공학석사 학위논문

조선소 메가블록 작업장의 공간일정계획 시스템 개발

2009년 2월

부경대학교 대학원

시스템경영공학과

장 정 희

공학석사 학위논문

조선소 메가블록 작업장의 공간일정계획 시스템 개발

지도교수 고 시 근

이 논문을 공학석사 학위논문으로 제출함.

2009년 2월

부경대학교 대학원

시스템경영공학과

장 정 희

장정희의 공학석사 학위논문을 인준함

2009년 02월 25일

주심 공학박사 오 수 철 인

위원 공학박사 구 평 회 인

위원 공학박사 고 시 근 인

목 차

표 목차

그림 목차

제 1 장 서 론

1.1 연구의 배경 및 목적

조선 산업은 우리나라의 경제를 견인하는 대표적인 산업으로 생산품의 거의 전량을 수출하고 있으며 기술과 자본이 집약되어 성립하는 선진국주도형 사업이다. 현재 우리나라가 세계의 선박시장을 지배하고 있으며 그중 탱커, 벌크화물선, 다목적화물선, 컨테이너선, LNG선 등은 우리나라가 실질적으로 시장을 지배하게 된 선박들이다(김효철 2006).

최근 금융경색으로 인하여 2009년 교역규모의 성장률은 4.4%로 2008년 5.2%보다 낮아질 것으로 예상되며, 이에 따라 조선에 대한 수요가 부정적으로 영향을 받을 가능성이 높을 것으로 지적된다. 하지만 이미 존재하고 있는 선박들의 노후화가 크며 해양오염 방지를 위한 규제로 인한 단일 선각 선박의 사용수명이 단축되고 있으며 이러한 요인들로 인해 선박수요가 결국은 증가 할 것으로 예상되어진다. 이러한 추세에 발맞추어 조선 산업에서 선박의 생산성 향상을 위한 연구에 대한 관심이 높아지고 있다(김효철 등 2007).

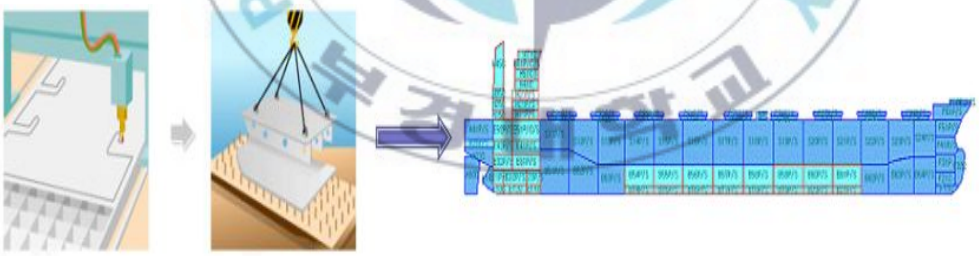
선박 시장의 동향을 살펴보면 유조선의 경우 1970년대 이후 중동전쟁으로 인한 수에즈운하 봉쇄, 수급상황 변화 그리고 환경오염 방지를 위한 각종 규제의 영향을 받아 발전하여 왔다. 특히 1973년 이후 초대형유조선(VLCC) 건조가 본격화되어 1975년에는 유조선 시장은 호황이었으나 1980년대에 급속 냉각되었다. 2000년대에 해난사고로 인한 해양오염을 방지하기 위하여 이중선각구조가 의무

사항이 되어 유조선 시장은 다시 활황을 맞게 되었으며 조선 산업의 발전은 유조선시장의 변동에 영향을 받고 있다. 2010년까지는 단일선각구조 유조선 사용금지로 새로운 수요가 있을 것이다. 컨테이너선의 경우 또한 마찬가지로 최근 대형화와 고속화를 필요로 하고 있다. 한국 조선소들은 대형화에 대비한 시설투자와 선형설계, 구조해석, 기자재 국산화 등에 노력한 결과 상세설계 및 생산설계 기술에서 일본보다 앞서있어 2010년에 이르면 한국은 모든 부문에서 일본 보다 우위에 있을 것이며 중국과는 8년 정도의 격차가 예상되므로 기술개발 노력을 계속해야 할 것이다. 다른 선박들의 경우도 이와 비슷한 이유로 앞으로 증가 하는 수요에 대처할 수 있는 연구가 필요하다(김효철 2006).

조선 산업에서의 선박 건조 과정을 살펴보면 우선 제일 먼저 철판의 절단으로부터 성형, 소조립, 중조립, 대조립 등의 과정을 거치면서 선박의 건조에 있어 기본단위가 되는 블록이 생성된다. [그림 1]에서 실선으로 나누어진 각 셀들이 하나의 블록이며, 이러한 블록들을 선박의 형태로 구성해가는 작업장인 도크(Dock)는 일반적으로 대부분의 조선소에서 병목자원으로 인식되고 있다. 이러한 문제를 해결하기 위한 가장 대표적인 방법이 선행탑재(PE; Pre-erection)이다. 선행탑재란 도크 내에서의 조립시간을 줄이기 위해 도크 옆에서 몇 개(2~3개 혹은 많은 경우 5~6개)의 블록을 미리 조립(Pre-erection)한 다음 골리앗 크레인으로 이 PE블록을 도크 내부로 이동하여 작업 중인 선박에 탑재(Erection)하는 방식이다. 현재 이 방식은 국내 거의 모든 조선소에서 사용되고 있다. 더 나아가 이러한 선행탑재 방식을 더욱 발전시킨 메가블록공법이 사용되고 있는데 이는 기존의 PE블록을 더욱 더 큰 블록으로 미리 조립하여 도크의 생산성을 증대 시키는 방법이다.

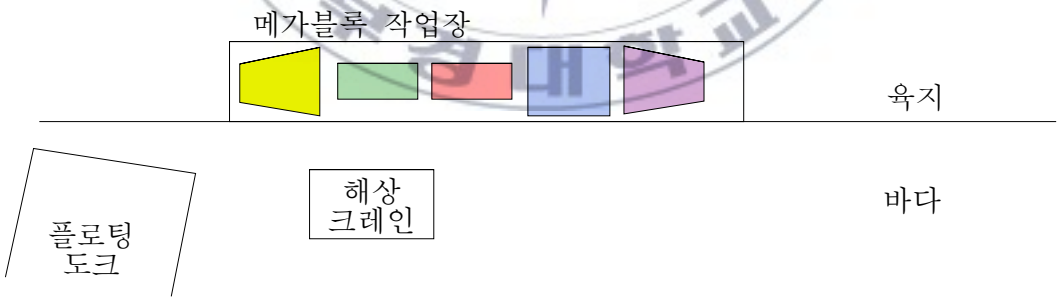
본 연구에서는 선박의 납기일을 준수하면서 경쟁력을 강화할 수 있도록 생산성의 향상이 요구되어 지는데 이러한 문제에 대응하기 위해 기존 시스템에 새로이 도입되고 있는 메가블록 작업장의 메가블록 배치 문제를 다루고자 한다. 본 연구에서 다루고자 하는 메가블록공법의 경우 이미 S중공업에서 실행하고 있는 시스템이다. 실제 메가블록공법을 도입한 결과 한 해 건조 선박수가 30척에서 50척으로 증가하여 매우 효율적이라는 것이 증명되었다. 이에 앞으로는 선박수요의 증가로 기가블록, 테라블록 시스템으로 발전할 것으로 보인다.

블록의 크기는 대략 15m X 15m, 중량은 100톤 내외이다. 선종 및 선형 그리고 공법에 따라 다르지만 [그림 1]에서 보듯이 한 척의 배를 건조하기 위해서는 대략 100여개 내지 150여개 이상의 블록이 결합되어진다. 메가PE시스템을 도입한 결과 10여개의 메가블록만으로 배 한척의 건조가 가능하게 되었고 그 만큼 도크(Dock)에서의 생산성을 높일 수 있게 되었다.



[그림 1] 선박의 건조과정

일반 블록 작업장의 경우 완성된 블록을 골리앗 크레인으로 옮기게 되는데 골리앗 크레인의 처리가능 중량이 보통 800톤 이하이므로 PE블록의 크기에는 제약이 있을 수밖에 없다. 이러한 문제를 해결하기 위해 일부 조선소에서는 도크를 육상에 두지 않고 [그림 2]와 같이 해상에 두는데 이를 플로팅 도크(Floating Dock)라고 하며 보통 선박 1척을 그 안에서 건조한다. 이 플로팅 도크를 사용하는 경우에는 골리앗 크레인이 아닌 [그림 3]과 같은 형태의 해상크레인을 사용하여 블록을 탑재하게 되는데 해상크레인의 경우 처리가능 중량이 3,600톤에 이를 정도이므로 PE블록의 크기제약이 크게 완화되어 메가블록 공법이 가능하게 되었다. 메가블록 작업장의 경우 기존의 다른 작업장과는 다른 특성을 가진다. 육상의 메가블록 작업장에서 블록이 일렬로 배치되지 않고 2열 이상 배치할 경우 해상크레인으로부터 먼 쪽에 있는 메가블록은 사실상 옮길 수 없기 때문에 메가블록 작업장의 블록들은 다른 블록 작업장과는 달리 모두 일렬로 배치된다. 이러한 상황 속에서 블록별 조립완료요구일(납기)을 준수하며 메가블록 작업장의 면적활용률을 최대화하는 것이 조선 생산성을 향상시키기 위해 필수적이다.



[그림 2] 본 시스템의 기본 구조

[그림 3] 해상 크레인

따라서 본 연구의 목적은 주어진 계획기간에 납기(도크 탑재일)를 가진 메가블록들을 작업장 안에 가장 효율적으로 배치하는 공간계획(Spatial Scheduling) 방법을 찾고자 한다.

1.2 기존 연구 분석

본 연구에서 다루고자 하는 문제는 메가블록 조립작업장의 효율적/효과적인 사용에 관한 것이다. 이 문제를 단순화하면 직사각형의 작업장 안에 여러 가지 형상(수직방향으로 투영된 2차원 평면형상)의 블록들을 배치하는 문제가 된다.

이러한 유형의 잘 알려진 문제에는 Nesting 및 Bin Packing 문제가 있다. 직사각형의 2차원 평면 위에 여러 가지 모양의 도형을 가능한 한 조밀하게 배치하고자 하는 Nesting 문제는 1970년대부터 본격적으로 연구되기 시작하였으며 국내외에서 많은 연구결과들

을 찾아볼 수 있다(정선교 & 전건욱 2008, Assaf 2003). 또한 배치대상의 형태를 직사각형으로 한정한 Bin Packing 문제에 대해서도 여러 가지 해법이 제시되어 있다(이상헌 & 이정민 2005, Lodi 등 2002).

그런데 이 두 가지 문제는 배치대상 도형들을 주어진 평면에 한 번 채우는 것으로 문제가 종료된다. 반면 조립 작업장의 경우에는 각 블록의 조립공기에 따라 배치된 블록들이 배치된 순서에 상관없이 작업장에서 빠져 나가므로 사용가능 면적이 매일매일 달라지게 된다. 즉, 작업장의 빈 영역이 각 일자별로 전혀 다른 형태로 나타나므로 조선소의 블록조립일정계획문제는 Nesting 및 Bin Packing 알고리즘을 적용할 수 없는 매우 복잡한 문제이다.

이러한 어려움으로 인해 수작업으로 이루어지던 조선소의 작업장배치문제를 알고리즘을 통해 해결하려는 시도는 Lee 등(1996)이 최초였던 것으로 생각된다. 그들은 조선사의 블록조립작업용 공간일정계획 전문가시스템을 개발하기 위하여 먼저 직사각형 작업장 내에 볼록다각형(convex polygon)의 배치문제를 연구하였는데, 배치가능 공간을 탐색하기 위하여 Lozano-Perez(1983)의 형상공간(configuration space) 기법을 사용하였다. 그러나 이론적으로 매우 의미가 있는 이 연구결과는 D 조선소에 구현되었으나 배치효율이 떨어져 현장에서 사용되지는 못하였다. 비슷한 시기에 Park 등(1996)은 특수한 형태의 조립작업장에 대한 배치알고리즘을 제안하였다. 이후 고시근 등(1999)과 고시근 & 채영명(2000)은 H 조선소를 대상으로 Lee 등(1996)의 연구를 응용해 블록조립일정계획 알고리즘을 개발하였다. 이들은 일자별 조립부하의 균형화 문제와 블록의 배치 문제를 분리하여 단계적으로 일정을 수립하였으나 배치효율성 측면에서는 만족할만한 성능을 얻지 못하였다. 비교적 최

근에 Park 등(2002)은 H 조선소의 도장(Paint)공장을 대상으로 한 배치알고리즘을 개발하였다. 이 연구에서는 고시근 등(1999)의 알고리즘에 회전탐색 전략과 특이배치간격 탐색전략 등 2가지 탐색전략을 추가함으로써 블록배치 효율성을 제고하였다.

조선소의 블록배치문제에 대해 이러한 연구들이 이루어졌음에도 불구하고 아직은 현장에서 사람이 직접 작성하는 계획에 비해 알고리즘의 성능이 떨어지는 것이 사실이다. 그 원인은 시간의 흐름에 따라 발생하는 변동성을 고려하기가 어렵기 때문이다. 즉, 블록의 배치효율성은 블록의 배치시점에 작업장의 빈 영역이 어떤 형태가 되어 있는가에 따라 좌우되는데, 각 블록의 배치 시점에 그 블록의 반출에 따른 빈 영역의 형태까지 고려하는 것이 거의 불가능하다는 것이다. 또한 블록의 배치순서도 배치효율에 영향을 미치는데 모든 가능성을 테스트해보는 것은 현재의 컴퓨터 성능으로는 현실적으로 불가능한 일이다. 조립작업장 배치문제에서 수작업보다 더 높은 배치효율성을 얻는 것은 앞으로도 매우 어려운 난제이다. 그런데 본 연구에서 다루는 메가블록 작업장은 일(day)별로 배치된다는 특징에 의해 일반블록 작업장이 2차원 형상을 모두 고려해야 하는 것과는 다르게 블록의 형상에 관계없이 가로길이만을 고려하는 1차원적인 문제가 되어 해결하기가 매우 쉬워진다.

실제 메가블록 작업장의 블록배치 문제는 대상물의 길이만을 고려한 1차원 배치에 시간 축을 더해 2차원 배치문제가 되어 항만에서의 Berthing Scheduling문제와 유사해 보인다. 실제 Berthing Scheduling문제에서는 정해진 길이의 선석에 각각의 선박들의 일정에 맞게 Scheduling하는 문제로써 일정한 길이의 선석은 메가블록 작업장으로, 일렬로 배치되는 선박들의 형태는 일렬로 배치되는 메가블록들과 유사한 특징을 가진다.

항만에서의 Berthing Scheduling문제는 Kim 등(2002)이 다루었다. 그들은 항만에서의 Berthing time와 Berthing position을 변수로 하여 화물 선적 시 비용을 최소화 하는 Scheduling을 목표로 하고 있다. 이 문제를 해결하기 위해 Sub-gradient Optimization Technique을 제시 하고 있다.

본 연구는 이러한 메가블록배치 일정계획문제를 풀기위해 메가블록 작업장에서의 자동배치시스템을 위한 모델을 제안하고, LINGO 소프트웨어와 유전알고리즘을 사용한다.

1.3 연구 내용

본 연구에서는 조선소의 메가블록 작업장에서 실제로 사용이 가능할 정도로 배치효율을 향상시키기 위한 메가블록 자동 배치 알고리즘을 개발하고자 한다.

본 연구의 구성은 다음과 같다. 서론에 이어 제 2 장에서는 항만에서의 Berthing Scheduling문제를 이용한 메가블록 작업장에서의 블록배치에 관한 모형을 설계하고, 각 부호와 수식에 대한 설명을 하였다. 제 3 장에서는 2장에서 다룬 모형의 문제를 LINGO 소프트웨어를 이용하여 최적해를 찾아보고, 간단한 예를 통해 그 해의 우수성에 대해 알아보았다. 제 4 장에서는 2장에서 다룬 모형의 문제를 LINGO가 아닌 유전알고리즘을 사용하여 해를 찾아보았다. 그리고 이를 이용하여 실제 조선소에서의 Data를 적용하여 해를 찾아보았다. 5장에서는 각 방법들을 비교분석하였고 마지막으로 6장에서 결론을 제시하였다.

제 2 장 메가블록 배치 문제 모형설계

이 장에서는 메가블록 배치 문제를 풀기위하여 이와 유사한 특징을 지닌 문제였던 항만에서의 Berthing Scheduling문제와 본 연구와의 차이점을 알아보고 이를 이용하여 메가블록 배치 문제의 모형을 설계하고자 한다.

2.1 기존연구의 문제점 및 본 연구의 방향

앞 장에서 지적한 바와 같이 본 연구의 대상문제는 길이와 시간을 두 축으로 하는 2차원 공간계획문제가 된다. 즉, 가로가 작업장의 길이이고 세로가 계획대상기간 전체인 큰 직사각형 내부에 가로가 메가블록의 길이이고 세로가 그 블록의 조립공기인 작은 직사각형들을 배치하는 문제가 되는 것이다. 이 문제는 Berth Scheduling 문제와 상당히 유사하나 다음과 같은 차이점을 가지고 있다.

(1) 외주 가능성 : 대부분의 Berth Scheduling 관련 연구에서는 접안일정을 유동적으로 처리하여 선박의 지연출항을 감안하였으나, 메가블록 조립계획에서는 조립완료일을 고정하고 배치가 불가능한 경우에는 외주로 처리하고 있다. 따라서 블록별로 우선순위를 부여하여 이 우선순위가 높은 블록들을 우선적으로 배치한다.
(2) 납기고정 및 공기단축 : 선박의 경우 납기지연이 발생하게 되

면 그에 따른 페널티가 상당히 크기 때문에 도크에서의 탑재일은 반드시 지켜져야 한다. 따라서 메가블록의 조립완료일(=탑재일)은 고정되어 있다. 반면 잔업 등을 통해 조립공기는 단축할 수 있으며, 이것은 조립착수일을 늦추는 것이 가능하다는 의미이다.

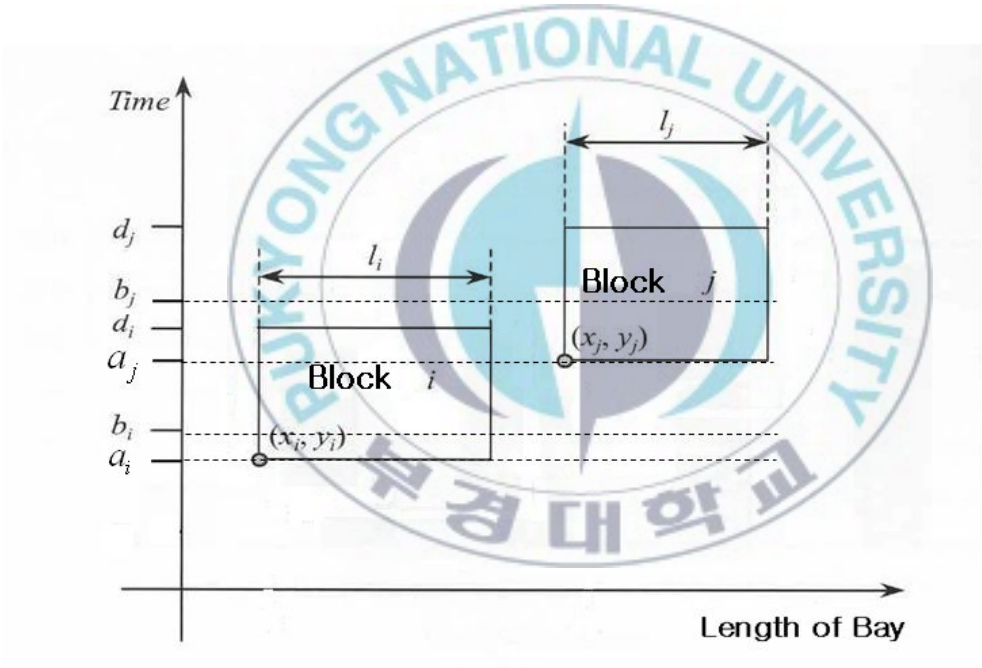
(3) 배치대상블록의 수 : 메가블록의 조립일정계획은 조선소 내에서 중장기에 해당하는 계획으로서 최소 3개월 길게는 1년 이상을 계획대상기간으로 하고 있다. 따라서 하나의 작업장에 대해 계획대상 메가블록의 수는 보통 100개 이상이다. 반면 Berth Scheduling 문제는 비교적 짧은 계획대상기간을 갖고 있으므로 계획대상 선박의 수가 본 연구에 비해 적고, 따라서 본 연구에서는 좀 더 효율적인 알고리즘을 개발해야할 필요가 있다.

(4) 목적함수 : 앞에서 소개하였듯이 Berth Scheduling 문제의 목적함수들은 필요한 안벽길이의 최소화, 지연출항의 최소화 등이었다. 본 연구에서는 주된 목적이 “외주의 최소화”이다. 즉, 계획기간 동안 가능하면 많은 블록들을 작업장 내에 배치하는 것이 제1의 목적이다. 또한 외주의 최소화와 더불어 작업비용을 최소화하기 위한 “공기단축의 최소화”와 계획기간 동안의 “부하균형화”도 필요하다. 이 두 가지 목적은 각 메가블록의 착수일 에 대한 조정을 필요로 한다.

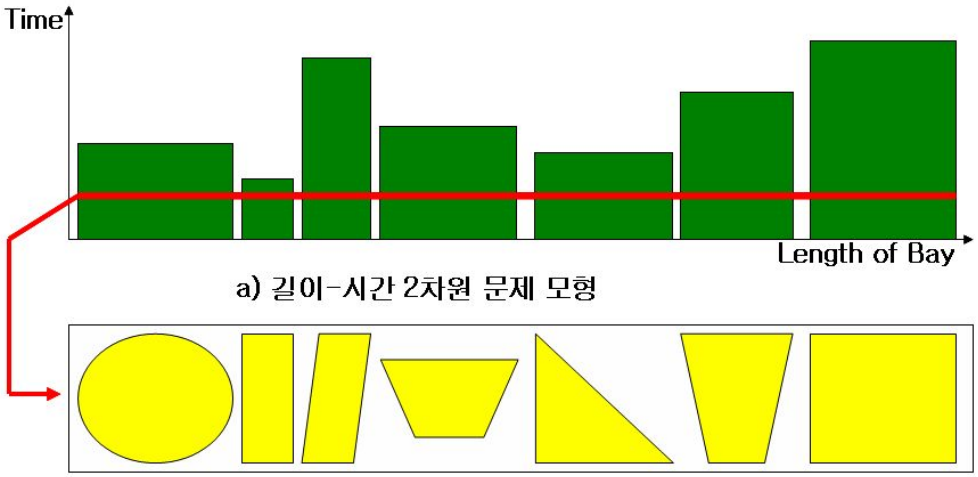
본 연구는 Berth Scheduling 문제를 기반으로 메가블록 작업장의 이러한 특성들을 고려하여 새로운 모형을 설계하였다.

2.2 부호설명

본 연구에서 대상문제의 모형화를 위해 메가블록 작업장과 그 블록들을 [그림 4]와 같이 표현하기로 한다. [그림 4]의 대상 문제의 표현방법에 따라 [그림 5(a)]와 같이 길이-시간 2차원 평면에 표현될 수 있다. 이 그림에서의 직사각형은 가로가 블록의 가로 길이를, 세로가 블록의 조립공기를 표시한다. 그리고 [그림 5(b)]는 [그림 5(a)]의 일정에 대한 실제 작업장의 배치 현황을 보여준다.



[그림 4] 대상 문제의 표현



[그림 5] 메가블록 조립일정계획 모형

[그림 4]에 표시된 기호 및 앞으로 모형에서 사용할 기호들을 종류별로 설명하면 다음과 같으며, 부호는 변수(variables)와 모수(parameters)로 나눌 수 있고, 그 중 모수에는 다음과 같은 것들이 있다.

<모수>

N : 계획대상 메가블록의 수

L : 계획대상 메가블록 작업장의 길이

T : 계획기간

d_i : 블록 i의 완료시간

a_i : 블록 i의 착수가능시간 (Earliest Start Time)

b_i : 블록 i의 최대지연착수가능시간 (Latest Start Time)

p_i : 블록 i의 자체생산 우선순위 점수
(낮은 점수부터 외주)

l_i : 블록 i의 길이

의사결정변수(decision variables)인 독립변수들은 다음과 같다.

<결정변수>

z_i : 블록 i를 자체 생산하면(즉, 계획기간 내에 배치되면) 1, 그렇지 않으면 0

(x_i, y_i) : 블록 i가 배치된 경우 [그림 5(a)]의 좌표계에서 블록 좌하점의 좌표

<모형화에 필요한 부가변수>

z_{ij}^x : 블록 i가 블록 j의 좌측에 배치되면 1, 그렇지 않으면 0

z_{ij}^y : 블록 i가 블록 j의 아래에 배치되면(즉, 블록 i 완료 후 블록 j를 착수하면) 1, 그렇지 않으면 0

2.3 가정

본 연구에서 모형을 개발하기 위해 사용된 가정에는 다음과 같은 것들이 있다.

(1) N 개의 계획대상 메가블록들은 착수가능시간(a_i)이 0 이상이고 지연착수시간(b_i)이 T 이하이다.

(2) 일단 배치된 블록은 반출 시까지 위치를 변경할 수 없다.

(3) 모든 블록들은 폭(세로길이)이 작업장 폭보다 작고, 둘 이상의 블록들을 세로방향으로 병렬 배치할 수 없다. 따라서 배치가능성은 블록과 작업장의 가로 길이만 고려하면 된다.

(4) 블록은 작업장 내부에 배치되어야 한다. 즉, $0 \le x_i \le L - l_i$ 를 만족하여야 한다.

(5) 주어진 기간($a_i \le y_i \le b_i$)에 배치되지 않는 블록들은 외주를 준다. 각 블록은 여러 가지 특성에 의해 정해진 자체제작 우선순위 값(p_i)을 갖는데, 이 값이 적은 블록을 우선적으로 외주를 준다.

(6) 각 블록의 반출시점(d_i)은 고정되어 있으나, 착수시점(y_i)은 착수가능시간(a_i)과 최대 지연착수가능시간(b_i) 사이에서 변동가능하다.

2.4 수리모형

위와 같은 부호와 가정을 사용하여 계획 기간 동안 전체 비용을 최소화 하는 'mixed integer programming(MIP)'로 나타내고자 한다. 그 모형은 다음과 같다.

$$\text{Maximize} \quad \sum_{i=1}^{N}\{p_i z_i - (y_i - a_i)z_i\} \tag{1}$$

subject to

$$0 \le x_i \le L - l_i \quad \text{for } all\ i \tag{2}$$

$$a_i \le y_i \le b_i \quad \text{for } all\ i \tag{3}$$

$$x_i + l_i - M(1-z_i) \le x_j + M(1-z_{ij}^x) + M(1-z_j)$$
$$\text{for } all\ i \text{ and } j,\ i \ne j, \text{ and } a\ large\ positive\ Number\ M \tag{4}$$

$$d_i - M(1-z_i) \le y_j + M(1-z_{ij}^y) + M(1-z_i)$$
$$\text{for } all\ i \text{ and } j,\ i \ne j, \text{ and } a\ large\ positive\ Nmber\ M \tag{5}$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \ge 1 \quad \text{for } all\ i \text{ and } j,\ i \ne j \tag{6}$$

$$z_i,\ z_{ij}^x,\ z_{ij}^y : 0 \text{ or } 1 \text{ integer} \quad \text{for } all\ i \text{ and } j,\ i \ne j \tag{7}$$

수식 (1)은 자체생산의 최대화(특히 자체제작 우선순위가 높은 블록의 외주를 최소화) 및 공기단축의 최소화를 동시에 달성하기 위한 목적함수이다. 두 목적의 중요도에 따라 우선순위 값(p_i)의 크기를 조정하는 것도 해결과제 중 하나이다. 수식 (2)와 (3)은 가정 (4)와 (6)을 표시한 것이다. 수식 (4)는 $z_i = z_j = 1$일때 x_i 값과 x_j

를 이용해 z_{ij}^{x} 값을 강제하는 제약, 다시 말해 i블록과 j블록이 x축 방향으로 서로 겹치지 않도록 하는 블록 길이의 제약이고 수식 (5)는 $z_i=z_j=1$일때 y_i 값과 y_j를 이용해 z_{ij}^{y} 값을 강제하는 제약, 즉 i블록과 j블록이 y축 방향으로 서로 겹치지 않도록 하는 시간적 제약이다. 마지막으로 수식 (6)에 의하면 z_{ij}^{x} , z_{ji}^{x} , z_{ij}^{y} , z_{ji}^{y} 등 4개 값 중 적어도 하나는 반드시 1이어야 한다. 이는 i블록과 j블록이 x축, y축 방향으로 서로 겹치지 않도록 하는 것으로 결국은 각각의 블록들은 작업장에 배치 시 서로 겹치지 않도록 하는 것이다.

위 모형은 Kim &Moon(2003)의 Berth Scheduling 모형을 참고하여 본 연구의 대상문제에 맞게 수정한 것이다. 위에서 제시한 모형의 타당성을 검증하기 위해 상용 최적화 소프트웨어 LINGO를 사용하여 최적해를 찾아보았다.

제 3 장 LINGO를 이용한 방법

3.1 LINGO로 해 찾기

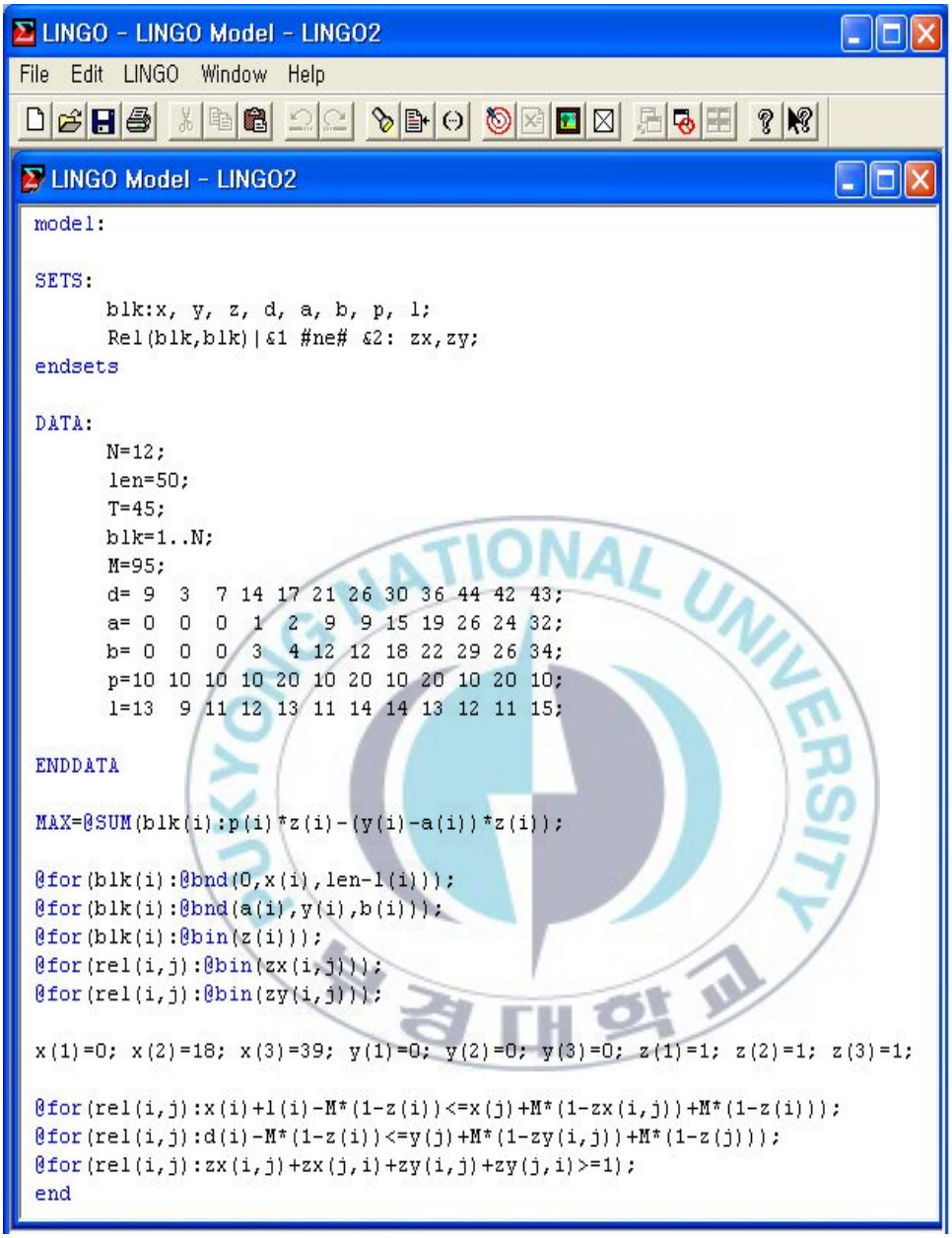
간단한 예제를 통해 위에서 제시한 모형을 LINGO를 이용하여 풀어보았다. 예제는 다음과 같다.

$N = 12$
$L = 50\,(m)$
$T = 45\,(day)$
$d = 9\ 3\ 7\ 14\ 17\ 21\ 26\ 30\ 36\ 44\ 42\ 43$
$a = 0\ 0\ 0\ \ 1\ \ 2\ \ 9\ \ 9\ 15\ 19\ 26\ 24\ 32$
$b = 0\ 0\ 0\ \ 3\ \ 4\ 12\ 12\ 18\ 22\ 29\ 26\ 34$
$p = 10\ 10\ 10\ 10\ 20\ 10\ 20\ 10\ 20\ 10\ 20\ 10$
$l = 13\ 9\ 11\ 12\ 13\ 11\ 14\ 13\ 13\ 12\ 11\ 15$

이 자료를 바탕으로 LINGO로 해를 찾아보면 다음과 같은 해를 얻을 수 있다. 100000번 이상의 반복을 통해 해가 얻어지며 해를 찾는데 걸리는 시간은 1분 30초 정도이다.

$z_i = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1)$

다시 말해 10번째 블록은 배치되지 못하였음을 알 수 있다.
위 예제를 풀기 위한 LINGO 입력창은 [그림 6]과 같다.

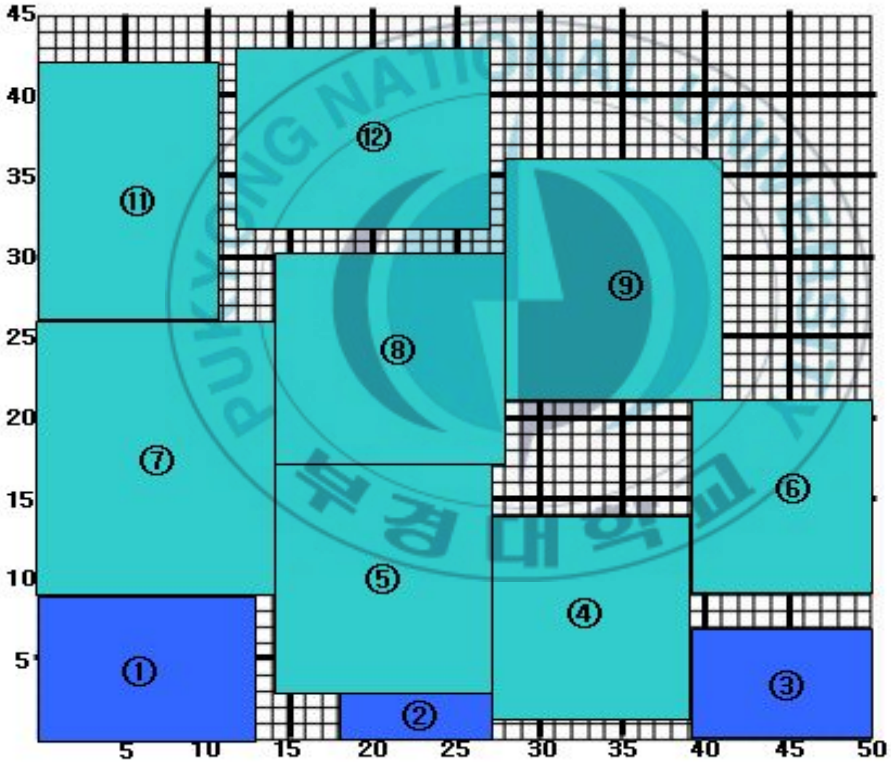


[그림 6] LINGO 입력창

3.2 LINGO 결과 분석

LINGO로 풀어본 예제를 앞서 설명한 [그림 4]와 [그림 5]와 같이 메가블록조립일정계획으로 나타내면 [그림 7]과 같다. 1번, 2번, 3번 블록은 기배치 블록이다.

계획 기간 중 총 12개의 메가블록들 중 일정이 겹치는 블록들의 경우 우선순위를 비교하여 배치가 이루어졌으며 배치되지 못한 10번 블록이 들어갈 자리가 있을 것으로 판단되지 않으므로 LINGO의 결과는 최적의 배치를 이루었음을 알 수 있다.



[그림 7] 예제의 결과

제 4 장 유전알고리즘을 사용한 해법

3장에서 LINGO를 이용하여 최적해를 찾아보았다. LINGO에서 얻은 결과를 보면 반복이 10만 번 이상으로 문제를 푸는데 얼마나 복잡한가를 잘 보여준다. 앞 예제는 계획대상 블록의 수 $N=12$인 문제이지만 이보다 블록의 수가 많아지게 될 경우 반복은 더 많아지게 된다. 실제 블록의 수가 15개만 되어도 LINGO로 풀 수 없다. 만약 LINGO가 없다면 이 문제를 어떻게 풀 것인가? 이번 장에서는 LIINGO를 사용하지 않고 해를 찾기 위해 유전알고리즘을 이용한 방법을 제시한다.

4.1 유전알고리즘(GA: Genetic Algorithm)의 개요

유전 알고리즘은 Holland(1975)에 의해 연구되기 시작하여 많은 조합최적화문제에 응용된 일종의 인공지능기법으로서, 자연의 진화과정을 모방하여 개발한 탐색알고리즘으로 초기에는 대규모 조합최적화 문제를 푸는데 주로 이용되었고, 그 후 점차적으로 다양한 연구 분야에서 폭넓게 적용되어왔으며 최근에는 스케줄링 분야에서 활발하게 응용되고 있다. 이러한 유전알고리즘 기법은 기존의 전통적 최적화 방법인 결정론적 방법에 의해 최적해를 도출하지 못할 경우에 그 유용성이 큰 것으로 알려져 특히 NP-Hard 문제군의 최적화 버전들, 즉 '순회 세일즈맨 문제' 등이 좋은 응용대상이며, 또한 해당 적용분야에 적합한 휴리스틱을 혼합 사용할 수 있어 모

형의 유연성을 제고시킬 수 있는 특징을 지니고 있다.

의사결정과정에 있어서 진화론적 관점이 적용된 유전알고리즘은 의사결정의 만족스러운 대안을 '적자생존의 원리'에 따라 탐색해 나가는 반복적인 절차를 말한다. 염색체(Chromosome)로 표현되는 대안의 해는 적합도(Fitness)를 만족시키는 유전자의 배열(Gene string)로 표현된다. 이러한 유전 알고리즘은 항상 최적해를 보장하는 것이 아니다. 의사결정자가 만족할 수 있는 만족해나 최적해에 근사한 대안을 탐색하는데 그럼에도 이 유전자 알고리즘은 문제해결의 기법이 알려져 있지 않거나 수학적으로 표현이 불가능한 경우에도 적합도 함수(Fitness Function)를 이용하여 만족스런 대안을 찾을 수 있기 때문에 단순하면서도 강력한 최적화 기법으로써 이용되고 있다.

유전 알고리즘은 기존의 최적화 방법과는 다른 다음의 특성을 갖는다.

(1) 병렬적 탐색

branch-and-bound search와 같은 탐색기법 및 기타 해석적 최적화 기법들은 모두 하나의 점에서 출발하여 다음의 점을 찾아가는 point-to-point 탐색방법으로서 local optima에 빠지기 쉬운 결점이 있다. 그러나 유전알고리즘은 모집단을 가지고 출발하여 적정 수준의 다양성을 유지해 가면서 병렬적으로 우수 후보해들 간에 정보를 교환해 가면서 탐색을 하므로 local optima를 피할 가능성이 높은 대단히 robust한 방법이다.

(2) 다양한 선택

단일해가 아닌, 최적점 근처에서 적합도가 우수한 다수의 해가 도출됨으로, 해에 대한 선택의 폭이 넓다.

(3) 목적함수 형태의 무제약성과 문제에 대한 지식 불필요

대개 최적화 방법들은 효율적인 탐색을 위해 대상문제 특유의 지식을 필요로 한다. 발견적 기법이 바로 이러한 경우의 단적인 예이다. 그러나 매번 문제가 주어질 때마다 필요한 지식을 이끌어 낸다는 것이 항상 용이하지만은 않으며 경우에 따라서는 불가능하다. 유전알고리즘은 목적함수의 값만을 필요로 하고, 미분 값이나 일체의 다른 정보를 요구하지 않으므로 매우 광범위한 적용이 가능하다. 직접 탐색 방법을 사용하기 때문에, 적용하고자 하는 문제에 대한 의사결정변수의 모델링이 명확하게 도출되면 복잡한 문제에 대해서도 용이하게 적용할 수 있다.

유전알고리즘을 적용하기 위해서는 대상문제의 특징에 맞는 알고리즘, 관련된 여러 조정변수들, 유전인자의 염색체(chromosome)의 표현방법, 적합도(fitness) 평가함수, 모집단의 초기화 방법, 염색체 선택전략 등을 결정해야 한다. 여기서는 우선 유전알고리즘의 절차를 기술하고 위의 여러 결정항목들을 본 문제의 특성에 부합되도록 수정된 내용을 소개한다.

4.2 표현방법(Representation)과 초기해 생성

유전알고리즘을 적용하기 위해서는 문제의 해가 개체표시자인 염색체로 적절하게 표현되어야 하며 이는 유전알고리즘의 탐색성능과 확장성에 있어서 중대한 영향을 미친다. 최적화할 대상 목적함수의 한 후보해는 하나의 비트열(bit string)로 부호화하여 표현되며, 다수의 비트열들이 모여 모집단(population)을 형성하게 되는데

주어진 모집단이 존재하는 시점을 그 집단의 세대(generation)라 부른다.

유전알고리즘을 사용하기 위해 우선 염색체 표현 방법을 결정해야 한다. 이 논문에서는 계획대상 기간 동안에 배치대상 블록들을 [0,1]사이의 랜덤 실수로 생성하여 배치 대상 블록의 배치 순서를 정하는 방법을 제시하고, 제시된 방법의 결과를 LINGO의 최적해와 비교하여 분석한다.

4.2.1 Procedure DECODE

앞서 언급한 바와 같이 계획대상기간동안의 대상 블록들의 Order 순서를 정하면 된다. 절차는 다음과 같다.

단계1. 각 블록에 대하여 생성된 실수를 오름차순으로 sorting한다. 이러한 결과로 생긴 order는 각 블록의 최대지연 착수 가능일 ($y_j=b_i$)을 이용하는 작업장 배치순서이다. 그 이유는 본 연구의 첫 번째 목적식이 작업장의 블록 배치를 최대화 하는 것이기 때문이다. 블록이 배치된 후에 두 번째 목적식인 공기단축의 최소화를 위해 단계 3이 다루어 질 것이다.

단계2. 단계 1의 order에 따라 작업장에 블록을 배치한다.

첫 번째 블록은 가능한 한 왼쪽에

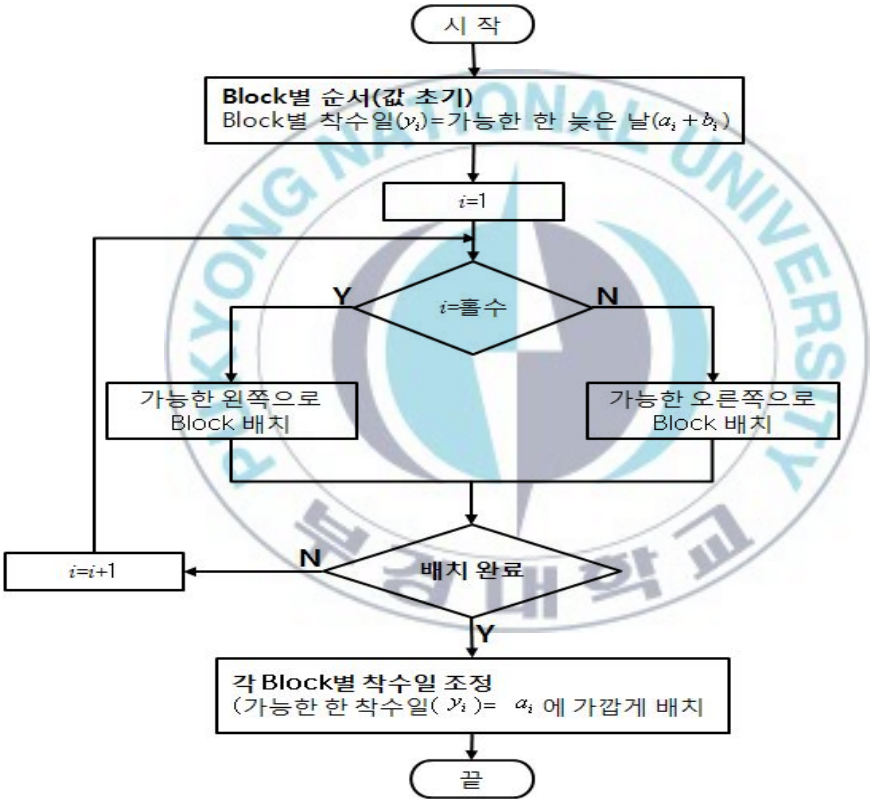
두 번째 블록은 가능한 한 오른쪽에

세 번째 블록은 가능한 한 왼쪽에

만약, i번째 블록을 배치하기 위한 공간이 없다면 $set\ z_i = 0$

단계3. 배치된 블록은 공기 단축을 최소화하기 위하여 블록 배치 착수일(y_j)을 최대한 배치 착수 가능일(a_i)에 가깝게 다시 조정한다.

이러한 Decoding 절차를 flow-chart로 살펴보면 [그림8]과 같다.

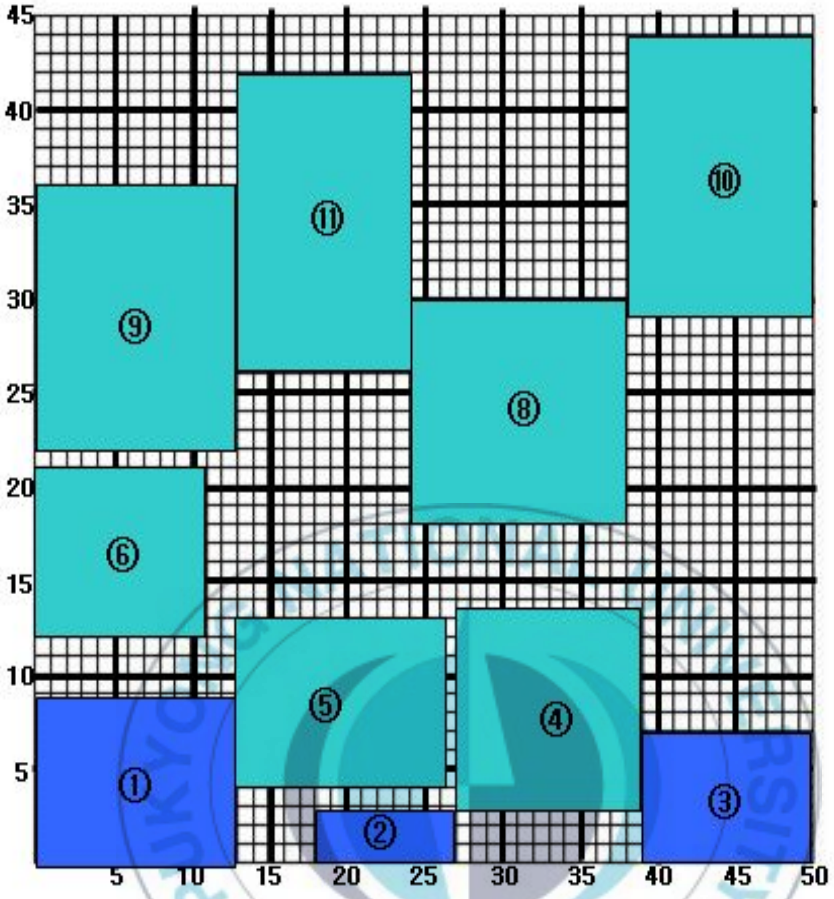


[그림 8] Decoding 절차

실제 3장에서 다루었던 문제를 위의 유전 알고리즘 단계를 통해 최적값을 구해보고 LINGO에서 얻은 최적값과 비교해 보았다. 총 12개의 블록 중 3개는 기배치 블록으로 9개의 랜덤 실수를 생성한다. 즉 첫 번째로 생성된 실수가 바로 4번째 블록을 뜻한다. 예를 들어 생성하면 (0.21, 0.16, 0.36, 0.85, 0.75, 0.46, 0.38, 0.86, 0.97)이 되고 아래와 같이 단계를 거친다.

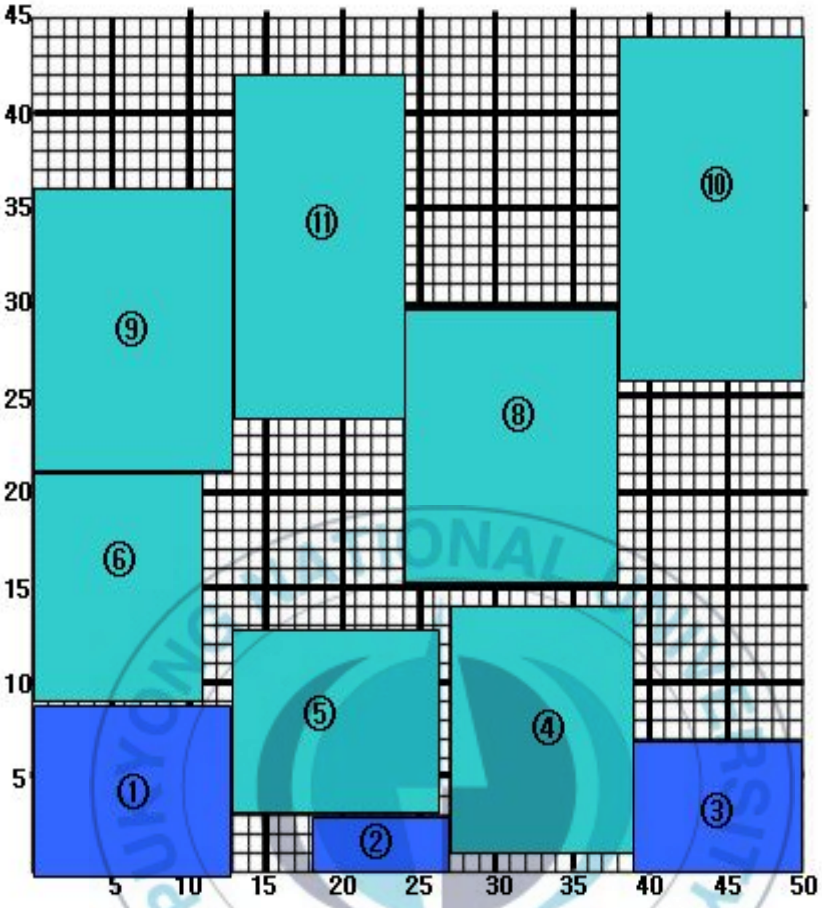
단계1. 각 블록에 대하여 생성된 실수를 오름차순으로 sorting한다. 즉, 각 블록의 배치 순서를 나열하면 다음과 같다.
(5, 4, 6, 10, 9, 8, 7, 11, 12)

단계2. 단계 1의 order에 따라 작업장에 블록을 배치한다. 첫 번째 배치 대상 블록은 5번 블록으로 블록 착수 시간은 4($y_5 = b_5$)이다. 블록 작업 완료 시간(d_5)은 17로써 가능한 한 비어있는 가장 왼쪽 공간에 배치하도록 한다. 하지만 배치 가능공간에 기배치 된 1번 블록에 의해 $x_5 = l_1 = 13$에 배치가 되고 $z_5 = 0$ 에서 $z_5 = 1$로 setting 된다. 두 번째 배치 대상 블록은 4번 블록으로 블록 착수 시간은 3 ($y_4 = b_4$)이고, 블록 작업 완료 시간(d_4)은 14이다. 오른쪽 공간을 탐색해 본 결과 기배치 블록 3번 블록에 의해 $x_4 = x_3 - l_4 = 27$에 배치되고 $z_4 = 0$ 에서 $z_4 = 1$로 업데이트 한다. 다음 배치 대상블록은 6번 블록, 다음은 10번 블록 순으로 order에 따라 블록을 배치한다. 배치결과는 [그림 9]와 같고 이 단계에서 7번 블록과 12번 블록은 미배치 되었으며 $z_7 = z_{12} = 0$ 이다.



[그림 9] 단계 2의 결과

단계3. [그림 9]에서 7개의 블록은 블록 배치 착수일(Y_i)을 조정할 수 있다. 이는 공기단축을 최소화하기 위한 것으로 조정된 배치 결과는 [그림 10]과 같다.



[그림 10] 단계 3의 결과

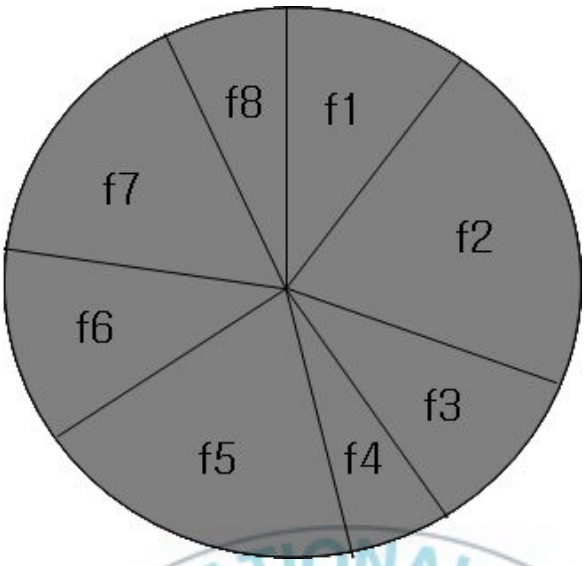
이 모든 단계를 거친 뒤 목적식을 구하여 보면 7번 블록과 12번 블록의 미배치로 인한 값과 5번 블록과 9번 블록의 공기 단축화로 인한 값으로 인해 127로 계산된다(160-20-10-2-1). 이는 LINGO보다 떨어지는 결과임을 알 수 있다.

4.3 목적함수와 적합도함수, 재생산(Reproduction)

이 논문에서 다루고 있는 문제는 블록 배치량의 최대화 문제이다. 목적함수(즉, Z_i)는 2.4절의 수식(1)이고, 적합도함수(즉, F_i)는 목적함수로 계산된다. 적합도 함수는 최대화 문제 형태로 표현해야 하므로 목적함수를 그대로 사용한다.

$$F_i = Z_i \tag{10}$$

수식 (10)에서 얻어진 적합도를 바탕으로 재생산이 이루어진다. 여기서 말하는 재생산이란 매 세대마다 모집단내의 각 염색체에 대하여, 높은 적합도 값(higher fitness value)을 갖는 염색체는 낮은 적합도 값(lower fitness value)을 갖는 염색체보다 선택될 기회를 더 많이 갖는다. 즉, 적합도가 높을수록 많은 수의 자기 복제를 허용함으로써 더 적합한 염색체를 더 많이 선택되는 것을 말한다. 이 논문에 사용된 방법은 사용자에게 폭넓게 인정되고 있는 방법인 룰렛 휠 선택(roulette wheel selection)을 사용하였다. 룰렛휠선택은 적합도와 그것이 차지하고 있는 비율을 이용하는 연산자로써 만약 총 적합도의 합이 100이고 어떤 유전자의 적합도가 10의 적합도 수치를 갖는다면 이것이 선택될 확률은 10/100, 즉 0.1 이 되는 것이다. 다시 말해 현재의 유전자가 차지하고 있는 비율에 따라 선택될 확률이 정해지는 것이다. 아래 [그림 11]에서처럼 계산한 8개의 각 염색체의 적합도를 모두 합한 값만큼의 크기를 가진 룰렛 휠을 가정하고 각 염색체의 적합도만큼 비례해서 룰렛 휠 상의 공간을 배정받은 모양의 예를 보이는 것이다.



[그림 11] 룰렛휠선택에 기초한 재생산 예

4.4 유전연산자(Genetic Operator)

모집단의 세대변천은 유전법칙을 적용하여 새로운 자손개체로 계속 진화함으로써 이를 통하여 최적해를 탐색한다. 유전연산자는 알고리즘의 성능에 결정적인 영향을 미치므로 문제의 특성을 감안하여 우수한 해를 효율적으로 탐색할 수 있도록 정의되어야 한다. 유전법칙은 기본적으로 두 개의 부모 세대로부터 하나의 자손개체로 생성되는 교차변이(Crossover)와 하나의 부모개체의 변화로 인해 자손개체가 생성되는 돌연변이(Mutation)가 있다.

4.4.1 교차변이(Crossover)

교차변이는 부모의 염색체의 일부분을 서로 바꿈으로써 부모의 특징을 결합하여 두 개의 유사한 자손들을 구성한다. 예를 들어 만일 부모가 5차원 벡터 (a_1,b_1,c_1,d_1,e_1)와 (a_2,b_2,c_2,d_2,e_2)에 의해 표현된다면, 두 번째 유전인자 염색체를 교차시키면 두 개의 자손 (a_1,b_1,c_2,d_2,e_2)와 (a_2,b_2,c_1,d_1,e_1)이 생성된다. 이 방법은 교차점을 랜덤하게 선별하여 부모의 염색체를 절반씩 그대로 물려받는 것이다.

본 연구에서 사용한 교차변이는 한 세대에서 각 부모개체마다 [0, 1]의 랜덤 실수를 생성해서 그 수가 교차변이 확률보다 낮으면 그 부모개체를 선택하고 다음 두 번째 부모 개체가 선택되면 교차변이가 일어난다. 교차변이 확률 $P_c=0.4$로 하였다.

4.4.2 돌연변이(Mutation)

교차변이는 기존에 생성된 유전자들을 반복해서 사용함으로 수렴성을 높여 주지만, 탐색 영역 확대에는 한계가 있다. 즉 교차변이를 통해 탐색한 해가 해공간(Solution space)에서 국지적 영역(Local area)으로 빠지는 것을 방지하기 위해 부모해에 없는 새로운 유전자를 도입함으로서 염색체 배열에 변화를 주는 유전 연산자이다. 따라서 임의의 유전자를 강제적으로 변화시킴으로써 탐색 공간에서 색다른 개체를 인위적으로 만들게 되며 탐색 범위 확대의 효과를 얻을 수 있다. 하지만 탐색의 방향이 지나치게 랜덤하게 바

뀌지 않아야 하므로, 돌연변이는 조심스럽게 매우 낮은 확률로 적용된다.

본 연구에서는 각 부모의 염색체마다 [0, 1]의 실수를 생성해서 돌연변이 확률보다 적은 경우 돌연변이를 발생시킨다. 돌연변이 확률 P_m=0.04로 정하였다.

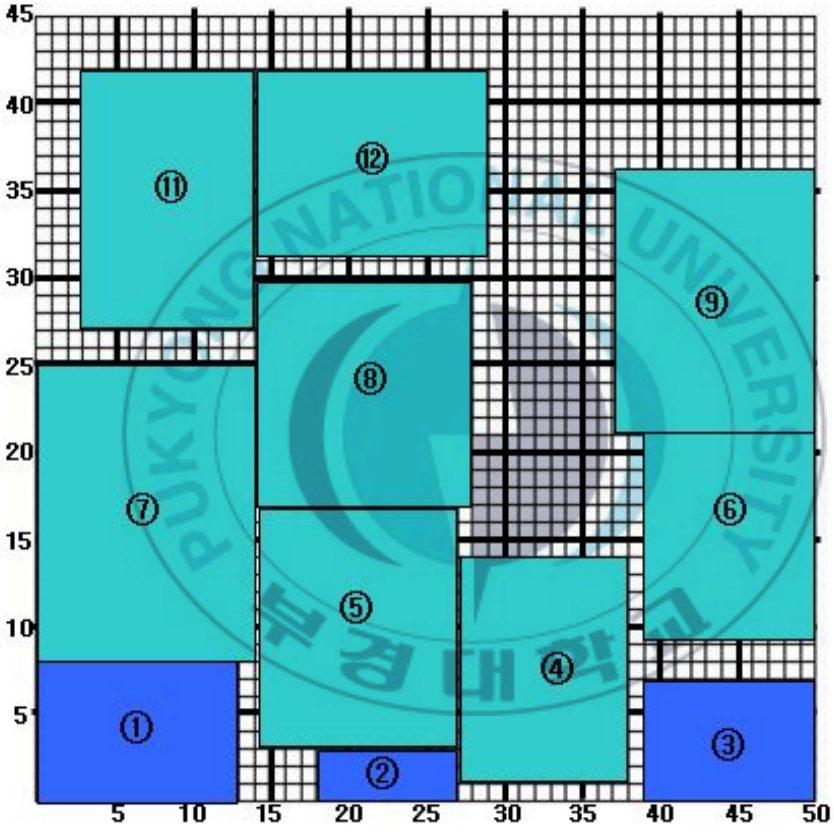
4.5 Stopping-Rule

유전알고리즘을 사용할 때 모집단이 세대변천을 계속함에 따라 그 모집단을 구성하고 있는 염색체들 사이에는 그것들이 취하는 bit string의 형태가 서로 유사해지는 경향이 있다. 이에 본 논문에서는 전체 진행의 최대 세대수를 제한하여 사용하는 방법을 선택하였다. 예를 들면 총 세대수를 1000세대로 하고 그 사이에서 가장 좋은 해를 선택하게 된다. 본 연구에서는 총 세대수를 1000세대로 하고 수렴하는 최적값이 100번 연속해서 나오면 멈추는 방법을 선택하였다.

4.6 결과

지금까지의 이론을 근거로 교차변이와 돌연변이를 통한 유전 알고리즘을 적용하여 최적값에 가장 근사한 해공간(Solution area)을 구할 수 있다. 앞선 [그림 10]의 배치는 만족할 만한 결과가 아니

었다. 이를 유전알고리즘을 적용한 결과 LINGO의 결과와 같은 최적값 143을 도출 할 수 있었으며 그 결과는 [그림 12]과 같다. 결과에서 보듯이 LINGO의 결과와는 조금 다른 배치 모습을 보이고 있지만 최적값이 143으로 같은 값을 보여 주고 있어 유전알고리즘의 결과가 타당함을 증명해준다.



[그림 12] GA의 결과

제 5 장 성능 평가

5.1 실험

이 장에서는 LINGO에 의한 결과와 유전 알고리즘의 결과를 비교해 보았다. 아래의 [표 1]은 LINGO로 얻은 최적값과의 백분율 오차를 분석한 것이다. 각 블록의 개수를 N=8, N=10, N=12의 3가지 유형으로 나누었고 각각 랜덤하게 10개의 문제를 생성하여 각 문제 별로 10번씩 반복하여 실험한 결과이다.

오차는 ((LINGO 최적값-GA 최적값)/LINGO 최적값)X100으로 계산되고 그 값들에 대한 평균과 표준편차, 최소치, 그리고 최대치를 나타내었다.

결과를 보면 배치대상 블록의 수가 증가 하여도 LINGO와 비교해서 아주 좋은 결과를 보여주고 있다. 배치 대상 블록의 수가 8개인 경우에는 평균 오차가 1% 내외이며 편차 또한 적어서 반복을 많이 하지 않아도 좋은 해를 찾을 수 있음을 알 수 있다. 배치 대상 블록의 수가 10개인 경우 또한 평균오차가 1% 내외이며 편차 또한 미미함을 알 수 있다. 마지막으로 배치 대상 블록의 수가 12개인 경우 평균 오차는 2% 내외이며 편차 또한 괜찮은 편이다. 그리고 전체적으로 볼 때 최소값이 0인 LINGO와 같은 최적해를 모든 GA값들이 찾을 수 있음을 보여주고 있다. 이상의 결과를 볼 때 LINGO 없이 문제를 풀어야 할 경우 또는 LINGO로 풀 수 없는 경우에 본 연구의 GA를 이용해도 거의 같은 결과를 얻을 정도로 타당함을 보여주고 있다.

[표 1] LINGO와 GA의 최적값 비교 오차 분석결과

Problem	blk No.	8	10	12
1	Mean	1.00	0.60	0.70
	SD	1.10	0.49	0.46
	Min	0.00	0.00	0.00
	Max	3.00	1.00	1.00
2	Mean	0.40	0.40	1.80
	SD	0.66	0.80	1.08
	Min	0.00	0.00	0.00
	Max	2.00	2.00	4.00
3	Mean	0.20	0.20	0.60
	SD	0.60	0.40	0.92
	Min	0.00	0.00	0.00
	Max	2.00	1.00	2.00
4	Mean	0.20	1.10	0.60
	SD	0.40	1.22	1.20
	Min	0.00	0.00	0.00
	Max	1.00	3.00	3.00
5	Mean	1.20	1.30	2.20
	SD	1.83	1.19	1.62
	Min	0.00	0.00	0.00
	Max	4.00	3.00	5.00
6	Mean	0.20	0.20	0.00
	SD	0.60	0.40	0.00
	Min	0.00	0.00	0.00
	Max	2.00	1.00	0.00
7	Mean	1.60	1.50	0.20
	SD	0.92	1.12	0.60
	Min	0.00	0.00	0.00
	Max	3.00	3.00	2.00
8	Mean	0.00	0.30	1.30
	SD	0.00	0.90	0.78
	Min	0.00	0.00	0.00
	Max	0.00	3.00	2.00
9	Mean	0.40	0.60	1.50
	SD	0.49	0.92	1.02
	Min	0.00	0.00	0.00
	Max	1.00	2.00	3.00
10	Mean	0.70	1.50	1.90
	SD	1.10	0.92	1.04
	Min	0.00	0.00	0.00
	Max	3.00	3.00	3.00

5.2 CPU-Time 분석

[표 2]는 각 배치대상 블록의 수에 따른 결과 도출 시간을 초단위로 나타낸 것이다. 각 방법의 테스트 환경은 AMD Athlon(tm)64 X2 Dual Core Processor 4200+ CPU와 메모리 2기가의 컴퓨터에서 실험하였다. LINGO의 경우 배치대상 블록의 수가 늘어날수록 그 시간이 급격하게 늘어남을 알 수 있다. 배치대상 블록의 수가 12개인 경우 반복이 많아져서 최적해를 찾는데 상당히 많은 시간이 소요된다. 하지만 유전 알고리즘의 경우 배치대상 블록의 수가 늘어남에도 소요시간은 크게 차이가 나지 않음을 알 수 있었다. 실제로 배치대상 블록의 수를 15개로 할 경우 LINGO로 해를 찾는 것은 1시간 내에 해결 할 수가 없었다. 하지만 유전 알고리즘은 짧은 시간에 좋은 해를 찾을 수 있었다.

[표 2] 평균 CPU-Time

blk 수	8	10	12	15
LINGO (초)	0.0	7.0	95.0	-
GA (초)	0.8	0.8	0.9	0.9

제 6 장 결론

우리나라는 세계의 조선시장을 석권하고 있고 전 세계의 절반 이상의 선박 수주량을 자랑하고 있다. 이러한 상황에서 생산성을 증대 시키는 것이 무엇보다 중요한 점 중의 하나이다. 현재 각 조선소에서는 나름의 특수한 방법을 이용하여 생산성을 향상시키고 있다. 특히 S 중공업의 경우 메가블록공법을 이용하여 생산성을 극대화 할 수 있었다. 조선업(shipbuilding industry)의 경쟁력 강화를 위해서는 생산자원의 효율 극대화를 통한 선박의 납기준수가 관건이다. 이러한 납기준수를 위해서는 일정계획이 무엇보다 중요하다. 일반적으로 작업일정계획은 생산계획을 기초로 하여, 구체적으로 어느 작업장에서 언제, 어떤 블록을 착수해야 할 것인가를 결정하는 상세한 시간적 공간 계획을 의미한다. 조선업은 작업자의 수작업에 의존하는 정도가 높아 예외상황 발생의 빈도가 잦기 때문에 이러한 예외상황을 고려하여 초기에 수립된 작업일정계획을 수정할 것이 요구되는데 매번 수작업으로 수정하는 것은 무리가 있다.

본 연구는 조선소에서 새롭게 다루어지는 병목공정인 메가블록 작업장 공간일정계획 알고리즘을 제안하였다. 최적해 도출을 위해 수학적 모델로 모형을 제시하고 LINGO 소프트웨어를 이용하였다. 또한 LINGO 소프트웨어를 사용하지 않고서 문제를 해결하기 위해 유전알고리즘을 개발하였다. 이 방법을 제시함으로서 메가PE작업장의 생산성 최대화를 이룰 수 있는 공간일정계획을 자동화 할 수 있었으며 매우 좋은 성과를 얻을 수 있었다. 앞으로 실제 조선소에서 적용할 수 있기 위해 조선소 내에 존재하는 여러 가지 제약사항을 고려하여 더욱더 효율적이고 현실상황과 더욱 비슷한 문제를 해결

하기 위한 연구가 되어야 할 것이다.

참 고 문 헌

[1] 고시근, 박주철, 최용선, 주철민, 1999, 고정정반 블록조립 작업장의 일정계획시스템 개발, 산업공학, 12, 586-594

[2] 고시근, 채영명, 2000, 플랜트 생산공장의 제품별 작업장사용계획 시스템 개발, 산업공학, 13, 266-272

[3] 김효철, 2006, 우리나라의 조선산업을 일으킨 선박들, 대한조선학회지, Vol.43, No.2, 84-91

[4] 김효철, 이창섭, 장 석, 이재욱, 이승희, 이영길, 유재문, 김상현, 김훈철, 2007, 특집: 조선산업 세계1위 지속을 위한 기술개발 협력방안, 대한조석학회지, Vol.44, No.2, 3-15

[5] 안남수, 강자영, 홍순익, 최화용, 박성수, 2007, 조선산업에서 안벽 배치 계획 문제에 관한 해법, 대한산업공학회 추계학술대회 논문집 pp.1-9

[6] 이상헌, 이정민, 2005, 타부서치를 이용한 2차원 직사각 적재문제에 관한 연구, 경영과학, 22, 167-178

[7] 정선교, 전건욱, 2008, 휴리스틱을 이용한 2차원 임의형상 부재 배치 문제, 산업공학 21, 8-17

[8] 한기호, 2007, 주요이슈: 세계 조선산업 전망, 포스코경영연구소, POSRI 철강 수급전망, Vol.2007, No.1, 161-168

[9] Assaf, Y., 2003, Human strategies based allocation of two-dimensional irregular shapes, *Journal of Intelligent and Fuzzy Systems*, 14, 181-190.

[10] Imai, A., Sun, X., Nishimura, E. and Papadimitriou, S., 2005, Berth allocation in a container port: using a continuous location space approach, *Transportation Research Part B*, 39, 199-221.

[11] Kim, K.H. and Moon, K.C., 2003, Berth scheduling by simulated annealing, *Transportation Research Part B*, 37, 541-560.

[12] Lee, K.J., Lee, J.K. and Choi, S.Y. , 1996, A Spatial Scheduling System and its Application to Shipbuilding: DAS-CURVE, *Expert System with Applications*, 10, 311-324.

[13] Lim, A., 1998, The berth planning problem, *Operations Research Letters*, 22, 105-110.

[14] Lodi, A., Martello, S. and Monaci, M., 2002, Two-dimensional packing problems: A survey, *European Journalof Operational Research*, 141, 241-252.

[15] Lozano-Perez, P., 1983, Spatial Planning : A Configuration Space Approach, *IEEE Transactions on Computers*, 32, 108-120.

[16] Park, C.K., Chung, K.H., Park, J.C., Cho, K.K., Baek, T.H. and Son, E.I., 2002, A spatial scheduling application at the block paint shop in shipbuilding: the HYPOS project, Production Planning & Control, 13, 342-354.

[17] Park, K.C., Lee, K.S., Park, S.S. and S.H. Kim, 1996, Modeling and solving the spatial block scheduling problem in a shipbuilding company, *Computers and Industrial Engineering*, 30, 357-364.

[18] Park, K.T. and Kim, K.H., 2002, Berth scheduling for container terminals by using a sub-gradient optimization technique, *Journal of the Operational Research Society*, 53, 1054-1062.

[19] 대한조선학회 홈페이지, 2008, http://www.snak.or.kr

[20] 월간 해양과조선 홈페이지, 2008, http://www.shipbuilding.or.kr

[21] 한국조선공업협동조합 홈페이지, 2008, http://www.kosic.or.kr

Case study on the Spatial Scheduling for Mega-block Assembly Yard in Shipbuilding Company

Jung Hee Jang

Department of Systems Management & Engineering,
The Graduate School, Pukyong National University

Abstract

Motivated by a software development project in a shipbuilding company, we propose a spatial scheduling algorithm for the mega-block assembly yard in a shipbuilding company. The blocks in a shipbuilding industry are the basic units in shipbuilding processes. They are usually made in the block assembly shop, and then are assembled into a ship in the dry dock. To improve the productivity of the dock, however, the shipbuilders usually assemble several blocks into a big block before the erection in the dock-side area. Our problem is to make an spatial schedule to use these dock-side pre-erection areas efficiently, and then elevate the productivity of the shipbuilding company. First of all, we formulated this situation into a mathematical model and found an optimal solution by a commercial optimization software. But it could not give optimal solutions for the practical sized problems since the problem is very complicated and time consuming. So we proposed a GA-based heuristic algorithm. Using real shipyard data, we showed that the spatial scheduling system based on the algorithm made a very good performance.

Keywords: shipbuilding, mega-block assembly, spatial schedule, pre-erection

감사의 글

대학원에 뜻을 담고 공부한지도 벌써 2년이 훌쩍 지나 버렸습니다. 때론 힘들고 저 자신에 대한 좌절도 느꼈지만 많은 고마우신 분들의 격려와 따뜻한 말 한마디로 지금까지 힘내서 달려오지 않았나 하는 생각이 듭니다. 막상 졸업을 앞두고 지난 일들을 생각해보니 진한 아쉬움만이 가득합니다.

먼저 지금의 저를 있게끔 관심 가져 주시고 이끌어 주신 고시근 교수님께 감사드립니다. 그저 의욕만으로 대학원에 들어온 저를 지금까지 이끌어주신데 대해서 어떤 감사의 인사를 드려야 할지 모르겠습니다. 사회에 나가서도 교수님의 은혜는 항상 가슴속에 간직하고 있겠습니다. 지난 2년동안 느껴온 것이지만 교수님의 그 부지런함은 어디가서도 잊지 않고 배울 수 있도록 노력하겠습니다.

바쁘신 와중에도 미흡한 제 논문을 심사해주시고 모자란 부분을 지적해 주신 오수철 교수님, 구평회 교수님께 고개 숙여 감사드립니다. 항상 교수님께 여쭈어보고 많이 배우라고 충고해주셨던 권혁무 교수님, 대학원 수업을 통해 항상 많은 가르침을 주실려고 노력하셨던 옥영석 교수님, 이운식 교수님, 박병무 교수님, 김민수 교수님, 대학원 수업은 없었지만 학부시절 많은 가르침을 주셨던 김병남 교수님, 마지막으로 좋은 자리를 알선해주신 김영진 교수님께 감사드립니다. 교수님들의 가르침과 따뜻한 격려를 잊지않고 사회에서 열심히 노력하겠습니다.

지금도 아니 앞으로도 묵묵히 지켜보시고 믿어주시는 어머니께 감사의 인사들 드리고 싶습니다. 어려운 가정 형편 속에서도 대학원에 들어가 좀 더 공부를 하고 싶다는 갑작스런 저의 말에도 놀라지 않으시며 만족할 때까지 공부해 보라며 언제나 버팀목이 되어 주신

어머니! 아직 부족한 것이 많은 저이지만 항상 어머니의 은혜만큼은 잊지 않고 있습니다. 당신이 있었기에 제가 있다는 생각으로 앞으로 더욱더 효도하며 살아가겠습니다. 어머니! 사랑합니다. 그리고 항상 말썽만 일으키고 걱정만 시키는 동생 정남이. 이제 새해도 되었고 결혼도 했으니 잘 해 나갈거라 믿는다. 그래도 이 형은 네가 어머니 곁에 있어줘서 너무너무 고맙다. 그리고 재수씨도 어머니께 잘 해 주셔서 너무 고맙습니다. 그리고 학부시절부터 항상 곁에서 지켜봐 주던 테니스 동아리 동기들 동호, 원재, 문석이, 두영이, 희진이, 현정이, 진희, 민영이, 선미, 채연이. 지금은 다들 만나지는 못하지만 언제나 너희들은 내 힘이 되어 주었어. 그리고 걱정스런 맘으로 지켜봐주었던 동아리 형, 누나, 동생들에게도 감사의 말을 전합니다.

짧은 2년 동안이었지만 함께 대학원 생활을 하면서 모르는 것은 누구보다 자상하게 가르쳐 주시고 항상 따뜻한 관심을 주면서 동생처럼 대해주시던 진아누님, 지금은 졸업하고 안계시지만 항상 조금이라도 더 가르쳐 줄려고 노력하고 맛있는 것도 사주셨던 철호선배, 원일선배, 찬호선배, 정수선배, 희영선배(?), 유진선배(?)에게도 감사의 맘을 전합니다. 그리고 많이 가르쳐 주지 못해 아쉽지만 항상 잘 따라주던 원봉이, 현애에게도 고맙다는 말을 전하고 싶습니다. 해주는 것 없이 부탁만 해도 웃는얼굴로 잘 들어주셨던 김쌤, 한쌤에게도 언제나 감사하는 마음 가지고 있습니다. 그래도 김쌤, 한쌤이 있었기에 부족함 없이 대학원 생활을 한 것 같습니다. ^^

항상 티격태격 싸우기도 하지만 서로를 걱정하고 위함이라는 것을 잘 알고 있습니다. 무엇보다도 지금까지 모자란 나를 지켜봐 주었고, 앞으로도 함께할 미영이에게 미안한 마음과 함께 고마움을 전합니다. 너무너무 사랑합니다.

장정희 드림