

효과적인 차량 선적을 위한 공간 배치의 최적화 기법

차주형¹ · 최진석² · 배유수³ · 우영운^{4*}

An Optimization Method of Spatial Placement for Effective Vehicle Loading

Joo Hyoung Cha¹ · Jin Seok Choi² · You Su Bae³ · Young Woon Woo^{4*}

¹Undergraduate Student, Division of Creative Software Eng., Dong-eui University, Busan 47340, Korea

²General Manager, Logistics System Institute, Total Soft Bank Ltd., Busan 47340, Korea

³Staff, Logistics System Institute, Total Soft Bank Ltd., Busan 47340, Korea

^{4*}Full Professor, Division of Creative Software Eng., Dong-eui University, Busan 47340, Korea

요 약

이 논문에서는 차량 운반선에서 선적 계획과 하적 계획에 따른 차량 선적을 효과적으로 진행하기 위하여, 선박 내 데크에 차량을 효율적으로 배치하는 최적화 기법을 제안하였다. 이를 위해, 선박의 공간 정보를 나타내는 XML 데이터의 변환, 병합 및 분할 알고리즘, 유전자 알고리즘을 활용하였으며, 또한 최적화된 차량 배치 결과를 시각화하는 기능까지 구현하였다. 기존의 전형적인 유전자 알고리즘에서 사용되는 선택, 교차, 변이, 엘리트 보존 등의 기법들을 활용하였으며, 특히 차량의 선적을 위한 선박 공간을 병합 및 분할하는 기법을 함께 제안하여 차량 배치 최적화 기법을 제안하였다. 실험 결과, 기존의 유전자 알고리즘만으로 최적화하기 힘든 부분에 제안한 병합 및 분할 기법을 적용하는 것이 최적화 과정에 효과적이었음을 확인할 수 있었다. 또한, 시각화 기법을 통해 차량 배치 결과를 도면 형태로 보여줌으로써 배치 결과의 효율성을 전문가가 쉽게 판단할 수 있도록 하였다.

ABSTRACT

In this paper, we proposed an optimization technique for efficiently placing vehicles on decks in a vehicle-carrying ship to efficiently handle loading and unloading. For this purpose, we utilized the transformation method of the XML data representing the ship's spatial information, merging and branching algorithm and genetic algorithm, and implemented the function to visualize the optimized vehicle placement results. The techniques of selection, crossover, mutation, and elite preservation, which are used in the conventional genetic algorithms, are used. In particular, the vehicle placement optimization method is proposed by merging and branching the ship space for the vehicle loading. The experimental results show that the proposed merging and branching method is effective for the optimization process that is difficult to optimize with the existing genetic algorithm alone. In addition, visualization results show vehicle layout results in the form of drawings so that experts can easily determine the efficiency of the layout results.

키워드 : 공간 배치, 차량 선적, 유전자 알고리즘, 병합 및 분할, 시각화

Keywords : Spatial placement, Vehicle loading, Genetic algorithm, Merging and branching, Visualization

Received 21 December 2019, Revised 25 December 2019, Accepted 3 January 2020

* Corresponding Author Young Woon Woo(E-mail: ywwoo@deu.ac.kr, Tel:+82-51-890-1712)

Full Professor, Division of Creative Software Eng., Dong-eui University, Busan 47340, Korea

Open Access <http://doi.org/10.6109/jkiice.2020.24.2.186>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

최근 차량을 운반하는 선박의 대형화가 진행되면서 하나의 차량 선적 화물선이 여러 항구를 거쳐 일주하면서 대량의 차량을 선적하고 하적하는 업무가 빈번히 발생하고 있다[1]. 하지만 선적하고자 하는 차량의 종류가 다양하고 선적, 하적이 항구마다 빈번히 일어남에 따라 선박의 어떤 공간에 어떤 목적지의 차량을 어떤 배치로 선적해야 하는 것은 복잡한 업무이다. 현재는 이 분야에 오래 근무한 전문가가 업무 경험을 통해 차량 선적, 하적 계획을 장시간에 걸쳐 수립하고 있다.

하지만 무역 구조가 복잡해짐에 따라 선박의 운항 도중에도 새로운 선적 요구와 하적 요구 사항이 발행하는 등 차량의 배치를 최적화하기 어려운 장애물이 많아 전문가라고 할지라도 차량 선적 계획을 체계적이고 효율적으로 하기가 어려운 현실이다[2].

그러나 최근 인공지능 기술이 발전함에 따라 인공지능 알고리즘에 의해 공장이나 선박에서 다양한 화물 배치를 자동으로 최적화하기 위한 기법들이 많이 개발되고 있다. 특히 선박에서 화물 배치를 자동으로 최적화하기 위해 유전자 알고리즘을 활용한 기법도 제안된 바 있다[3]. 하지만 이 논문에서 제안된 기법에서는 규칙적인 사각형 형태의 대형 화물을 하나의 데크에 배치하기 위한 기법으로서 여러 개의 데크를 갖는 차량 운반선 등에는 적용하기 어려운 점이 있다.

또한, 공장 자동화와 관련하여 화물 팔레트를 사각형 구조의 공장 내부에 자동으로 최적 보관 면적을 산출하는 기법도 제안된 바 있다[4]. 그러나 이 논문에서 제안된 기법 또한 대상 공간도 사각형이고 화물 팔레트 또한 사각형으로 가장 단순한 형태에서의 최적화 문제로 접근하고 있기 때문에, 일반적인 배치 최적화 문제에는 적용하기 어렵다는 단점을 가지고 있다.

따라서 이 논문에서는 이러한 제약 사항을 극복하고 데크를 여러 개 갖는 대형 차량 운반선을 위한 인공지능 기법을 제안하였다. 먼저, 차량 운반선에서 선적 계획과 하적 계획에 따른 차량 선적을 효과적으로 진행하기 위하여, 선박 내 데크에 차량을 효율적으로 배치하기 위하여 선박의 공간 정보를 나타내는 XML 데이터를 변환하는 모듈을 구현하였고, 선박 데크 공간의 효율적인 처리를 위한 병합 및 분할 알고리즘을 제안하였으며, 유전자 알고리즘을 활용하여 차량 배치를 최적화했다. 더욱이

최적화된 차량 배치 결과를 시각화하는 기능을 구현하여 배치 결과의 효율성을 전문가가 쉽게 판단할 수 있도록 하였다. 기존의 전형적인 유전자 알고리즘에서 사용되는 선택, 교차, 변이, 엘리트 보존 등의 기법들을 그대로 활용하였으며, 특히 차량의 선적을 위한 선박 공간을 병합 및 분할하는 새로운 기법을 제안하였다. 2장에서는 이 논문에서 제안한 차량 배치 최적화 기법에 대해 제시하고, 3장에서는 실험 결과를 제시하였다. 끝으로 4장에서 결론 및 향후 연구 계획을 제시하였다.

II. 제안한 차량 배치 최적화 기법

여기에서는 차량 운반선에서 각 항구별 선적, 하적 계획에 따른 차량 선적 위치를 자동으로 결정하기 위한 제안 기법에 대하여 단계별로 설명하였다.

2.1. 데이터 전처리(Data Preprocessing)

차량 운반선의 전체 구조를 정의한 데이터는 데크 번호, 차량을 선적하기 위한 공간의 크기, 접촉해서는 안 되는 부분, 장애물들의 위치, 데크간 이동을 담당하는 램프 등의 영역 정보들이 포함되어 있으며, 모든 데이터들이 XML 형식으로 구성되어 있다[5]. 이들 중 선박의 벽면 형태와 각 층으로 이동하는 경로와 같은 상세한 데이터가 폴리곤(polygon) 형식으로 데이터가 구성되어 있다. 폴리곤으로 구성된 데이터를 x, y 좌표(미터 단위) 형식으로 입력받아 이 좌표를 이용해 시각화를 진행하면 선박 형태에 대한 도면을 생성할 수 있다. 이를 통해 실제 현업에서 사용되는 데이터를 활용하여 차량을 선적할 수 있는 최대 공간을 탐색하게 된다. XML 데이터를 이용하여 생성한 선박 구조에 대한 도면은 그림 1과 같다.

현업에서는 차량 선적을 위한 기본 단위로 AEU라는 것을 사용하는데 이 단위의 크기는 4.115m x 1.62m 크기를 의미한다. 모든 차량 운반선 선박에서는 AEU 단위를 활용하여 필요 공간을 할당하고, 차량을 배치한다. 선박 구조가 파악된 이후에는 그림 1과 같은 선박 구조도에서 실제 차량을 배치할 수 있는 선적 가능 공간을 파악하여 AEU 단위로 몇 대의 차량을 배치할 수 있을 것인지를 결정하게 된다.

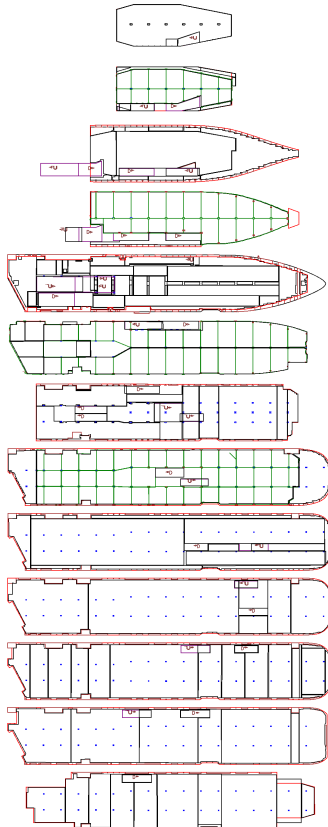


Fig. 1 Overall Deck Shapes of a Car-carrying Ship

그림 1의 선박 도면에서 선적 가능한 공간만을 대상으로 AEU 단위의 차량을 선적한다고 할 때 차량간 간격 (margin)을 어떻게 결정하는가에 따라 선적 가능한 차량 대수가 달라짐을 알 수 있다. 그림 2는 장애물 영역을 제외하고 간격을 주지 않았을 경우와 1m의 간격을 할당한 경우의 최대 가능 선적 배치도를 나타내고 있다. 그림에서 알 수 있듯이 차량간 간격을 1m로 설정한 경우에는 그렇지 않은 경우보다 선적 가능 차량 대수가 절반 정도로 줄어드는 것을 알 수 있다.

다음으로 수행하게 되는 전처리하는 그림 3에 나타나 있는 것처럼, 산재되어 있는 선박의 운항 경로 및 차량의 POL(Port Of Loading), POD(Port Of Discharge)[6] 데이터를 모은 뒤 정렬하여 차량의 크기에 따른 필요 공간 분량을 계산하는 알고리즘을 구현하기 위한 선박 운항 경로를 운항 순서에 따라 정렬된 XML 형식으로 변환하는 것이다.

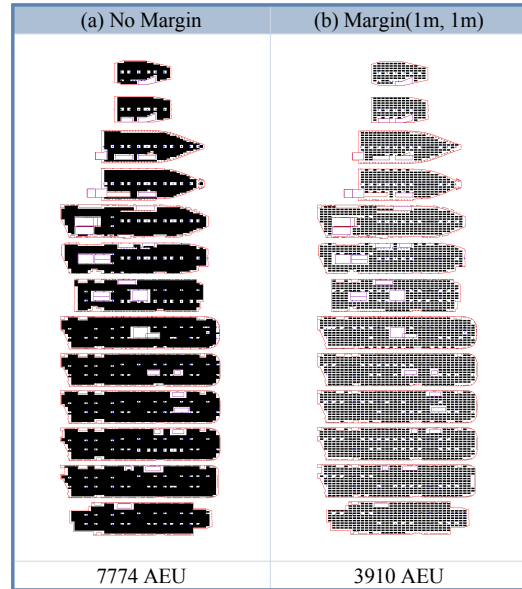


Fig. 2 Comparison of Number of Loadable Cars by No Margin and 1m Margin

```

<Sequence>
  <Port CountryCode="KR" POL="True" POD="False">KRUSN</Port>
  <Port CountryCode="CN" POL="True" POD="False">CNXGG</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRPTK</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRKUV</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRMOR</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRKAN</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRMAS</Port>
  <Port CountryCode="KR" POL="True" POD="False">KRINC</Port>
  <Port CountryCode="PH" POL="False" POD="True">PHBTG</Port>
  <Port CountryCode="SG" POL="True" POD="True">SGSIN</Port>
  <Port CountryCode="ZA" POL="False" POD="True">ZADUR</Port>
  <Port CountryCode="ZA" POL="False" POD="True">ZAPLZ</Port>
  <Port CountryCode="PH" POL="False" POD="True">PHDAK</Port>
  <Port CountryCode="ZA" POL="False" POD="True">ZAVIG</Port>
  <Port CountryCode="ZA" POL="False" POD="True">ZABRS</Port>
  <Port CountryCode="FR" POL="False" POD="True">FRLEH</Port>
  <Port CountryCode="BE" POL="False" POD="True">BEANR</Port>
  <Port CountryCode="BE" POL="False" POD="True">BEZEE</Port>
  <Port CountryCode="DE" POL="False" POD="True">DEBRV</Port>
</Sequence>
    
```

Fig. 3 Sail Route of a Car-carrying Ship

마지막 전처리 과정은 이동 항해 경로에 따른 필요 AEU와 POL을 기준으로 먼저 항구에 도착하는 곳을 기준으로 정렬하고 POL 내에서 POD로 가장 근접한 곳에 하역하는 곳을 기준으로 정렬하는 것이다. 그 결과는 그림 4와 같으며 데이터를 정렬한 후 작업자가 쉽게 알아볼 수 있도록 XML 형식의 데이터로 표현하고 관련 프로토콜을 정의하게 된다.

```

    <Port-Info>
    <POL>KRPTK</POL>
    <Arrives>
    <POD TotalAeu="195">PHBTG</POD>
    <POD TotalAeu="116">SGSIN</POD>
    <POD TotalAeu="344">PHDAK</POD>
    <POD TotalAeu="358">FRLEH</POD>
    </Arrives>
    </Port-Info>
    <Port-Info>
    <POL>KRKUV</POL>
    <Arrives>
    <POD TotalAeu="292">SGSIN</POD>
    <POD TotalAeu="537">ZADUR</POD>
    <POD TotalAeu="14">PHDAK</POD>
    </Arrives>
    </Port-Info>
    <Port-Info>
    <POL>KRMOK</POL>
    <Arrives>
    <POD TotalAeu="44">PHBTG</POD>
    <POD TotalAeu="33">SGSIN</POD>
    <POD TotalAeu="37">FRLEH</POD>
    </Arrives>
    </Port-Info>
  
```

Fig. 4 Organized Data for Loading and Unloading of Cars Using POD, POL Lining Up

2.2. 제안한 차량 배치 최적화 알고리즘

선박 공간에 대한 차량 선적을 최적화하기 위하여 딥러닝을 사용을 시도하였으나 샘플 데이터의 부족으로 인하여 적합도(fitness) 함수를 활용하여 최적의 모습을 찾아가는 알고리즘인 유전자 알고리즘을 선택하게 되었다[7].

기본적인 유전자 알고리즘을 그대로 사용하게 될 경우, 차량 선적 최적화를 1AEU 단위로 하게 되므로 많은 시간이 소요될 것으로 예상하여 새로운 기법을 제안하였다. 즉, 1AEU 단위로 최적화하는 것이 아닌 10AEU × 10AEU 또는 5AEU × 5AEU 크기와 같이 영역, 즉 구역 별로 공간을 크게 만들어 빠른 시간 내에 최적해를 찾아내는 알고리즘을 제안하였다. 예를 들어 1층 데크의 공간을 5AEU × 5AEU 크기로 영역을 분할하였다고 가정할 경우에 분할된 공간의 좌표와 선적 가능 차량 대수를 변환한 XML 데이터는 그림 5와 같이 생성된다.

```

    <Floor-1>
    <Block X="0" Y="0">23</Block>
    <Block X="1" Y="0">24</Block>
    <Block X="2" Y="0">14</Block>
    <Block X="3" Y="0">1</Block>
    <Block X="0" Y="1">21</Block>
    <Block X="1" Y="1">21</Block>
    <Block X="2" Y="1">20</Block>
    <Block X="3" Y="1">5</Block>
    <Block X="0" Y="2">22</Block>
    <Block X="1" Y="2">11</Block>
    <Block X="2" Y="2">7</Block>
    <Block X="3" Y="2">1</Block>
    </Floor-1>
  
```

Fig. 5 XML Data Generated by Dividing 1st Floor of the Decks by 5AEU × 5AEU Block Size

모든 영역의 배치 가능 차량 대수가 동일하게 산출되지 않는 것은, 같은 크기의 5AEU × 5AEU 영역이라 하더라도 선적할 수 없는 장애물 공간인 램프(ramp), 기둥 (beam) 등과 같이 선적이 불가능한 공간이 포함되어 있는 경우, 이것을 배제하고 결과값을 도출하기 때문이다.

일반적인 유전자 알고리즘은 선택, 교차, 변이, 엘리트 보존 등을 활용하는 알고리즘[8]이지만, 공간 최적화를 수행하는 과정에서는 일반적인 교차 방법을 수행하는 것이 목적에 부합하지 않는 결과값이 도출되는 경우가 있어, 이를 배제하고 교차 역할과 동일한 기능을 수행하는 새로운 기법을 제안하였다.

이 논문에서 제안한 유전자 알고리즘에서의 교차 역할을 대신하는 제안 알고리즘은 공간의 병합 및 분할 기법을 활용하는 것이다. 공간의 병합 및 분할을 반복하여 공간 최적화를 찾아가는 이 알고리즘을 이 논문에서는 병합 및 분할(merging and branching) 알고리즘이라고 명명하였다.

병합 및 분할에서 병합은 A 공간이 B 공간으로 합쳐진다고 할 때 POD, 즉 도착 지점이 같은 데이터 기준으로 묶는다. 만약에 B 공간이 부족하다면, 즉 더 이상 병합이 불가능하다면 다시 A 공간을 병합하다가 실패한 크기를 저장한다. 분할은 A 공간의 50%를 자른 뒤 B 공간으로 이동하는 것을 의미한다. 분할은 이미 존재했던 공간에 병합하는 것이 아니라 완전히 새로운 공간에 할당한다. 만약 새로운 공간이 없어 할당이 불가능한 경우라면 분할을 하였던 A 공간으로 다시 복귀한다.

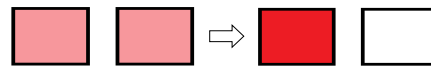


Fig. 6 Merging Process

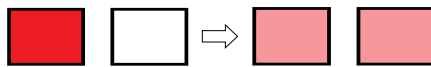


Fig. 7 Branching Process

그림 6과 7은 병합 및 분할 과정을 설명하는 그림이다. 각 그림에서, 색이 옅은 공간은 차량이 일부만 배치되어 적은 양을 차지하고 있다는 것을 의미하며, 색이 짙은 공간은 차량이 많이 배치되어 거의 모든 공간을 차지하고 있다는 것을 의미한다. 이 논문에서의 배치 최적화 기법은 이 병합 및 분할 알고리즘이 핵심이며, 이 알고리즘과 유전자 알고리즘의 적합도 함수값을 함께 이용하여 최적화를 진행하게 된다.

III. 실험 결과

이 논문에서 실험을 위하여 차량 배치 공간을 최적화 하기 위한 유전자 알고리즘에서 적합도 함수는 사용된 블록의 수에 따라서 값의 변화가 생기도록 수식 (1)과 같이 정의하였다. 이 수식에서 ExitFloor는 선적한 화물이 바깥으로 나가는 층을 의미하며, $Block_N$ 의 Floor 층을 뺀 값을 제공하여 먼 지역의 POD 화물이 ExitFloor에 가까이에 있다면 매우 높은 가중치를 주었으며, 사용되지 않은 블록이 많으면 500이라는 추가 가중치를 부여하였다. 여기에서 가중치는 낮을수록 좋은 유전자를 의미한다.

$$Fitness = \sum_{N=0}^{useBlockCount} (Block_N - ExitFloor)^2 + (unUseBlockCount * 500) \quad (1)$$

즉, 병합 및 분할 알고리즘을 활용하고 거리가 가까울수록 좋은 유전자를 의미하는 적합도 함수를 통하여 최적화된 차량 선적 데이터를 찾아내는 알고리즘을 구현하였다. 구현된 알고리즘에 의한 최적화 과정의 1세대 결과와 1,000세대 진화 결과는 그림 8과 같다. 또한, 실제 현업에서 사용하는 화물 예약 정보 데이터를 추가로 활용하되 공간을 10AEU × 10AEU와 15AEU × 15AEU로 분할하여 실행한 최종 진화 결과는 그림 9와 같이 나타났다.

IV. 결론

병합 및 분할 알고리즘과 유전자 알고리즘을 활용하여 차량 배치를 최적화하는 알고리즘을 제안하고 실험 결과를 제시하였다. 최적화를 위한 진화 과정을 통하여 최선의 적합도 값으로 바뀌는 변화를 확인할 수 있었다. 향후 적합도 함수를 더욱 개선하여 제약 조건, 상태 조건과 같이 여러 가지의 상태와 필요한 조건을 추가 구현한다면 더욱 우수한 결과를 나타낼 수 있을 것으로 예상된다. 또한, 병합 및 분할 알고리즘에서 공간 위치를 강제적으로 교환하는 알고리즘 등을 추가하는 방식을 통해 부족한 공간은 그 공간만큼 분할하도록 개선한다면 더욱 빠르게 우수한 유전자를 찾아갈 수 있을 것으로 예상된다.

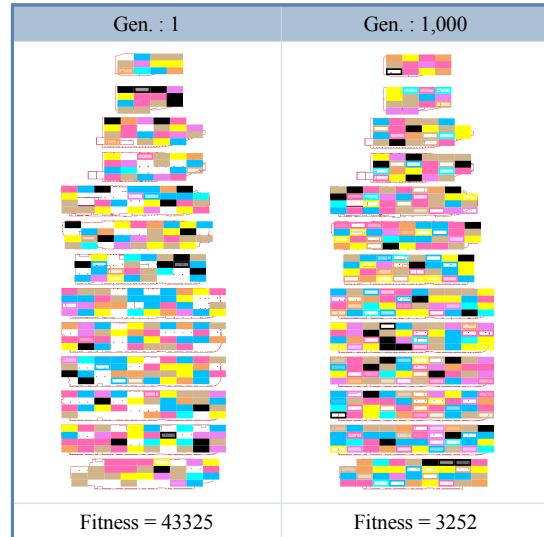


Fig. 8 Results of Optimization on Car Placement in a Car-carrying Ship having Multi-deck

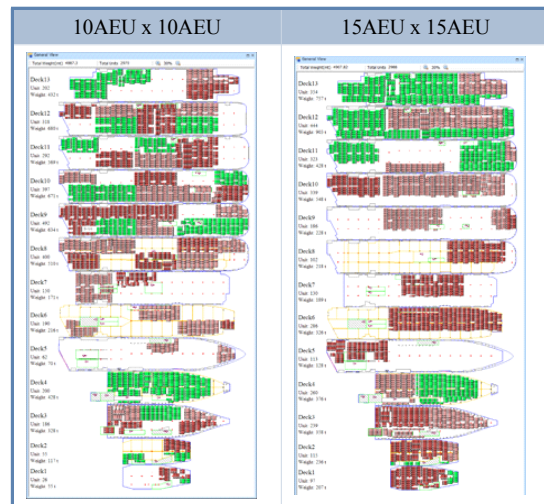


Fig. 9 Results of Optimization on Car Placement by 10AEU × 10AEU and 15AEU × 15AEU

ACKNOWLEDGEMENT

This work was supported by National IT Industry Promotion Agency(NIPA) grant funded by the Korea government(MSIT). (S0249-19-1021, Development of loading optimization and ship operation safety support service for vehicle carriers).

REFERENCES

- [1] Shipping, Hyundai Glovis [Internet]. Available: <https://www.glovis.net/Eng/business/contentsid/304/index.do>.
- [2] S. K. Kim, K. S. Kim, J. W. Hong, S. Ha, and M. I. Roh, "Integrated Method for Optimal Layout of Offshore Plant Topside," *Proc. of The Proceedings of Korean CAD/CAM Winter Conference*, pp. 287-291, 2015.
- [3] K. Y. Lee, S. N. Han, and M. I. Roh, "Optimal Compartment Layout Design for a Naval Ship Using an Improved Genetic Algorithm," *Marine Technology*, vol.39, no.3, pp. 159-169, Jul. 2002.
- [4] J.-H. Park, D.-M. Kim, and G.-T. Yeo, "A Study on the Estimation of Optimal Storage Area in the Distribution Center," *Journal of Digital Convergence*, vol.17, no.7, pp. 59-71, Jul. 2019(<https://doi.org/10.14400/JDC.2019.17.7.059>).
- [5] XML Tutorial, w3schools.com. [Internet]. Available : www.w3schools.com/xml/.
- [6] Shipping Terms, Carry Cargo International. [Internet]. Available : www.carrycargo.com/shipping-terms/.
- [7] J. H. Cha, Y. W. Woo, and I. Lee, "An Effective Method for Generating Images Using Genetic Algorithm," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 8, pp. 896-902, Aug. 2019 (<https://doi.org/10.6109/jkiice.2019.23.8.896>).
- [8] A. J. Umbarkar, and P. D. Sheth, "Crossover Operators in Genetic Algorithms: A Review," *ICTACT Journal on Soft Computing*, vol.6, no.1, pp. 1083-1092, Oct. 2015 (<https://doi.org/10.21917/ijsc.2015.0150>).



차주형(Joo Hyoung Cha)

동의대학교 창의소프트웨어공학부(2018~현재)

※ 관심분야 : 인공지능, 유전자 알고리즘, 신경망



최진석(Jin Seok Choi)

부산외국어대학교 컴퓨터학과(1999)
부산외국어대학원 전산교육학과 수료(2002)
(주)토탈소프트뱅크(2010~현재)

※ 관심분야 : 소프트웨어 아키텍처, 빅데이터, 인공지능, 유전자 알고리즘



배유수(You Su Bae)

부산대학교 IT응용공학과(2018)
(주)토탈소프트뱅크(2018~현재)

※ 관심분야 : 인공지능, 빅데이터, 항만물류시스템



우영운(Young Woon Woo)

연세대학교 전자공학과 공학석사(1991)
연세대학교 전자공학과 공학박사(1997)
동의대학교 창의소프트웨어공학부(1997~현재)

※ 관심분야 : 인공지능, 빅데이터, 퍼지이론, 의료정보