 그리고 각 작업장의 작업유형 사이의 상호관

계에 관련된 가정을 완화한 휴리스틱 규칙의 개발에 관한 연구가 활발히 이루어져야 할 것으로 생각된다.

참 고 문 헌

- [1] 김여근, 광재승, "관련 작업을 고려한 혼합 모델 조립라인 밸런싱," 「한국경영과학회지」, 제18권 제2호(1993), pp. 1-22.
- [2] 최종열, "혼합모델조립라인(MMAL)의 작업순서결정," 경북대학교 박사학위 논문, 1993. 12.
- [3] Bird, L. D., "Sequencing Vehicles for Assembly Under Precedence Constraints," *Paper Presented at ORSA Conference, Los Angeles, CA.*, April 15 (1986).
- [4] Bolat, A., "Generalized Mixed Model Assembly Line Sequencing Problem," *unpublished Ph. D. Dissertation*, Department of Indust. and Operations Engineering, The University of Michigan, Ann Arbor (1988).
- [5] Burns, L. D. and C. F. Daganzo, "Assembly Line Job Sequencing Principles," *Int. J. Prod. Res.*, Vol. 25, No. 1(1987), pp. 71-99.
- [6] Coffman, P. E., S. E. Hoffman and S. A. Weiner, "An O. R. View of Assembly Plant Modeling," *Working Paper, Ford Motor Company* (1985).
- [7] Dannenbring, D. G., "An Evaluation of Flow Shop Sequencing Heuristics," *Management Science*, Vol. 23, No. 11(1977), pp. 1174-1183.

- [8] Dar-El, E. M. and R. F. Cother, "Assembly Line Sequencing for Model-Mix," *Int. J. Prod. Res.*, Vol. 13, No. 5(1975), pp. 463-477.
- [9] Dar-El, E. M. and S. Cucuy, "Optimal Mixed-Model Sequencing for Balanced Assembly Lines," *Omega*, Vol. 5, No. 3 (1977), pp. 333-342.
- [10] Dar-El, E. M., "Mixed-Model Assembly Line Sequencing Problems," *Omega*, Vol. 6, No. 4(1978), pp. 313-323.
- [11] Dar-El, E. M. and A. Nadivi, "A Mixed-Model Sequencing Application," *Int. J. Prod. Res.*, Vol. 19, No. 1(1981), pp. 69-84.
- [12] Garey, M. R., and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.
- [13] Garfinkel, R. S. and G. L. Nemhauser, *Integer Programming*, John Wiley and Sons, New York, 1972.
- [14] Graves, S., "A Review of Production Scheduling," *Operations Research*, Vol. 29, No. 4(1981), pp. 646-675.
- [15] Monden, Y., *Toyota Production System*, Industrial Engineering and Management Press, Atlanta, 1983.
- [16] Okamura, K. and H. Yamashina, "A Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor," *Int. J. Prod. Res.*, Vol. 17, No. 3(1979), pp. 233-247.
- [17] Park, Y. B., C. D. Pegden and E. E. Enscore, "A Survey and Evaluation of Static Flowshop Scheduling Heuristics," *Int. J. Prod. Res.*, Vol. 22, No. 1(1984), pp. 127-141.
- [18] Reddi, S. S. and C. V. Ramamoorthy, "On the Flow Shop Sequencing Problem with No Wait in Process," *Operational Research Quarterly*, Vol. 23, No. 3 (1972), pp. 323-331.
- [19] Thomopoulos, N. T., "Line Balancing- Sequencing for Mixed-Model Assembly," *Management Science*, Vol. 14, No. 2 (1967), pp. 59-75.
- [20] Thomopoulos, N. T., "Mixed-Model Line Balancing with Smoothed Station Assignments," *Management Science*, Vol. 16, No. 9(1970), pp. 593-603.
- [21] Yano, C. A. and R. Rachamadugu, "Analytical Tools for Assembly Line Design and Sequencing," *Working Paper*, Department of Indust. and Operations Engineering, The University of Michigan, Ann Arbor (1990).
- [22] Yano, C. A. and R. Rachamadugu, "Sequencing to Minimize Work Overload in Assembly Lines with Product Options," *Management science*, Vol. 37, No. 5(1991), pp. 572-586.