

A hybrid genetic algorithm for the optimal transporter management plan in a shipyard

Jun-Ho Park*, Yung-Keun Kwon**

*Student, School of IT Convergence, University of Ulsan, Ulsan, Korea

**Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea

[Abstract]

In this study, we propose a genetic algorithm (GA) to optimize the allocation and operation order of transporters. The solution in the GA is represented by a set of lists each of which the operation order of the corresponding transporter. In addition, it was implemented in the form of a hybrid genetic algorithm combining effective local search operations for performance improvement. The local search reduces the number of operating transporters by moving blocks from a transporter with a low workload into that with a high workload. To evaluate the effectiveness of the proposed algorithm, it was compared with Multi-Start and a pure genetic algorithm through a simulation environment similar in scale to an actual shipyard. For the largest problem, compared to them, the number of transporters was reduced by 40% and 34%, and the total task time was reduced by 27% and 17%, respectively.

▶ **Key words:** Transporter Management, Genetic Algorithm, Block Transport Scheduling

[요 약]

본 연구에서는 트랜스포터의 할당 및 운행 순서를 최적화하기 위한 유전 알고리즘을 제안한다. 유전 알고리즘의 해는 리스트의 집합으로 표현되는데 각 리스트는 해당 트랜스포터가 작업할 순서를 나타낸다. 또한 성능 향상을 위해 효과적인 지역 탐색 연산을 결합한 혼합형 유전 알고리즘의 형태로 구현하였다. 지역 탐색 연산은 작업량이 적은 트랜스포터에서 작업의 블록을 꺼내어 다른 트랜스포터의 작업 목록에 삽입함으로써 트랜스포터 운용 대수의 감소를 유도한다. 제안하는 알고리즘의 효율성을 평가하기 위해 실제 조선소와 유사한 규모의 시뮬레이션 환경을 통해 Multi-Start 및 순수 유전알고리즘과 비교하였다. 가장 큰 규모의 문제에 대해 그들에 비해 트랜스포터 수는 각각 40% 및 34%, 총작업 소요 시간은 27% 및 17% 감소시켰다.

▶ **주제어:** 트랜스포터 운영, 유전 알고리즘, 블록운반 스케줄링

-
- First Author: Jun-Ho Park, Corresponding Author: Yung-Keun Kwon
 - *Jun-Ho Park (qkrwnsgh71w@gmail.com), School of IT Convergence, University of Ulsan
 - **Yung-Keun Kwon (kwonyk@ulsan.ac.kr), Department of Electrical, Electronic and Computer Engineering, University of Ulsan
 - Received: 2023. 10. 26, Revised: 2023. 11. 28, Accepted: 2023. 11. 29.

I. Introduction

선박 건조는 다양한 크기의 블록들을 조립하여 완성하는 방식으로 이루어진다. 선박 건조 기본단위인 블록은 조립, 의장, 도장 등의 단계를 거치며, 각 단계는 조선소 내의 여러 플랜트에서 수행된다. 블록들을 조립할 때까지의 과정 중 생산 공정 및 적치에 필요한 운반은 트랜스포터(Transporter)라는 블록 운반용 특수 차량을 이용한다. 트랜스포터의 효율적 운영은 선박 인도 일정을 준수하는 데 중요한 역할을 하며, 비효율적인 운반 계획은 트랜스포터 운영 비용과 물류비용 증가 및 전체 공정의 지연을 유발한다.

이러한 트랜스포터 운용 최적화를 위해 기존 연구[1][2]에서는 개미 군집 알고리즘을 적용하여 초기 블록의 할당 및 운반 순서를 결정한 뒤 유전 알고리즘(Genetic Algorithm; GA)을 통해 개선하는 2단계 알고리즘을 제시하였다. 하지만 트랜스포터의 이용 가능 대수 최소화에 대해서는 다루지 않았다. 또 다른 연구[3]에서는 염색체 표현 방법을 달리하는 두 유전 알고리즘을 제시한 뒤 비교하였다. 그러나 이 연구에서는 트랜스포터의 대수가 최대 8대로 제한되었다. 따라서 보다 실제 조선소 환경을 고려한 트랜스포터 운용 최적화 방법은 연구할 필요성이 있다.

본 논문에서는 단위 계획 기간 동안 각 생산 공정 및 적치 위치에 따라 운반해야 할 블록들의 모든 요구량이 미리 알려진 있는 환경이라고 가정한다. 이러한 환경과 제약사항을 만족시키며 계획된 모든 블록을 운반하는 데 필요한 트랜스포터의 대수와 이동 거리를 최소화할 방법을 제안한다. 이를 위해 사용할 트랜스포터들의 목록과 각 트랜스포터가 운반할 블록들의 리스트로 해를 표현하고 최적해를 탐색하기 위한 유전 알고리즘을 고안하였다. 특히 효과적인 지역 최적해를 탐색하여 성능을 개선한다고 알려져 있는 혼합형 유전 알고리즘[4][5][6]을 사용하였다.

본 논문의 2장에서 기존 관련 연구와 문제 정의를 설명하고, 3장에서는 본 논문에서 제안하는 방법에 대하여 기술한다. 4장에서는 실험 결과, 5장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

II. Preliminaries

1. Related Works

기존 연구에서 트랜스포터의 대수와 이동 거리를 최소화하기 위해 유전 알고리즘, 타부서치 알고리즘, Dijkstra 알고리즘 등 다양한 방법이 제안되었다.

Shin, et al.[7]은 경로 생성 알고리즘으로 초기해를 구성하고 타부 서치 알고리즘을 이용한 3가지의 휴리스틱 알고리즘을 제안하여 트랜스포터 보유량을 최소화하고 할당된 트랜스포터들의 운행 시간을 평준화하는 최적화 방법을 제안하였다. Lee, et al.[8]은 조립 공정과 블록 운반 과정을 통합적으로 고려하여 Makespan을 최소화하는 일정계획을 수립하였다. 블록 운반 과정에서 새로운 이웃 탐색 방법을 적용한 타부 탐색을 제안하여 적용하였다. Moon, et al.[9]은 경로 탐색 기법 방법인 Dijkstra, A*, Floyd 알고리즘들의 계산 비용과 정밀도를 비교하였다. 회전, 경로 손상에 따른 총 공사 이동 시간을 각 알고리즘 별로 계산하여 Dijkstra 알고리즘이 적합함을 확인하였다. 또 다른 유사한 연구[10]에서는 트랜스포터의 공주행 거리를 최소화하기 위해 Dijkstra 알고리즘을 사용하여 각 절점 간의 최단 경로를 구하였고 트랜스포터들에 블록 할당 및 운반 순서를 결정을 하는데 전역 최적화 방법 중 하나인 Simulated Annealing(이하 SA) 기법을 사용하였다. 하지만 운용 트랜스포터 수의 최소화를 고려하지 않았다. Yim et al.[1]은 유전 알고리즘을 통해 공주행을 최소화 하는 운반 계획 알고리즘을 제안하였다. 유전 알고리즘의 초기해를 구성하기 위해서 개미 군집 알고리즘을 활용하였고 운반 거리 결정을 위해서 Dijkstra 알고리즘을 사용하였다. 하지만 운용 트랜스포터 수의 최소화를 고려하지 않았다. Joo, et al.[3]은 최소수의 트랜스포터를 결정하고, 블록 운반 완료 시간의 최소화를 위한 일정을 수립하기 위하여 유전 알고리즘의 염색체 구성을 새로 하였다. 하지만 블록과 트랜스포터의 수가 제한적이어서 실험 내용이 실제 환경을 제대로 반영하지 못하였다.

위의 연구 방법들은 트랜스포터의 블록 운반 문제를 해결하기 위한 다양한 알고리즘들을 적용하였으며, 각 방법은 소수의 트랜스포터 운용이나 블록 이동과 같은 제한된 문제 환경에 적합한 접근 방식을 사용하였다. 그러나 수십대의 트랜스포터와 매일 수백 개의 블록을 이동하는 실제 운영 환경에서의 적용 가능성에 대해서는 성공적인 결과를 보이지 못한 한계점이 있다.

본 연구에서는 운용 트랜스포터의 최소화와 트랜스포터의 총 주행 시간의 최소화를 목표로 최적화를 진행한다. 이를 위해 기존 유전 알고리즘에서 지역 탐색을 추가한 혼합형 유전알고리즘을 통해 추가적인 성능 개선 방안을 제안한다. 실험은 두 알고리즘의 성능 차이를 확인하기 위해서 제안하는 모델로 프로그램을 구축한 뒤 실험 결과를 비교한다. 또한 실제 운영 환경과 비슷하도록 블록 수와 트랜스포터 수를 맞추어 실험 데이터를 구성하였다.

2. Problem Formulation

본 연구에서 트랜스포터 스케줄링 문제는 트랜스포터 정보표와 블록 운반 일정표 등 2가지 표를 입력으로 사용하며 Table 1과 2는 그 예를 각각 보여준다.

Table 1. Example of Transporter Information Data

Transporter No	Loadable Weight (t)	Loaded Speed (m/s)	Unloaded Speed (m/s)
1	500	2	5
2	600	1	4
3	600	1	4
..
M	500	1	5

Table 2. Example of Block Transport Schedule Data

Block No	Weight	Starting Point	Destination Point	Start Time	End Time
1	200	1	9	12:00	17:00
2	100	4	5	08:00	13:00
3	500	3	10	09:30	18:30
..
N	150	2	6	10:30	17:30

Table 1에서 보듯이 트랜스포터 정보표는 총 M 대의 가용 트랜스포터 각각에 대해 탑재 최대 중량, 주행 및 공주행 속력에 대한 정보를 담고 있다. 또한 Table 2에서 보듯이 블록 운반 일정표는 운반해야 하는 총 N 개의 블록 각각에 대해 블록 중량, 출발지, 도착지, 운반 시작 시각과 종료 시각에 대한 정보를 담고 있다. 이때 실제 운반은 시작 시각 이후에 시작되어 종료 시각 이전에 종료되어야 함을 의미한다. 또한 각 트랜스포터는 할당된 블록들을 한번에 하나의 블록만을 운반하며 하나의 블록을 운반 완료한 후 다음 블록을 운반할 수 있다고 가정한다. 한편, 트랜스포터 스케줄링 문제의 출력은 각각의 블록 운반 일정(즉 Table 2의 각 행)에 대해 운반을 담당할 트랜스포터를 할당하고 운반 순서를 스케줄링하는 것이다.

Table 3. Example of Schedule Program Results

Block No	Assigned Transporter No	Order
1	6	1
2	20	3
3	10	2
..
N	56	2

Table 3은 그러한 출력의 예를 보여준다. 즉 운반해야 할 N 개 블록 각각에 대해 운반을 담당할 트랜스포터의 번호와 해당 트랜스포터가 운반할 순서를 출력한다. 할당된

트랜스포터와 순서에 따라 작업의 효율성이 달라지기 때문에 본 연구에서는 제약 조건을 만족하면서 가장 적은 수의 트랜스포터를 할당하고 운영시간을 최소화하도록 Table 3을 최적화하는 것이 본 연구의 목표가 된다. 실제 조선소 상황을 고려하여 본 연구에서 다음과 같은 3가지 제약사항을 반영하였다.

- 운송 블록의 중량이 할당된 트랜스포터의 탑재 최대 중량보다 작아야 한다.
- 각 블록은 계획된 운반 시작 시각과 종료 시각 사이에 운반되어야 한다.
- 모든 작업은 계획된 일과 종료 시각 전까지 완료되어야 한다.

III. The Proposed Scheme

본 논문에서 다루는 트랜스포터 운용 문제는 각 트랜스포터에 운반할 블록들을 할당하고 운반 순서를 스케줄링하는 조합 최적화 문제이다. 이러한 종류의 문제를 풀기 위해 유전 알고리즘 기반의 방법을 제안하며 특히 순수 유전 알고리즘의 낮은 지역 탐색 능력을 보완하기 위해 지역 탐색이 결합한 혼합형 유전 알고리즘 모델을 활용한다. Figure 1은 본 논문에서 제안하는 혼합형 유전 알고리즘의 흐름도를 나타내며, 유전 알고리즘의 초기 해 생성 과정, 선택, 교차, 변이 연산 그리고 지역 탐색 연산에 대한 과정을 보여주고 있다. 다음 절부터 각각에 대해서 자세히 설명한다.

3.1 Chromosomal Representation

유전 알고리즘은 해 집단을 진화시켜 최적해를 탐색한다. 이때 해는 염색체로 표현되며 탐색에 적절하도록 표현하는 것이 중요하다. 본 연구에서 염색체는 가용할 트랜스포터들의 리스트로 구성되며 각 트랜스포터에 대해 운반하도록 할당된 블록들의 리스트가 표현된다. Figure 2는 이러한 염색체 예를 보이는데 m 대의 트랜스포터를 운용하여 n 개의 블록을 운반하는 운용을 나타낸다. 구체적으로 트랜스포터 T_1 은 5개의 작업(블록 1, 6, 2, 14, 9를 순서대로 운반)이 할당되었지만 T_2 는 3개의 작업(블록 8, 13, 5를 순서대로 운반)만 할당됨을 의미한다. Figure 2의 염색체 표현은 Table 3에서 보인 프로그램 출력 표와 표현 형태는 다르지만, 실질적으로 동일한 의미임을 알 수 있다.

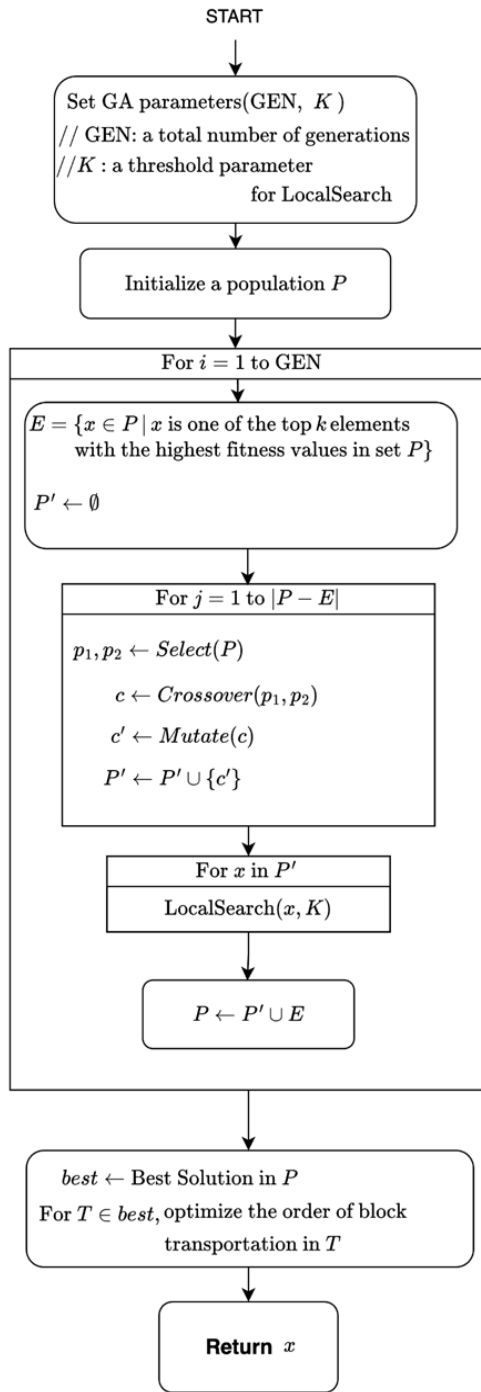


Fig. 1. Flow of the Proposed Method

Transporter No	Block No				
	1	6	2	14	9
T_1	1	6	2	14	9
T_2	8	13	5	-	-
...	...				
T_{m-1}	$n-1$...			$n-12$
T_m	n	...			-

Fig. 2. Example of a Chromosomal Representation

3.2 Fitness Evaluation

임의의 해 i 에 대한 품질을 수치화하기 위해 아래와 같은 적합도 함수 $fitness_i$ 를 정의하였다.

$$fitness_i = (\alpha p_i + (1 - \alpha) \frac{1}{t_i}) / s_i.$$

이때 p_i 는 총 트랜스포터 중 사용되지 않은 트랜스포터의 대수를, t_i 는 모든 트랜스포터가 블록을 운반하는 데 걸린 시간을 의미한다. 또한 α 는 두 요인 사이의 가중치로서 본 연구에서는 0.9로 설정되었다. 따라서 적합도 값이 클수록 사용된 트랜스포터가 적거나 작업 완료 시간이 빠르다는 것을 의미한다. 한편, 블록 중량과 운반 시작/종료 시각 등에 관한 제약 사항(2장 문제 정의 참조)을 위반하는 해에 대해서는 위반 요소의 갯수 s_i 만큼 적합도에서 나누어 적합도가 제약 사항 위반 횟수에 반비례하도록 정의하였다. 이를 통해 제약 조건을 일부 어기더라도 그 해가 생존하여 해집단 내의 다양성을 높이도록 하였다.

3.3 Selection

선택 연산은 두 개의 해를 교배시키기 위해 해 집단에서 해를 선택하는 연산이다. 일반적으로 자주 사용되는 룰렛 휠 방식의 선택 연산을 본 연구에서도 적용하였다. 룰렛 휠 선택 연산은 적합도 값에 따라 자식 해로 선택될 가능성을 높이는 품질에 비례하는 선택 방식이다. 다만 선택 확률이 단순히 적합도에 비례하는 경우 적합도가 좋은 일부 해의 선택 확률이 지나치게 높아져 생성되는 자식해의 다양성이 떨어질 수 있다. 이러한 문제를 해결하기 위해 적합도가 가장 높은 해와 낮은 해의 선택 확률을 k 배로 일정하게 유지하도록 해 i 의 선택 확률 p_i 를 다음 식과 같이 정의하였다 (본 연구에서는 $k = 4$ 로 설정하였다).

$$p_i = fitness_i - fitness_{min} + \frac{fitness_{max} - fitness_{min}}{k - 1}.$$

이때 $fitness_{max}$ 와 $fitness_{min}$ 은 해 집단에서 각각 적합도가 가장 높은 해와 낮은 해의 적합도를 의미한다.

3.4 Crossover

교차 연산은 두 개의 부모 해에서 새로운 자식 해를 생성하는 연산이다. 본 연구에서는 교차 연산의 과정에서 자식 해에 중복되는 블록 번호가 생기지 않도록 균일 교차(uniform crossover) 연산을 구현하였다. Figure 3은 두 부모 해 i 와 j 의 교차 연산 예를 보여준다. 그림에서 보듯

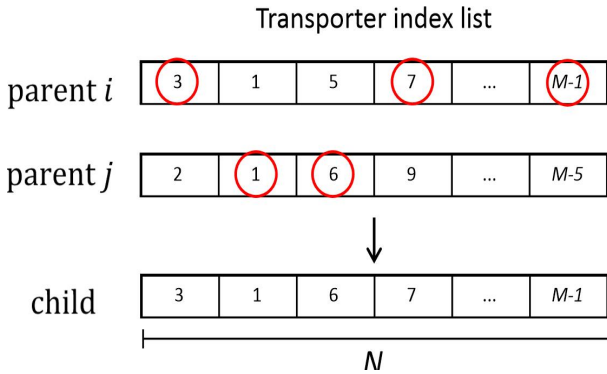


Fig. 3. Example of Crossover Operation

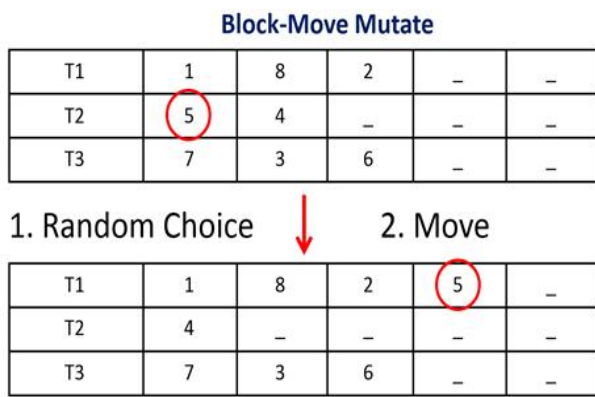


Fig. 4. Example of Block-Move Mutate

이 두 부모 해는 먼저 3.1에서 설명한 염색체 표현법에 의하여 길이가 N (총 블록 개수)인 1차원 배열 블록 기반 염색체로 표현된다. 이때 k 번째 원소의 값 p 는 번호가 k 인 블록이 번호가 p 인 트랜스포터에 할당되어 운반될 것임을 의미한다. 교차 연산자는 각 유전자에 대해 P 의 확률로 i 로부터, $(1 - P)$ 의 확률로 j 로부터 해당 유전자를 자식 해로 복사한다 (본 연구에서 $P=0.5$ 로 설정되었다). 예를 들면 블록 3과 7은 부모 해 i 로부터, 블록 1과 6은 부모 해 j 로부터 자식 해로 전달되었음을 의미한다.

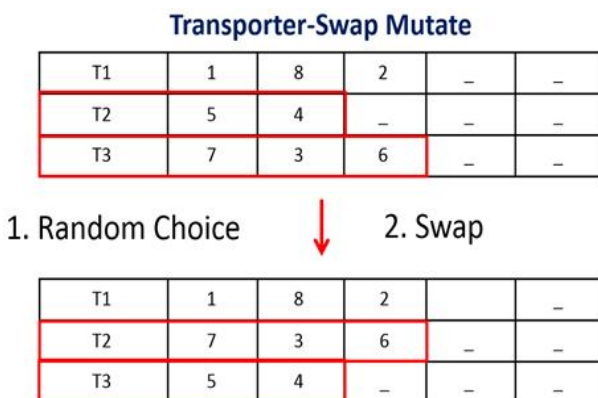


Fig. 5. Example of Transporter-Swap Mutate

3.5 Mutation

변이 연산은 교차 연산으로부터 얻은 자식 해를 비교적 작은 확률로 변형시킨다. 변형된 새로운 해는 변형 전보다 적합도가 떨어질 수도 있지만 새로운 문제공간을 탐색하는 데 도움이 된다. 본 연구에서 두 가지 변이 연산 Block-Move Mutate와 Transporter-Swap Mutate를 구현하였다. 전자는 어느 한 블록이 할당된 트랜스포터 및 운반 순서를 변경하는 것이고 후자는 한 쌍의 트랜스포터들에 대해 역할을 교환하는 것이다. Figure 4와 5는 이러한 변이 연산의 예를 각각 보여준다.

Figure 4에서 보듯이 자식 해에서 트랜스포터 T_2 의 블록 5번이 임의로 선택되어 트랜스포터 T_1 으로 이동되게 된다. 반면 Figure 5에서는 T_2 와 T_3 이 한 쌍으로 낮은 확률로 선택되어 운반 블록 리스트 전체가 교환된다.

3.6 Local Search

앞서 언급한 바와 같이 본 연구에서는 유전 알고리즘의 낮은 지역 탐색 효과를 보완하기 위해 Figure 6과 같은 지역 탐색 연산을 구현하였다. 지역 탐색 연산은 주어진 해의 주변 문제 공간에서 탐색을 통해 지역 최적해로 개선하는 연산이다. 본 연구에서 제안하는 Local Search 연산은 작업량이 적은 트랜스포터에서 작업의 블록을 꺼내어 다른 트랜스포터의 작업 목록에 삽입함으로써 트랜스포터 운용 대수의 감소를 유도한다. Figure 6은 Local Search 연산이 수행되는 과정을 의사코드로 나타낸 것이다.

한편, K 는 지역 탐색 연산에서 블록을 옮길 두 트랜스포터를 선정할 때 결정되는 후보 트랜스포터의 개수를 의미하는 파라미터이다. Figure 7은 K 값에 따른 해의 다양성을 확인하기 위해 세대 변화에 따른 적합도의 그래프를 나타낸 것이다. 세대 변화를 확인 할 수 있도록 산점도에서 각 세대별로 점의 색상을 다르게 설정하였다. K 값이 낮게 설정되면 해가 수렴하는 정도가 높은 반면 (Fig. 7(a)), K 값이 크게 설정되면 수렴하는 정도가 낮음을 확인하였다 (Fig. 7(b)). 이는 K 값에 따라 Local Search에서 고려되는 해의 다양성이 결국 해 집단의 다양성에도 영향을 미치는 것을 의미한다.

3.7 Block Order Optimization

Figure 1에서 설명한 바와 같이 트랜스포터의 작업 할당이 최적화된 후 작업 순서를 최종적으로 다시 최적화를 진행한다. 작업 할당 탐색 과정에서도 작업 순서를 고려하지만, 정교한 성능 향상을 얻기 어렵다. 따라서 작업이 결정된 최종 해에서 작업 순서를 최적화하여 추가적인 성능

Function LocalSearch(x, K)// x : a solution// K : a threshold parameter to choose transporters in x $x' \leftarrow x$ $T \leftarrow$ List sorted in ascending order based on the number of blocks in x' $T' \leftarrow T[:K]$ // K transporters with the smallest number of assigned blocksRandomly choose $T_a, T_b \in T'$ such that $|T_a| < |T_b|$ // $|T|$ denotes the number of assigned blocks to T Let (n_1, n_2, \dots, n_k) and (m_1, m_2, \dots, m_l) a list of blocks in T_a and T_b , respectivelyRandomly choose $i \in \{1, 2, \dots, k\}$ and $j \in \{1, 2, \dots, l\}$ Move block n_i from T_a to T_b by changing $(n_1, n_2, \dots, n_{i-1}, n_{i+1}, \dots, n_k)$ and $(m_1, m_2, \dots, m_{j-1}, n_i, m_j, \dots, m)$ **if** fitness(x) > fitness(x')**return** x **else****return** x'

Fig. 6. Pseudo-Code of the Local Search Algorithm

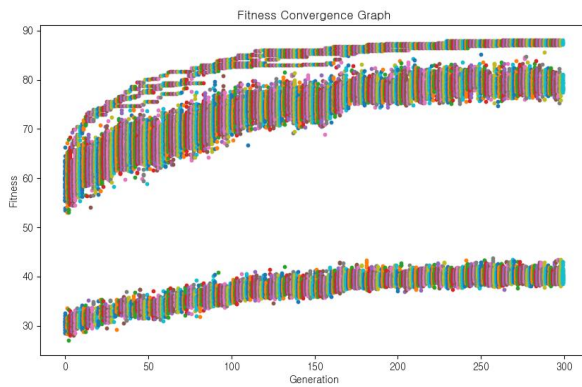
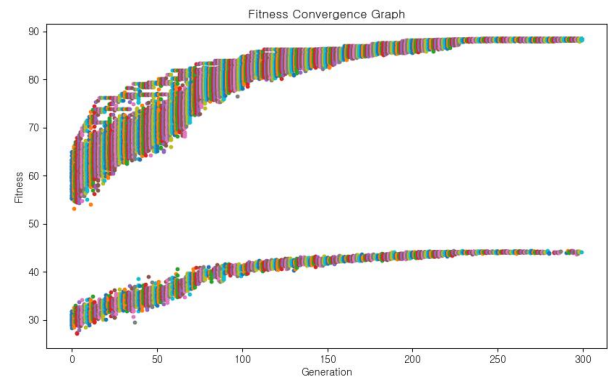
(a) $K = 15$ (b) $K = 2$

Fig. 7. Fitness-Generation Graph

향상을 얻을 수 있도록 하였다. 트랜스포터의 작업 순서를 최적화하는 작업은 순회 판매원 문제(Travelling Salesman Problem: TSP) 문제[11]로 표현된다. 본 연구에서는 트랜스포터에 할당되는 블록의 개수가 현실적으로 크지 않은 점을 고려하여 간단한 별도의 유전 알고리즘을 구현하여 문제를 간단하게 해결하였다.

IV. Experiment Results

본 연구에서 제안한 혼합형 유전 알고리즘의 효율성을 검증해보기 위해서 시뮬레이션을 진행할 가상 환경을 개발하여 실험하였다. 실제 조선소와 유사한 환경을 만들기 위해 Figure 8과 같이 특정 조선소의 지도 데이터를 이용하여 그래프로 변환하였다. 이 때 Start 노드는 모든 트랜스포터가 출발하는 노드로 가정한다. 또한 각 블록에 대해서 블록 중량, 출발지, 도착지, 운반 시간을 임의로 생성하고, 각 트랜스포터에 대해서 가용 중량, 주행 속도, 공주행 속력을 생성하였다. (입력데이터 양식은 Table 1, 2 참고)

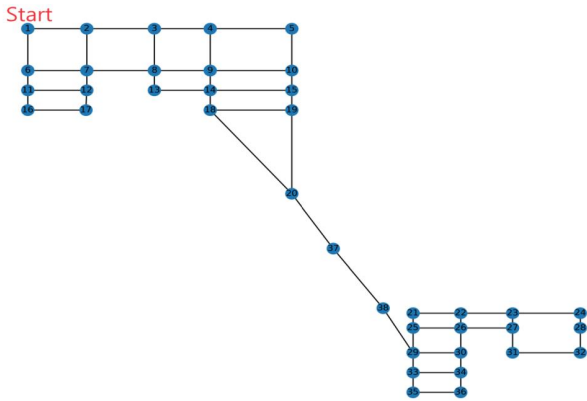


Fig. 8. Example of a Graph in a Shipyard

한편 해를 탐색할 때마다 불필요한 연산을 하지 않도록 다익스트라 알고리즘[12]을 사용하여 각 노드 간의 최단 거리 및 경로를 미리 계산해 두어 Fitness를 구할 때 활용하였다. 그 이외의 세부적인 환경설정은 Table 4, 5와 같이 설정하였다.

Table 4. Simulation Parameters

Start Time (hour)	09
End Time (hour)	18
Number of Transporters (unit)	100
Number of Nodes	30

Table 5. Our HGA Parameters

Generations	500
Population Size	100
Mutation Rate	0.05
Elite Rate	0.01

본 논문에서 제안하는 방법(HGA)을 검증하기 위해 다른 알고리즘들과 성능을 비교하였다 (Table 6). 이 표에서 Multi-Start 방법은 다수의 임의 해를 탐색한 후 그 중 최선 해를 반환하며 공정한 비교를 위해 HGA와 총탐색 횟수가 동일하도록 설정하였다. 또한 GA 방법은 기존 HGA 방법에서 지역 탐색 연산을 제거하고 최적해를 찾아가는 방법이며 그 외의 연산은 HGA와 동일하게 구성하였다. 다양한 문제 크기에 대한 비교를 위해서 블록의 개수 ($N=100, 200, 300$)를 변경하며 진행하였으며 각각에 대해 5회 실험을 통해 트랜스포터 대수와 소요 시간에 대한 평균값을 비교하였다. 표에서 보듯이 소요 시간의 경우 GA는 Multi-Start에 비해 $N=100, 200, 300$ 각각의 경우 13.74%, 12.19%, 11.77% 평균적으로 향상되었다. 이는 유전 알고리즘의 탐색이 상당히 효과가 있음을 의미한다.

다. 또한 HGA는 GA보다 26.49%, 19.26%, 17.36% 평균적으로 향상되었는데, 이는 지역탐색이 유전 알고리즘 탐색과 결합하여 더욱 효과가 있음을 의미한다. 한편 트랜스포터 대수 측면에서는 GA가 Multi-Start에 비해 $N=100, 200, 300$ 각각의 경우에 대해 26.29%, 19.05%, 8.64% 평균적으로 향상시켰고 HGA는 GA보다 42.04%, 36.4%, 33.78% 평균적으로 향상되었다. 총운행 소요 시간은 트랜스포터 운용 대수와 크게 상관이 없는 척도임을 고려할 때 HGA가 매우 안정적인 해를 탐색했음을 알 수 있다. 또한 N 값이 커짐에 따라 성능 향상률이 감소하는 데 이는 문제의 난이도가 매우 커졌기 때문으로 추정되며 향후 개선의 여지가 클 것으로 예상된다.

V. Conclusions

본 논문에서는 조선소에서 블록 운송 작업에 투입되는 트랜스포터의 대수를 최소화하기 위해 트랜스포터 작업 할당을 최적화하는 방법으로 HGA 방법을 제안하였다. 조립 공장간 최단 거리를 얻기 위해 Dijkstra 알고리즘을 적용하였으며, 유전 알고리즘에서 국부적 탐색을 강화하기 위해 지역 탐색 기능을 탑재하였다. 그리고 제안한 알고리즘에 대한 효용성을 검증하기 위해, 다양한 크기의 문제를 설정하고 이를 유전 알고리즘과 Multi-Start 두 알고리즘과 비교 실험해보았다. 그 결과 본 연구에서 개발한 알고리즘이 다른 두 알고리즘보다 효과적으로 작용함을 확인할 수 있었다. 하지만 본 논문에서는 실제 도로와 교차로에서의 충돌 상황, 트랜스포터가 블록을 옮기는 작업의 효율 등 현실적인 요소를 반영하는 데 한계가 있었다. 또한 실제 상황에서 도로가 막히는 등의 동적인 상황을 고려해야 하는 경우가 생긴다. 향후 연구에서는 트랜스포터가 이동하는 공간을 다차원적으로 고려하여 실제 환경에서도 적용가능한 방법에 대하여 진행할 예정이다.

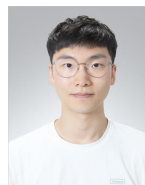
ACKNOWLEDGEMENT

This work was supported by Institute of Information & Communication Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2020-0-00869, Development of 5G-based Shipbuilding & Marine Smart Communication Platform and Convergence Service). This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2023RIS-003).

REFERENCES

- [1] Yeon mi Suh, Myung-Il Roh, and Doo-Yeon Cho, "A Study on the Optimal Operation Planning of a Transporter Using the Optimization Technique," *Korean Journal of Computational Design and Engineering Conference Paper Collection*, pp. 187-192, Jan. 2011.
- [2] Sun-Bin Yim, Myung-Il Roh, Ju-Hwan Cha and Kyu-Yeul Lee, "Optimal Block Transportation Scheduling Considering the Minimization of the Travel Distance without Overload of a Transporter," *Journal of the Society of Naval Architects of Korea*, Vol. 45, No. 6, pp. 646-655, Dec. 2008. DOI: 10.3744/snak.2008.45.6.646
- [3] Cheol Min Joo, "Determination of the number and schedule of transporters for block transportation in shipbuilding," *Journal of the Korean Society of Supply Chain Management*, Vol. 13, No. 2, pp. 143-150, Oct. 2013.
- [4] Ahmed Maghawry, Rania Hodhod, Yasser Omar and Mohamed Kholief. "An approach for optimizing multi-objective problems using hybrid genetic algorithms," *Soft Computing*, Vol. 25, pp. 389-405, 2021. DOI:10.1007/s00500-020-05149-3
- [5] Hossam Magdy Balaha, Mohamed Saif, Ahmed Tamer and Ehab H. Abdelhay, "Hybrid deep learning and genetic algorithms approach (HMB-DLGAHA) for the early ultrasound diagnoses of breast cancer," *Neural Computing and Applications*, Vol. 34, pp. 8671-8695, 2022. DOI:10.1007/s00521-021-06851-5
- [6] Gagan Deep Singh, Manish Prateek, Sunil Kumar, Madhusi Verma, Dilbag Singh and Heung-No Lee, "Hybrid Genetic Firefly Algorithm-Based Routing Protocol for VANETs," *IEEE Access*, Vol. 10, pp. 9142-9151, 2022, DOI: 10.1109/ACCESS.2022.3142811.
- [7] Jae-Young Shin and Na-Hyun Bak, "Transporter Scheduling with Transporter Combination in Shipbuilding," *Journal of Navigation and Port Research*, Vol. 38, No. 3, pp. 299-305, Jun. 2014. DOI: 10.5394/KINPR.2014.38.3.299
- [8] Woon-Seek Lee, Won-Il Lim, and Pyung-Hoi Koo, "Transporter scheduling based on a network flow model for dynamic block transportation environment," *IE interfaces*, Vol. 22, No. 1, pp. 63-72, Mar. 2009.
- [9] Jong-Heon Moon, Won-Sun Ruy and Doo-Yeoun Cho, "An Operation Scheduling of Transporters Considering Turns and Passing Delay at the Intersection Roads on the Shipyard," *Journal of the Society of Naval Architects of Korea*, Vol. 54, No. 3, pp. 187-195, Jun, 2017. DOI: 10.3744/SNAK.2017.54.3.187
- [10] Jong-Heon Moon, Won-Sun Ruy, and Ju-Hwan Cha, "Comparison of Optimal Path Algorithms and Implementation of Block Transporter Planning System," *Journal of the Society of Naval Architects of Korea*, Vol. 53, No. 2, pp. 115-126, Apr. 2016. DOI: 10.3744/snak.2016.53.2.115
- [11] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi, "The traveling salesman problem," *Handbooks in operations research and management science*, Vol. 7, pp.225-330, 1995. DOI: 10.1016/S0927-0507(05)80121-5
- [12] Edsger Wybe Dijkstra, "A note on two problems in connexion with graphs," *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pp. 287-290, July 2022. DOI: 10.1145/3544585.3544600

Authors



Jun-Ho Park is supposed to receive the B.S. degree in Computer Science and Engineering from University of Ulsan, Korea, in 2024. He is currently a B.S student in School of IT Convergence, University of Ulsan.

He is interested in genetic algorithm and optimization.



Yung-Keun Kwon received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Seoul National University, Korea, in 1999, 2001 and 2006, respectively. He is currently a Professor in Department of

Electrical, Electronic and Computer Engineering, University of Ulsan. He is interested in genetic algorithm and optimization.