

고정정반 일정계획시스템의 개발

박경철* · 이경식* · 박성수* · 김성환**

황규옥*** · 김완희**** · 유달호**** · 이동렬**** · 오명준****

Development of Bay Schedulers for Shipbuilding

K. Park · K. Lee · S. Park · S. Kim

K. O. Hwang · W. H. Kim · D. H. Yu · D. R. Lee · M. J. Oh

1. 서론

조선공정은 크게 선체건조(hull construction), 의장(outfitting) 및 도장(painting)으로 분류할 수 있다. 블럭조립공정은 선체건조공정의 일부로서 강재가공 후 취부(fitting), 용접(welding)하여 도크(dock)에서 탑재할 블럭을 만드는 공정이다. 일반적으로 블럭조립과정은 소조립, 중조립, 대조립으로 나뉜다. 블럭의 조립은 크게 이동정반과 고정정반에서 이루어진다. 이동정반의 경우는 블럭의 조립과정을 몇 개의 단계로 나누어 각 단계가 완료되면 다음 단계의 작업을 위해 블럭이 이동하는 형태로 작업이 이루어진다. 이동정반을 사용하면 작업의 진척상황 통제가 용이하고 일정계획 수립도 각 일자별로 투입되는 블럭의 집합만 결정하면 된다. 그러나, 모든 블럭을 이동정반에서 작업할 수 있는 것은 아니며, 특히 블럭의 형태가 불규칙한 곡블럭의 경우는 대부분 고정정반을 이용하여 조립작업을 수행하여야 한다. 고정정반에서의 작업은 블럭이 한 위치에 고정되고 작업자들이 각 단계별로 해당위치에서 교대로 작업하게 된다. 따라서, 이동정반의 경우는 투입되는 블럭의 작업시간이 동일하여, 동일한 일자에 투입된 블럭들은 동일한 일자에 완료될 수 있고, 블럭의 위치에 따라 각 블럭의 작업단계의 파악

이 용이하다. 고정정반의 경우는 동일한 시점에 동일한 작업장에 배치된 블럭들도 각 블럭에 따라 작업의 진척도 및 완료일의 차이가 발생한다. 또한, 고정정반의 경우는 계획되는 블럭의 기하학적인 형태를 고려하여 작업장내 실제 배치가능성을 보장할 수 있어야 한다. 이로 인해 고정정반의 일정계획문제는 상대적으로 이동정반에 비해 매우 복잡한 문제가 된다.

본 논문에서는 S 중공업의 조선소를 대상으로 대조립블럭에 대한 고정정반의 일정계획을 수립하는 시스템의 개발에 대해 연구한다. 먼저 고정정반 일정계획 문제의 특성에 대해 고찰하고 이를 해결하기 위해 사용된 접근방법을 소개한다. 고정정반일정계획 문제는 대상 조선소의 특수한 상황에 따라 그 특성이 변화할 수 있는 면이 있다. 따라서, 본 연구에서는 대상조선소의 블럭조립공장의 작업 특성에 따라 문제를 정의하고 해법을 개발하도록 하였다.

2. 문제정의

고정정반 일정계획문제는 각 블럭에 대해 조립될 작업장을 결정하고 투입시점 및 작업장내의 배치위치를 정하는 것이다. 따라서, 보통의 일정계획문제와는 달리 시간 및 공간에 대한 일정계획을 동시에 수립하

* 한국과학기술원 산업공학과

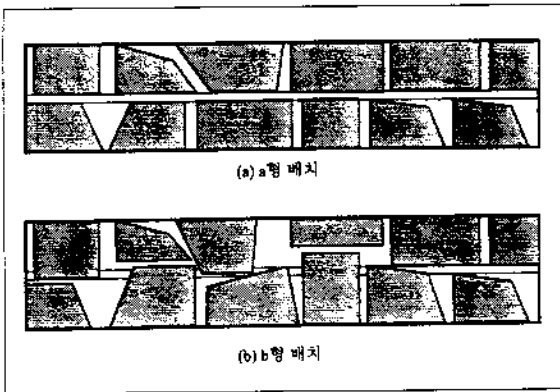
** 동양 SHL

*** 삼성중공업

**** 삼성데이터시스템

여야 하는 특성이 있다. 각각의 블록은 기하학적 형태, 중량 등의 특성치를 가지고 있다. 또한, 각 작업장은 가로, 세로의 길이, 설치되어 있는 크레인의 용량, 치구장비 등의 제약을 가지고 있다. 따라서, 블록의 크기, 중량 및 기타 작업특성에 따라 블록별로 작업이 가능한 작업장의 집합이 제한된다. 본 연구에서는 각 블록별로 작업가능한 작업장의 집합이 사전처리되어 입력되는 것으로 가정한다.

고정정반의 일정계획문제에서 중요한 요소는 블록들이 현재 시점에서 여유공간에 배치가 가능한가를 검토하는 것이다. 배치가가능성의 검토는 블록의 배치형태에 따라 다양한 접근방법을 사용할 수 있다[4]. 본 연구에서는 <그림 1>에 나타난 바와 같은 2열 배치형태의 고정정반만을 대상으로 한다. 대상 고정정반의 구성은 중간에 통로가 있고, 2열 형태로 구성되어 있으며, 각 열에 대해서 1열로 블록을 배치하게 된다. 이와 같은 배치형태는 연구대상인 S 중공업에서 주로 사용하는 형태이다.



<그림 1> 배치형태

연구대상 일정계획문제는 이러한 형태의 작업장이 여러 개 있고, 배치대상 블록들이 주어졌을 때, 다양한 제약조건을 만족하면서 일정계획 및 공간배치계획을 수립하는 것이다.

고정정반 일정계획문제의 입력은 계획대상 블록의 집합, 계획대상 작업장의 집합, 각 블록별 자료, 각 작업장별 자료 및 P,S 블록 리스트, P.E. 블록리스트 등이다. 각 블록별 자료로는 각 블록의 기하학적인 형

태를 블록다각형으로 근사하여 입력한 좌표정보, 선행공정의 작업완료일로부터 구한 가장 빠른 작업개시시점인 투입가능시점, 블록의 탑재일로부터 역으로 계산하여 구한 기준일정, 작업시간 및 작업가능한 작업장집합이 입력된다. 작업장별 자료로는 가용면적과 가용시수(manpower)가 입력된다. P,S 블록리스트는 선수, 선미를 구성하는 대칭적 모양을 가진 블록들의 쌍의 집합이며, P.E. 블록리스트는 선행탑재되는 블록들의 집합이다.

고정정반 일정계획문제의 출력은 각 블록별 작업개시일정 및 작업장과 각 작업장별로 할당된 블록들의 배치계획이 된다.

작업장별 블록의 배치형태는 각 작업장별로 블록들을 1 - 2열로 배치한다. <그림 1>에 나타난 바와 같이 2열 배치는 다시 두 가지 경우로 나눌 수 있다. (a)의 경우와 같이 1열(상부)과 2열(하부)을 독립적으로 배치할 수 있는 경우와 (b)와 같이 그렇지 않은 경우이다. a형의 경우는 대상블록이 통로를 침범하지 않고 배치될 수 있다. 즉, a형의 작업장에서 작업가능한 블록들은 사전처리를 통해 위의 가정을 만족한다고 가정한다.

일정계획 작성시의 제약조건은 크게 작업개시시점에 대한 제약, 작업장면적 제약, 기타배치제약으로 나눌 수 있다. 첫째, 각 블록의 작업개시시점은 투입가능시점과 기준일정 사이에 위치하여야 한다. 또한, 각 블록의 계획시점은 기준일정순으로 작성되어야 한다. 즉, 블록1의 기준일정이 블록2의 기준일정보다 빠르면, 블록1의 작업개시시점은 최소한 블록2의 작업개시시점과 같거나 빨라야 한다. 기준일정순서를 지키는 이유는 조립공정 이후의 후속공정에서의 투입순서를 준수하여 작업을 원활하게 하기 위함이다. 이를 기준일정에 의한 우선순위계약이라 한다. 둘째, 각 작업장별로 한 시점에 배치되어 있는 블록들은 배치가가능해야 한다. 단, 초기에 위치가 정해지면 작업 중 위치의 변경은 불가능하다. 마지막으로 기타배치제약에는 P, S 제약과 P.E. 제약이 있다. P,S 제약은 P,S 블록들의 작업형태 및 부재공급 등의 조건이 동일하므로 인접하여 배치하여야 한다는 것이다. P.E. 제약은 P.E. 블록들은 탑재 전 선행탑재공정을 거쳐야 하므로, 작업

개시시점이 동일하도록 (따라서, 작업종료시점이 일치하도록) 계획하는 것이 바람직하다는 것이다.

위에 열거된 모든 제약조건을 만족하는 일정계획의 존재여부를 판단하는 것은 매우 어려운 문제이다. 따라서, 실제 계획작성시에는 위의 제약조건을 완화하여 해를 작성하고, 제약조건을 위배하는 정도를 최소화 하는 것을 첫번째 목적함수로 사용한다. 각 제약조건 별 중요도는 의사결정자의 판단에 의해 결정될 수 있으며, 실제 적용시에는 다양한 형태의 제약조건 순위를 반영할 수 있도록 하였다. 첫번째 목적함수에 대해서 동일한 품질을 갖는 해에 대해서는 두번째 기준으로 면적활용률 등의 경영상의 목표를 반영할 수 있도록 한다. 그러나, 실제 우수한 일정계획의 선택은 의사결정자의 의견을 최대한 반영하는 것이 바람직하므로 1개의 최적해를 구하는 것보다 다수의 바람직한 해들을 구하여 제시할 수 있도록 시스템을 구성하였다.

3. 일정계획시스템의 구조

개발된 일정계획시스템은 크게 선행계획시스템과 배치계획시스템의 두 부분으로 나눌 수 있다. 선행계획시스템은 근사적으로 면적제약 및 기타제약을 만족하는 해를 구하는 기능을 담당하고, 배치계획시스템은 실제 작업장내에서의 배치 및 일정계획을 작성하는 기능을 한다.

선행계획시스템의 주된 기능은 현재 계획대상 블럭들에 대해서 작업장을 결정한다. 즉, 각 블럭별로 작업가능 작업장 집합 중에서 하나를 선정한다. 이때, 각 작업장에 대해서 배치된 블럭들의 면적요구량 및 시수요구량 등을 고려하여 각 작업장별로 평균화가 이루어질 수 있도록 한다. 각 블럭별 작업장이 정해지면, 배치계획시스템에서는 각 작업장별로 블럭들의 기하학적인 모형을 고려하면서, 구체적인 일정계획 및 배치계획을 작성하게 된다. 이러한 접근방법은 배치문제에 의해 발생하는 문제의 복잡도를 감소시킬 수 있다. 또한, 일정계획작성시 전체작업장에 대한 평균화는 주로 선행계획시스템을 통해서 만족되도록 하고, 배치계획시스템에서는 주로 배치가능성에 초점을 맞

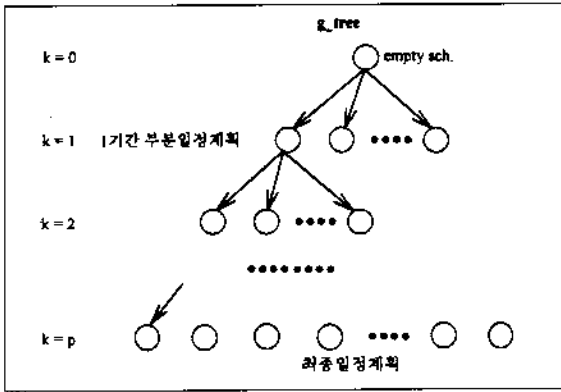
추게 된다.

선행계획시스템과 배치계획시스템은 기준계획기간 별로 반복적으로 수행되며, 이전기간까지의 일정계획 결과를 반영하게 된다. 기준계획기간은 전체 일정계획 기간을 단위기간으로 나눈 것으로 주로 15일 또는 1개월을 사용한다. 일정계획기간을 p 개월이라 하고, 단위기간으로 1개월을 사용하면, 전체일정계획은 p 단계를 거쳐 작성된다 (그림 2). 단계 $k = i$ 에서는 먼저 현재시점(i 달초)부터 p 번째 월말까지의 기간에 대해 계획대상 블럭들의 면적 요구량 및 시수 요구량을 각 작업장별로 평균화할 수 있는 초기일정계획을 선행계획시스템을 통해 작성한다. 이의 결과로 각 작업장별로 수행하여야 할 블럭들의 집합이 결정된다. 다음으로 배치계획시스템은 각 작업장별로 블럭들을 배치하면서 일정계획을 수립하게 된다. 이때 계획기간은 i 번째 달이 되며 부분적인 열거법(partial enumeration)을 사용하여 기준일정을 최대한 만족하는 일정계획을 탐색하게 된다. 이의 결과로 얻어지는 일정계획 중 최선의 계획 m 개를 선택한다. 단계 $k = i+1$ 에서는 단계 $k = i$ 에서 선정된 m 개의 부분일정계획 중 하나를 입력으로 하여 $i+1$ 번째 달의 일정계획을 수립한다. 먼저 선행계획시스템은 해당 부분일정계획에서 계획되지 않은 블럭들을 대상으로 현재시점부터 계획기간말까지의 $(p-i)$ 개월동안의 일정계획을 작성한다. 이를 입력으로 배치계획시스템이 $i+1$ 번째 달의 일정계획 및 배치계획을 구한다. 이러한 절차가 각 기간에 대해 반복되어 최종일정계획이 얻어진다.

이상의 일정계획 작성방법을 개략적으로 도시하면, 다음의 <그림 2>와 같다. 먼저 전체일정계획은 단위기간 동안의 부분일정계획을 통한 부분열거법을 사용하여 작성되며, 탐색방향은 depth first search를 우선으로 하고, 다른 기준도 가능하도록 하였다. <그림 2>의 g_tree 의 root에서 leaf($k = p$)까지의 하나의 path가 완성된 하나의 일정계획이 된다.

다음 <그림 3>은 일정계획알고리즘의 개략적인 구조를 나타낸다.

먼저 전체일정계획을 완성하기 위해 탐색하는 search tree를 g_tree (그림2)라 하자. g_tree 는 초기에



〈그림 2〉 일정계획작성 방법

```

MAIN_SCHEDULER
Init_g_tree:
Do
    Choose_one_node_in_g_tree:      ← cur_node
    Prescheduler (cur_node):
        각 bay, 배치 type, schedule 대상 block 집합 B에 대해
        Bay_scheduler (bay, type, B):
    Evaluate_par_sch:
    Make_new_node_in_g_tree:
While (stopping criteria)
Evaluate_final_sch:
    
```

〈그림 3〉 일정계획알고리즘의 구조

empty schedule로 초기화된다 (Init_g_tree routine). g_tree의 각 node들은 해당 단위기간동안의 부분일정계획을 나타내며, 자신의 parent들과 결합하여 해당시점까지의 부분일정계획이 된다.

g_tree의 탐색절차는 위에서 설명한 바와 같이 depth first search를 기준으로 하되, 동일한 level에서의 부분일정계획의 선택은 탐색 parameter에 의해 결정된다. 이때, 탐색 parameter로는 현재까지의 부분일정계획에서 기준일정을 어긴 블럭들의 수, 기준일정을 어긴 블럭들 중 최대 tardiness, 작업장별 면적활용률 등이 사용되며, 선택사항으로 처리할 수 있게 되어 있다. Stopping criteria로는 현재까지 작성된 최종 schedule의 수, 탐색시간, 현재까지의 best schedule의 품질조건 또는 그들의 조합을 사용할 수 있게 되어 있다. 위의 Do loop는 g_tree내에서 하나의 node (부분일정계획)를

선택하여, 해당 node의 child node들을 생성하는 기능을 한다. 먼저 선행계획시스템(Prescheduler routine)은 현재시점(즉, 현재 선택된 부분일정계획의 최종시점)까지 계획되지 않은 블럭들의 집합에 대해 전체 작업장에 대한 자원사용을 최적화할 수 있는 각 작업장별 블럭의 할당을 구한다. 이의 출력으로 각 작업장별로 작업할 블럭들의 집합이 정해진다. 배치계획시스템(Bay_schedule routine)은 각 bay 및 배치형태, schedule대상 block들에 대해 단위기간동안의 후보 부분일정계획들을 작성하고 작성된 부분일정계획들을 평가하여, 정해진 갯수의 우수한 부분일정계획들을 구하는 기능을 한다. 이때, 각 작업장별로 부분적인 열거법을 사용하여 여러개의 대안 schedule을 작성하게 된다. Evaluate_par_sch routine은 Bay_scheduler에서 작성된 대안 schedule들의 전체작업장에 대한 조합을 구하고, 이를 평가하여 해당 기간에 대한 부분일정계획들을 완성하게 된다. 이들이 g_tree에서 해당시점에 대한 대안 schedule들로 현재 선택된 부분schedule cur_node의 child node들이 된다. Evaluate_final_sch routine은 현재까지 탐색된 최종 schedule들을 평가하여 관련 평가 자료들을 계산하고 출력하는 기능을 한다.

4. 선행계획시스템

선행계획시스템의 주된 목적은 현재시점부터 전체 계획기간을 대상으로 현재까지 계획되지 않은 물량들을 각 작업장에 대해서 최대한 평균화할 수 있는 각 작업장별 작업블럭의 집합을 구하는 것이다. 선행계획시스템의 결과를 이용하여 배치계획시스템에서는 일정계획문제를 각 작업장별로 분할하여 해결할 수 있게 된다.

선행계획시스템의 입력은 현재까지 계획되지 않은 블럭들의 집합, 각 블럭별, 각 작업장별 자료가 된다. 배치가능성은 각 블럭별 근사면적을 사용하여 근사적으로 고려하게 된다. 이때, 실제 배치가능성을 최대한 반영하기 위해 각 작업장별 가용면적을 수정하여 사용하게 된다. 즉, 직사각형으로 근사하는 과정에서 발생하는 여유분을 반영하기 위해 작업장의 가용면적을 실제보다 증가시켜 사용할 수 있게 되어 있다.

선행계획시스템의 구성은 초기해 구성 및 근접해 탐색을 통한 해의 개선으로 이루어져 있다. 목적함수는 각 작업장에 대해서 면적사용율을 평준화하는 것이다. 이때, 가능해(feasible solution)를 작업개시시점 제약, 우선순위제약 및 기타제약을 모두 만족시키는 것으로 정의하고 면적한계는 넘을 수 있는 것으로 본다.

선행계획시스템의 구성절차를 정형화하면 다음과 같다. 먼저 초기해는 이진탐색을 이용하여 각 블록의 가장 빠른 작업개시시점을 결정함으로써 작성된다. 이때, 작업장은 각 블록의 작업장집합 중 최초의 것(즉, 최선호 작업장)을 사용한다. 현재 임의의 가능해에 대해서 다음의 값을 정의하자.

- t 시점에 작업장 W_j 에 할당된 블록의 집합 $B(t, j)$

- t 시점에 작업장 W_j 의 면적사용량 :

$$A(t, j) = \sum_{i \in B(t, j)} (\text{블록 } i \text{의 면적요구량})$$

- t 시점에 작업장 W_j 의 수정된 면적사용량 :

$$\bar{A}(t, j) = A(t, j) / (\text{작업장 } j \text{의 가용면적})$$

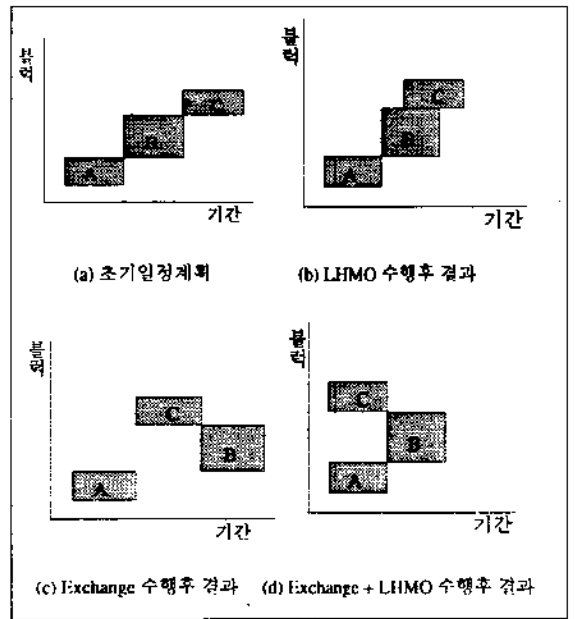
- 작업장 W_j 의 최대면적사용량 :

$$PEAK(j) = \max_t A(t, j)$$

- 최대면적사용량 : $PEAK = \max_{j \in W} PEAK(j)$

현재의 해를 개선시키려면, PEAK의 값을 감소시켜야 한다. 이를 위해, 현재해에서 PEAK가 발생한 작업장 및 해당 시점에 계획된 블록들의 계획시점을 변경하거나 작업장을 변경하여 해의 개선을 도모한다. 이때, 기준일정에 의한 우선순위제약에 의해 계획시점을 변경할 수 있는 범위에 제약이 따르게 된다. 또한, 작업장의 변경도 작업장집합에 명시된 작업장들로 제한된다. 해의 개선방법은 계획시간을 조정하는 두 가지 연산(LHMO, RHMO)과 블록의 위치교환(INTERCHANGE), 작업장 변경 들로 이루어져 있다 (그림 4).

해의 탐색절차는 크게 작업장내 평준화 및 작업장별 평준화로 구성되어 있다. 작업장내 평준화는 현재 해당 작업장에 할당된 블록들의 일정을 조정함(LHMO, RHMO, EXCHANGE)으로써 자원사용율의 평준화를 추구한다. 한편, 작업장별 평준화는 PEAK가 발생한 작업장에서 일부 블록의 할당을 변경함으로써 자원의



(그림 4) 해의 개선절차

평준화를 추구하게 된다. 이상의 두 가지 평준화절차를 반복적으로 수행함으로써 해의 개선을 도모한다.

5. 배치계획시스템

선행계획시스템에서 주어진 각 작업장별 블록할당을 입력으로 배치계획시스템에서는 실제 작업장내 배치 및 작업일정을 정하게 된다. 배치계획시스템은 일정계획문제를 각 작업장별로 분할하여 해결한다. 즉, 주어진 시간 및 작업장에 대해 반복적으로 부분일정계획을 작성함으로써 주어진 시간에 대한 전체 작업장에 대한 일정계획을 완성하게 된다.

작업장별 부분일정계획은 일정계획시점별로 몇개의 가능한 대안을 탐색하는 부분적인 열거법을 사용하여 작성된다. 여기서 일정계획시점은 작업장내에 계획가능한 여유공간이 발생한 시점이 되며, 구체적으로 현재 배치된 블록 중에서 작업을 마치고 작업장을 나가는 블록이 발생하는 시점이 된다. 각 계획시점에서는 대안으로 몇 개의 블록의 집합을 선택하게 된다.

대안의 작성은 현재 시점에서 계획되지 않은 블록들에 대해서 우선순위를 고려함으로써 이루어진다. 우

선순위는 기준일정을 주된 값으로 하고 면적요구량, 선행계획시스템의 일정계획등을 고려하여 동물의 경우 순서를 정하게 된다. 이때, 우선순위로 몇 개의 다른 기준을 사용함으로써 대안이 생성되는 것이다.

이제 배치계획시스템(Bay_scheduler routine)의 구성에 대해 좀더 자세히 설명하도록 한다. Bay_scheduler routine은 위에 설명한 바와 같이 해당 작업장 및 대상 블럭들에 대해 단위기간 동안의 부분일정계획들을 작성하는 기능을 한다. 부분일정계획의 작성은 역시 tree search를 통하여 이루어지는 데, 이때 작성되는 tree를 s_tree라 하자. s_tree는 초기에 하나의 node로 초기화되는데 해당 node는 계획시점 초기에 대상 작업장에 배치된 블럭들의 정보를 포함하게 된다. 이러한 정보는 g_tree내에서 현재 선택된 부분일정계획의 정보로부터 얻을 수 있다. s_tree의 각 node는 배치가능시점, 즉, 현재 배치된 블럭의 출동으로 인하여 여유공간이 발생한 시점에 새로 선택된 블럭들의 정보 및 작업장 내 배치정보를 포함하게 된다. 따라서, 현재 node 및 parent node들을 결합하면, 현재시점까지의 일정계획 및 배치계획을 알 수 있다. s_tree내에서 현재까지 탐색되지 않은 node들에 대한 정보는 s_queue에 수록된다. 최초로 s_queue는 초기 node로 초기화된다.

Bay_scheduler routine의 구성을 요약하면 다음 <그림 5>와 같다.

Bay_scheduler

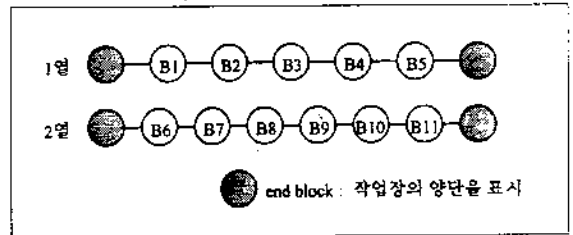
```
Init_s_tree, Init_s_queue;
while (s_queue != empty)
  Pop_a_node: ---> cur_node
  Make_hole_list(cur_node): ---> Hole_list
  Make_blk_insertion(Hole_list);
  Update_s_tree, Update_s_queue;
Evaluate_s_sch;
```

<그림 5> 배치계획시스템의 구조

<그림 5>에서, Pop_a_node routine은 s_queue에서 하나의 node를 선택하는 기능을 한다. Make_hole_list routine은 선택된 node에서 블럭의 최초 완료시점을 계산하고, 해당 시점이 현재의 계획기간말보다 빠르면, 해당시점에 발생하는 작업장내 여유공간(hole)의 리

스트를 작성하게 된다. 이때, 해당시점이 기간말을 넘으면, loop의 끝으로 간다. Make_blk_insertion routine은 작성된 Hole_list에 대해 현재까지 계획되지 않은 블럭들의 배치대안들을 생성하는 기능을 한다. 이때, 몇 개의 우선순위를 고려함으로써 여러 개의 배치대안들이 작성된다. 이들이 s_tree에서 현재 node의 child node들이 된다. Evaluate_s_sch routine은 작성된 최종 schedule들을 평가하여 우수한 부분 schedule들을 선택하는 기능을 한다.

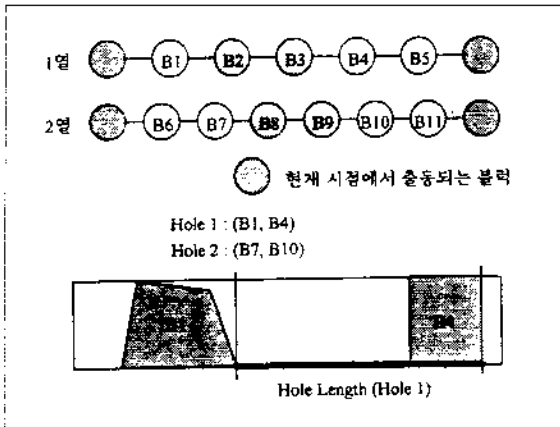
다음은 배치대안을 생성하는 부분에 대한 설명을 하도록 하겠다. 먼저 현재 작업장내의 블럭들의 배치정보를 요약하기 위한 자료구조들을 소개한다. 현재 작업장내 배치된 블럭들의 상대적 위치를 표현하기 위해 배치그래프라는 구조를 사용한다. 배치그래프는 좌에서 우로 현재 배치된 블럭들의 블럭번호를 수록하며, <그림 6>과 같은 형태를 가진다. 배치된 블럭들의 중요한 정보는 배치정보 list에 수록되어 있다. 배치정보 list에는 블럭들의 일정정보 및 배치정보가 수록된다. s_tree내의 각 node에서는 배치그래프와 배치정보 list가 작성되고 갱신된다.



<그림 6> 배치그래프

다음은 배치공간에 관련된 자료들을 소개하기로 한다. 배치가능공간은 현재 배치된 블럭의 출동으로 발생하며, Hole이라 정의한다. Hole은 해당 배치공간의 좌,우 블럭들로 정의되며, 이를 배치그래프에서 살펴보면 <그림 7>과 같다.

각각의 Hole에는 발생시점, 길이 등의 정보가 수록되며, 동일한 시점에 발생한 Hole들의 리스트를 Hole_List라 한다. Hole의 길이는 x축상에서 해당 Hole의 좌편 블럭의 최우측 종점에서 우편 블럭의 최우측 종점까지의 거리를 말한다 (그림 7). Hole의 길이는 블



(그림 7) Hole의 생성

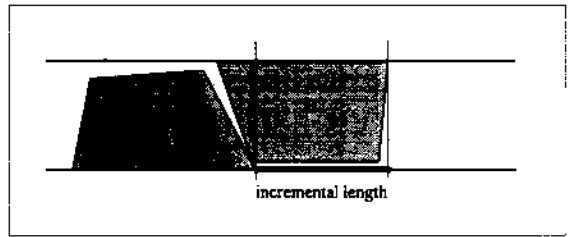
력들의 배치가능성을 조사하는 데 사용된다.

이제 Hole에 블록들의 할당을 구하는 Cand_blk_insertion routine에 대해 설명한다. Cand_blk_insertion routine은 작업장의 배치형태에 따라 다소의 차이가 발생한다. 여기서는 a형의 작업장에 대해서만 설명하기로 한다.

Cand_blk_insertion의 입력은 이전단계에서 구한 Hole_list이다. Hole_list에는 현재 가능한 Hole들의 모든 정보가 기록되어 있다. Cand_blk_insertion routine에서는 현재까지 계획되지 않은 블록들을 몇가지 기준에 의해 우선순위를 구하여, 각각의 우선순위별로 블록을 Hole들에 할당하게 된다. 이러한 할당은 더이상 블록의 배치가 불가능한 시점까지 계속된다. 블록 B의 Hole H에 대한 배치가능성은 incremental length를 계산함으로써 이루어진다. Incremental length는 두 개의 블록 B1, B2에 대해 블록 B1을 블록 B2의 우편에 최대한 인접시켰을 때 x축상에서 증가되는 거리로 <그림 8>과 같이 계산된다.

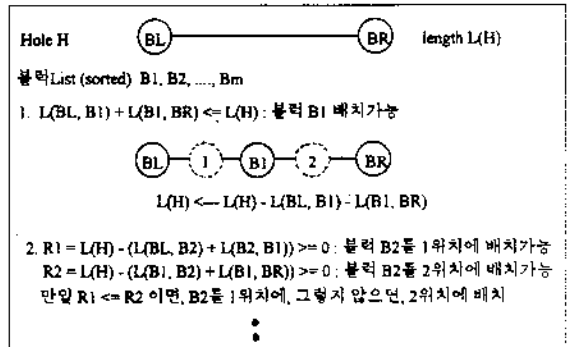
이제 블록 B1과 B2사이의 incremental length를 $L(B_1, B_2)$ 로 표시하기로 하자. 이값은 두 블록의 좌표값을 이용하여 손쉽게 계산될 수 있다. 현재의 Hole H의 좌편 블록은 BL, 우편 블록을 BR이라 할 때, 집합 $\{B_1, \dots, B_n\}$ 에 포함된 모든 블록들이 해당 순서대로 배치가능할 조건은

$$L(BL, B_1) + L(B_1, B_2) + \dots + L(B_m, BR) \leq \text{HOLE LENGTH}(H)$$



(그림 8) Incremental length의 계산

이면 된다. 따라서, 배치되는 블록의 집합은 위의 조건이 만족되는 범위내에서 우선순위에 따라 블록의 집합을 하나씩 증가시키면서 결정되게 된다. 이때, 블록의 배치방향, 배치위치, 배치되는 Hole 등의 대안이 존재하는데, 이 경우에는 각각의 경우에 대해서 incremental length를 조사하여 최소가 되는 대안을 선택한다. 이상의 과정을 그림으로 요약하면, <그림 9>와 같다.



(그림 9) 배치대안의 생성

위의 배치대안생성 과정에서 배치가능성을 조사하면서 다른 제약조건의 만족여부도 조사하게 된다. 예를 들어 P,S 블록 제약은 해당블록들의 조합을 구하여 결과되는 통합블록의 배치가능성을 조사하여, 배치가 가능하면 모두 배치하고 그렇지 않으면 모두 배치하지 않음으로써 만족되게 된다. 가용시수 제약조건도 쉽게 만족되도록 할 수 있다.

6. 결론

본 연구에서는 조선공정의 일부인 고정정반에서의

일정계획문제를 정의하고 일정계획알고리즘에 대해 논하였다. 고정정반의 일정계획문제는 다른 분야의 일정계획문제와 달리 시간 및 공간에 대한 계획을 동시에 작성하여야 하는 특징이 있으며, 본 연구에서는 이를 해결하기 위한 접근방법을 소개하였다.

개발된 일정계획시스템은 이동정반의 일정계획시스템과 결합하여 통합된 조립공정의 일정계획시스템을 구성하며, 이는 다시 선포, 옥외, 의장, 탑재 시스템과 결합하여 통합된 조선생산계획 시스템을 구성하게 된다.

[참고문헌]

- [1] 이재동, 홍유신, "블럭조립공장의 부하평준화를 위한 생산일정계획", 산업공학, 7권 2호, pp. 75-85, 1994.
- [2] 최형림, Erection Scheduling at Shipbuilding Using Constraint Directed Graph Search, 박사학위논문, 한국과학기술원, 1993.
- [3] 서민수, 일정계획 및 통제를 위한 계층적 전문가 시스템: KAIS-3, 박사학위논문, 한국과학기술원, 1991.
- [4] Preparata, F.P. and Shamos, M. Computational Geometry: an Introduction, Springer-Verlag, New York, 1985.



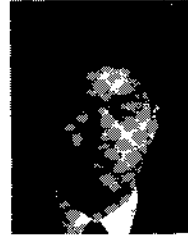
박경철
서울대학교 산업공학과 졸업
한국과학기술원 산업공학과 박사
과정
관심분야: Integer Programming & Combinatorial Optimization



이경식
서울대학교 산업공학과 졸업
한국과학기술원 산업공학과 박사
과정
관심분야: Integer Programming & Combinatorial Optimization



박성수
Cornell University. OR 박사
한국과학기술원 산업공학과 교수
관심분야: OR



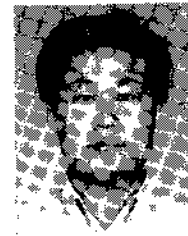
김성환
한국과학기술원 산업공학과 졸업
(석사)
동양 S.H.L.(주)S.I. 사업본부
관심분야: VRP & Network Topology Design



황규욱
삼성중공업 생산기술 연구팀 수석
연구원
조선 자재관리 일관시스템(MACOS)
개발
현재 통합 생산관리 시스템(TOP-COS) 개발책임



김완희
Indiana Univ. 경영학(생산관리전공)박사
SDS CIM 사업부 기술팀장(수석)
관심분야: Technology Management, Productivity Management, Manufacturing Policy



유달호
연세대학교 물리학과 졸업
SDS CIM 개발팀 선임연구원
관심분야: 생산 스케줄링 전문가 시스템



이동렬

Univ. of Michigan 산업공학 박사
SDS CIM 개발팀 선임연구원
관심분야: OR



오명준

고려대학교 수학과 졸업
SDS CIM 개발팀 연구원
삼성석유화학 CIM 구축, 삼성조선
공정계획시스템 구축