University of Ibadan
Journal of Science and Logics in ICT Research

Genetic Algorithm Based Space-Optimised Arrangement of Containers and Stability in Containerships

[*1]Ojo Abayomi Fagbuagun, [1]Olaiya Folorunsho, [2]Obinna Nwankwo, [1]Timilehin Vincent Adewole, and [3]Gregory Ugorji Eke
[1]*Department of Computer Science, Federal University Oye-Ekiti, Oye, Nigeria*
[2]*Department of Computer Science, Novena University, Ogume, Delta State, Nigeria*
[3]*Department of Artificial Intelligence and Data Science, University of Hull, United Kingdom*

Abstract

Space optimization in a container terminal to prevent space wastage and ensure the stability of containers is an optimization problem frequently experienced at container terminals. Ineffective planning of the arrangement of containers creates the problem of stability, loading and unloading containers from one port to another, leading to time wastage. In this research, the optimization of containers stowage in containership is achieved by using genetic algorithm processes of selection, crossover, mutation, and fitness in order to determine the optimal arrangement of containers in container ships. The objective function, the solution representation, and the constraints were set, and a fitness function determines how good a candidate solution is. The algorithm is able to optimize space, thereby ensuring stability and preventing time wastage that comes with loading and unloading at different ports. The experiment was conducted six times on the population size of one hundred 20-feet containers and one hundred 40-feet containers. The results show that an average percentage space utilization of 98.55% and an average time of 17.73 seconds were achieved. The algorithm is efficient for arranging any type and size of containers on containerships.

Keywords— Containerships, Crossover, Genetic Algorithm, Space-Optimized, Stowage.

1 INTRODUCTION

The maritime sector is key for countries bordered by the sea and involved in international trade with other countries of the world. The sector is a major infrastructure where international trading occurs in countries bordered by the sea [1]. The maritime sector is a major source of employment and a means of earning foreign exchange for many world countries.

In Nigeria, about 11 billion tons of goods are shipped yearly through international trade with other countries. In the year 2019 review by United Nations Conference on Trade and Development (UNCTAD), the volume of maritime trade was about 11 billion tonnes in the year 2018 [2]. Most of these trades were done through container ships throughout the world. The capacity of container ships has been increasing gradually over the years due to technological breakthroughs. According to Marine [3], Ever Ace is the world's largest container ship in the year 2022. This information is based on the ship's capacity, measured in twenty-foot equivalent units (TEUs). When such a large capacity of ships is carefully considered, space optimization of the arrangement of ships is essential to maximizing efficiency in handling containers, which is important to minimization of loading and unloading tasks at all ports of operation [4].

Furthermore, according to Gamal *et al.* [5], container terminal is a system that is open and which usually has two interfaces. The first interface is the quayside, where the loading and offloading of ships takes place; the landside, where containers are either loaded or unloaded on or off the trucks carrying them. Several logistics are involved in handling large container terminals, and this has raised a challenge that has to do with improving the techniques used in the operations at the quayside and landside by leveraging on information technology and other scientific techniques.

Ojo Abayomi Fagbuagun, Olaiya Folorunsho, Obinna Nwankwo, Timilehin Vincent Adewole, and Gregory Ugorji Eke (2023). Genetic Algorithm Based Space-Optimised Arrangement of Containers and Stability in Containerships. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 9 No. 1, pp. 41 – 51.

International trade in Nigeria and other countries of the world relies so much on shipping as a means of conveying local products such as cocoa and crude oil to the international markets. The activities involved include the containerization of these products before export. In order to facilitate easy and bulk movement of goods from one country to another, the use of containers is very important. Countries have utilized containers for many years and potentially for an extended period, and it is being continuously used to facilitate the movement of goods from source to destination. The products are initially loaded into containers before being transported to container terminals, where they are placed in a container yard. At the arrival of a ship at the quayside of the container terminal, cranes are used to load the containers onto truck that will eventually move the containers to quayside where a quay crane unloads the containers into a container ship.

In Ahmed [6], containers were described as cuboidal steel frames with a suitable length and strength to support heavy cargo in stowage and in transit. Containers come in different types and shape such as general-purpose containers, open-top containers, double-door containers, high cube containers, international standard organization (ISO) reefer containers, and tank containers. Containers also come in different dimensions, though with the same frame, which determines their usage. For containers that conform to ISO standards, the exterior dimensions are often 20ft long and 8ft wide and 8ft 6 inches high or 9ft 6 inches high for certain cube containers.

Containers always come in standardized units where the length is usually 20 feet or 40 feet. Still, longer containers are often used predominantly in the United States trade which are 45 feet, 48 feet, and 53 feet long, but their width is usually 8 feet. Some other standard shape containers can have a size of 9 feet 6 inches as stated earlier, while other comes in 10 feet 6 inches. Every vessel has a bay plan, a numbering that depicts the cross-sectional view of stacking containers above and under the deck, which helps choose the best stowage position for containers according to a plan. A row refers to the position that containers can be set across the width of a vessel, and they are often given numerical values such as 00, 02, 04, and 08 and so on, while on the starboard side, the rows are given odd numerical values such as 01, 03, 05, and so on. The tallness of containers on the deck or under the deck is indicated by the tier whose numbering starts from 8, 84, 86, 88 and goes upwards from the main tier. For containers stowed under the deck, the numbering is usually 04, 06, 08, beginning from the base level.

Different stowage plans in ships often create uncertainties in ship scheduling, making it necessary for ports to have rational stowage plans that are very effective in solving contingency problems that can come up during the operation of the ports [7]. In Huiling *et al.* [8], it is maintained that the operational efficiency of container stowage could be improved when there is a plan to reduce the number of operations required during loading and unloading containers to or from a container ship. This plan can also be used to optimize the economy of shipping as well as safety on board.

Problems of overstow and understow often arise when containers are not stowed in a way to allow for easy unloading of containers when they reach their destination. A further advantage of having a good container stowage plan is to prevent containers from falling off from ships. A model can be proposed to achieve all the needs as mentioned earlier to have an adequate stowage plan for containers on ships.

2 LITERATURE REVIEW

Genetic algorithm is a search and optimization algorithm that is solely based on the biological principle of natural selection, Haldurai *et al.* [9] and it is applicable in getting the solution to problems where the available information is limited [10]. It was proposed to optimize land use in Ding *et al.* [11], where land was allocated based on land use type. Many problem-solving approaches have been implemented when it comes to storage space allocation problem for containers on ships.

For instance, Wilson *et al.* [12] proposed a method that is based on Tabu search algorithm which ensures that a container is correctly allocated a stowage position. In Ayachi *et al.* [13], GA was implemented in the determination of optimal arrangement of containers with a view to maintaining deadlines for delivery of goods, reduction of the number of operations required on containers before delivery, and minimizing stop time of the container ship. It was discovered from the algorithm that the population size of the containers determines the fitness function of the genetic algorithm. The higher the population, the better the fitness functions. A drawback of this method is that the execution time for the algorithm increases with the population of the containers thereby making the algorithm inefficient in handling larger population of containers on container ships when timing is of essence.

Wang *et al.* [14] proposed a non-dominated sorting GA (NSGA) III that was combined with some local search components to provide a solution to container stowage problem. The results obtained shows that the hybridized GA was effective in comparison with NSGA-II and random weighted Gas when applied to many-objective stowage problems. Stability of the containers was not considered and it is a major limitation of the work.

Furthermore, GA and simulated annealing (SA) was implemented in solving container stowage, Yurtseven *et al.* [15]. The optimization problem of stowing plans was investigated for container vessels that have to visit several ports of call before reaching the final destination. The aim of the work is to minimize the number of times containers have to be moved during the journey before getting to its destination. Movement of containers comes with additional costs and the aim is to reduce the cost associated with container movement. In the paper, one type of container, specifically 40 inches long was used in the experiment. The weight and height of the containers were not considered in the work. A special feature of the work is that the number of variables used in their implementation is equal to the exact number of containers.

The result obtained showed that SA produced better result when compared to GA because SA got the optimum result faster than GA does with shorter computation period. A major limitation of the work is that the number of variables used is in direct proportion to the number of containers. Moreover, different sizes of containers should have been considered instead of single size container because a port handles several different types of containers.

In Huiling *et al.* [8], a solution to containership stowing problem was presented. In an attempt to solve the problem, integer linear programming (LP) model was considered. Four models were established named model I, model II, model II, and model IV. Model I was extended to model II and so on until model IV which gives the optimal solution. While model IV was reported to have achieved the optimal solution, a limitation of the method is that with the growing number of variables and constraints, the time taken to reach the optimal solution grows rapidly with the number of variables. In addition to this, stability of the containers was not considered in the work.

Li *et al.* [16] proposed a multi-stage hierarchical decomposition method of stowage planning in solving container stowage problem. In the work, stowing plan was implemented for each port based on the present port. When a port is considered, the stowing problem is decomposed into two: master bay planning problem (MBPP), and the slot planning problem (SPP). In MBPP, heuristic algorithms, greedy adaptive search procedure were implemented for the optimisation of the MBPP for the entire route of the ship. A heuristic evolutionary algorithm was also used for the SPP and the two produced good results but the case of stability of the containers was not considered.

3 RESEARCH METHODOLOGY

The problem of packing containers on a ship's bay is a three-dimensional (3D) packing problem, but if we assume that all the containers have the same height, we can decompose it into two-dimensional (2D) problems. We can solve each level or 'layer' independently, starting from the bottom layer and moving up until the maximum height is reached or the list of containers to be placed has been exhausted. The problem is to pack as many containers in the space as possible. Thus, for the overall setup, we want to minimize the number of unplaced containers, and for each level, we want to maximize space utilization. In a situation where all the containers are of the same length and breadth, the solution to this problem is trivial, but when different sizes of containers are involved, determining the optimal arrangement is more difficult.

3.1 The Objective

We define the objective for the overall (3D) problem, while the fitness function is defined for the sub-problem (2D). Let $C = \{C_1, C_2, \dots, C_n\}$ be the list of containers yet to be placed. Initially this is the set of all containers to be placed. As each container is placed, it is removed from C. The objective can be defined as area of the number of containers placed to the total area of all available bay area (all layers) as shown below:

$$f(x) = \frac{\sum_{i=1}^{n} w_i \times l_i}{W \times L \times H} \quad \text{............} \quad (1)$$

Where, W and L are the width and length of the bay area respectively, and H is the number of layers allowed. w_i and l_i are the width and length of individual containers respectively. The closer the value of the objective function is to 1, the better the solution.

3.2 Solution Representation (Chromosome Representation)

The choice of solution representation is vital for the task at hand. Each layer is divided into cells across both length and breadth thereby forming a grid. Each cell (gene) contains an integer which indicates which container is placed on it. A value of zero indicates that the cell is unoccupied.

Let m be the number of cells in a column and n be the number of cells in a row. Let $I = \{I_1, I_2, \dots, I_n\}$ be the list of ID of containers in C Each grid, G is defined as:

$$G_{m\times n} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{bmatrix} \quad \text{.......} \quad (2)$$

Where $g_{i,j} \in G$.

3.2.1. Constraints

Having defined the objective, the next step is to define the constraints. The most obvious constraint is that containers placed cannot overlap, that is, two containers may not be assigned to the same cell. Each g_{ij} must be single valued.

The containers are marked for different destinations (ports). The principle of last-in-first-out must be followed. This means that the list of containers marked for the longest distance must be exhausted before placing those going to the next farthest port, etc.

3.2.2. Container Placement

Having selected the grid representation for the chromosomes, there is a need to place containers in the chromosomes in the process of creating our initial population, $P(0)$, of solutions. We employ a hybrid of First Fit and Random Placement techniques.

The First Fit algorithm places the containers in the first location that fits. While this can generate solutions that are closer to optimal than some other techniques, there is tendency for the variability among the candidate solutions to be low. Therefore, to create more diverse solutions, all possible container sizes and orientations that can be placed in a position are selected, then a random choice is made from these. The pseudocode is presented below.

```
For each cell in the grid:
        Create empty list for possible placements
        For each combination of container size and orientation:
                If size and orientation fits in position:
                        Add to list
        If list is not empty:
                Randomly choose one out of list
                Place container
```

3.2.3. Fitness Function (Selection)

The fitness function measures how good a candidate solution is. One possible choice is to use the ratio of the area that has been covered by the placed containers to the area of bay as shown below.

$$f(x) = \frac{\sum_{i=1}^{n} w_i \times l_i}{W \times L} \quad \text{.....................} \quad (3)$$

Where w_i and l_i are the width and length of individual containers respectively, and W and L are the width and length of the bay respectively. This function is similar to the objective defined earlier. However, the grid representation of solutions provides a more natural measure of fitness. Since we represent unoccupied space with zeros, we can simply use the number of zeros as measure of fitness, where lesser is better.

Minimise: $f(x) = |\{x : x \in G,\ x = 0\}|$ … (4)

The relationship between the fitness function and the objective function is clearly inverse in nature.

3.2.4. Crossover Operation

The crossover operator takes chromosomes from the population at time t, $P(t)$, and creates the chromosomes for the next population $P(t+1)$. The selected technique is a one-point crossover in which two chromosomes exchange halves. That is, the first half of the first parent is joined with the second half of the second parent, while the first half of the second parent is joined with the second half of the first parent. Containers that are truncated because of this cut-and-join process are removed. A check is then performed to ensure that the resulting chromosome is valid. If the resulting chromosome has more of a size of container than available, then remove the excess. The crossover operation is presented in Figure 1.

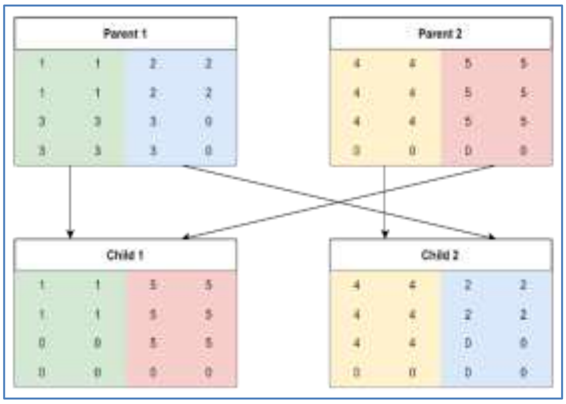


Figure. 1: Chromosome crossover

3.2.5. Mutation Operation

The mutation operation attempts to fill holes that may occur after the crossover by placing more containers using the technique. The mutation operation is thus designed to improve the candidate solution's fitness wherever possible, thereby helping the process to converge faster.

3.2.6. Stopping Criteria

More generations are iteratively created until the stopping criteria is reached. Two stopping criteria are defined:

i. A perfect chromosome with optimal fitness has been found. That is, there is no unfilled space in the chromosome. The fitness function returns 0:

$$|\{x : x \in G,\ x = 0\}| = 0 \ldots\ldots\ldots\ldots (5)$$

ii. A specified maximum number of generations has been reached. The best solution found is restored.

num_generations = max_iter

3.2.7. Aggregating Solutions

The above steps are repeated for each layer until either all the containers have been exhausted or the topmost layer has been filled. The flowchart for the system is shown in Figure 2.

4 SYSTEM TESTING

The system testing was carried out based on two criteria:

i. Percentage of used space: This is the percentage of allocated space to allocable space.

$$p_u = \frac{area\ of\ placed\ containers}{\min(area\ of\ all\ containers, area\ of\ bay)} \times 100\%. \quad \ldots (6)$$

ii. Computation time: This is the amount of time taken to arrive at a final solution. This depends on the parameters selected for the algorithm.

4.1. IMPLEMENTATION AND RESULT

The container stowage problem in this research work was implemented using python programming language version 3.6, NumPy numerical library, and Turtle graphics library. The user inputs are entered as requested at run time. At the completion of the computation, the number of containers used is reported while the arrangement of the containers is saved in a compressed file format where it can be viewed by executing the visualization script. The interface for the computation is presented in Figure 3.

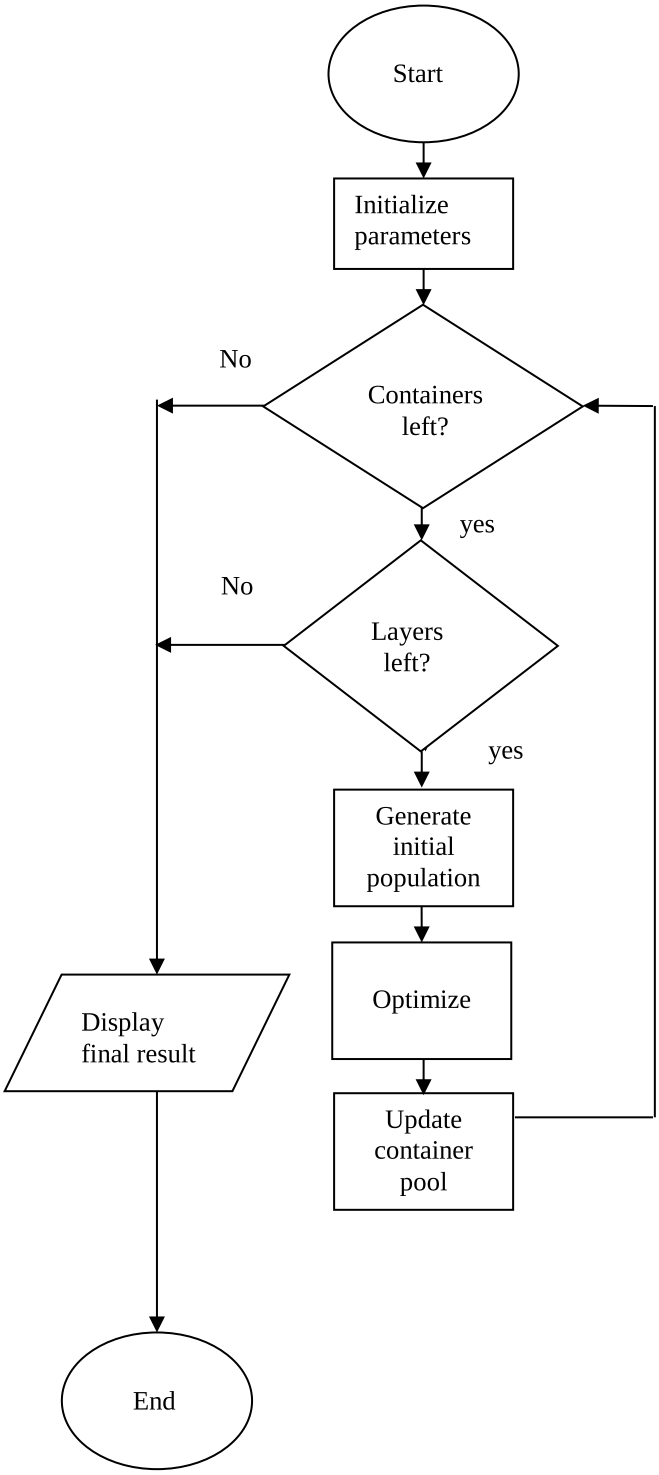
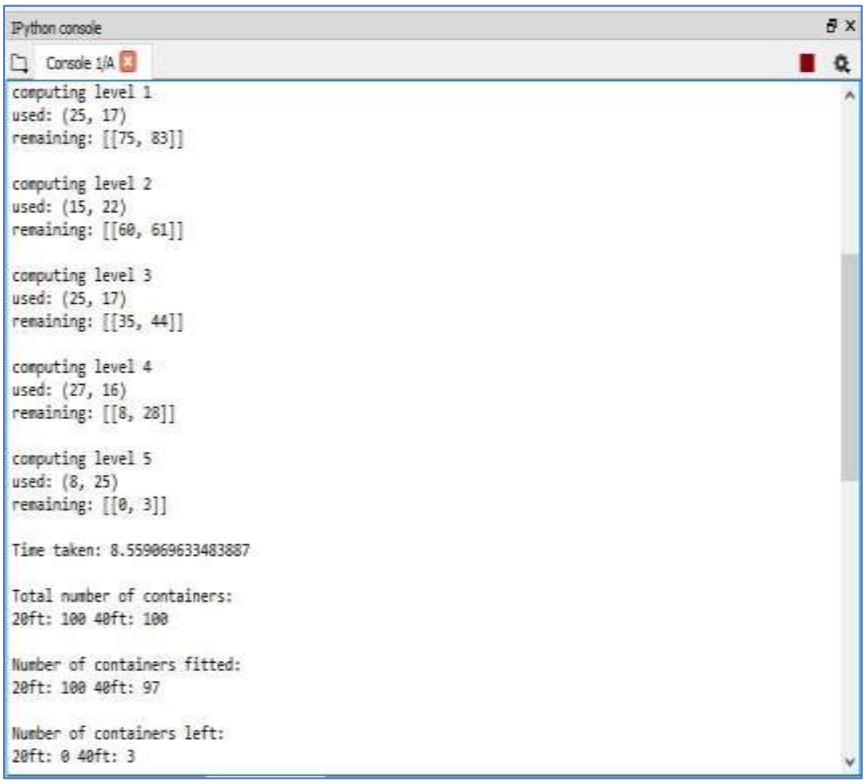


Figure. 2: Flowchart of the system

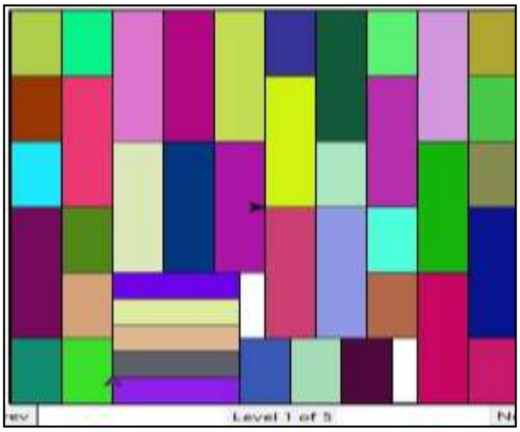


```
computing level 1
used: (25, 17)
remaining: [[75, 83]]

computing level 2
used: (15, 22)
remaining: [[60, 61]]

computing level 3
used: (25, 17)
remaining: [[35, 44]]

computing level 4
used: (27, 16)
remaining: [[8, 28]]

computing level 5
used: (8, 25)
remaining: [[0, 3]]

Time taken: 8.559069633483887

Total number of containers:
20ft: 100 40ft: 100

Number of containers fitted:
20ft: 100 40ft: 97

Number of containers left:
20ft: 0 40ft: 3
```

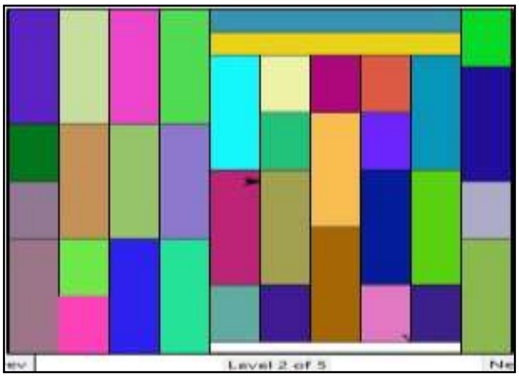
Figure 3: Genetic Algorithm computation interface

4.2. Visualization of Containers Arrangement

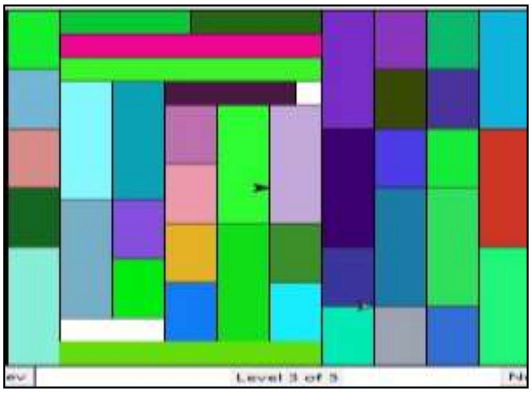
The size of the containers to be stowed, the number of layers and other inputs are entered. After the system completes executing, the arrangements of containers in each level are displayed and can be visualized. For a five-layer arrangement, for example, the visualization in presented in Figure 4(a) – (e). The colours were used to differentiate each container. White space indicate no container placed, while size indicates different container sizes.



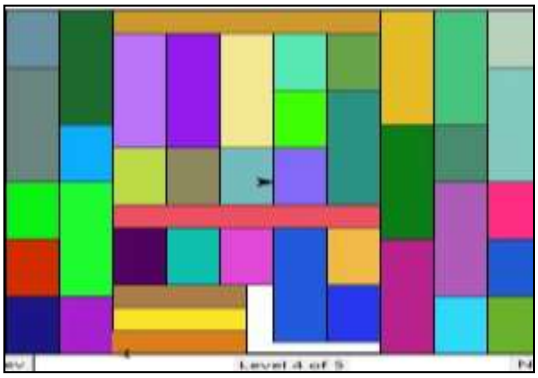
(a)


ev
Level 2 of 5
Ne

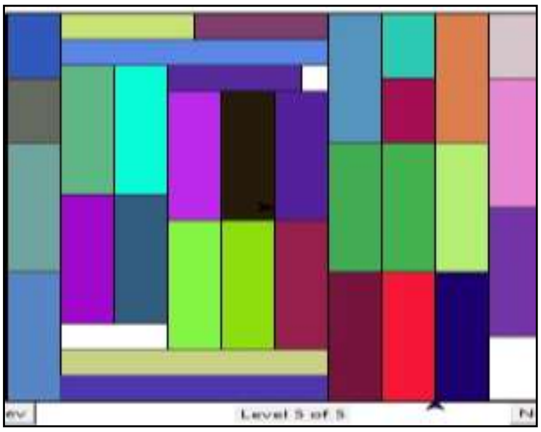
(b)


ev
Level 3 of 5
Ne

(c)


ev
Level 4 of 5
N

(d)


ev
Level 5 of 5
N

(e)

4.3. SYSTEM EVALUATION

The experiment was carried out on a hypothetical ship of bay length of 120ft, bay width of 80ft, a maximum layer of 5, and a population size of 100. The maximum number of generations is 20 while the number of 20ft containers is 100 and the number of 40ft containers is 100. The experiment was conducted 6 times. The percentage of space used and the time taken by the algorithm are recorded in Table 1. The algorithm was executed on a system with 8GB RAM, Intel Core i5 (2430) 2.4 GHz Processor, and a 64-bit Windows Operating System. The percentage space and time used in the experiment is presented in Table 1.

It can be seen from Table 1 that the algorithm achieved an average percentage space utilization of 98.55% and an average processing time of 17.73 seconds.

4.4. SYSTEM BENCHMARKING

The average time required by the Genetic algorithm is this work is 17.73 seconds while the percentage of space utilization is 98.55%. On comparison with the work of Ayachi *et al* (2010), after running for average of five instances, the average time required is 44.8 seconds. Furthermore, in [17] the average time required is 612. The benchmark is presented in Table 2.

Table 1. Percentage Space and Time used for the experiment

Trials	No. of 20ft containers	No. of 40ft containers	Space used (ft)	% space used	Time (s)
1	100	98	47360	98.66	16.92
2	100	96	46720	97.33	17.42
3	100	97	47040	98.00	19.09
4	99	100	47840	99.66	16.71
5	100	99	47680	99.33	18.34
6	99	98	47200	98.33	17.89
Average				98.55	17.73

Table 2: System Benchmark

Author	No of Containers	Instances	Average Time (secs)
Fagbuagun et al (2023)	200	6	17.73
Ayachi et al (2010)	50	5	44.8
Ural & Yunus (2016)	200	100	612

5 CONCLUSIONS AND FUTURE WORK

In this paper, genetic algorithm was used to optimize the stowage of containers on containership. The stability of containers was a major factor considered in this work as well as ensuring that containers destined for a specific port are kept for easy access in order to reduce the tedium associated with loading and unloading during transit. The proposed system can be used for the arrangement of different kinds of containers simply by supplying the attributes of the containers, such as height, width, and length, which represent the dimensions of the containers. Also, the number of containers is required, as well as the number of decks on the containership.

The proposed system is flexible because it can easily be adapted to suit the arrangement of any size of the container. Based on previous studies, the genetic algorithm can effectively prevent space wastage and ensure container stability in containerships. However, the effectiveness of this optimization technique may vary depending on some factors, including the fitness function, encoding scheme, population size, crossover and mutation rate, selection method, and termination criterion.

Additionally, previous studies have primarily relied on choosing a minimum population size for the container, thereby reducing the genetic algorithm's fitness. Future research should be on developing a mobile application for solving the problem of computer stowage on containerships.

Furthermore, the running time for the algorithm can still be improved by considering other algorithms or hybridised algorithms for better results.

6. REFERENCES

[1] Sriphrabu, P., Sethanan, K., & Arnonkijpanich, B. (2013). A solution of the container stacking problem by genetic algorithm. International Journal of Engineering and Technology, vol. 5, No.1, pp. 45-49.

[2] UNCTAD (2019). Review of maritime transport 2019; In United Nations Conference on Trade and development (2019)'. Retrieved 24 June 2022, from https://unctad.org/system/files/official-document/rmt2019_en.pdf.

[3] Marine Insight (2022). Top 10 World's Largest Container Ships in 2022' (2022). Retrieved 24 June 2022, from https://www.marineinsight.com/know-more/top-10-worlds-largest-container-ships-in-2019/.

[4] Consuelo, P. T., Hatise, C., Ramon, A. V., Reuben, R. (2021). Solving the generalized multi-port container stowage planning problem by a matheuristic algorithm, *Computers & Operations Research*, 133 (2021), 1-14. https://doi.org/10.1016/j.cor.2021.105383.

[5] Gamal, A. E., Said, A., El-Sayed, M., El-Horbaty (2015). An Optimization Methodology for Container Handling Using Genetic Algorithm, *Precedia Computer Science, 65(2015)*, 662-671. https://doi.org/10.1016/j.procs.2015.09.010.

[6] Ahmed, H. R. (2015). Containers Architecture: Reusing Shipping Containers in making creative Architectural spaces', *International Journal of Scientific & Engineering research*, 6, 11(2015), 1562-1577.

[7] Jian., J., Weijian, M. (2019). An AIMMS-Based Decision-Making Model for Optimizing the Intelligent Stowage of Export Containers in a Single Bay'; *Discrete and continuous dynamical systems series*, 12, 4 & 5 (2019), 1101-1115. https://doi.org/10.3934/dodss.2019076.

[8] Huiling., Z., Mingjun., J., Wenwen., G. (2020). Integer Linear programming models for the Containership Stowage Problem, *Hindawi: Mathematical Problems in Engineering*, (2020), 1-14. https://doi.org/10.1155/2020/4382745.

[9] Haldurai, L., Madhubala, T., Rajalakshmi, R. (2016). A Study on Genetic Algorithms and its Applications, *International Journal of Computer Science and Engineering*, 4, 10(2016), 139-143.

[10] Chen., H-C., Widodo, A. M., Wisnujati., A. (2021). Using Genetic Algorithms to Optimize The Loading Space in a Container'; *UEU-Journals* (2021). Retrieved 25th June from https://digilib.esaunggul.ac.id/public/UEU-Journal-19173-11_0947.pdf.

[11] Ding., X., Zheng, M., Zheng, X. (2021). The Application of Genetic Algorithm in Land Use Optimization Research: A review, *Land* https://doi.org/10.3390/land10050526.

[12] Wilson., I. D., Roach, P.A. (1999). Principles of combinatorial Optimization Applied to Container-Ship Stowage Planning'; *Journal of Heuristics*, 5, 4(1999), 403-418. htpps://doi.org/10.1023/A:1009680305670.

[13] Ayachi, I., Kammarti., R., Ksouri., M., Borne., P. (2010). A genetic algorithm to solve the container storage space allocation problem, *In proceedings – International conference on Computational Intelligence and vehicular system* CIVC (2010), pp.1-4.

[14] Wang, Y., Shi, G., Hirayama, K. (2022). Many-Objective Container Stowage Optimization Based on Improved NSGA-III, *Journal of Marine Science and Engineering*, 10, 517(2022), https://doi.org/10.3390/jmse10040517.

[15] Yurtseven, M.A., Boulougouris, E., Turan, O., Papadoupolous, O. (2017). Optimization of container stowage using simulated annealing and genetic algorithms, *Maritime Transportation and Harvesting of Sea Resources – Guedes Soares & Teixeira* (Eds). Retrieved 8 July 2022, from https://www.researchgate.net/profile/EvangelosBoulougouris/publication/320538568_Optimization_of_container_stowage_using_simulated_annealing_and_genetic_algorithms/links/59ea78494585151983c7f1e1/Optimization-of-container-stowage-using-simulated-annealing-and-genetic-algorithms.pdf?origin=publication_detail.

[16] Li, J., Zhang, Y., Ji, S., Xu, J. (2020). Multi-stage hierarchical decomposition approach for stowage planning problem in inland container liner shipping, *Journal of the Operational Research Society,* 71, 3(2020), 381-399. http://www.doi.org/10.1080/01605682.2018.1561162.

[17] Ural, G. C., & Yunus, K. (2016). A Genetic Algorithm for the Allocation of Dangerous Goods Containers in a Stowage yard for Freight Villages and Dry Ports. Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, pp. 264-282