조선소 선행의장 공정의 스케줄링 최적화를 위한 알고리즘 비교 분석 연구

김기현[1], 나영진[1], 박세호[1],
이동녕[2], 장종환[2], 임민중[1]
[1]동국대학교, [2]한화오션

rlsrlgus123@naver.com, dudwls976@naver.com, 323psh@naver.com,
dnlee26@hanwha.com, jhjang36@hanwha.com, minjoong@dongguk.edu

A Comparative Study of Scheduling Optimization Algorithms for the Pre-Outfitting Process in Shipyards

Ki-Hyun Kim[1], Young-Jin Na[1], Seho Park[1],
Dongnyeong Lee[2], Jonghwan Jang[2], Minjoong Rim[1]

[1]Dongguk University, [2]Hanwha Ocean

요 약

본 논문은 조선소 선행의장 공정에서의 효율적인 스케줄링과 원활한 공장 흐름을 위한 최적화 알고리즘을 제시하고, 여러 알고리즘을 비교 · 분석함에 따라 실제 환경에 적용할 수 있는 공정 자동화의 기반을 마련한다. 선행의장은 선체 블록 제작 후 탑재 이전 단계에서 진행되는 건조 공정의 중요한 단계 중 하나로 이후 진행되는 공정에 직접적인 영향을 미친다. 그러나 현행 스케줄링 체계는 스케줄러의 수작업에 의존하여 숙련도에 의한 차이점과 휴먼 에러 발생 가능성 등의 문제점이 존재한다. 이를 해결하기 위해 본 논문에서는 다양한 알고리즘 및 휴리스틱 알고리즘을 기반으로 한 여러 최적화 로직을 비교 · 분석하여 실제 환경에서 더욱 효과적인 방법을 찾고자 한다.

Ⅰ. 서 론

조선소에서 선행의장은 선체 블록 제작 후 탑재 이전 단계에서 배관, 설비 등을 미리 설치하는 공정으로, 후속 탑재 일정의 효율성과 전체 리드타임에 직접적인 영향을 미친다. 특히 블록 별 크기, 협력사 배정 등 제약이 복잡하게 얽혀 있고, 자원 활용률을 높일 수 있는 중요한 단계로 간주된다. 이 단계의 계획 수립이 비효율적일 경우, 전체 선박 건조 일정의 지연, 인력 · 설비의 부하 불균형, 블록의 납기 지연 등 복합적인 리스크가 발생한다. 따라서 효율적이고 일관된 스케줄링 체계를 통한 일정 관리가 필수적이다.

현재 조선소에서는 대일정 및 중일정 수준의 계획은 팀 단위 검토를 통해 관리되고 있으나, 소일정은 여전히 전문 스케줄러 개인의 경험에 의존하고 있다. 이로 인해 계획 품질이 작업자의 숙련도에 따라 달라지거나 휴먼 에러가 발생할 수 있다는 한계점이 존재한다. 기존의 조선소 스케줄링 연구들은 주로 특정 공정에 한정된 최적화 문제에 집중하거나, 단일 휴리스틱을 적용하는 수준에 머물렀다. 또한, 선행의장 공정의 복합적인 제약(공장 별 제약, 협력사 부하 분배 등)을 통합적으로 다루지 못했다는 한계가 있다. 이러한 점을 보완하여 여러 최적화 알고리즘을 동일 조건에서 비교함으로써 실제 조선소 적용 가능성을 검증한다는 점에서 차별성을 갖는다. 따라서 본 논문에서는 실제 조선소의 의장 데이터와 제약 조건을 기반으로, 그리디, 랜덤 그리디 서치, 유전 알고리즘 등 서로 다른 탐색 전략을 비교 · 분석함으로써 실무에서 용이한 접근 방식을 도출하고자 한다.

Ⅱ. 본론

본 논문의 목표는 선행의장 공정의 소일정을 자동으로 수립하고, 납기 준수 · makespan 최소화등 다중 목표를 고려한 최적 스케줄을 탐색하는 알고리즘을 개발하고 비교하는 것이다. 이를 위해 먼저 입력 데이터 · 제약조건 · 목적함수를 정의하였다.

1. 입력 데이터

<표 1> 입력 데이터 구분

구분	주요내용
블록 데이터	블록 ID, 크기, 인력 시수, 착수가능일, 납기일 등
정반(bay)데이터	정반 ID, 공장/라인 구분, 크기, capacity
자원/조직 데이터	협력사 및 직영 구분, 선 · 후행 조직 정보

2. 제약 조건

1. 공장/정반 수용 제약
   - 블록의 크기<정반의 크기 이어야 함
   - 각 정반은 정반마다 수용할 수 있는 블록의 최대 개수가 정해져 있음
2. 공장 별 제약 조건
   - 선박의 종류(LNGC 등)에 따라 정해진 블록은 B 공장 배량 불가
   - 특정 작업이 이루어지는 블록은 A 공장 배량

3. 목적함수 설계

알고리즘 내의 최적화 목표는 여러 가지 항목의 가중합으로 구성된다.

<표 2> 목적함수 구분

구분	설명
f_1	납기를 초과한 기간 최소화
f_2	전체 일정 makespan 최소화
f_3	협력사 부하 평준화
f_4	정반 재배치 최소화
f_5	선행 조립에 따른 선호 조건
f_6	특정 블록 옥내 배치 선호 조건
f_7	시수 적은 블록 옥내 배치 선호 조건

이러한 항목들을 통합하여 최종 목적함수는 다음과 같이 구성된다.

$$\text{Minimize } F = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4 + \alpha_5 f_5 + \alpha_6 f_6 + \alpha_7 f_7$$

<식 1>

가중치는 현업 우선순위에 따라 조정할 수 있으며, 본 논문에서는 납기 지연을 1 순위로 그 이후 부하평준화, makespan, 기타 제약 등을 우선으로 최적화를 진행한다.

다음과 같이 3 가지 휴리스틱 알고리즘을 기반으로 최적화 알고리즘의 기반을 마련한다.

(1) Greedy Algorithm

- 매 단계에 현재 시점에서 가장 우수하다고 판단되는 선택을 반복적으로 수행하는 탐색 기반 알고리즘

- 블록을 납기 순으로 정렬 후, 가능한 후보 정반 중 가장 높은 score 를 가지는 정반에 배정

- Score 는 납기초과, 부하차이, 공장 선호도 등의 목적함수의 내용들을 정규화 하여 종합

- 계산속도 빠르지만 지역 최적해에 머물 가능성이 있음

(2) Randomized Greedy Search

- 일정 시간을 정하여 그리디 기반 알고리즘 여러 번 실행 후 최고 점수를 찾는 방식

- 그리디 대비 품질이 개선되지만,탐색시간이 길어질수록 초반에 비해 개선폭이 감소하여 적절한 제한이 필요

(3) Genetic Algorithm (GA)

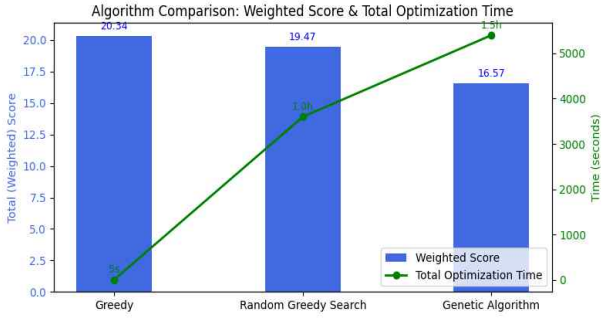
- 자연의 진화 과정을 모방하여 여러 해를 동시에 진화시키는 알고리즘

- 초기 개체군(50 개) 생성 -> 교차 및 변이 수행 -> 세대 반복(30 세대)

- 알고리즘을 통해 최적의 블록 순서를 결정 후, score 기반으로 정반에 배정

- 다양한 해탐색을 통해 품질이 개선되지만 시간이 오래 걸림

4. 결과 분석



<그림 1> 각 알고리즘의 종합 점수 및 소요 시간

세 가지 최적화 알고리즘을 비교한 결과, 복합 목적함수를 종합한 총합 점수는 Greedy, Random Greedy Search, Genetic Algorithm 순으로 감소하였다. 점수가 낮을수록 일정 품질이 우수함을 의미하며, Genetic Algorithm 이 가장 높은 품질의 결과를 도출하였다.

Ⅲ. 결론

본 논문에서는 선행의장 공정의 스케줄링 최적화를 위해 그리디, Random Greedy Search, 유전 알고리즘 기반 최적화 알고리즘을 통해 기법을 비교하였다. 유전 알고리즘 기반 최적화가 가장 좋은 품질의 일정을 도출하였지만, 실제 현업에서 활용하기 위해서는 파라미터 조정 등을 통해 일정 도출 시간을 단축 시킬 필요가 있다.

향후 연구에서는 다음과 같은 확장을 고려할 수 있다.

- 다양한 메타 휴리스틱 기법 적용 및 소요 시간 단축

- 실제 생산데이터 기반의 시뮬레이션 검증

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원 - 학석사연계 ICT 핵심인재양성 사업의 지원을 받아 수행된 연구임 (IITP-2024-00436744)

참 고 문 헌

[1] K. Oh, J. Lee, H. Kim, et al., “Ship block scheduling optimization using genetic algorithm considering dock capacity and delivery deadline,” Ocean Engineering, vol. 266, pp. 113250, 2023.

[2] S. Lee, M. Ryu, and H. Park, “A hybrid heuristic algorithm for shipyard block scheduling considering resource leveling and delivery constraints,” Computers & Industrial Engineering, vol. 175, pp. 109022, 2023.