



저작자표시-비영리 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

컨테이너 터미널에서 안벽 크레인을
고려한 선석 계획 수립에 관한 연구



2010년 2월

부경대학교 대학원

시스템경영공학과

이 원 봉

공학석사 학위논문

컨테이너 터미널에서 안벽 크레인을 고려한 선석 계획
수립에 관한 연구

지도교수 구 평 회

이 논문을 공학석사 학위논문으로 제출함

2010년 2월

부경대학교 대학원

시스템경영공학과

이 원 봉

이원봉의 공학석사 학위논문을 인준함

2010년 02월 22일



주심 공학박사 고 시 근 인

위원 공학박사 김 영 진 인

위원 공학박사 구 평 회 인

목 차

제 1 장 서 론-----	1
1.1 연구의 배경 및 목적 -----	1
제 2 장 기존 연구 및 문제 정의-----	3
2.1 기존 연구 분석-----	3
2.1 문제 정의-----	7
제 3 장 수리모형-----	11
3.1 최적화 모형-----	11
3.2 최적해 도출-----	15
3.3 수리모형 결과 분석-----	17
제 4 장 유전 알고리즘을 이용한 해법-----	18
4.1 유전알고리즘 -----	18
4.2 유전알고리즘의 진행구조-----	19
4.3 초기해 생성-----	20
4.4 적합도 평가-----	25
4.5 유전 연산자-----	26
4.6 종료조건-----	28
4.7 실제 구현 시 진행 절차-----	29
제 5 장 실험 분석-----	33

제 6 장 결론-----	41
참고문헌-----	43
Abstract-----	46



표 목차

[표 1] 선박별 입력 데이터 -----	15
[표 2] 결과 분석 입력 데이터-----	34
[표 3] FCFS 결과 분석(주간 평균 화물량)-----	35
[표 4] 배 할당 순서 변경 시(주간 평균 화물량)-----	35
[표 5] FCFS 결과 분석(평균 화물량의 2배) -----	37
[표 6] 배 할당 순서 변경 시(평균 화물량의 2배)-----	37



그림 목차

[그림 1] BAPD에서 선석의 선박 배정 방법-----	4
[그림 2] BAPC에서 선석의 선박 배정 방법-----	4
[그림 3] 크레인 할당 대안에 따른 작업 시간의 변화 -----	8
[그림 4] 선석 위치 변화에 따른 작업 시간의 변화 -----	8
[그림 5] LINGO 입력창 -----	16
[그림 6] 예제의 결과 -----	17
[그림 7] 초기해 생성 및 유전 알고리즘 진행 절차-----	20
[그림 8] 초기 세대 생성 예 -----	22
[그림 9] 룰렛휠선택에 기초한 배 우선 순위 예 -----	23
[그림 10] 선박의 배정 방법-----	24
[그림 11] 2-point crossover-----	27
[그림 12] 돌연변이 방법-----	28
[그림 13] 8대의 배의 초기해-----	30
[그림 14] 대기 시간에 따른 위치변경-----	31

[그림 15] 최적화 모형과 유전알고리즘의 실험 비교 분석-----	32
[그림 16] 각 상황별 결과 비교(주간 평균 화물량)-----	36
[그림 17] 각 상황별 결과 비교(주간 평균 화물량의 2배)-----	38
[그림 18] 실제 일정 계획 및 구현된 알고리즘 일정계획-----	39



제 1 장 서 론

1.1 연구의 배경 및 목적

오늘날 항만에 입출항 하는 선박은 매년 증가하고 있으며 선박 수의 증가에 비례하여 물동량수도 현저히 증가하고 있다. 세계 컨테이너 물동량은 지속적으로 증가하여 2000년 약 2억 TEU(1TEU는 20 feet 컨테이너 1개)에서 2010년에는 4억 TEU를 넘어설 것으로 전망하고 있다. (삼성경제연구소, 2005년)

이러한 상황에서 현재 각국은 항만 컨테이너 터미널을 중심으로 물류 선진화를 달성하기 위해 노력하고 있다. 이를 위해 하드웨어적인 항만 인프라 구축뿐만 아니라 기초 자원을 최대한 효율적으로 활용하기 위한 운영과 관리도 함께 이루어져야 하지만 현재까지는 항만 컨테이너 터미널의 운영의 효율화를 위한 소프트웨어적인 연구는 미비한 상태이다.

본 논문에서는 터미널의 처리능력 및 생산성을 결정하는 병목 자원인 선석과 안벽 크레인의 효율화를 위한 운영 방안에 대해 연구한다. 컨테이너 터미널에서 경쟁력은 선박의 빠른 처리에 달려있다. 컨테이너 터미널의 경쟁력 확보는 체류 시간의 최소화를 통해 이루어질 수 있는데 여기서 체류시간이란 선박이 항만에 도착해 서비스를 마치고 출발할 때까지의 시간을 뜻하며 도착 후 선석이 할당될 때까지 기다리는 대기 시간과 선석이 할당된 후 접안해 서비스가 이루어지는 접안 시간으로 나눌 수 있으며 즉, 선박 체류시간 = 선석 대기시간 + 접안시간이 된다.

대기 이론을 통해 서비스 시간(접안 시간)을 빠르게 하면 대기 시간(선석 대기시간)을 줄일 수 있는 것을 알 수 있다. 이러한 체류시간의 최소화를 위해 자원을 최대한 활용하여 서비스 시간을 줄이는 것이 필요하다.

현재 국내의 터미널의 효율성은 다른 해외 컨테이너 터미널에 비해 뒤떨어져 있는 상황이다. 부산항의 안벽 크레인의 경우 시간당 약 30개의 컨테이너를 처리하나 시간당 약 35개를 생산하는 중국항이나 55개를 처리하는 ETC항(로테르담)에 비하면 생산성이 현저히 낮다. 삼성경제연구소의 자료에 따르면 2006년 기준 홍콩항은 2,320만 TEU의 물동량을 처리해 부산항이 처리한 1,200만 TEU의 2배에 가까운 컨테이너를 처리했다. 홍콩항은 부산항과 유사하게 공간의 확보면에서 불리한 위치에 있으나, 항만 운영의 효율성을 높이는 것으로 이를 해결하고 있다.

이를 통해 항만의 하드웨어적 인프라뿐 아니라 소프트웨어 측면인 기초 자원의 운영방안에서 효율성이 얼마나 중요한 지 알 수 있다. 항만에서의 운영적 측면은 선석의 일정계획(berth scheduling)을 통해 이루어질 수 있는데 선석 일정계획이란 일정 기간 동안 입항 예정인 선박들을 대상으로 선석을 배정하고 선사의 요청에 따라 선박의 접안 위치와 시간 및 접안 기간을 결정하고 각 선박의 컨테이너를 싣고 내릴 크레인을 배정하는 일련의 과정을 말한다. 선박의 위치 지정은 각 선박마다 요구하는 시간대와 기간 그리고 컨테이너 작업 시에 선박의 선호 위치 등이 있을 수 있는데 일정 계획 기간 내에 다수의 선박들이 있을 경우에는 선박들 간의 모든 요구 조건을 다 만족시킬 수 있는 계획을 작성하는 것은 사실상 매우 어려운 문제이다.

예를 들어 한 선박의 접안 위치 및 접안 기간이 변경될 경우 변경된 선박의 이후 작업과 인접한 선박들의 작업에 또한 변경이 이루어지며 결국 전체 선석 계획의 변경을 초래하게 된다. 또한 컨테이너 크레인은 선석에서 이동 범위가 한정되어 있기 때문에 선박의 위치가 변경될 시에는 배정 가능한 크레인의 수가 달라지며 이로 인해 소요 작업 시간이 변경될 경우에는 전체 기간의 선석 계획도 변경이 이루어지게 된다. 이렇게 각 선박들과 크레인들은 서로의 연관성에 의해 시공간적으로 얽혀 있으므로 부분적인 변경이 결국 전체적인 변경을 초래하게 되는 경우가 생긴다. 이러한 이유로 선석과 안벽크레인이 바로 컨테이너 터미널의 병목자

원이며 이 자원들을 어떻게 활용하는가에 따라 컨테이너 터미널의 경쟁력이 좌우된다고 할 수 있다.

본 논문에서는 컨테이너 터미널에서 선박에 대한 선석과 크레인의 최적 일정 계획을 위한 알고리즘을 개발하고자 한다. 지금까지는 대부분 선석 스케줄과 크레인 할당문제가 문제의 복잡성으로 인하여 각기 별개로 의사결정이 수행되어 왔지만 본 연구에서는 이 두 가지 의사결정 내용을 동시에 수행하는 방안을 제시하고자 한다.



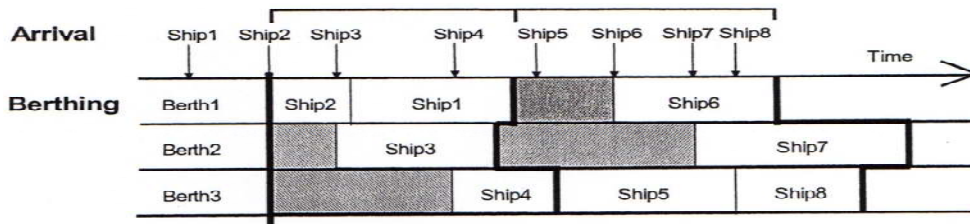
제 2 장 기존 연구 및 문제 정의

이 절에서는 본 논문에서 연구하고자 하는 문제에 대한 기존 연구에 대한 분석 및 문제에 대해 정의 및 본 연구의 목적에 대해 소개하고자 한다.

2.1 기존 연구 분석

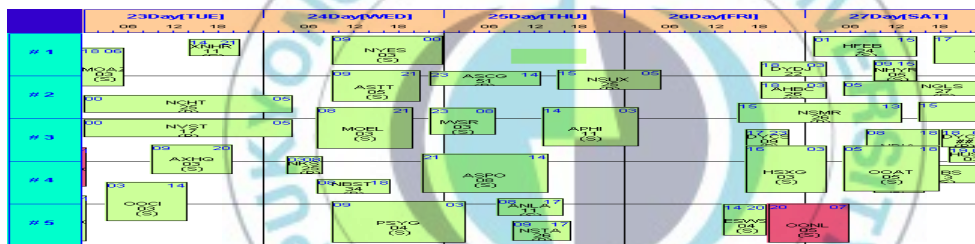
앞 장에서 기술한 바와 같이 본 연구의 대상문제는 길이와 시간을 두 축으로 하는 2차원 공간계획문제로 한정된 공간에 선박 및 크레인 일정 계획을 통해 전체적인 효율성을 최대화 시키고자 함이 본 논문의 문제이다. 하지만 기존 연구들 중 현재까지 선박과 크레인 일정 계획을 동시에 수립하는 방안은 그 문제의 복잡성으로 인해 국내 및 국외에서의 연구 사례가 거의 없으며 대부분의 연구는 선박 일정 계획과 크레인 일정계획을 각기 따로 나누어 연구하고 있다.

이 중 선박 일정 계획 문제 BAP(Berth Allocation Problem)는 NP-hard 문제이므로 현실적인 문제 해결이 어려워 휴리스틱 알고리즘 및 시뮬레이션 등으로 해를 도출하는 방법이 널리 사용되고 있다(Imai et al 2005). 이러한 선박 스케줄 문제는 일반적으로 비연속적(Discrete)인 상황과 연속적 (Continuous)인 상황으로 각각 연구되며 비연속적 선박 입항 위치 지정 문제는 BAPD(Discrete Berth Allocation Problem)라 하며 이 문제는 한 선박에 한 대의 배만 위치함을 뜻한다. (Imai et al 1997, 2001, 2003 : Nishimura et al) 이 문제는 배가 도착 전에 위치가 지정되며 [그림 1]과 같이 한 대의 배는 한 선박에만 위치함을 뜻한다.



[그림 1] BAPD 에서 간트차트로 표현된 선석 계획 (Nishimura et al 2001)

이와 반대로 BAPC(Continuous Berth Allocation Problem)로 불리는 연속적 선석 지정 문제가 있는데 이는 주로 선석 일정 계획의 수정이 자주 이루어지는 공용 터미널을 대상으로 한다. BAPC는 [그림 2]와 같이 배가 선석의 구분 없이 어느 위치에든지 배정될 수 있으며 매우 큰 유연성을 가지며 선석 운영에 높은 효율성을 획득할 수 있지만 NP-hard 문제이므로 현실적인 문제에서 최적해를 도출하기가 어렵다(Imai et al 2005).



[그림 2] BAPC 에서 간트차트로 표현된 선석 계획 (신선대 부두 일정계획)

또한 BAP는 어떤 선박을 대상으로 하는지에 따라 Static와 Dynamic의 두 가지 상황으로 나뉘어 연구 된다. Static의 경우는 계획 선상에서 모든 선박이 입항 후 계획이 시작되며 모든 선박들의 도착 후에 선박들의 입항 시간이 결정되는 상황을 말한다. 주로 선박의 크기가 크며 많은 선박이 드나드는 전용 터미널의 상황에 대해서 연구가 이루어지고 있다(Imai et al 2003, 2005).

Dynamic의 경우에는 계획 시 도착하지 않은 선박과 이미 도착한 선

박 들 다를 고려하며 계획 시작 후에도 도착하는 선박들을 고려하여 계속적으로 스케줄이 수정되는 상황을 말한다. 주로 소규모의 공용 터미널에서의 선석 일정 계획을 말하며, 이 때 선석 일정 계획은 Dynamic로 푸는 것이 보다 효율적이며 현실적이지만 문제 해결이 Static에 비해 어렵다는 단점이 있다.

현재까지 BAP에 관한 연구는 앞서 분류한 상황에 관계없이 최적 및 최소화에 중점을 두었으며 체류 시간이나 위치 혹은 선박의 우선순위 등 여러 가지 기본 자원이자 병목 요소들을 이용하여 최적의 일정계획을 수립하고자 하였다. 이에 관련해 가장 많은 연구가 (Imai et al. 2001, 2005) 현재까지 이루어졌다. 그들은 BAP를 Dynamic으로 확장하여 연구하였다.

그들의 연구는 처음엔 모든 선석의 수심이 같다는 가정으로 문제를 연구 하였는데 Nishimura et al. (2001)은 선석에서의 수심의 깊이에 따른 선박 접안 장소 제약을 추가하여 연구를 확장하였다. 각각의 연구는 각 기 연구의 목적에 따라 Lagrangian relaxation 및 subgradient method를 이용하거나 혹은 유전자 알고리즘과 같은 최적화 탐색기법들을 이용하여 현실적 문제의 해를 도출하였다. Kim and Moon(2003)은 BAPD에 대해 선형 계획 모델을 수립하였다. 이 후 Guan and Cheng(2004)에 의해 Tree search를 통해 선박의 총 체류시간의 최소화를 위한 2가지 수리모델을 성립하여 선석 일정계획 문제를 풀이 하였다.

또한 BAP의 또 다른 관점인 선석에서의 연속적인 선박 배정 문제 (BAPC)가 연구되었는데 Park and Kim(2002)은 배의 출발 지연에 걸리는 비용의 최소화를 목적으로 하는 BAPC에 의해 연구하였으며 Kim and Moon (2003)은 Park and Kim의 연구에 subgradient method를 적용시켜 해를 도출하였다. 또한 Imai et al.(2005)은 Imai et al.(2001, 2005)에서 언급했던 BAPD를 Multi-user container terminal에서 Continuous적인 상황에서의 문제로 확장하고 해를 구하기 위해 유전자 알고리즘을 사용하였다. Cordean et al.(2005)은 DBAP에서 연속적, 불연속적 두 가지 상황 모두에 연구 하였고 선박 일정 계획을 위한 2가지 수리모델과 tabu search

heuristic를 통해 해를 도출하는 방법을 제시하였다. 이 연구들은 선석의 활용과 생산성에 대해 불연속적인 상태에 비해 더 높은 효율성을 획득함을 보여주었다.

컨테이너 터미널의 안벽크레인을 이용해 일정계획에 최적화를 이루고자하는 문제를 CAP(Crane Allocation problem)라 한다. 대부분의 CAP 연구들은 BAP와 개별 문제로 다루어지고 있다. 대다수의 연구들이 선석 스케줄링을 미리 계획된 상태에서 연구 하였고 문제가 풀려나가는 동안 선박에 관련된 부분에 대해선 전혀 고려하지 않아 현실적이지 않으며 CAP가 중점인 연구들 또한 많지 않다. 현재 이에 관련된 연구로는 Daganzo(1989)은 Branch and Bound 기술을 이용해 부두에서 배의 지연에 따른 비용의 최소화를 위한 효율적 크레인 스케줄링에 대해 연구하였고, Kim and Park(2004)는 배의 turnaround time의 최소화를 위해 특정 배를 대상으로 크레인의 수를 예약하는 스케줄 문제에 대해서 연구하였다.

그리고 BAP와 CAP의 동시 연구로는 매우 극소수의 연구가 존재하는데 Schonfeld et al.(1985)가 항만 운영비용을 최소화하는 선석의 수와 선석에 배치될 Crane수의 최적 조합에 대해 연구하였다. 김홍배 등(1998)은 컨테이너 크레인 할당을 위한 수리모형과 유전 알고리즘을 이용한 해법을 제시하였다. Park and Kim(2003)이 선석과 겐트리 크레인에 대한 연구를 했었고 이 후 류광렬(2000)에 의해 휴리스틱 기법을 이용해 최적 선석 및 크레인 일정계획에 관한 연구와 Imai et al.(2007)에 의해 다중 사용자 터미널에서 불연속적인 상황에서의 위치 지정에 관해 선석과 겐트리 크레인의 동시 작업에 대한 연구가 이루어졌다. 위와 같은 연구들은 각기 문제의 상황이 틀리며 선석과 크레인을 함께 연구하였지만 크레인의 작업량에 따른 선박의 작업시간 혹은 정해진 크레인 할당대수 등에서 본 논문에서와의 차이가 있다.

본 논문에서는 앞선 연구들과는 달리 동적이며 연속적인 상황으로 선박의 위치와 크레인 할당 계획 문제를 동시에 고려하며 배의 위치와 크레인의 에 따른 최적 배치 조합에 관해 연구하였고 해의 도출을 위해 수

리모델을 제안하고 현실적인 문제의 해결을 위하여 유전 알고리즘을 기반으로 하는 선석 계획을 제시한다.

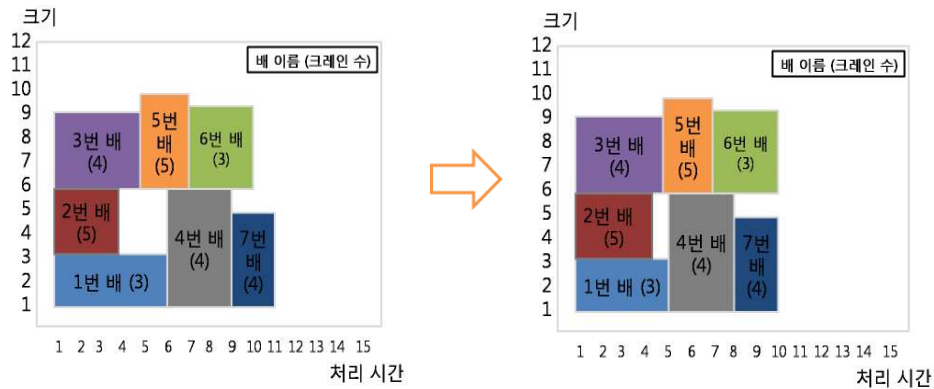
2.2 문제 정의

앞서 기술한 바와 같이 본 연구에서의 목적은 컨테이너 터미널의 정해진 공간 내에서 선박들의 체류시간을 최소화함으로 보다 효율적인 운영을 하고자 함이 목적이며 아래와 같은 가정들로 하여 문제에 대해 보다 현실적인 접근을 하고자 하였다.

본 연구에서 선박에 대한 시간은 다음과 같이 정의한다. 선박의 체류 시간은 대기 시간 + 작업 시간으로 정의 할 수 있는데 대기 시간이란 선박이 선석에 위치를 배정받고 안벽 크레인을 할당 받기 까지 대기 하는 시간을 뜻하며, 작업 시간은 선박이 선석을 배정 받은 후 안벽 크레인이 작업을 시작하여 컨테이너 양 적하를 마치고 출항하기까지 걸리는 시간을 말한다.

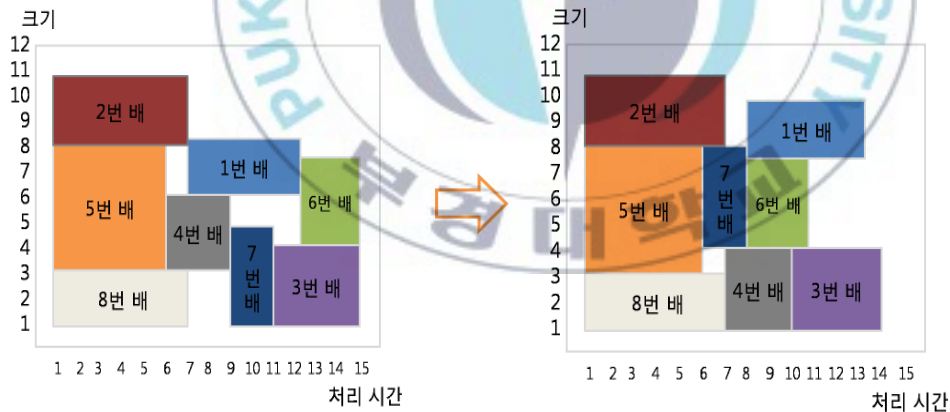
본 논문에서는 선석 배정 문제에서 크레인을 고려하지 않거나 크레인의 할당 수에 따른 작업 처리 시간의 변화를 고려하지 않은 기존 연구와는 달리 크레인의 할당과 크레인 할당 수에 따른 선박의 체류 시간의 변화를 고려하였다. 선박의 체류 시간의 변화에 선박 위치 배정과 크레인 할당 배정이 어떠한 영향을 미치는 지는 아래 [그림 3]을 보면 알 수 있다. 먼저 크레인 할당에 따른 체류시간의 변화는 기존의 선박 1과 선박 2에 할당된 크레인 수를 조정하여 선박 2의 크레인을 5대에서 14대로 줄여 처리시간을 증가 시키고, 여기서 남은 크레인 1대는 선박 1에 추가로 배정해 선박 1의 처리시간을 줄여 이 두 선박의 작업이 동시에 마쳐질 수 있도록 한다. 그 결과 1번 선박의 체류 시간이 줄어들면서 생기는 빈 공간에 4번 선박과 7번 선박의 작업이 앞당겨 지게

되어 전체 작업 완료 시간이 줄어들게 된다. 이와 같이 선박의 크레인 할당 대 수에 따라 체류 시간이 변하는 것을 확인 할 수 있다.



[그림 3] 크레인 할당 수에 따른 작업 시간의 변화

또한 선박의 위치 배정에 따른 체류시간의 변화는 아래 [그림 4]에서 나타난 바와 같이 7번 배의 위치를 4번 배와 바꾸었을 때 생기는 빈 공간에 의해서 뒤의 위치한 배들의 위치가 조정되며 전체 작업 시간이 줄어드는 것을 알 수가 있다.



[그림 4] 선석 위치 변화에 따른 작업 시간의 변화

따라서 본 논문에서는 주어진 제한된 자원 안에 선석의 위치 변화와

크레인 할당 수에 대해선 선택 가능한 대안들을 제시하고 그 대안들의 최적 조합을 통해 체류 시간의 최소화를 구하고자 한다.

본 연구에서는 아래와 같은 가정들을 기초로 하여 선박에 대해 선석과 크레인 배정에 대한 모형을 수립하였다.

1. 배의 체류 시간을 최소화하여 일정 시간 동안에 최대한의 배를 할당하고자 함이 목적이다.
2. 배와 선석간의 물리적인 제한(물의 깊이와 배의 폭 같은 경우)은 제외한다.
3. 배의 작업이 시작 된 후 (배가 위치를 지정받고 크레인의 작업이 완료될 때까지), 작업이 완료 될 때까지 어떠한 간섭도 생기지 않는다.
4. 배는 도착 시간이 미리 알려져 있고 작업이 마침과 동시에 출항한다.
5. 선박이 이미 할당된 공간에서는 중복하여 위치배정을 할 수 없다.
6. 한 시점에서 배치되는 선박들은 선석의 총 길이를 넘어 배치 될 수 없다.
7. 배정 받은 크레인의 수에 의해 배의 작업 처리 시간은 의존적이다.
8. 크레인의 작업속도는 동일하며 배의 작업 처리 시간은 배의 물동량에 의해 좌우된다.
9. 한 시점에서 총 할당 가능한 크레인의 수는 정해져 있는 수를 넘을 수 없다.
10. 각 선박은 크레인 할당에 대해 변동적으로 선택 가능한 할당 대안을 가지며 선택 가능한 할당 대안 안에서 체류 시간을 최저로 하는 최적의 조합을 찾는다.

위의 가정들을 통해 선박에 대한 선석 배정 및 크레인의 할당을 연속적이며 동적인 상황으로 나타내었고 이전 연구들에 비해 보다 현실적으로 각 자원들을 할당하도록 하였다. 선석의 위치와 크레인 할당 수를

입력된 값 들 중에 선택 가능하도록 하여 실제 컨테이너 터미널에서의 배치 상황과 같은 상황에서 제한된 자원 안에서 최적 조합 배치를 찾아내는 점이 이전의 논문들과의 가장 큰 차이점이다.



제 3 장 수리모형

3.1 BAP 최적화 모형

이 절에서는 선석 및 크레인 최적 일정 계획 문제를 풀기 위해 사용된 기호들과 최적화 모형을 소개한다.

3.1.1 Notation

본 연구에서 사용되는 각 기호들은 변수(variables)와 모수(parameters)로 나눌 수 있고, 그중 모수에는 다음과 같은 것들이 있다.

<모수>

w_i = 배 i 의 가중치 (배들의 우선순위)

a_i = 배 i 의 도착 시간

d_i = 배 i 의 출발 요구 시간

s_i = 배 i 의 길이

j_i = 배 i 에 할당된 j 번 째 크레인

p_{ij} = 배 i 에 j 번 째 크레인 할당 대안에 따른 처리 시간

q_{ij}	= 배 i 에 j 번 째 크레인 할당 시 할당되는 크레인 수
q_t	= 시점 t 에서 할당되는 크레인 수
α	= 체류 시간의 가중치
β	= 출발 요구 초과 시간의 가중치
M	= 양의 큰 상수값
T	= 계획기간
S	= 계획 대상 작업장의 길이
N	= 계획 대상 선박의 수

의사결정변수(decision variables)인 독립변수들은 다음과 같다.

<결정변수>

c_i	= 배 i 의 완료 시간
x_{ijst}	= 1 (if lower left corner of vessel rectangle is located at grid (s,t) by using j^{th} crane assignment) 0 otherwise
y_{ist}	= 1 (if grid (s,t) is covered by vessel rectangle i) 0 otherwise

3.1.2 수리모형

위와 같은 부호와 가정을 사용하여 계획기간 동안 선박의 체류 시간을 최소화 하는 최적 선석 스케줄을 아래의 모형에 의해서 구할 수 있다.

$$\text{Minimize } z = \alpha \sum_{i=1}^N w_i (c_i - a_i) + \beta \sum_{i=1}^N w_i (d_i - c_i) \quad (1)$$

,

subject to

$$\sum_{j \in J_i} \sum_{s=1}^{S-s_i+1} \sum_{t=a_i}^{T-p_{ij}+1} x_{ijst} = 1 \quad \forall i \quad (2)$$

$$\sum_{i=1}^N y_{ist} \leq 1 \quad \forall s, t \quad (3)$$

$$\sum_{j \in J_i} \sum_{s=1}^{S-s_i+1} \sum_t t \cdot x_{ijst} + \sum_{j \in J_i} \sum_{s=1}^{S-s_i+1} \sum_t p_{ij} \cdot x_{ijst} = c_i \quad \forall i \quad (4)$$

$$\sum_{j \in J_i} \left(\sum_{m=s}^{S+s_i-1} \sum_{n=t}^{T-p_{ij}-1} y_{imn} - p_{ij} \cdot s_i \cdot x_{ijst} \right) - \left(\sum_{j \in J_i} x_{ijst} - 1 \right) M \geq 0 \quad \forall i, s, t \quad (5)$$

$$\sum_{i=1}^N \sum_{j \in J_i} \sum_{s=1}^{S-s_i+1} \sum_{t=-p_{ij}+1}^{pt} q_{ij} \cdot x_{ijst} \leq q_t \quad \forall t \quad (6)$$

$$x_{ijst}, y_{ist} = 0 \text{ or } 1 \quad \forall i, j, s, t \quad (7)$$

목적함수식 (1)은 계획 기간 동안 모든 선박의 체류 시간을 최소화하기 위한 목적식이다. 여기에는 배의 *weight*와 배의 작업 완료시간, 그리고 배의 도착시간 및 배의 출발 요구시간 등으로 구성되어 있다. 이 때 배의 *weight*는 각 배들의 우선순위를 나타낸다. 이 식에서는 전체 선박의 체류 시간과 출발 요구 시간을 초과하는 시간의 최소화를 목적으로 한다. 제약식(2)는 모든 배들은 선석에서 한 공간에 한 대의 배만 할당됨을 나타내는 수식이다. 제약식(3)에 의해 시공간면에서 한 공간에 최대 한 대의 배만 할당 될 수 있음을 나타내며 중복할당을 방지하는 제약식이다. 제약식(4)는 완료시간을 정의한다. 제약식(5)는 배의 LL(lower left corner)이 1이면 ($x_{ijst} = 1$) 이에 대응하는 영역은 모두 1 ($y_{ist} = 1$)이 되어 중복을 배제토록 하는 제약식이다. 이 때 제약식 x_{ijst} , y_{ist} 에서 사용된 *grid* 란 계획선상에서 선박이 위치하는 (space, time)을 뜻하며, *rectangle* 은 한 선박이 차지하는 공간에서의 선박의 끝부분으로, 이 부분을 시작으로 선박이 차지하는 값만큼 공간을 점유한다. 제약식 (6)은 QC수의 제약을 나타내며 이 수식으로 인해 배의 할당 시 한 시점에서 정해진 수 이상의 크레인 할당을 방지하도록 한다.

이전 연구(Imai et al 2008)에서 이와 같이 선석 및 크레인을 동시에 지정하는 문제에 대한 수리모형을 제시하고 문제를 풀었지만 크레인의 수와 그에 비례하는 작업처리 시간 및 각 선박의 화물량에 따른 처리 시간 차이 등을 고려하지 않았다. 본 연구에서는 위의 부분들을 고려하여 최적 수리 모형을 제시하고 이 수리 모형을 통해 최적해를 도출했다. 그러나 작은 크기의 문제에서는 최적해를 찾을 수 있었지만 현실적인 문제를 풀 수 없었고 이를 위해 유전자 알고리즘을 이용하여 해를 도출하고자 한다.

3.2 최적해 도출

간단한 예제를 통해 위에서 제시한 수리모형의 해를 도출하기 위해 상용 최적화 소프트웨어인 LINGO를 이용하여 풀어보았다. 예제로 공간의 크기를 12로 하며 한 시점에 9대의 크레인을 초과할 수 없다는 가정 하에 모든 같은 우선순위를 가졌다고 가정하여 총 8대의 배에 대해 아래 [표 1]과 같이 데이터들을 입력하고 풀어보았다.

[표 1.] 선박별 입력 데이터

No.	선박명	입항예정일시	출항예정일시	선적용량	선박 Size	컨테이너 크레인 1	처리시간 1	컨테이너 크레인 2	처리시간 2	컨테이너 크레인 3	처리시간 3
1	1	2009-10-01 3:00	2009-10-2	321	4	2	5	3	4	4	3
2	2	2009-10-01 2:00	2009-10-2	322	3	2	5	3	4	4	3
3	3	2009-10-01 4:00	2009-10-1	415	3	3	5	4	4	5	3
4	4	2009-10-01 2:00	2009-10-1	162	3	1	5	2	3	3	2
5	5	2009-10-01 1:00	2009-10-2	534	5	3	6	4	5	5	4
6	6	2009-10-01 5:00	2009-10-2	251	4	2	4	3	3	4	2
7	7	2009-10-01 3:00	2009-10-3	252	5	2	4	3	3	4	2
8	8	2009-10-01 1:00	2009-10-2	384	3	2	6	3	5	4	3

이 자료를 바탕으로 최적화 모형으로 해를 찾아보면 다음과 같은 해를 얻을 수 있다. 2십만번 정도의 반복을 통해 해가 얻어지며 해를 찾는 데 걸리는 시간은 4분 36초 정도이다. 1~8번까지의 배들은 대기시간을 포함하여 총 작업완료 시간이 다음과 같이 (5, 5, 8, 2, 11, 3, 3, 3시간) 가지며 작업완료까지 총 40시간을 가지게 된다. 위 예제를 풀기 위한 LINGO 입력창은 [그림 5]와 같다.

```

Model:
Sets:
Ship: a, s, w, c;
opt;
time;
loc;
assign(ship, opt, loc, time) : x;
grid(ship, loc, time) : y;
alt(ship, opt) : nqc, ptime;
jst(opt, loc, time);
endsets

data:
shp = 10;
len = 12;
tim = 15;
tqc = 9;
m = 99999;
ship = 1..shp;
opt = 1..3;
loc = 1..len;
time = 1..tim;
a = 3 2 4 2 1 5 3 1 3 2 ;
s = 4 3 3 3 5 4 5 3 5 3 ;
w = 1 1 1 1 1 1 1 1 1 1 ;
d = 10 9 9 10 12 13 14 11 ;
nqc = 2 3 4
      2 3 4
      3 4 5
      1 2 3
      3 4 5
      2 3 4
      2 3 4
      2 3 4
      2 3 4
      2 3 4 ;
ptime = 5 4 3
        5 4 3
        5 4 3
        5 3 2
        6 5 4
        4 3 2
        4 3 2
        6 5 3
        6 5 3
        6 5 3 ;
enddata

min = @sum(ship(i):w(i)*(c(i)-a(i)));

@for(ship(i): @sum(opt(j):@sum(loc(k)|k#LE#(len-s(i)+1):
@sum(time(t)|t#GE#a(i)#and#t#le#(tim-ptime(i,j)+1):x(i,j,k,t)))=1);
@for(loc(k): @for(time(t):@sum(ship(i):y(i,k,t))<=1));
@for(ship(i):@sum(opt(j):@sum(loc(k)|k#LE#(len-s(i)+1):
@sum(time(t)|t#GE#a(i)#and#t#le#(tim-ptime(i,j)+1):(t+ptime(i,j))*x(i,j,k,t))) = c(i));
@for(ship(i):@for(opt(j):@for(loc(k):@for(time(t): (@sum(loc(kk)|kk#GE#k#and#kk#LE#(k+s(i)-1):
@sum(time(v)|v#GE#t#and#v#le#(t+ptime(i,j)-1):y(i,kk,v))-ptime(i,j)*s(i)-(x(i,j,k,t)-1 * m) >= 0)))));
@for(time(t):@sum(ship(i):@sum(opt(j):@sum(loc(k)|k#LE#(len-s(i)+1):
@sum(time(v)|v#GE#(t-ptime(i,j)+1)#and#v#le#t:nqc(i,j)*x(i,j,k,v))))<=tqc);
@for(ship(i):@for(opt(j):@for(loc(k):@for(time(t):@BIN(x(i,j,k,t)))));
@for(ship(i):@for(loc(k):@for(time(t):@BIN(y(i,k,t)))));

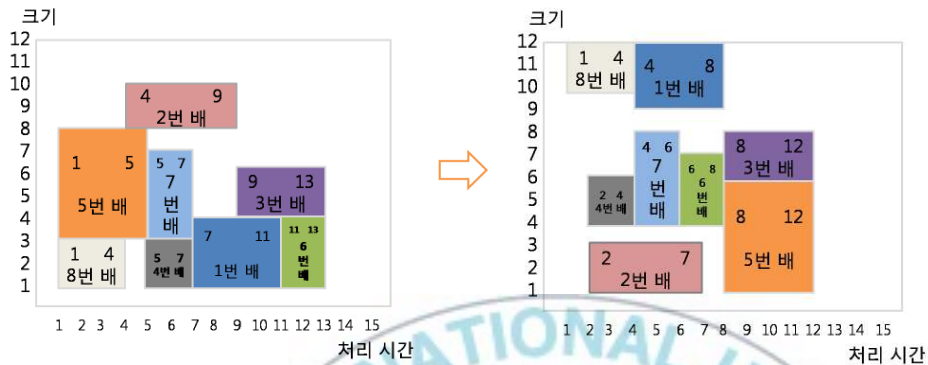
end

```

[그림 5] LINGO 입력창

3.3 수리모형 결과 분석

상용 최적화 소프트웨어LINGO로 풀어본 예제를 앞서 설명한 [그림 1]에서 나타낸 것처럼 일정계획을 간트차트로 표현하면 아래 [그림 6]와 같다.



(a : FCFS의 결과)

(b : 최적화 모형의 결과)

[그림 6] 예제의 결과 (FCFS와 최적화모형의 결과)

계획 기간 중 실험마다 배의 위치 및 크레인 할당 수가 약간의 차이는 있었지만 최적해를 찾을 수 있었다. [그림 6]의 결과에서 크레인 할당 대수가 같은 경우에서 배의 위치를 FCFS로 했을 경우 최적화 모형에 비해 Makespan(모든 선박이 작업을 완료 하는 시점)이 13시로 대상 선박들의 전체 체류시간은 48시간이 걸렸지만 최적화모형을 통한 결과에서는 배의 위치가 바꿈으로 같은 크레인 수를 할당 받지만 Makespan은 12시며 선박들의 전체 체류 시간은 40시간이 걸렸다. 현재 결과 그림을 살펴보면 공간이 있음에도 불구하고 크레인과 공간의 제약에 의해 배의 도착 시간 직후 작업이 이루어지지 않고 대기 시간을 가지는 것을 확인할 수 있다. (ex.5번 배의 경우 도착 시간은 1시이며 작업시간은 4시간이지만 제약 조건에 의해 8시에 작업이 시작되어 12시에 작업이 완료되며 도착에서 완료까지 총 11시간이 걸린 것을 확인할 수 있다.)

제 4 장 유전자알고리즘을 이용한 BAP

3절에서 최적화 모형을 이용하여 최적해를 찾아보았다. LINGO에서 얻은 결과를 보면 반복이 223413 번으로 문제를 푸는데 얼마나 복잡한가를 잘 보여준다. 위 예제는 입력 데이터가 8개뿐인 문제이지만 현실적인 문제에서는 데이터의 수가 더 많아지게 되며 최적해를 찾을 수 없게 된다. 따라서 이와 같은 NP-hard 문제에서 현실적인 문제에 대한 해결을 위해 유전 알고리즘을 이용하여 최적해에 근사한 해를 찾고자 하며 이 절에서는 유전 알고리즘을 이용해 해를 도출하는 절차를 제시하고자 한다.

4.1 유전알고리즘의 개요

유전알고리즘은 Holland(1997)에 의해 연구되기 시작하여 많은 조합 최적화 문제에 응용된 일종의 인공지능기법으로서, 자연의 진화과정을 모방하여 개발한 탐색알고리즘으로 최근에는 스케줄링 분야에 활발하게 응용되고 있다. 유전알고리즘을 적용하기 위해서는 대상문제의 특징에 맞는 알고리즘, 관련된 여러 조정변수들, 유전인자의 유전염색체(chromosome)의 표현방법, 적합도(fitness) 평가함수, 모집단의 초기화 방법, 염색체 선택전략 등을 결정해야 한다.

유전자 알고리즘은 다른 탐색이 최적화 방법과 다음과 같은 점에서 차이점이 있다

1. 파라미터를 코딩한 것을 직접 이용한다.
2. 점(point)이 아닌 다점(multi points, : 군(population)) 탐색 방법이다.

3. 탐색에 비용 정보(fitness function)를 이용하며, blind search를 한다. (미분값이나 다른 부가적인 지식을 요구하지 않는다.).
4. 결정론적인 규칙이 없고 확률적 연산자를 사용하여 수행된다.

이와 같은 특징을 가짐으로 다른 최적해 탐색기법이나 혹은 최적화 방법들과 같이 계산에 의존한 방법에 비해 전역적 해를 구할 가능성이 높으며 다른 최적해 탐색 방법에 비해 빠른 처리시간을 가짐으로 효율적이다.

본 연구에서는 여러 가지 제약조건들이 복잡하게 얹혀있고 NP-Hard 문제로 알려진 본 문제의 해결을 위해 유전 알고리즘을 사용하였다.

4.2 유전알고리즘의 진행 구조

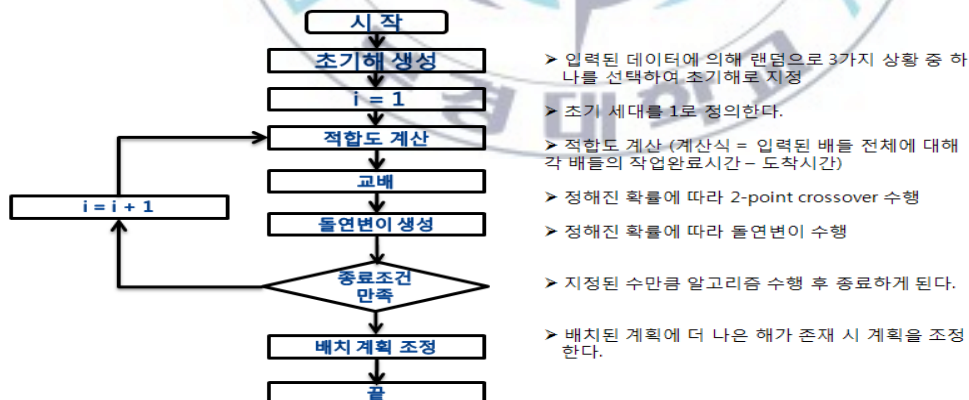
유전 알고리즘의 진행단계는 풀고자하는 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현한 다음, 이들을 점차적으로 변형함으로써 점점 더 좋은 해들을 생성하게 된다. 즉 풀고자 하는 문제에 대한 가능한 해들을 염색체로 표현한 다음 이들을 점차적으로 변형함으로써 점점 더 좋은 해들을 생성한다. 각각의 가능한 해를 하나의 유기체 (organism) 또는 개체 (individual) 로 보며 이들의 집합을 개체군 (population) 이라 한다. 하나의 개체는 보통 한 개 또는 여러 개의 염색체로 구성되며 염색체를 변형하는 연산자들을 유전연산자 (genetic operator) 라 한다. 기본적인 연산자는 다음의 3가지가 있다. 선택 (selection, 집단 중에서 적응도의 분포에 따라서 다음의 단계로 교배를 행하는 개체의 생존 분포를 결정한다. 적응도의 분포에 기초하고 있기 때문에 적응도가 높은 개체일수록 보다 많은 자손을 남기기 쉽게 된다),

교배 (crossover, 2개의 염색체 사이에서 유전자를 바꾸어 넣어 새로운 개체를 발생시킨다), 돌연변이 (mutation, 유전자의 어떤 부분의 값을 강제로 변화시킨다)의 연산자를 거친 후 적합도를 평가하고 조건에 부합되는지를 확인 후 진행을 완료하게 된다.

4.3 초기해 생성

유전 알고리즘을 사용하기 위해서는 우선 염색체 표현 방법을 결정해야 한다. 본 연구에서는 계획 대상 기간 동안에 배정대상 선박들과 크레인의 수를 [0,1]사이의 랜덤 실수로 생성하여 배정 대상 선박의 배치 순서를 정하는 방법을 제시하고, 제시된 방법의 결과와 수리모형의 최적해와 비교하여 분석한다. 입력되는 변수 값으로는 배 도착 시간, 컨테이너 수, 출발시간 및 선박 접안 위치와 컨테이너 크레인의 서비스 가능 여부 및 시작과 완료시간들이 있다. 이 알고리즘에서 목적식으로는 컨테이너 크레인의 할당을 고려한 배의 체류 시간의 최소화로 앞선 수리 모형에서의 목적식을 사용한다.

본 연구의 전체 진행도는 아래 [그림 7]과 같이 표현할 수 있다.



[그림 7]. 초기해 생성 및 유전 알고리즘 진행 절차

본 알고리즘의 진행 절차에 앞서 초기해 생성을 위해 먼저 계획상에 있는 모든 선박들과 컨테이너 크레인에 대한 데이터를 작성한다. 이 후 선박의 체류 시간을 크레인에 따라 계산 한 후 초기해를 생성하게 된다. 입력될 데이터들은 앞서 3.2절의 [표-1]과 동일한 값을 입력했다.

표 1. 선박별 입력 데이터

No.	선박명	입항예정일시	출항예정일시	선적용량	선박 Size	컨테이너 크레인 1	처리시간 1	컨테이너 크레인 2	처리시간 2	컨테이너 크레인 3	처리시간 3
1	1	2009-10-01 3:00	2009-10-2	321	40	2	5	3	4	4	3
2	2	2009-10-01 2:00	2009-10-2	322	30	2	5	3	4	4	3
3	3	2009-10-01 4:00	2009-10-1	415	30	3	5	4	4	5	3
4	4	2009-10-01 2:00	2009-10-1	162	30	1	5	2	3	3	2
5	5	2009-10-01 1:00	2009-10-2	534	50	3	6	4	5	5	4
6	6	2009-10-01 5:00	2009-10-2	251	40	2	4	3	3	4	2
7	7	2009-10-01 3:00	2009-10-3	252	50	2	4	3	3	4	2
8	8	2009-10-01 1:00	2009-10-2	384	30	2	6	3	5	4	3

각 선박의 체류 시간은 초기에 3가지 대안으로 크레인의 할당 대수를 지정하여 할당된 크레인 대수에 의해 좌우되도록 한다. 처리 시간으로 는 선박의 컨테이너 물량을 할당된 크레인 개수로 나누도록 하였고 컨테이너 크레인의 평균 시간당 처리량은 30으로 하였다. (ex. 1번 선박의 컨테이너 처리물량이 523개 일 때 크레인이 3대가 할당되었다면 처리 시간은 반올림을 위해 엑셀에서 다음과 같이 입력하여 $\text{round}[(\text{화물량} / (3 * 30) + 1) / 2]$ 로 계산하여 시간 단위로 표현하도록 하였다.)

초기해 생성 절차 및 유전 알고리즘 진행 절차

Step 1. 선박별 크레인 할당 대안 선택과정 :

선박의 닻 수와 각 선박이 가지는 대안의 수만큼 랜덤으로 실수를 발생시킨다. 아래 [그림 8]과 같이 선박은 각각 3개의 랜덤 실수를 가지며 각기 다른 실수로 각 선박 별 데이터를 구별하며 이 후 단계로 진행한다.

생성된 실수들 중 선박과 가정된 선박의 크레인 대수에 해당되는 실수들 중 가장 높은 값을 가진 값 (ex. 만약 1번 선박이 위의 그림과 같이 0.31, 0.78, 0.68 의 실수를 가질 경우 이 중 0.78을 해로 선택)을 선택하며 이와 같이 선택된 실수 값의 크레인 대수를 초기해로 선택한다.



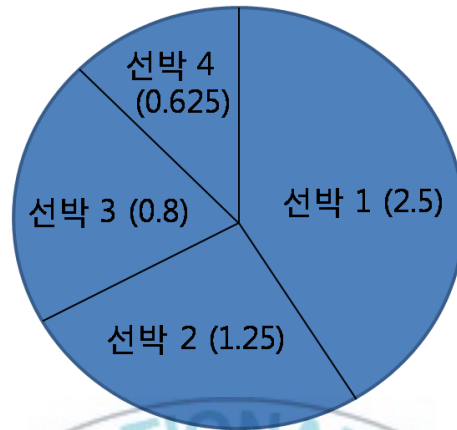
[그림 8] 초기 세대 생성 예

Step 2. 선박별 집안 시간 및 위치 지정

컨테이너 크레인이 배정된 후 각각의 배들은 도착 시간 이후 시점에서 위치를 배정받는다. 모든 선박에 대해 시공간적으로 중복되지 않고 한 시점에서 제한된 크레인 수 이상을 초과하지 않도록 선박을 배치한다. 이 때 선박의 위치 배정 방법은 유전자 알고리즘의 룰렛 휠 선택과 같은 방법을 사용하여 위치를 배정하였고 이 후 도착 시간에 따라 선박별 우선순위를 결정할 수 있도록 하였다.

Step 2-1. 모든 선박들은 도착 시간과 출발 요구 시간에 따라 일정 비율을 가지며 그 비율에 따라 선택될 확률이 정해진다. 전체 선박의 도착시간의 합이 10이고 도착한 시간에 따라 각각의 선박은 1, 2, 3, 4의 도착 시간의 값과 4, 4, 5, 7의 출발 요구 시간을 가진다면 (전체 선박들의 도착시간의 합/전체 선박 수/도착 시간)+(전체 선박들의 출발 요구 시간의 합/전체 선박 수/(출발 요구 시간-도착시간))으로 각각 (4.2, 3.75, 3.3, 2.325)의 값을 가지게 된

다. 위와 같은 방법으로 아래 [그림 9]에서와 같이 계산한 4개의 각 염색체의 비율을 모두 합한 값만큼의 크기를 가진 공간을 가정하고 각 염색체의 값만큼 비례해서 룰렛 휠 상의 공간을 배정 받도록 한다.



[그림 9] 룰렛 휠 선택에 기초한 선택 우선순위 예

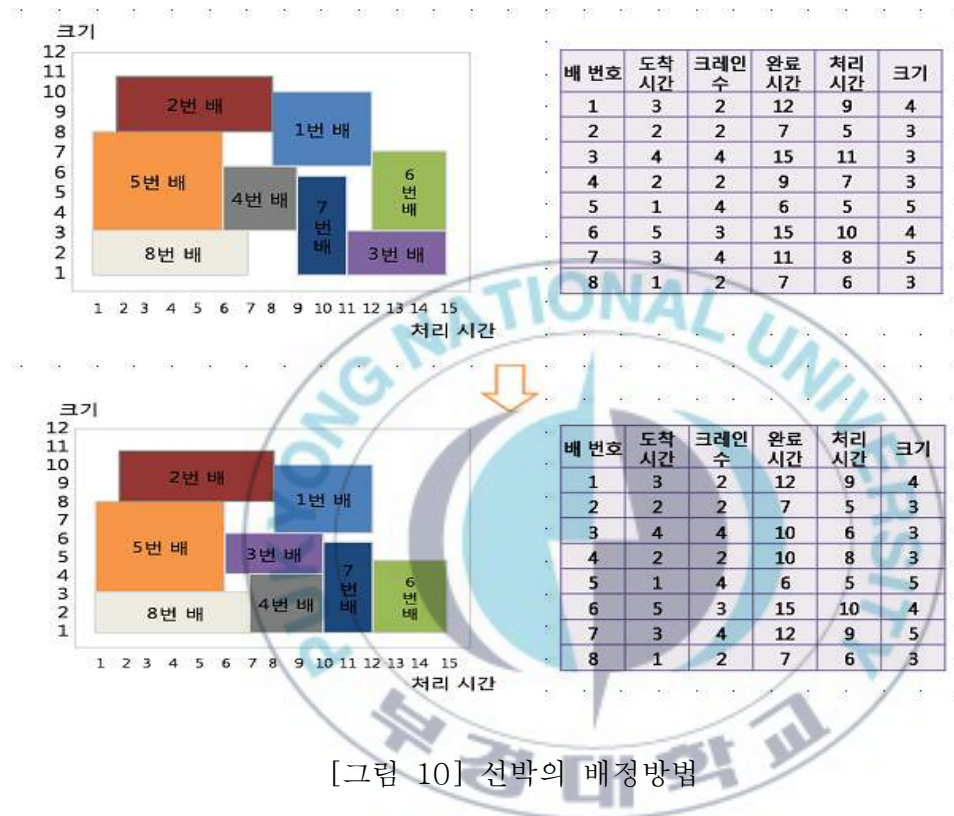
Step 2-2. 시공간 배정 방법

공간을 탐색하고 배정하는 방법은 전역 탐색 기법을 사용하며 각 배들의 배치는 배의 시작 순서에 상관없이 (0, 0) 에서부터 시작하여 배의 크기에 따라 들어갈 수 있는 공간을 찾고 할당한다. 이 때 이미 할당 된 공간에 재배치는 불가능하도록 하였다. 배치 도중 선택이 같은 시간대에 작업을 시작 하는 경우에는 공간 축을 늘려가며 배치하고 만약 같은 시점에서 더 이상 할당 가능 공간이 없을 경우에는 시간 축에서 늘려가며 배치하도록 한다.

Step 2-3. 배치된 선택들 중 대기 시간이 발생하는 배들에 대해 리스트를 작성한다.

Step 3. 해 개선 단계

대기 시간이 가장 긴 선박을 기준으로 대기 시간이 발생한 모든 배들에 대해 위치를 교환하게 한다. 대기 시간 행렬의 수가 $W = n$ 일 경우 대기 시간 행렬의 $w = n-1$ 의 위치까지 모두 반복하며 그 중 나온 최적해를 저장한다.



[그림 10] 선박의 배정방법

(ex. 위의 [그림 10]와 같이 도착시간의 비율로 위치를 배정 했을 때 1번, 3번, 4번, 6번, 7번 배가 대기 시간이 발생 하며 이중 대기 시간이 가장 긴 배는 3번 배로 7시간의 대기 시간을 가진다. 이를 앞서 설명한 방법처럼 대기 시간이 발생한 배 중 가장 짧은 배인 1번 배 (4시간의 대기 시간 발생)부터 위치를 교환 할 시 58 값이 나왔다. 이와 같

은 방식으로 $n-1$ 까지 선박의 위치를 교환하고 그 중 최적값을 가지는 위치를 해로 선택하게 한다.

Step 4. 위와 같은 과정으로 정해진 수만큼의 초기해를 만든 후, 초기해들을 이용해 유전자 알고리즘을 수행하여 최적해를 도출하도록 한다.

Step 4-1. 초기해로 생성된 랜덤실수의 수만큼 다시 0~1사이의 랜덤 실수를 생성시킨다.

Step 4-2. 재생산을 위해 부모세대를 토너먼트 방식으로 선택 후 정해진 확률에 따라 2-point crossover로 교배 및 돌연변이를 통한 유전자 알고리즘을 수행한다.

Step 5. 종료조건
주어진 종료 조건(전체 진행 횟수)을 모두 만족하는 경우 도출된 최적해를 저장하고 종료하며 그렇지 않은 경우 Step 2에서 Step 6의 상황을 종료 조건까지 반복하도록 한다..

4.4 적합도 평가

유전알고리즘에서는 새 연산을 거쳐 새로운 집단이 완성될 때마다 개체들의 적합도가 평가되는데 이 논문에서는 앞서 수리모델에서 정의한 목적함수를 적합도 평가로 사용하였다.

$$\text{Minimize } z = \alpha \sum_{i=1}^N w_i (c_i - a_i) + \beta \sum_{i=1}^N w_i (d_i - c_i)$$

위 식에서 얻어진 적합도를 바탕으로 재배열이 이루어지는데 재배열이란 매 세대마다 모집단 내 염색체에 대해 낮은 적합도 값을 갖는 염색체가 높은 염색체보다 선택될 기회를 더 많이 갖게 되는 과정이다. 즉, 적합도가 낮은 값을 갖게 하는 염색체 배열은 높은 값을 가지는 배열에 비해 더 많이 선택되는 것을 말하며 이 논문에서 토너먼트 선택(tournament selection)을 사용하였다. 이 방법은 개체군 중에서 일정한 개수의 개체를 임의로 선택하여 그 중에 더 나은 적합도를 가지는 개체를 다음 세대에 남기는 방법이다. 이 선택법을 사용하여 종료조건으로 정해진 세대의 개체수가 종료조건에 도달할 때까지 해를 선택하며 작업을 수행한다.

4.5 유전연산자

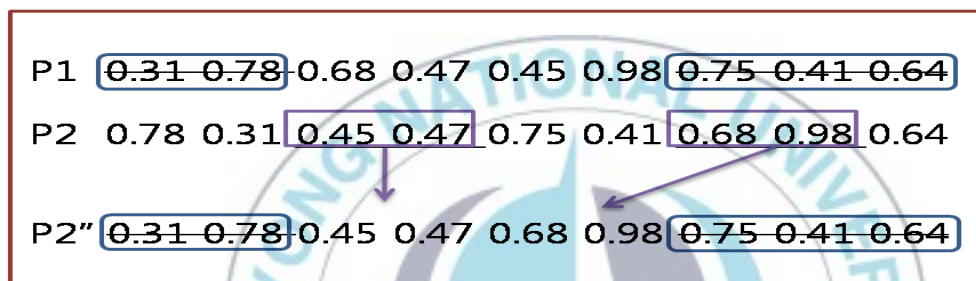
다음 세대를 결정하는 것이 유전 연산자이다. 유전 법칙으로는 교차(Crossover)와 돌연변이(Mutation)가 있다.

4.5.1 교배(Crossover)

일반 생명체는 세대 내에서의 교배를 통해 다음 세대를 생성한다. 일반적으로 두 개의 해가 교배를 해서 다음 세대의 해를 생성하게 되며, 새로운 해는 각각의 부모 해로부터 서로 겹치지 않는 위치의 유전체를

받아 새로운 유전자를 구성하게 된다. 이때 염색체가 재조합되는 과정에서 부모 염색체의 일부분이 특정 위치를 기준으로 서로 바뀌어 결합되는 경우가 있는데 이 현상을 교배라고 한다. 유전 알고리즘에서는 이 교차 현상을 의도적으로 이용, 부모해를 교배시켜서 자식해를 만들어낸다. 탐색 공간상의 가능한 새로운 점을 찾기 위하여 부모 염색체 쌍을 임의로 선택하고, 교배점 이 후의 유전자들을 서로 교환 결합함으로써 자손을 생성한다. 이때 유전자를 절단하는 교배점은 염색체 내에서 임의로 선택된다. 이러한 연산은 교배된 자손의 수가 부모 집단의 크기와 같을 때까지 반복된다,

본 연구에서는 교차 변이는 crossover을 사용하여 표현하였고 진행 단계는 [그림 11]과 같다



[그림 11] 2-point crossover

위의 그림과 같이 이점교차는 원소 사이의 두 절단점을 임의로 선택한 다음 두 절단점에 의해서 나누어진 중간부분의 유전인자를 상호 교환하여 자손을 생성하는 방법이다. 실험적으로 흔히 이점교차가 일점교차보다 더 좋은 해를 유도하는 것으로 알려져 있다.

4.5.2 돌연변이 (Mutation)

진화가 계속 되는 동안 재생산과 교배 연산자는 집단을 더욱 강하게 하며 이로 인해 염색체들은 서로 닮아가게 되지만 이러한 현상이 세대

초기에 발생하게 되면 유전자의 다양성이 결핍되어 최적해를 구하는데 어려움이 생길 수 있다. 이러한 경우에서 벗어나기 위한 방법으로 돌연변이를 사용하는데 염색체 내의 비트를 정해진 돌연변이 확률을 토대로 변경시킨다. 이렇게 함으로 초기 세대에서 모든 염색체의 특정 비트가 고정되는 것을 방지하고 탐색영역을 확대할 수가 있다.

교배확률과 같이 돌연변이 확률도 그 빈도, 즉 반전되는 비트 수를 조절한다. 일반적으로 돌연변이 확률을 높게 설정하면 돌연변이가 일어나기 쉽고 반대로 낮게 설정하면 일어나기 어렵다.

본 연구에서는 돌연변이 방법으로 [그림 12]와 같은 교환방법을 사용하였다.



[그림 12] 돌연변이 방법

위의 그림에서와 같이 교환 방법은 랜덤으로 두 유전자가 선택되고 그 선택된 위치의 유전자 실수값을 교환하여 위치를 재배열 하도록 한다.

4.6 종료조건

본 논문에서는 전체 진행의 최대 세대수 및 연속된 최적값을 제한함으로써 유전 알고리즘을 종료하도록 하였다. 예를 들어 총 세대수가 1000 이라면 1000번의 알고리즘 수행 후 종료하게 되며 이 때 만약 100번

이상 연속된 최적값이 나오는 경우 그 값을 최적해로 도출하도록 하며 100번 이상 연속된 최적값이 나오지 않는 경우는 1000번의 알고리즘 수행 시 나온 값 중 가장 좋은 값을 해로 선택하도록 하였다.

4.7 실제 구현 시 진행 절차

앞선 최적해 모형을 통해 풀이한 예제를 유전 알고리즘을 이용하여 최적값을 구해보았다. 입력하는 데이터는 아래와 같이 동일 날짜에 도착하는 배들을 시간 순으로 오름차순으로 입력한다. 이 실험에서는 앞선 3.2절과 예제에서와 같이 배들의 대한 데이터 및 제약 조건들을 동일하게 설정하였다.

Start Year	Start Month	Start Day	Start Time	End Year	End Month	End Day	End Time	SIZE	Container	CC1	HT1	CC2	HT2	CC3	HT3	Name
2009	9	1	3	2009	9	2	2	40	321	2	5	3	4	4	3	1
2009	9	1	2	2009	9	2	2	30	322	2	5	3	4	4	3	2
2009	9	1	4	2009	9	2	21	30	415	3	5	4	4	5	3	3
2009	9	1	3	2009	9	2	23	30	162	1	5	2	3	3	2	4
2009	9	1	1	2009	9	2	7	50	534	3	6	4	5	5	4	5
2009	9	1	5	2009	9	2	6	40	251	2	4	3	3	4	2	6
2009	9	1	3	2009	9	2	2	50	252	2	4	3	3	4	2	7
2009	9	1	1	2009	9	2	20	30	384	2	6	3	5	4	3	8

위의 데이터에서 크레인의 할당 수에 대해 랜덤실수를 생성 시 이와 같이 24개의 랜덤실수(0.31, 0.45, 0.27 / 0.62, 0.35, 0.59 / 0.30, 0.91, 0.81 / 0.52, 0.61, 0.74 / 0.37, 0.44, 0.52 / 0.49, 0.55, 0.93 / 0.43, 0.51, 0.70 / 0.22, 0.34, 0.90)가 생성되며 이 후 아래와 같은 단계를 거친다.

1. 각 배에 대해 생성된 3개의 실수 중 가장 높은 수를 가지는 실수를 선택한다. (0.45, 0.31, 0.27 / 0.62, 0.35, 0.59 / 0.30, 0.91, 0.81

/ 0.52, 0.74 , 0.61, / 0.37, 0.52, 0.44, / 0.49, 0.93 0.55,/ 0.43, 0.51, 0.70 / 0.90 ,0.22, 0.34,) 각 숫자들의 위치는 배들에 할당된 컨테이너 크레인의 수를 뜻하게 된다. 즉 1번 배는 1번 째 컨테이너 크레인 할당 대안을, 3번째 배는 2번째 대안을 선택하게 되며 이와 같은 방법으로 마지막 배까지 선택하도록 한다. 이와 같은 식으로 선박들은 각각 1, 1, 2, 2, 2, 2 ,3 ,1 대의 컨테이너 크레인을 초기해로 할당 받았다.

2. 8대의 배에 대해 도착시간을 백분율로 나타내고 비율이 낮은 값들을 우선 배치하였다. (선박 별 비율 : 14.3, 9.5, 19, 9.5, 4.8, 23.8, 14.3, 4.8) 이 때 같은 값의 경우 랜덤으로 우선 배치하도록 하였고 한 시점에서 9대의 크레인 이상을 할당하지 않도록 하였다. 첫 번째 배치 대상 배는 크레인 할당 대안이 1번째 일 경우로 배에 할당되는 크레인의 수는 2대로 총 처리시간은 5시간이 걸린다. 이때 배의 크기는 40으로 40만큼의 크기를 차지한다. 위와 같은 단계로 아래 [그림 13]과 같이 모든 배에 대해 크레인을 전부 할당하였고 배들의 위치를 배정 하였다.



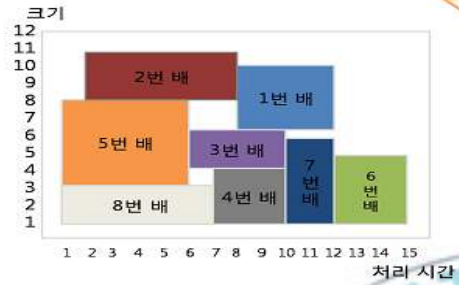
였다.

[그림 13] 8대의 배의 초기해

3. 위와 같이 배치 시 적합도는 66이 나왔고 대기 시간은 5, 8, 2번



배 번호	도착 시간	크레인 수	완료 시간	처리 시간	크기
1	3	2	12	9	4
2	2	2	7	5	3
3	4	4	15	11	3
4	2	2	9	7	3
5	1	4	6	5	5
6	5	3	15	10	4
7	3	4	11	8	5
8	1	2	7	6	3



배 번호	도착 시간	크레인 수	완료 시간	처리 시간	크기
1	3	2	12	9	4
2	2	2	7	5	3
3	4	4	10	6	3
4	2	2	10	8	3
5	1	4	6	5	5
6	5	3	15	10	4
7	3	4	12	9	5
8	1	2	7	6	3

배를 제외한 모든 배에서 대기 시간이 발생하였다.

[그림 14] 대기 시간에 따른 위치변경 예

대기 시간이 가장 긴 선박을 기준으로 대기 시간이 발생한 모든 배들에 대해 위치를 교환하게 한다. 위의 [그림 14]와 같이 대기 시간이 가장 긴 6번배를 가장 짧은 배와 위치를 교환 하였을 경우 5시간이 감소하여 총 61시간이 걸렸다. 이와 같은 과정을 모든 배에 반복하여 그 중 나온 최적값을 저장한다.

3. 앞선 과정들을 통해 정해진 숫자만큼 만들어진 각각의 초기해들에 다시 8대의 배에 대해 24개의 랜덤실수를 발생하도록 한다. 이 후 랜덤으로 초기해 중 2개를 선택하고 둘 중 적합도가 더 좋은 해를 생성된 자식 세대와 정해진 확률에 따라 교배를 수행하도록 하였다. 교배는 선택된 초기해와 재생산된 해를 서로 간에 랜덤으로 임의의 위치를 선택하고 선택된 위치를 교환하여 교배를 수행하도록 하였다. 또한 교배와 마

참가지로 정해진 확률에 따라 돌연변이를 수행하였고 돌연변이는 앞서 설명한 바와 같은 방법으로 만약 1번 배가 돌연변이가 일어날 경우에 0.31, 0.45, 0.27 의 값에서 가장 높은 실수를 가장 낮은 실수와 교환하도록 하였다. (ex 0.31, 0.45, 0.27 → 0.31, 0.27, 0.45) 이와 같이 돌연변이 수행 시에는 원래 크레인 할당 대안에 2번째가 해로 선택되었어야 하지만 돌연변이 수행 시 위치가 바뀜으로 3번째가 해로 선택되었다. 이와 같이 선택, 교배, 돌연변이를 통해 알고리즘을 수행하며 크레인의 대 수를 할당하며 배의 위치는 앞서 설명한 바와 같이 배정하였다.

4. 동일한 조건하에 200번의 작업이 수행되도록 하여 아래의 [그림 15]와 같이 배치 및 크레인 할당 수는 서로 상이하지만 앞서 최적해 모형의 결과와 같이 전체 선박의 작업이 40의 체류시간을 가지는 조합을 찾아내었다.



(a : 최적화 모형 결과) (b : 유전 알고리즘 결과)

[그림 15] 최적화 모형과 유전알고리즘의 실험 결과

제 5 장 실험 분석

이 절에서는 구현된 알고리즘의 효율성을 검증하기 위하여 선석 일정 계획 시 기존에 사용되던 방식과 구현된 알고리즘을 이용한 방식을 비교 분석하였다.

실제 신선대 컨테이너 터미널 경우 선박에 대해 할당된 크레인의 수나 선박의 대기시간 등에 대해 정확한 데이터를 얻기 어려워 정확한 비교 분석이 어렵기 때문에 비교 분석하기엔 적합하지 않다. 따라서 기존의 방식과 구현된 알고리즘을 비교하는 것으로 실험을 비교 분석 하였다.

기존의 방식으로는 배가 들어오는 순서에 따라 선석의 위치를 배정받는 FCFS (First-Come-First-Service)를 따르는 경우이며 구현된 알고리즘은 체류 시간의 최소를 위해 도착 시점 이후에서 배의 위치가 변하는 상황이다. 이 후 컨테이너 크레인의 할당 대안에 대해서는 기존의 할당 방안에 대한 정확한 데이터가 없어 이 실험에서는 할당 대안이 하나로 고정되어 있을 때와 선택 가능한 할당 대안의 경우가 3가지 일 때의 결과를 비교해봄으로 크레인의 할당 대안에 따른 시간 차이를 비교 분석 한다. 실험에 사용된 변수값은 신선대 컨테이너 터미널과 같이 총 선석의 길이는 1500m 이며 컨테이너 크레인은 총 15기가 가동된다고 가정 하였다. 각 실험에서 선박 수는 실제 터미널에서 주 일정계획 시 평균 50대로 본 논문에서도 50대로 지정하였다. 신선대 컨테이너 터미널의 데이터는 시뮬레이션 패키지의 ARENA Input Analyzer 이용하여 분석 하였고 분석 결과 선박의 화물량 및 선박 크기 등은 베타분포를 따른다는 것을 알았다. 실험을 위해 평균 190m의 크기와 850개 정도의 베타 분포를 따르는 데이터를 생성시켰고 선박의 대 수는 실제 컨테이너 터미널의 주간 평균 입항 대수인 50대를 입력하였다. 각각의 실험 데이터들은 아래 [표-2]와 같이 입력된다. 결과를 비교하기 위하여 [표-2]의 크

레인 할당 대안이 하나로 고정된 경우와 선택 가능한 여러 개의 대안이 있을 때를 각각 실험하였다. 또한 컨테이너 터미널의 화물처리량의 차이에 따른 변화를 알아보기 위해 화물처리량에 변화를 주어 결과를 비교하였다.

Arrival Year	Arrival Month	Arrival Day	Arrival Time	demand Year	demand Month	demand Day	demand Time	SIZE	Container	CC1	HT1	CC2	HT2	CC3	HT3	Name
2009	9	1	13	2009	9	2	4	20	1045	2	12	3	12	4	12	NVST-27
2009	9	1	19	2009	9	2	2	31	1530	2	18	3	18	4	18	APCI-6
2009	9	1	2	2009	9	2	21	29	1277	2	15	3	15	4	15	NGLS-36
2009	9	1	22	2009	9	2	23	22	874	2	10	3	10	4	10	MOTA-4
2009	9	1	7	2009	9	2	7	35	1129	2	13	3	13	4	13	XNHR-9
2009	9	1	10	2009	9	2	6	18	1224	2	14	3	14	4	14	HVDK-1
2009	9	1	10	2009	9	3	2	24	580	2	7	3	7	4	7	AKAO-2
2009	9	1	15	2009	9	3	20	18	1294	2	15	3	15	4	15	NBST-54
2009	9	2	12	2009	9	3	20	30	1214	2	14	3	14	4	14	ASPO-12
2009	9	2	9	2009	9	3	21	21	1576	2	18	3	18	4	18	AMAY-3
2009	9	2	16	2009	9	3	7	33	1186	2	14	3	14	4	14	HHLO-4
2009	9	2	21	2009	9	3	7	19	1616	2	18	3	18	4	18	ASCG-71
2009	9	2	20	2009	9	4	7	20	917	2	11	3	11	4	11	ANLA-16
2009	9	2	8	2009	9	4	7	18	637	2	8	3	8	4	8	MUBR-1
2009	9	2	8	2009	9	4	4	18	1264	2	15	3	15	4	15	HSXG-5
2009	9	2	17	2009	9	4	15	32	750	2	9	3	9	4	9	NHST-56
2009	9	3	8	2009	9	4	6	29	775	2	9	3	9	4	9	NCHT-5
2009	9	3	21	2009	9	4	23	21	1356	2	16	3	16	4	16	NNLL-4
2009	9	3	6	2009	9	4	22	16	663	2	8	3	8	4	8	APHI-15
2009	9	3	17	2009	9	4	15	33	819	2	10	3	10	4	10	OOQD-6
2009	9	3	22	2009	9	4	8	21	638	2	8	3	8	4	8	NKST-35
2009	9	3	21	2009	9	5	14	19	648	2	8	3	8	4	8	ESKO-5
2009	9	3	22	2009	9	5	0	21	928	2	11	3	11	4	11	ASNZ-2
2009	9	3	10	2009	9	5	22	32	629	2	7	3	7	4	7	SJFC-1
2009	9	4	21	2009	9	5	0	26	1508	2	17	3	17	4	17	AHBC-36
2009	9	4	15	2009	9	5	22	31	537	2	6	3	6	4	6	VICT-1
2009	9	4	22	2009	9	6	0	32	801	2	9	3	9	4	9	DYDJ-32
2009	9	4	5	2009	9	6	7	30	749	2	9	3	9	4	9	CAPR-2
2009	9	4	8	2009	9	6	0	23	760	2	9	3	9	4	9	NSUX-35
2009	9	4	10	2009	9	6	23	20	1429	2	16	3	16	4	16	OONL-7
2009	9	4	15	2009	9	6	1	16	816	2	10	3	10	4	10	NHYR-15
2009	9	4	16	2009	9	6	10	18	776	2	9	3	9	4	9	OOAT-7
2009	9	5	17	2009	9	6	20	30	1518	2	17	3	17	4	17	NBST-55
2009	9	5	4	2009	9	6	3	29	1352	2	16	3	16	4	16	NPPY-23
2009	9	5	14	2009	9	6	8	24	798	2	9	3	9	4	9	NSMR-6
2009	9	5	21	2009	9	6	9	23	632	2	8	3	8	4	8	DYCS-19
2009	9	5	8	2009	9	6	14	33	1379	2	16	3	16	4	16	NLBS-18

[표 2] 결과 분석을 위한 입력 데이터

이 후 신선대 컨테이너 터미널의 데이터를 분석하고 분석 결과와 동일한 분포를 따르는 데이터를 가진 총 10개의 문제를 생성하고 구현된 알고리즘을 1000번 반복시킨 후의 결과를 아래의 표들에서 나타낸다. 이때 실험에서 앞서 제시한 수리모형의 목적식의 α (선박의 체류 시간에 대한 가중치)와 β (선박의 출발 요구 초과 시간)등은 전부 같은 값으로 입력하였다.

아래 [표-3]과[표-4]는 실제 신선대 터미널의 평균 주간 화물량으로 화물량을 배정하고 실험한 결과이다.

[표-3] FCFS 결과 분석 (신선대 터미널의 현재 주간 평균 화물량)

상황별	크레인 할당 대안이 고정적일 경우				크레인 할당 대안이 변동적으로 선택 가능한 경우			
Scenario	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)
1	5.81	0.21	0.04	0	5.2	0	0	13
2	7.01	0.5	0.04	1	6.31	0	0	14
3	5.46	0.12	0.2	1	5.12	0	0.12	12
4	5.9	0.16	0.08	2	5.53	0.18	0.02	14
5	5.78	0.19	0.16	1	5.41	0	0.08	13
6	6.54	0.12	0.2	1	6.37	0.05	0.1	10
7	5.59	0.2	0.12	1	5.07	0	0	12
8	5.67	0.18	0.08	0	5.15	0	0	11
9	5.71	0.21	0	2	5.32	0	0	11
10	5.74	0.3	0.06	1	5.14	0	0	10
평균	5.921	0.219	0.098	1	5.462	0.023	0.032	12

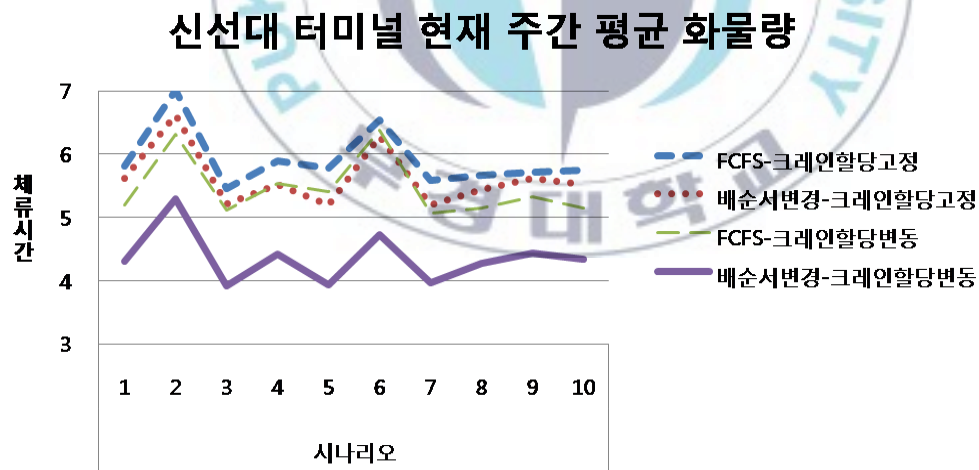
[표-4] 배 할당 순서 변경 시 (신선대 터미널의 현재 주간 평균 화물량)

상황별	크레인 할당 대안이 고정적일 경우				크레인 할당 대안이 변동적으로 선택 가능한 경우			
Scenario	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)
1	5.62	0.14	0.04	0	4.3	0	0	13
2	6.62	0.4	0.04	1	5.3	0	0	14
3	5.22	0.06	0.2	1	3.92	0	0.12	12
4	5.52	0.1	0.08	2	4.42	0.12	0.02	14
5	5.2	0.12	0.16	1	3.94	0	0.02	13
6	6.28	0.08	0.2	1	4.72	0.02	0.1	10
7	5.2	0.16	0.12	1	3.96	0	0	12
8	5.44	0.1	0.08	0	4.28	0	0	11
9	5.62	0.14	0	2	4.44	0	0	11
10	5.52	0.26	0.06	1	4.34	0	0	10
평균	5.624	0.156	0.098	1	4.362	0.014	0.026	12

현재 경제상 컨테이너 터미널의 화물량이 많지 않아서 실제 신선대 컨테이너 터미널의 주간 평균 화물량으로 화물량을 배정하고 일정 계획을 수립 한 결과 전체적인 대기 시간 및 출발 요구 지연 시간 등이 거의 발생 하지 않는 것을 위의 [표-3]과 [표-4]에서 확인 하였다.

각 실험의 결과를 보면 각각 선박 당 체류시간은 배 할당 순서와 크레인 할당 대안이 변동적인 경우를 제외하면 평균적으로 5~6시간이며 배 할당 순서의 변경 및 크레인의 할당 대안이 선택 가능할 경우 선박들의 작업 시간이 1시간 이상 줄어드는 것을 확인 할 수 있다. 또한 선박들의 평균 출발 요구 시간에 비해 짧은 시간 안에 작업이 완료됨을 대기 시간과 초과 시간에서 확인 할 수 있다. 기존의 방안으로 배가 배치되고 크레인 할당 대안이 고정적인 경우엔 평균적으로 0.15~0.21의 값을 가지지만 배의 위치가 바뀌고 크레인 할당 대안이 변동적으로 선택 가능한 경우엔 평균적으로 0.01~0.02의 값으로 대기 시간 및 출발 요구 초과 시간이 거의 발생하지 않음을 확인 하였다.

다음 [그림-16]에서는 상황별 평균 체류 시간의 차이를 그림으로 나타내고 있다. 배 순서 변동에 크레인 할당이 변동적일 경우 가장 낮은 체류시간을 가짐을 확인 할 수 있다.



[그림 16] 각 상황별 결과 비교 (신선대 터미널 현재 주간 평균 화물량)

앞서 비교 분석한 데이터는 화물량이 많아 바쁜 컨테이너 터미널의 상황을 반영할 수 없기 때문에 화물량을 증가시키고 화물 처리량이 많은 상황에서 구현된 알고리즘이 어떤 결과를 나타내는 지에 대해 실험하였다. 이 후 아래 [표-5]와 [표-6]에서 그 결과를 나타내고 있다.

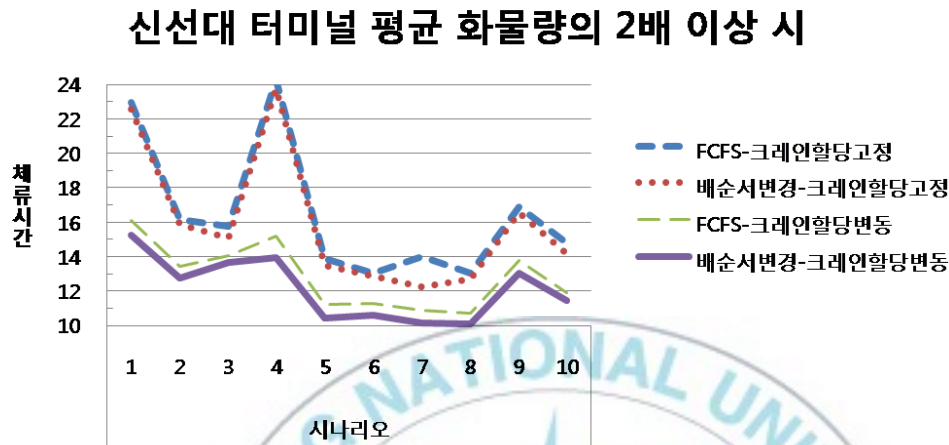
[표-5] FCFS 결과 분석 (신선대 터미널의 평균 화물량의 2배)

상황별 Scenario	크레인 할당 대안이 고정적일 경우				크레인 할당 대안이 변동적으로 선택 가능한 경우			
	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)
1	22.94	9.02	2.04	3	16.1	3.2	1.35	11
2	16.14	3.51	1.78	1	13.45	3.15	0.74	13
3	15.76	2.92	2.54	2	14.06	1.77	0.92	12
4	24.26	10.7	2.99	3	15.17	3.12	0.88	13
5	13.91	2.55	1.65	1	11.21	2.02	0.78	10
6	13.03	1.85	1.97	2	11.3	1.16	1.02	12
7	14.02	1.4	1.75	1	10.9	0.71	1.21	11
8	13.05	2.03	1.78	3	10.7	1.2	0.98	14
9	16.91	4.45	1.54	2	13.78	3.01	0.72	12
10	14.77	2.01	1.32	1	11.91	1.56	1.01	11
평균	16.5	4	1.86	1.9	12.9	2.1	0.96	12

[표-6] 배 할당 순서 변경 시 (신선대 터미널의 평균 화물량의 2배)

상황별 Scenario	크레인 할당 대안이 고정적일 경우				크레인 할당 대안이 변동적으로 선택 가능한 경우			
	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)	선박 당 체류시간	대기 시간	초과 시간	계산 시간(초)
1	22.54	8.68	1.88	3	15.26	3.16	0.72	12
2	15.86	3.08	0.64	1	12.76	2.12	0.34	15
3	15.06	2.58	0.42	2	13.66	1.56	0.18	13
4	23.76	10.12	2.04	3	13.92	2.4	0.48	15
5	13.5	2.08	0.64	1	10.46	1.32	0.48	13
6	12.88	1.44	0.94	2	10.58	1.1	0.42	10
7	12.26	0.7	0.54	1	10.14	0.66	0.38	12
8	12.7	1.64	0.6	3	10.1	1.22	0.32	11
9	16.54	3.78	0.86	2	13.02	2.02	0.34	10
10	14.18	1.76	0.5	1	11.48	1.48	0.18	11
평균	15.9	3.586	0.906	1.9	12.1	1.704	0.384	12.2

위의 [표-5]와 [표-6]에서의 결과를 보면 화물량 증가 시에도 선박의 위치와 크레인의 할당 대안을 변동적인 경우가 기존의 방식과 크레인의 할당 대안이 고정적인 경우에 비해 더 효율적임을 확인 할 수 있다. 아래 [그림-17]에서와 같이 각 상황별 평균 체류 시간에 대해 그림으로 나타내고 있다.



[그림 17] 신선대 터미널 평균 화물량의 2배 이상 시

위의 [그림-17]에서와 같이 각 상황별 평균에서 배 할당 순서의 변경과 크레인 할당 대안이 변동적인 경우가 각 시나리오 별 가장 낮은 체류 시간인 평균 12.1의 체류 시간을 가졌다. 또한 크레인의 할당에 따른 체류 시간의 차이에서도 FCFS 및 배의 할당 순서가 변경될 때도 크레인의 할당 대안이 변동적인 경우가 전체적으로 선박의 체류 시간을 3시간 정도 줄여 주는 것을 확인 할 수 있다. 위의 실험들을 통해 선박의 작업 순서와 크레인 할당 대수에 선택의 폭이 커질수록 보다 빠른 처리 시간을 가지는 조합을 찾을 수 있다는 걸 알 수 있다.

아래 [그림 18]에서는 실제 신선대 부두에서 일정 계획을 화면상에서 나타낸 것과 본 알고리즘을 구현 한 뒤 나타난 결과를 확인할 수 있

는데 실제 일정계획에 비해 본 알고리즘에서 나타난 결과 배치는 전체적으로 한쪽으로 기울어져 배치되어진 모습을 볼 수 있다.



[그림 18] 신선대 컨테이너 터미널과 구현된 알고리즘의 일정계획

하지만 실제 신선대에서 쓰이는 일정계획 또한 담당자가 수동으로 배치한다는 것을 고려했을 때 자동 배치 측면에서 알고리즘이 충분히 활용적임을 보여주고 있다. 또한 실제 수동으로 작업 작성 시 걸리는 시간은 시간 단위이지만 알고리즘 수행시간이 10~20초로 차후 작업자의 수동 보정 시간을 합하더라도 많은 시간이 단축된다.

각 실험의 테스트 환경은 윈도우 XP, AMD Athlon(tm) 64 X2 Dual Core Processor 4200+ CPU와 2.00GB 메모리의 컴퓨터 환경에서 실험하였다. 앞서 제시한 수리모형으로 최적화 소프트웨어를 통해 해를 도출할 경우 선박의 수가 늘어날수록 해의 도출에 필요한 시간이 급격하게 늘어나며 11개를 넘어설 경우는 해를 도출하지 못 하였다.

본 논문에서 제시한 알고리즘 및 프로그램 구현은 C++, Flash MX 2004를 이용하여 구현하여 실제 컨테이너 터미널에서 쓰이는 프로그램과 같이 유전 알고리즘을 통해 배치된 결과 값을 실시간으로 화면상으로 나타내도록 하였다.



제 6장 결 론

해외 수출입에 의존적인 우리나라 경제구조에서 수출입 물류 산업 역할을 수행하는 컨테이너항만 하역 산업의 발전은 국가 경제성장의 바탕이 된다. 현재 우리나라 컨테이너항만은 중국 경제의 급격한 성장과정에서 발생 하는 대규모의 컨테이너 화물들을 처리 하면서 새로운 부가가치 및 고용을 창출하는 경제성장의 역할도 담당하고 있다. 이처럼 컨테이너항만의 생산성 증대는 곧 국가 경제 성장과 연관된다고 할 수 있다. 최근 전세계적으로 부두가 동시 다발적으로 개발되고 있는 현 시점에서 컨테이너 터미널의 경쟁력 강화를 위해서는 항만 자원의 효율 극대화를 통한 최적 일정계획이 관건이다. 컨테이너 터미널에서 일정계획은 선박의 입/출항 시간을 기초로 하여, 선박을 어느 선석에 언제, 몇 대의 크레인을 할당할 것인가를 결정하는 상세한 시공간적 계획을 의미한다. 컨테이너항만에서의 일정계획은 배의 입/출항 시간을 기초로 하여 계획하는데 이러한 일정계획 단계에서 배의 입/출항의 예외 상황 발생의 빈도가 잦아 계획 수정이 이루어져야 하는 경우가 많다. 이러한 계획의 수정을 매번 수작업으로 수정하는 것은 많은 무리가 따르곤 한다. 본 연구에서는 일정계획에 중요한 결정 요소인 크레인 대수의 제한과 할당 대안의 변화를 배와 함께 고려하는 수리 모형을 제시하고 이 수리 모형의 최적해를 최적화 프로그램의 일종인 LINGO 소프트웨어를 이용하여 도출 하였다. 그리고 최적해의 성질을 근간으로 일정계획을 효율적으로 찾을 수 있는 유전 알고리즘을 제안하고 예제를 통해 알고리즘의 적용절차를 설명 하였다. 또한 시뮬레이션 실험을 통해 제안된 유전자 알고리즘의 효율성을 검증 하였다. 실험에서 문제가 커질수록 휴리스틱 알고리즘에 비해 최적화 모형을 이용해 LINGO에서 최적해를 구하기 위해 많은 계산 시간이 요구됨을 알 수 있었다. 실제 부산 신선대 컨테이너 터미널의 데이터를 이용하여 성능 실험을 할 수 있었으면 더욱 정확한 비교 결과를 할 수 있었지만 실험을 위한 모든 데이터를 정확히 알 수가 없으므로 실제 데

이터와 비교분석은 하지 못하였지만 여러 가지 현실과 같은 상황으로 비교 분석을 해본 결과 기존의 결과에 비해서 확실히 개선이 이루어짐을 확인 하였다. 면적 활용률을 비롯하여 제시된 알고리즘을 통한 자동 배치는 최적화 소프트웨어와 비교 하여 짧은 수행 시간으로 충분히 자동 일정 계획 시스템에 적용될 수 있으리라 생각된다. 향후 연구과제로는 실제 컨테이너 터미널에서 발생하는 여러 가지 예외 상황 및 제약 조건들 (선박의 우선순위, 선호 위치, 야드 작업, 항만의 수심) 등을 고려하여 보다 현실적인 연구의 확장이 필요하며, 향후 실제 도출된 결과를 수동으로 재배치 가능하게 하여 보다 원활하게 일정 계획을 개선 할 수 있도록 해야 할 것이다.



참 고 문 헌

- [1] Guan, Y., Xiao, W.-Q., Cheung, R.K., Li, C.-L., A multiprocessor task scheduling model for berth allocation heuristic and worst-case analysis. Operations Research Letters 30, 2002, 343-350.
- [2] Imai, A., Nagaiwa, K., Chan, W.T., Efficient planning of berth allocation for container terminals in Asia. Journal of Advanced Transportation 31, 1997, 75-94.
- [3] Imai, A., Nishimura, E., Papadimitriou, S., The dynamic berth allocation problem for a container port. Transportation Research Part B 35, 2001, 401-417.
- [4] Imai, A., Nishimura, E., Papadimitriou, S., Berth allocation with service priority. Transportation Research Part B 37, 2003, 437-457.
- [5] Imai, A., Nishimura, E., Papadimitriou, S. "The dynamic berth allocation for a container port" Transportation Research Part B35 . Transportation Research Part B, , doi: 2004. 03.004
- [6] Kim, K.H., Moon, K.C. Berth scheduling by simulated annealing. Transportation Research Part B 37,2003, 541-60.
- [7] Lai, K.K., Shih, K., A study of container berth allocation.

Journal of advanced Transportation 26, 1992, 45-0.

- [8] Lim, A., The berth planning problem. Operations Research Letters 22, 1998, 105-10.
- [9] Nishimura, E., Imai, A., Papadimitriou, S., Berth allocation planning in the public berth system by genetic algorithms. European Journal of Operational Research 131, 2001, 282-92.
- [10] Park, K.T., Kim, K.H., Berth scheduling for container terminals by using a sub-gradient optimization, Journal of the Operational Research Society 53, 2002 1054-062.
- [11] Park, Y.-M., Kim K.H., A scheduling method for berth and quay cranes. OR Spectrum 2003 25, 1-3.
- [12] Imai, Hsieh, Nishimura, Papadimitriou. The simultaneous berth and quay crane allocation problem. Transportation Research Part E 44 2008 900-20
- [13] Eliyi, Özçam, Ada, Güldogan, Sevil, Isık Ö. Yumurtacı .Simultaneous allocation of continuous berths and mobile with variables hqndling times International Conference 20th EURO Mini Conference “continuous Optimization and Knowledge -Based Technologies” EurOPT-2008 20-3,
- [14] Imai. Xin , Nishimura, Papadimitriou. Berth allocation in a container port: using a continuous location space approach.

- [15] 이홍걸, 이철영. 발견적 알고리즘에 의한 컨테이너 선석배정에 관한 연구. 한국항만학회지 (Journal of Korean Port Research) 1995, 9-2.
- [16] 홍동희. 시뮬레이션과 유전자 알고리즘을 이용한 선석 계획과 야드의 통합. 해운물류연구(구 한국해운학회지) , 2005, 44, 77-93.
- [17] 김홍배, 양성민, 송경동, 이혜경. 유전자 알고리즘을 이용한 container crane 스케줄링에 관한 연구. 대한산업공학회 학술대회 논문집 1984 A 08.3.
- [18] 류광렬, 김갑환, 백영수, 황준하, 박영만. 제약만족 탐색과 휴리스틱 교정기법을 이용한 최적 선석 및 크레인 일정 계획. 한국지능정보시스템학회지논문집 2000, 6-2.
- [19] 윤영철, 문성혁. 컨테이너 터미널 사용자비용을 최소로 하는 선석과 크레인의 최적 구성에 관한 연구. 한국항만학회지 . 1995,9-2. 3~9.

The berth allocation problem with quay crane in container terminal

Won Bong Lee

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

Abstract

This study considers how to assign berth and crane in a container terminal to a ship for more efficient schedule. This study is a constraint satisfaction problem in which cranes and berths should be assigned in such a way all the space and time constraints are satisfied without any interference. First we made the mathematical model for the decision making. It includes the three CC (Container crane) assignment options for determining which one is more efficient allocation. Then, using LINGO software, the optimal solution for the mathematical model can be found. A numerical example is given to verify the model and its solution. Furthermore we developed a GA (Genetic Algorithms). And we compared the results of GA with LINGO optimal solutions using a real data from Pusan East Container Terminal and it showed that our study can derive a schedule of satisfactory quality.

감사의 글

대학원에 뜻을 담고 입학한지가 엇그제 같은데 벌써 2년이라는 시간이 훌쩍 지나버렸습니다. 새로운 학교에서의 생활과 학부 때와는 다른 전공으로 인해 많은 어려움이 있었지만 여러분들의 도움으로 지금 이 글을 쓸 수 있게 되었습니다. 졸업을 앞두고 지난 일들을 돌이켜보니 제 자신이 부끄러울 정도로 그 동안 대학원 생활에 충실하지 못했던 것 같아 후회와 아쉬움이 가득합니다.

제가 가장 먼저 감사의 말씀을 드리고 싶은 분이 계십니다. 지금의 제가 졸업을 할 수 있게 되기까지 2년간 항상 많은 관심과 가르침을 주신 구평희 교수님입니다. 학부 때 전공과는 틀려 적응하기에 많은 어려움이 있었지만 교수님의 따뜻한 보살핌에 여기까지 올 수 있었습니다. 교수님께는 어떻게 감사의 인사를 드려야 할지 모르겠습니다. 그동안 교수님께 배웠던 가르침을 맘속 깊이 새기며 꼭 실천 하도록 노력하겠습니다. 부족한 저를 2년 동안이나 보살피 주셔서 너무 감사드립니다. 교수님, 항상 건강 조심하시고 하시는 일마다 좋은 결과 있으시길 빌겠습니다.

바쁘신 와중에도 한없이 부족한 제 논문을 심사해주시고 모자란 부분을 지적해 주셨던 고시근 교수님, 김영진 교수님께 고개 숙여 감사드립니다. 또한 처음 대학원에 들어왔을 때 모든 것이 생소하고 서툴렀던 제가 쉽게 적응할 수 있도록 많은 가르침을 주셨던 옥영석 교수님, 단순히 이론이 아닌 실전 경험을 쌓도록 지도해주신 권혁무 교수님, 그리고 대학원 수업을 통해 항상 많은 가르침을 주실려고 노력하셨던 이운식 교수님, 고시근 교수님, 수업은 없었지만 타교에서 온 절 따듯이 맞아주셨던 여러 교수님들에게 정말 감사드립니다. 교수님들의 가르침, 잊지 않고 사회에서도 열심히 노력하겠습니다.

그그리고 이제까지 항상 절 믿어주시고 바른 길로 이끌어주셨던 부모님 그 동안의 은혜에 정말 감사드리며 앞으로 보다 자랑스러운 아들이 되도록 노력하겠습니다. 또한 항상 어려울 때나 힘들 때 많은 도움을 주던 나원이누나, 진희, 대경이, 강욱이, 병주를 비롯한 여러 소중한 친구들과 해준 것도 없이 늘 부탁만 하는 제게 항상 웃는 얼굴로 따뜻히 맞이

해주던 학사 사람들과 2년이란 시간동안 가장 많은 시간을 함께했던 대학원 식구들, 언제나 따뜻한 관심을 주시며 귀여워해주셨던 진아누님, 지금은 졸업하고 안 계시지만 함께 한 1년이란 시간동안 자상히 이끌어 주셨던 정희 선배, 비록 나보다 늦게 들어왔지만 그 열정과 노력으로 항상 나보다 선배 같았던 현애누나에게도 감사의 맘을 전합니다. 특히 같은 대학원 생활을 하면서 힘든 시기나 즐거웠던 시기나 항상 함께 하며 더더욱 정이 들었고 서로가 있어 힘이 되었던 현애누나. 그간 누나의 보살핌에 정말 감사드립니다. 2년이란 시간 동안 당신들과 함께했던 했던 시간들이 길었던 만큼 정도 많이 들었습니다.

가끔 티격태격 싸우기도 했지만 그만큼 미운 정도 많이 들었기에 더더욱 가족 같은 우리 대학원 식구들, 평생 잊지 못할 추억을 간직하고 떠납니다. 모두 몸 건강히 꼭 성공하시길 빌겠습니다. 감사합니다.

이원봉 드림

