

## 골리앗 크레인의 공주행 거리와 와이어 교체 최소를 고려한 최적 블록 리프팅 계획

노명일\*, 이규열\*\*

### Optimal Block Lifting Scheduling Considering the Minimization of Travel Distance at an Idle State and Wire Replacement of a Goliath Crane

Myung-Il Roh\* and Kyu-Yeul Lee\*\*

#### ABSTRACT

Recently, a shipyard is making every effort to efficiently manage equipments of resources such as a gantry crane, transporter, and so on. So far block lifting scheduling of a gantry crane has been manually performed by a manager of the shipyard, and thus it took much time to get scheduling results and moreover the quality of them was not optimal. To improve this, a block lifting scheduling system of the gantry crane using optimization techniques was developed in this study. First, a block lifting scheduling problem was mathematically formulated as a multi-objective optimization problem, considering the minimization of travel distance at an idle state and wire replacement during block lifting. Then, to solve the problem, a meta-heuristic optimization algorithm based on the genetic algorithm was proposed. To evaluate the efficiency and applicability of the developed system, it was applied to an actual block lifting scheduling problem of the shipyard. The result shows that blocks can be efficiently lifted by the gantry crane using the developed system, compared to manual scheduling by a manager.

**Key words:** Block lifting scheduling, Gantry crane, Meta-heuristic optimization algorithm, Genetic algorithm, Multi-objective optimization problem, Production planning, Shipbuilding

## 1. 서 론

### 1.1 연구 배경 및 필요성

선박과 같은 거대 구조물은 자동차와 같이 한 번에 제작할 수 없다. 따라서 다수의 작은 블록들로 먼저 분할하여 각각 조립하고, 이 블록들을 다시 여러 개씩 합쳐 보다 큰 블록들을 만들게 된다. 그 후 도크(dock)라 불리는 작업 공간 내에서 이 블록들을 다시 일정한 순서로 탑재(erection, 블록들을 도크 내에서 크레인을 이용, 서로 조립 또는 용접하는 것)하여 하나의 선박을 완성하게 된다. 즉, 선박의 건조 과정은 마치 레고 블록을 조립하여 하나의 큰 제품을 만드는

것이라고 볼 수 있다.

각 조선소의 가장 중요한 자원은 도크와 블록을 탑재하기 위한 크레인이라 할 수 있다. 일반적으로 하나의 도크 내에는 동시에 여러 선박이 함께 건조되기 때문에 하나의 선박을 건조할 때 크레인이 블록을 들지 않고 이동하는 거리인 공주행 거리가 길어지면 그만큼 해당 선박의 도크 점유 시간은 증가하게 되어 조선소의 생산성에 막대한 영향을 끼친다. 도크 내에서 블록을 탑재할 때 일반적으로 이용되는 크레인은 갠트리 크레인이다. 갠트리 크레인은 문 또는 다리 모양의 크레인으로서 거더(girder)의 양끝에 다리를 설치하고, 지상에 설치한 레일 위를 주행하도록 한 것이다. Fig. 1은 선박 건조를 위해 일반적으로 사용되고 있는 갠트리 크레인의 예이다.

갠트리 크레인은 도크 주변의 PE(Pre-Erection)장에 임시로 놓여져 있는 블록들을 하나씩 들어 도크 내의 해당 위치로 이동 후 블록을 내려 놓게 된다. 이때 블

\*교신저자, 정회원, 울산대학교 조선해양공학부

\*\*중신회원, 서울대학교 조선해양학과

- 논문투고일: 2009. 08. 23

- 논문수정일: 2009. 11. 02

- 심사완료일: 2009. 11. 18

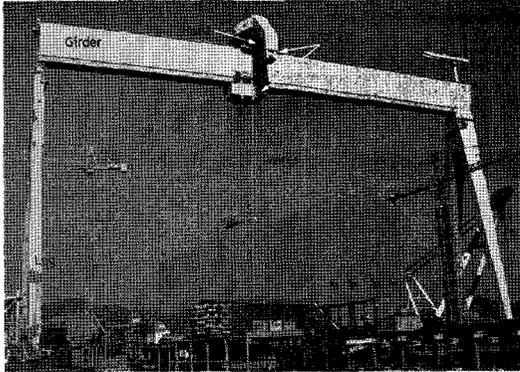


Fig. 1. Example of a gantry crane for erecting ship blocks.

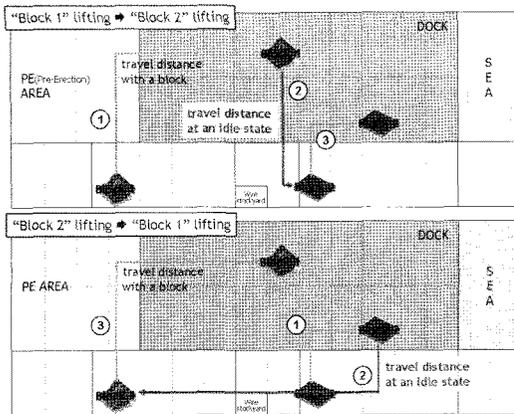


Fig. 2. Example of a travel distance being at an idle state of a gantry crane according to the block lifting sequence.

물들을 어떤 순서로 들어서 내려 놓는가에 따라 공주행 거리가 달라진다. Fig. 2는 리프팅 순서에 따른 공주행 거리의 차이를 보여주는 예이다. 그림에 나타나 있듯이 블록 1을 운반한 후 블록 2를 운반할 경우, 그렇지 않은 경우에 비해 공주행 거리가 짧음을 알 수 있다. 따라서 도크 내에서 건조되는 선박의 도크 점유 시간 최소화화를 위해서는 갠트리 크레인에 의한 블록 리프팅 순서 관리가 무엇보다도 중요하다. 비효율적인 갠트리 크레인의 운영으로 인한 작업 시간의 증가는 전체적인 생산 공정의 지연을 초래할 수 있기 때문이다.

지금까지 조선소 내의 갠트리 크레인을 체계적으로 관리하기가 어려웠다. 갠트리 크레인의 운영 담당자는 하루 동안 탑재해야 할 블록들의 리프팅 순서를 자선에 경험에 의해 결정해 왔다. 그러나 하루에 수십

개의 블록들이 탑재되는 환경에서 갠트리 크레인의 운영 담당자가 리프팅 계획을 효율적으로 처리하는 데에는 어려움이 있다. 만일 이러한 과정을 지원하는 전산 시스템이 있다면, 담당자는 보다 짧은 시간 내에 많은 내안을 검토해 봄으로써 최적의 블록 리프팅 계획을 얻을 수 있을 것이다.

## 1.2 관련 연구 현황

조선소 내의 효율적인 갠트리 크레인 운영을 통해 생산성 향상과 생산 비용을 절감할 수 있음에도 불구하고 지금까지 이와 관련된 연구는 거의 수행되지 않았다. 하지만 갠트리 크레인의 블록 리프팅 계획 문제와 관련된 연구로서, 컨테이너 터미널에서의 크레인 운용 최적화에 대한 기존의 여러 가지 연구가 제시되어 왔다. Daganzo<sup>[12]</sup>, Peterkofsky와 Daganzo<sup>[13]</sup>는 안벽 크레인(quay crane)의 최적 운용 계획과 관련하여 휴리스틱 알고리즘을 개발하였으며, Kim과 Kim<sup>[14]</sup>은 주어진 작업 일정 및 컨테이너 적재 상태 하에서 트랜스퍼 크레인(transfer crane)의 최적 운용을 위한 알고리즘을 개발하였고, Ng와 Mak<sup>[15]</sup>은 서로 다른 작업 준비 시간을 가지는 양, 적하 작업에서 트랜스퍼 크레인의 최적 운용 알고리즘을 개발하였다. 그 외에 Kim과 Park<sup>[16]</sup>은 야드 베이(yard bay)에서의 컨테이너 적하시 컨테이너 재배치의 수를 최소화하기 위한 컨테이너의 최적 저장 위치를 결정하는 알고리즘을 개발하였다. 또한 김후곤 등<sup>[7]</sup>은 컨테이너 터미널에서 트랜스퍼 크레인의 야드 베이 이동 시간과 야드 베이에서의 작업 준비 시간을 최소화하는 최적 라우팅 알고리즘을 제안하였다. 즉, 장치장(berth)에 적재되어 있는 컨테이너들을 주어진 작업 일정에 따라 옮기기 위해 트랜스퍼 크레인이 방문해야 할 야드 베이의 최적 방문 순서와 야드 베이에서 운반할 최적의 컨테이너 수를 결정하는 알고리즘을 제안하였다.

그러나 이들 알고리즘을 조선소에서의 선박 건조를 위해 이용하는 갠트리 크레인에 그대로 적용하기에는 한계가 있다. 먼저 갠트리 크레인은 상대적으로 넓은 범위를 자유롭게 움직일 수 있는 안벽 크레인, 트랜스퍼 크레인과는 달리 도크 주위에 설치된 레일을 따라 제한적으로만 움직일 수 있기 때문에 이동 방식이 다르다. 또한 컨테이너 터미널에서 이용되는 안벽 크레인, 트랜스퍼 크레인 등의 경우, 일정한 크기와 중량을 가지는 규격품인 컨테이너만을 운반하기 때문에 운반 도중에 크레인의 와이어(wire, 크레인과 블록을 연결), 샤클(shackle, 와이어와 블록을 연결) 등을 교체할 필요가 없다. 하지만 조선소의 갠트리 크레인의 경

우 서로 다른 크기와 중량을 가지는 블록들을 운반해야 하므로 크레인의 와이어, 사륜 등의 교체가 빈번히 이루어지면 이것이 리프팅 계획 사 반드시 고려되어야 한다.

따라서 본 연구에서는 기존 연구들이 가진 한계를 극복하기 위해, 최적화 기법을 기반으로 한 블록 리프팅 계획 시스템을 개발하였다. 이를 위해 블록 리프팅 계획 문제를 최적화 문제로 정식화 하였고, 이를 풀기 위한 최적화 알고리즘을 제안하였다. 그리고 나서 개발된 시스템의 효율성과 효용성을 검증하기 위해 이를 조선소의 실제 블록 리프팅 계획 문제에 적용하였다.

## 2. 최적 블록 리프팅 계획 문제

### 2.1 블록 리프팅 계획 문제

본 연구에서의 블록 리프팅 계획 문제는 갠트리 크레인(이하 크레인)에 의한 블록의 최적 리프팅 순서를 결정하는 것이다. 목적 함수는 각 블록의 탑재 시간 제한, 탑재 우선 순위(탑재 네트워크 상에 명시된 탑재 순서) 등을 만족하면서 크레인의 공주행 거리와 와이어(사륜 포함) 교체 회수를 최소화하여 전체 블록 리프팅 시간을 최소화하는 것이다. 블록 리프팅 계획 시 크레인 운영자에 의해 주어지는 입력 정보는 다음과 같다.

#### (1) 크레인 정보

- 크레인의 사양(최대 리프팅 중량, 이동 속도, 운영 시간 등)
- 크레인의 초기 위치

#### (2) 블록 정보

- 전체 리프팅 블록의 수, 크레인에 의해 리프팅될 각 블록의 ID
- 각 블록의 중량
- 각 블록의 리프팅 시 소요 시간
- 각 블록의 이동 전 현재 위치와 이동 후 목표 위치
- 각 블록의 리프팅 시간 제한(리프팅 가능 시간(lower bound), 리프팅 예정 시간(upper bound))
- 블록들 사이의 리프팅 우선 순위
- 각 블록의 리프팅을 위한 와이어 및 사륜의 개수 및 사양

#### (3) 기타 정보

- 도크 정보(도크 배치, 지면 정보 등)

- 와이어 적치장의 위치
- 작업 시간 정보(전체 작업 완료 시간, 크레인 유휴 시간 등)

### 2.2 블록 리프팅 계획 문제의 수학적 정식화

본 연구에서는 블록 리프팅 계획 문제의 정식화를 위해 다음과 같은 변수와 기호가 사용된다.

$N$  : 블록들의 총 수

$D_i$  :  $i$ 번째 블록의 현재 위치에서 목표 위치까지의 주행 거리

$d_{i,i+1}$  :  $i$ 번째 블록에서 다음( $i+1$ 번째) 블록까지의 공주행 거리

$d_{i,w}$  : 와이어 교체를 위해  $i$ 번째 블록에서 와이어 적치장까지의 공주행 거리

$d_{w,i+1}$  : 와이어 교체를 마친 후 와이어 적치장에서  $i+1$ 번째 블록까지의 공주행 거리

$T_i$  :  $i$ 번째 블록의 현재 위치에서 목표 위치까지의 주행 시간

$t_{i,i+1}$  :  $i$ 번째 블록에서 다음( $i+1$ 번째) 블록까지의 공주행 시간

$t_{i,w}$  : 와이어 교체를 위해  $i$ 번째 블록에서 와이어 적치장까지의 공주행 시간

$t_{w,i+1}$  : 와이어 교체를 마친 후 와이어 적치장에서  $i+1$ 번째 블록까지의 공주행 시간

$r_{i,i+1}$  :  $i$ 번째 블록에서  $i+1$ 번째 블록으로 이동 시 와이어 교체 여부(교체 불필요=0, 교체 필요: 1)

$V_x$  : 크레인의 도크 길이 방향 이동 속도

$V_y$  : 크레인의 도크 폭 방향 이동 속도

$T_c$  : 와이어 교체를 위해 소요되는 단위 작업 시간

$s_i$  :  $i$ 번째 블록의 리프팅 시작 시간

$f_i$  :  $i$ 번째 블록의 리프팅 완료 시간

$f_N$  : 전체 작업 완료 시간

$l_i$  :  $i$ 번째 블록의 리프팅 가능 시간(리프팅 시간 제한의 lower bound)

$u_i$  :  $i$ 번째 블록의 리프팅 완료 예정 시간(리프팅 시간 제한의 upper bound)

$p_j$  : 높은 우선 순위를 가지는 블록  $j$ 의 리프팅 시작 시간

$p_k$  : 낮은 우선 순위를 가지는 블록  $k$ 의 리프팅 시작 시간

$T_c$  : 전체 작업 완료 시간

$a$  and  $b$  : Weighting factors

$R_w$  : Penalty coefficient

이제 2.1점에서 서술된 블록 리프팅 계획 문제를 수

학적으로 정식화하면 다음과 같다.

Minimize

$$F_1 = \sum_{i=0}^{N-1} \{(1-r_{i,i+1}) \cdot t_{i,i+1} + r_{i,i+1} \cdot (t_{i,W} + t_{W,i+1})\}$$

; Total travel time of the crane being at an idle state (1)

and

Minimize

$$F_2 = \sum_{i=0}^{N-1} (r_{i,i+1} \cdot T_r)$$

; Total time for wire replacement of the crane (2)

Subject to

$$g_1 = t_i - s_i \leq 0 \quad (3)$$

$$g_2 = f_i - u_i \leq 0 \quad (4)$$

$$g_3 = p_j - p_k \leq 0 \quad (5)$$

$$g_4 = f_N - T_o \leq 0 \quad (6)$$

$$i = 0, \dots, N-1 \text{ and } j, k = 0, \dots, N$$

위 정식화에서, 식 (3)은 블록의 리프팅 시작 시간( $s_i$ )이 리프팅 가능 시간( $t_i$ , 리프팅 시간 제한의 lower bound) 이후여야 함을 나타낸다. 식 (4)는 블록의 리프팅 완료 시간( $f_i$ )이 리프팅 완료 예정 시간( $u_i$ , 리프팅 시간 제한의 upper bound) 이전이어야 함을 나타낸다. 식 (5)는 리프팅 우선 순위가 높은 블록이 우선 순위가 낮은 블록보다 먼저 리프팅되어야 함을 나타낸다. 여기서 리프팅 우선 순위는 각 선박의 탑재 네트워크에 명시된 각 블록의 탑재 순서를 의미한다. 식 (6)은 모든 블록들의 리프팅 작업(마지막 블록의 리프팅 완료 시간)이 조선소의 전체 작업 완료 시간 내에 완료되어야 함을 나타낸다.

식 (1)-(6)으로 표현된 문제는 두 개의 목적 함수( $F_1, F_2$ )를 가지기 때문에 이것은 다목적 최적화 문제이다. 하지만 이 문제는 Weighting 방법<sup>[6]</sup>에 의해 단일 목적 최적화 문제로 변환이 가능하다. 식 (1), (2)의 목적 함수에 weighting factors( $\alpha$  and  $\beta$ )를 적용하면 다음과 같은 식을 얻을 수 있다.

Minimize

$$F = \alpha \cdot F_1 + \beta \cdot F_2$$

$$= \alpha \cdot \sum_{i=0}^{N-1} \{(1-r_{i,i+1}) \cdot t_{i,i+1} + r_{i,i+1} \cdot (t_{i,W} + t_{W,i+1})\} + \beta \cdot \sum_{i=0}^{N-1} (r_{i,i+1} \cdot T_r) \quad (7)$$

Weighting factors  $\alpha$ 와  $\beta$ 는 크레인의 전체 공주행 시간과 와이어 교체를 위한 전체 작업 시간 사이의 일종의 트레이드오프에 해당한다. 따라서  $\alpha$ 와  $\beta$ 의 비율에 따라서 최적화 결과가 달라질 수 있다. 본 연구에서는 반복적인 시행 착오를 거쳐서 두 개의 목적 함수가 동일한 영향을 갖도록 하는  $\alpha$ 와  $\beta$ 의 값(예,  $\alpha = 0.67, \beta = 0.33$ )을 구하여 최적화에 이용하였다. 블록 리프팅 계획 문제에서의 이들의 영향은 3.5절에 상세히 설명하였다.

Penalty function 방법<sup>[9],[10]</sup>을 이용하면, 식 (3)-(7)로 표현된 제약 최적화 문제를 다음과 같이 비제약 최적화 문제로 변환할 수 있다. Penalty function 방법은 제약 조건을 위배하는 해들에 대해 벌점(penalty)을 부과함으로써 이들이 최적해가 되는 것을 막는 방법이다. 식 (8)에 나타나 있듯이 각 제약 조건의 위배 정도에 벌점 상수(penalty coefficient)를 곱한 벌점 항을 기존 목적 함수에 더하게 되면 제약 최적화 문제가 간단히 미제약 최적화 문제로 변환될 수 있다. 즉, 주어진 해가 모든 제약 조건들을 만족할 경우( $g_u \leq 0, u = 1, 2, 3, 4$ ) 벌점 항은 0이 되어 주어진 해의 목적 함수 값은 변화 없으며, 제약 조건을 하나라도 위배할 경우( $g_u > 0, u = 1, 2, 3, 4$ ) 그에 대응되는 양수의 벌점 항이 추가되어 원래의 목적 함수 값보다 더 큰 값을 가지게 되고 결국 이 해는 최적해로 선택될 수 없게 된다.

Minimize

$$F' = \alpha \cdot \sum_{i=0}^{N-1} \{(1-r_{i,i+1}) \cdot t_{i,i+1} + r_{i,i+1} \cdot (t_{i,W} + t_{W,i+1})\} + \beta \cdot \sum_{i=0}^{N-1} (r_{i,i+1} \cdot T_r) + \sum_{u=1}^4 \{R_u \cdot \max(g_u, 0)\} \quad (8)$$

이상의 과정을 거쳐 2.1절에서 서술된 블록 리프팅 계획 문제는 하나의 목적 함수를 가지는 단일 목적 최적화 문제로 변환이 되었다. 3.1절에서 자세히 설명하겠지만 초기해 집합(initial population)을 구성하는 블록 리프팅 계획에 대한 각각의 해들은 자신의 적합성 값(fitness value)에 따라 최적해로서의 적합 여부가 판단되며, 이 적합성 값은 식 (10)에서처럼 각 해의 목적 함수 값으로부터 계산될 수 있다. 따라서 본 연

구에서는 블록 리프팅 계획에 대한 각 해의 적합성 값을 계산할 때 식 (8)을 이용하게 된다.

### 3. 블록 리프팅 계획을 위한 알고리즘 제안

#### 3.1 최적 블록 리프팅 계획 알고리즘

본 연구에서 제안된 메타-휴리스틱 최적화 알고리즘은 공간 배치, 계획 등을 위해 현재 널리 사용되고 있는 유전 알고리즘을 기반으로 한다. 유전 알고리즘은 진화론에 의한 탐색 및 최적화 기법으로 분류된다. 이 알고리즘은 설계 과정을 진화 과정으로 볼 수 있다는 전체를 기반으로 한다. 이 알고리즘에 대한 보다 상세한 사항은 많은 참조 문헌에서 찾아 볼 수 있다<sup>[11][12]</sup>. 유전 알고리즘을 기반으로 하여 본 연구에서 제안된 알고리즘은 C++ 언어로 구현되었고, 그 개략도는 Fig. 3과 같다. Fig. 3에 나타나 있듯이, 먼저 블록 리프팅 계획에 대한 다수의 해를 포함하는 초기해 집합을 생성하고(Fig. 3의 (1)) 이들의 적합성 값을 계산한다(Fig. 3의 (2)). 적합성 값은 각 해가 얼마나 최적해에 가까우지를 나타내는 지표로서 식 (10)에서처럼 각 해의 목적 함수 값으로부터 계산이 가능하다. 따라서 초기해 집합에 포함되는 각 해에 대해 와이어 교체 여부를 판단하여 이에 따른 주행 및 공주행 거리를 계산하고 이로부터 각 해에 대한 목적 함수 값 및 최종적으로는 적합성 값을 구한다(Fig. 3의 (3)). 초기해 집합으로부터 두 개의 부모해를 선택한 후(Fig. 3의 (4)) 교배 및 돌연변이 과정을 거쳐 새로운 자식해를 생성하고(Fig. 3의 (5), (6)), 이 과정을 반복하면(Fig. 3의 (7)) 개선된 초기해 집합을 얻을 수 있다

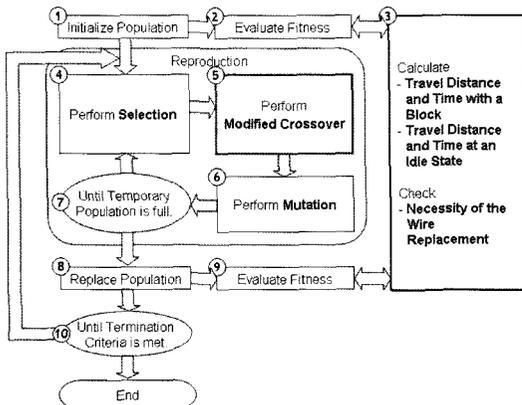


Fig. 3. Flow diagram of the proposed algorithm for block lifting scheduling of a gantry crane.

(Fig. 3의 (8)). 개선된 초기해 집합에 대해 다시 적합성 값을 계산하고(Fig. 3의 (9)), 최적화 종료 조건(예, 최대 반복 회수 등)이 만족할 때까지 이상의 과정을 반복하면 주어진 블록 리프팅 계획 문제에 대한 최적해(적합성 값이 가장 큰 해)를 얻을 수 있다(Fig. 3의 (10)).

#### 3.2 블록 리프팅 순서의 표현 방법

블록 리프팅 계획, 즉 블록의 리프팅 순서는 유전 알고리즘의 인코딩(encoding) 과정에 의해 염색체로 표현될 수 있다. 그리고 그 염색체는 디코딩(decoding) 과정에 의해 다시 블록 리프팅 계획으로 표현될 수 있다. 본 연구에서는 블록의 리프팅 순서를 포함하는 1차원 배열 형태의 염색체를 이용하였다. 즉, 염색체 내 유전자들의 개수는 크레인에 의해 리프팅될 블록의 총수와 같다. Fig. 4는 블록 리프팅 계획과 이에 대한 염색체 표현의 예를 나타낸다. 즉, 블록 리프팅 계획에 대한 각 해는 두 가지 표현 형태를 가진다. 첫 번째는 Fig. 4의 (1)과 같은 리스트(list) 표현이며 두 번째는 Fig. 4의 (2)와 같은 염색체 표현이다. 따라서 갠트리 크레인의 블록간 이동 거리 계산(3.3절) 또는 블록 리프팅 계획의 가시화 등을 위해서는 블록 리스트 표현 방법이 사용되고, 최적해 도출을 위한 개선된 유전자 연산의 적용(3.4절) 시에는 염색체 표현 방법이 사용된다.

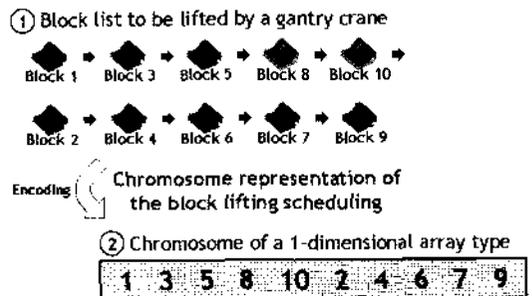


Fig. 4. Example of the block lifting scheduling and the corresponding representation of the chromosome.

#### 3.3 갠트리 크레인의 블록간 이동 거리 계산

갠트리 크레인은 블록을 도크의 길이 방향과 폭 방향으로 이동시킬 수 있다. 도크의 길이 방향으로 이동시킬 때는 본체 하부에 장착된 바퀴를 회전시켜 크레인 전체를 움직이며, 폭 방향으로 이동시킬 때는 본체는 고정된 상태에서 상부의 트롤리(trolley)만을 움직인다(Fig. 5 참조). 이와 같은 크레인의 이동 방법에

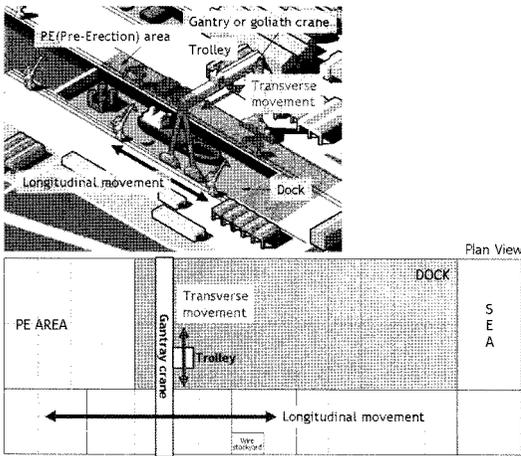


Fig. 5. Movement of a gantry crane on a dock of a shipyard.

따라 블록간 이동 거리, 즉 주행 거리 또는 공주행 거리는 직각 거리 방법(rectilinear distance method)에 의해 간단히 계산할 수 있다. 크레인의 길이 방향 및 폭 방향 이동 속도를 각각  $V_x, V_y$ 라고 할 때 블록 1의 리프팅을 위한 주행 시간( $T_1$ )과 블록 1의 리프팅을 마친 후 블록 2까지 이동하는데 소요되는 공주행 시간( $t_{1,2}$ )는 Fig. 6에서와 같이 계산할 수 있다.

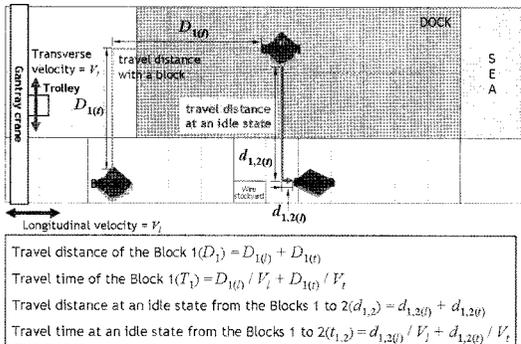


Fig. 6. Example of calculating the travel distance of a gantry crane using the rectilinear distance method.

한편, 크레인은 블록과 직접 연결된 상태에서 블록을 이동시키는 것이 아니라 크레인, 와이어, 샤클, 러그(lug), 블록의 순서로 연결된 상태에서 블록을 이동시킨다. 즉, 크레인과 블록 사이에는 와이어, 샤클, 러그가 위치한다(Fig. 7 참조). 여기서, 러그는 리프팅 전에 블록과 이미 용접되어 있는 부재이다.

이때, 리프팅할 블록의 중량에 따라서 와이어 및 샤

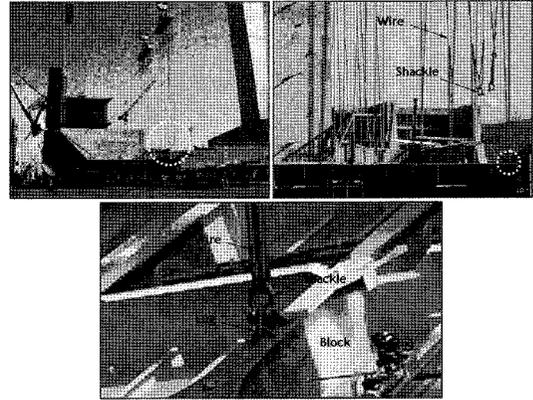


Fig. 7. Connection between a gantry crane and a block means of wires, shackles, and lugs.

클의 개수와 사양이 달라진다. 만일 연속해서 리프팅되는 두 블록의 중량이 많이 다를 경우 후속 블록의 리프팅을 위해서는 이전 블록의 리프팅을 위해 사용된 와이어 및 샤클을 그대로 이용하지 못하고 교체해야 한다. 특히, 이의 교체를 위해서 크레인은 와이어 적치장(wire stockyard)을 방문해야 한다. 즉, 한 블록의 리프팅을 마친 후 다음 블록을 리프팅할 때 와이어 교체가 필요한지를 확인한 후 필요하다면 크레인은 와이어 적치장을 방문해야 한다. 이 경우, 블록간 이동 거리 내에 와이어 적치장까지의 이동 거리가 포함되어야 한다. Fig. 8은 와이어 교체가 필요한 경우에 대한 블록간 이동 거리의 계산 예를 나타낸다. 그림에 나타나 있듯이, 블록 1과 블록 2의 중량 차이가 많이 나서 블록 1의 리프팅 후 블록 2의 리프팅을 위해 와이어 교체가 필요하며, 따라서 크레인이 와이어 적치장을 방문함에 따라 이동 거리가 늘어남을 알 수 있다.

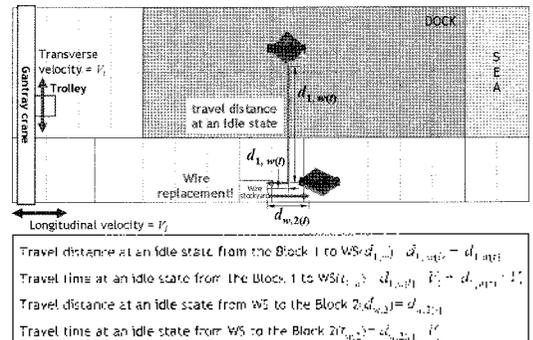


Fig. 8. Example of calculating the travel distance of a gantry crane considering the wire replacement.

### 3.4 개선된 유전자 연산

앞서 언급하였듯이, 본 연구에서는 블록 리프팅 계획을 위한 유전 알고리즘 기반의 메타-휴리스틱 최적화 알고리즘이 제안되었다. 유전 알고리즘에서는 새로운 개체(individual, 자손; child)를 생성하기 위해 선택(selection), 교차(crossover), 그리고 돌연 변이(mutation)라는 세 개의 유전자 연산이 일반적으로 사용된다. 본 연구에서는 블록 리프팅 계획 문제를 효율적으로 풀기 위해 이들 연산을 개선한 후 제안된 알고리즘에 통합하였다.

#### 3.4.1 선택(selection) 연산

선택 연산은 이후의 유전자 연산 단계를 위해 모집단(population)으로부터 두 개의 부모 개체를 선택하는 과정으로 제안된 알고리즘에서는 비례 선택(proportionate selection, roulette wheel selection) 방법이 사용된다. 비례 선택 방법에서 각 개체가 부모 개체로서 선택될 확률  $p_{selection}(i)$ 는 다음과 같이 결정된다.

$$p_{selection}(i) = \frac{F(i)}{\sum_j F(j)} \quad (9)$$

여기서,  $F(i)$ 는  $i$ 번째 개체의 적합성 값을 나타낸다. 위 식으로부터 각 개체의 선택 확률은 적합성 값의 함수임을 알 수 있으며, 적합성 함수는 다음과 같이 나타낼 수 있다.

$$F_t = -F' \text{ or } F_t = \frac{1}{F'} \quad (\text{if } F' > 0) \quad (10)$$

본 연구에서의 적합성 함수는 식 (10)의 두 번째 방법이 이용되었다. 이러한 선택 방법에 의해 적합성 값이 높은 개체일수록 부모로서 선택되어 다음의 유전자 연산에 이용될 확률이 높아지게 된다. 식 (10)에 따라 각 개체의 적합성 값을 계산할 때 식 (8)에 나타낸 목적 함수가 이용된다.

#### 3.4.2 교배(crossover) 연산

교배 연산은 선택 연산을 통해 선택된 두 개의 부모 개체로부터 새로운 자손 개체를 생성하는 과정으로서 본 연구에서는 modified crossover가 이용된다. Modified crossover는 두 부모 중 적합성 값이 높은 개체가 그렇지 않은 개체보다 더 많은 유전 인자(gene)를 자손에게 상속해야 한다는 가정을 기반으로 하는 유전자 연산이다. 첫 번째 자손 개체를 생성하기 위해, 먼저 첫 번째 부모 개체의 염색체에서  $s_1$ 개의

위치가 임의로 선택된다. 여기서  $s_1$ 의 값은 첫 번째 부모 개체의 인자들 중에서 두 번째 부모 개체의 것으로 대체될 인자들의 수를 나타낸다.  $s_1$ 의 값은 두 부모 개체의 적합성 값에 따라 다음과 같은 식에 의해 결정된다.

$$s_1 = \frac{\{F_t(p_1) + F_t(p_2)\} - F_t(p_1)}{F_t(p_1) + F_t(p_2)} \times n$$

(discard decimals)

$$s_2 = n - s_1 \quad (11)$$

여기서,  $F_t(p_1)$ 과  $F_t(p_2)$ 는 각각 첫 번째 및 두 번째 부모 개체의 적합성 값을 나타내고,  $s_2$ 는 첫 번째 부모 개체에서 첫 번째 자손 개체로 그대로 전달되는 인자들의 수를 나타낸다. 그리고  $n$ 은 부모 개체의 염색체 내 인자들의 수를 나타낸다. 식 (11)로부터, 앞서 언급한 가정에 의해 적합성 값이 높은 개체가 그렇지 않은 개체보다 더 작은  $s_1$ 의 값을 가짐을 알 수 있다. 다음 단계에서는 첫 번째 부모 개체의  $s_2$  위치에 있는 인자들이 첫 번째 자손 개체의 해당 위치로 그대로 전달된다. 마지막 단계에서는 첫 번째 부모 개체의  $s_1$  위치에 있는 인자들이 두 번째 부모 개체에서의 그 인자들의 순서에 따라 정렬된 뒤 첫 번째 자손 개체의 해당 위치로 전달된다. 이와 유사한 과정이 두 번째 부모 개체에 대해 두 번째 자손 개체를 생성하기 위해 수행된다. Fig. 9에는 두 자손 개체를 생성하기 위해 두 부모 개체에 적용된 modified crossover의 예가 나타나 있다.

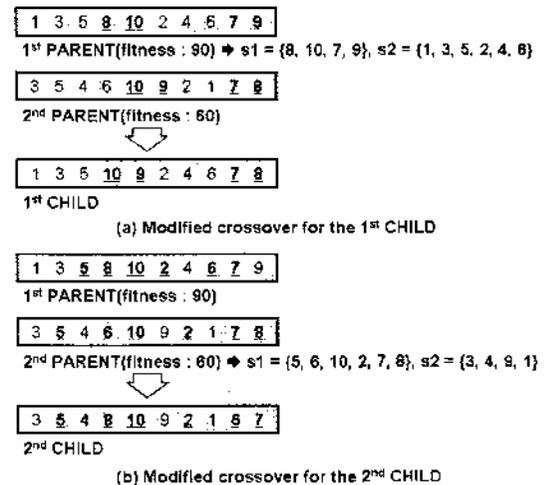


Fig. 9. Example of the modified crossover operation for generating the first and second children.

3.4.3 돌연 변이(mutation) 연산

돌연 변이 연산은 교배 연산을 통해 얻어진 각 자손 개체에 대해 염색체 내 인자들 중에서 각각 두 개의 인자를 임의로 선택하여 상호 교환함으로써 새로운 자손 개체를 생성하게 된다. 제안된 블록 리프팅 계획 알고리즘에 이용된 돌연 변이 연산이 Fig. 10에 나타나 있다.

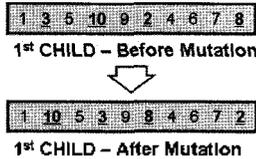


Fig. 10. Example of the mutation operation applied to the first child.

3.5 Weighting factors의 영향 검토

블록 리프팅 계획 문제의 목적 함수(식 (8))에서 weighting factors( $\alpha$  and  $\beta$ )의 영향을 평가하기 위해서 이들에 대한 파라메트릭 테스트를 수행하였다. 앞서 언급하였듯이 weighting factors  $\alpha$ 와  $\beta$ 는 크레인의 전체 공주행 시간(식 (1)의  $F_1$ )과 와이어 교체를 위한 전체 작업 시간(식 (2)의  $F_2$ ) 사이의 일종의 트레이드오프에 해당한다. 따라서  $\alpha$ 와  $\beta$ 의 비율에 따라서 최적화 결과가 달라질 수 있다.  $\alpha$ 와  $\beta$ 를 변경하여 최적화를 수행할 경우 'Pareto optimal set'이라는 많은 수의 최적해를 얻을 수 있다. 본 연구에서는 5장의 Table 1에 나타난 예제에 대해  $\alpha$ 와  $\beta$ 의 값을 0부터 1까지 변화시켜가면서 최적해를 구하는 파라메트릭 테스트를 수행하였다. Fig. 11은 이로부터 얻어진 Pareto optimal set을 나타낸다. 이 그림에서 점 A는 크레인의 전체 공주행 시간(식 (1)의  $F_1$ )만이 목적 함

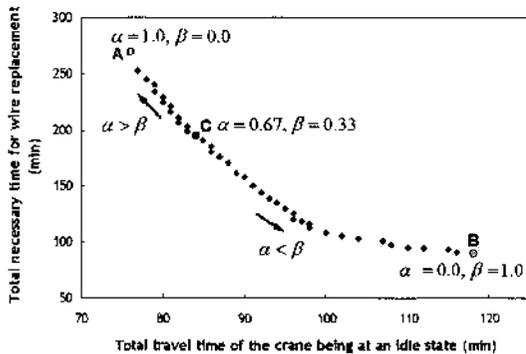


Fig. 11. Pareto optimal set obtained from the parametric test for the weighting factors.

수로 고려되었을 때( $\alpha = 1.0, \beta = 0.0$ )의 결과이고, 점 B는 와이어 교체를 위한 전체 작업 시간(식 (2)의  $F_2$ )만이 목적 함수로 고려되었을 때( $\alpha = 0.0, \beta = 1.0$ )의 결과이며, 점 C는 크레인의 전체 공주행 시간과 와이어 교체를 위한 전체 작업 시간이 함께 고려되었을 때( $\alpha = 0.67, \beta = 0.33$ )의 결과이다. 점 C에 대한 이들 값은 반복적인 시행 착오를 거쳐서 두 개의 목적 함수(식 (1)의  $F_1$ , 식 (2)의  $F_2$ )가 전체 목적 함수(식 (8)의  $F$ )에 비교적 동일한 영향을 갖도록 하는  $\alpha$ 와  $\beta$ 의 값을 구한 것이다.

4. 블록 리프팅 계획 시스템의 개발

본 연구에서는 제안된 알고리즘의 효용성을 검토하기 위해 이를 기반으로 한 블록 리프팅 계획 시스템을 개발하였다. Fig. 12는 개발된 시스템의 구성도를 나타낸다. 그림에 나타나 있듯이, 개발된 시스템은 유전 알고리즘을 기반으로 하는 메타-휴리스틱 최적화 알

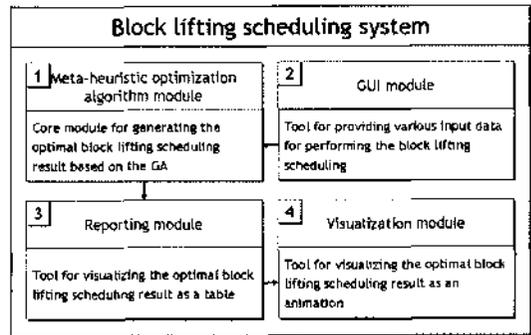


Fig. 12. Configuration of the block lifting scheduling system developed in this study.

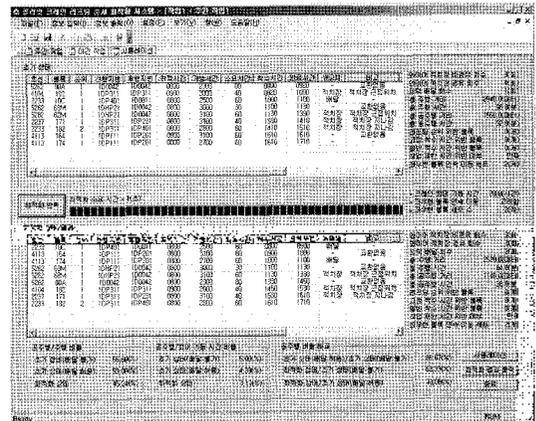


Fig. 13. Screenshot of the block lifting scheduling system developed in this study.

고리증 모듈, 블록 리프팅 계획의 수행을 위해 각종 입력 정보를 넣기 위한 사용자 인터페이스(GUI) 모듈, 최적화 결과를 보여주기 위한 레포팅 모듈, 최종 블록 리프팅 순서를 애니메이션으로 보여주는 가시화 모듈로 구성된다. Fig. 13은 개발된 프로그램의 실행 화면을 나타낸다.

### 5. 블록 리프팅 계획 시스템의 적용 예

본 연구에서 개발된 시스템의 효용성을 검토하기 위해 이를 조선소의 실제 블록 리프팅 계획 문제에 적용해 보았다. 그리고 적용 결과를 조선소의 크레인 운영 담당자에 의한 수작업 결과와 비교해 보았다.

Table 1은 특정일에 대한 블록 리프팅 계획을 수작업과 본 연구에서 개발된 프로그램에 의해 수행한 결과를 비교한 것이다. 표에 나타나 있듯이 수작업에 의한 결과로 블록 리프팅을 수행하였을 경우, 크레인의 총 주행 시간은 3시간 20분이고, 이중 공주행 시간은 2시간 6분(총 주행 시간의 63%)이었으며 와이어 교체 회수는 9회였다. 이에 반해 본 연구에서 개발된 시스템을 이용했을 경우, 크레인의 총 주행 시간은 2시간

38분이고, 이중 공주행 시간은 1시간 24분(총 주행 시간의 53%)이었으며 와이어 교체 회수는 5회였다.

Table 2는 6일간 수작업과 본 연구에서 개발된 프로그램에 의한 결과를 서로 비교한 것이다. 표의 값은 일일 작업 시간 중 공주행 시간이 차지하는 비율(%)을 나타낸 것이다. 표에서 알 수 있듯이 본 연구에서 개발된 프로그램을 이용할 경우 크레인의 공주행 시간을 상당 부분 단축시킬 수 있음을 알 수 있다.

### 6. 결론 및 향후 연구 계획

본 연구에서는 조선소에서 선박 건조를 위해 이용되고 있는 갠트리 크레인의 생산성 향상을 위해 최적 블록 리프팅 계획 알고리즘을 제안하고 이를 기반으로 한 시스템을 개발하였다. 이를 위해, 크레인의 공주행 거리와 와이어의 교체 회수를 고려하여 블록 리프팅 계획 문제를 다목적 최적화 문제로 정식화하였고, 이를 해결하기 위한 유전 알고리즘 기반의 메타-휴리스틱 최적화 알고리즘을 개발하였고, 특히, 크레인의 블록간 이동 거리를 계산할 때 와이어 교체 여부를 고려하였다. 마지막으로 본 연구에서 제안된 알고리즘 및 개발된 시스템의 효용성과 효용성을 검증하기 위해 조선소의 실제 블록 리프팅 계획 문제에 적용해 보았다. 그 결과, 본 연구에서 개발한 시스템을 이용할 경우, 수작업에 의한 결과에 비해 크레인의 공주행 시간을 상당 부분 감소시킬 수 있을 확인하였고, 따라서 본 시스템이 조선소의 블록 리프팅 계획 업무에 충분히 활용될 수 있음을 확인하였다.

본 연구에서는 블록 리프팅 계획과 관련된 초기의 연구로서 정적인 환경에 대한 문제를 우선 고려하였다. 하지만 실제 조선소의 블록 리프팅 계획 업무에서는 계획된 블록 리프팅 작업 사이에 시급히 처리해야 할 리프팅 작업이 간혹 추가된다. 따라서 이와 같은 동적인 환경에서도 신뢰성 있는 결과를 내 줄 수 있는 알고리즘 및 시스템의 개발이 필요하다.

### 감사의 글

- 본 연구는
- a) 울산대학교 조선해양공학부 세계일류화사업단 (SOTOP)
  - b) 서울대학교 해양시스템공학 연구소 및 BK 21 해양기술인력양성사업단
  - c) 국방과학연구소 수중유동체기술특화센터 SM-11 과제 “수중 유동체의 체계/부체계 기능 및 성능 시뮬

**Table 1.** Comparison of the performance results of the manual scheduling by the manager and the automatic scheduling by the developed system on the specific day

	Result of manual scheduling	Result of the developed system
Total travel time	3 hr 20 min	2 hr 38 min
Total travel time at an idle state	2 hr 6 min	1 hr 24 min
No. of the wire replacement	9	5

**Table 2.** Comparison of the performance results of the manual scheduling by the manager and the automatic scheduling by the developed system during six days

	Result of manual scheduling	Result of the developed system
Day #1	15.8%	7.9%
Day #2	11.1%	5.7%
Day #3	20.3%	13.2%
Day #4	12.2%	9.8%
Day #5	13.2%	7.8%
Day #6	13.1%	8.1%
Avg.	14.3%	8.8%

레이션을 위한 네트워크 기반의 가상(Virtual) 복합 시스템 모델 구조(Architecture) 연구”

d) 대우조선해양

의 지원과 2009년 정부(교육과학기술부)의 새원으로 한국연구재단의 지원을 받아 연구되었음(KRF-2008-314-D00494)을 밝히며, 이에 감사 드립니다.

**참고문헌**

1. Daganzo, C. F., “The Crane Scheduling Problem”, *Transportation Research B*, Vol. 23, No. 3, pp. 159-175, 1989.
2. Daganzo, C. F., “Crane Productivity and Ship Delay in Ports”, *Transportation Research Record*, Vol. 1251, pp. 1-9, 1990.
3. Peterkofsky, R. I. and Daganzo, C. F., “A Branch and Bound Solution Method for the Crane Scheduling Problem”, *Transportation Research B*, Vol. 24, No. 3, pp. 159-172, 1990.
4. Kim, K. H. and Kim, K. Y., “An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals”, *Transportation Science*, Vol. 33, No. 1, pp. 17-33, 1999.
5. Ng, W. C. and Mak, K. L., “An Effective Heuristic for Scheduling a Yard Crane to Handle Jobs with

- Different Ready Times”, *Engineering Optimization*, Vol. 37, No. 8, pp. 867-877, 2005.
6. Kim, K. H. and Park, K. T., “A Dynamic Space Allocation Method for Outbound Containers in Carrier-Direct System”, *Proceedings of the Third Annual International Conference on Industrial Engineering Theories, Applications and Practice*, Hong Kong, Vol. 2, pp. 859-867, 1998.12.28-31.
7. 김후곤, 김철한, “컨테이너 터미널에서의 트랜스퍼 크레인의 최적 운영 방안에 관한 연구”, *한국경영학회지*, Vol. 33, No. 1, pp. 35-49, 2009.
8. Cohon, J. L., *Multiobjective Programming and Planning*, Academic Press, New York, 1978.
9. Gill, P. E., Murray, W. and Wright, M. H., *Practical Optimization*, Academic Press, New York, 1981.
10. Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M., *Engineering Optimization: Methods and Applications*, John Wiley and Sons, New York, 1983.
11. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, MA, 1989.
12. Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand-Reinhold, New York, 1991.
13. 한성남, 이규열, 노명일, “개선된 유전자 알고리즘을 이용한 최적 공간 배치 설계에 관한 연구”, *한국 CAD/CAM학회 논문집*, Vol. 6, No. 3, pp. 174-183, 2001.



**노 명 일**

1998년 서울대학교 조선해양공학과 학사  
 2000년 서울대학교 조선해양공학과 석사  
 2005년 서울대학교 조선해양공학과 박사  
 2005년~2007년 서울대학교 공학연구소  
 해양시스템공학연구소 선임연구원  
 2007년~현재 울산대학교 조선해양공학과  
 부 조교수

관심분야: Computer-Aided Ship Design and Manufacturing, Simulation-Based Design/Manufacturing, 최적 설계, CAD/CAM/CAE, CAGD



**이 규 열**

1971년 서울대학교 공과대학 조선공학과 학사

1975년 독일 하노버 공과대학 조선공학 석사(Dipl.-Ing.)

1982년 독일 하노버 공과대학 조선공학 박사(Dr.-Ing.)

1975년~1983년 독일 하노버 공과대학 선박설계 및 이론연구소, 주정부 연구원

1983년~1994년 한국기계연구원 선박해양공학연구센터, 선박설계, 생산자동화 연구사업(CSDP)단장

1994년~현재 서울대학교 공과대학 조선해양공학과 교수

관심분야: CAD, CAGD, 선박 설계, 시뮬레이션, CALS