Using Genetic Algorithm to Optimize The-Loading Space in a Container

Hsing-Chung Chen[1,2], Agung Mulyo Widodo[1,3], Andika Wisnujati[1,4], Mosiur Rahaman[1]

[1] Dept. of Computer Science and Information Engineering, Asia University, Taiwan,
[2] Dept. of Medical Research, China Medical University Hospital, China Medical University, Taiwan,
[3] Dept. of Computer Science, Esa Unggul University, West Jakarta, Indonesia,
[4] Dept. of Machine Technology, Universitas Muhammadiyah Yogyakarta, Indonesia,
cdma2000@asia.edu.tw, shin8409@ms6.hinet.net, agung.mulyo@esaunggul.ac.id, andikawisnujati@umy.ac.id, mosiurahaman@gmail.com

Abstract

The problems often occur in loading the contents of goods into a container, so it is needed an optimization of the goods to be loaded into a container. The settlement of the problem which generally occurs is based on the experience of the actors who arrange the layout of the goods in order that they can be contained in containers. But the loading process of goods experiences problems in the number of items that do not fit due to less effective drafting patterns. This is certainly less effective, so it is proposed a solution using an artificial intelligence in the form of a genetic algorithm to optimize the layout of goods in containers. This algorithm aims to determine the order of optimal laying of goods based on the dimensions of goods data (length, width, height), weight of goods and types of goods. From the simulation that is carried out, obtained the result the optimal layout of goods in container.

Keywords: Genetic algorithm, optimization, goods, container

1 Introduction

Genetic Algorithms (GAs) are the most efficient procedure to understand and solve problems for which have limited information. These algorithms are able to effectively handle both unconstrained and constrained optimization problems depending on a process of natural selection through biological evolution. The working mechanism of GAs is linked with a search space that contains all possible solutions. There are two significant points in the GA process: one is starting point initialization in search space and other is assigning of fitness function. GA starts with the initialization of a population or potential solutions of the problems. The initialization is represented by the chromosomes (individuals), which are a set of genes, with each gene carrying the features of the dataset [1]. The location–allocation

H. C. Chen *et al.* (Eds.): FICT 2021
https://easychair.org/cfp/FutureICT2021

problem is a difficult optimization problem because multiple local minima may exist. For this reason, it is hard to be solved with the traditional algorithms. A genetic algorithm (GA) is a meta-heuristic method based on the mechanics of copying strings according to their strings to generate successive populations that improve over time [2].

The process of loading and shipping goods is an important thing and very influential for the running of a company's business processes especially manufacturing companies or shipping service companies. The pattern of arranging goods in three-dimensional space, for example containers, must be as optimal as possible to reduce the shipping costs in one shipment. There is a manual item calculation data that is used as a reference in the cargo of goods, for example, if a 20 feet container should be able to load 300 product packages. But it is often the cargo of goods that does not meet the specified reference criteria. This is often caused due to the lack of an optimal arrangement of goods in containers. The arrangement does not utilize empty space properly and not based on the type of goods. Therefore, the factory requires a system that is useful for calculating and optimizing the load capacity of goods on a container that will be sent to avoid losses that can be fatal for companies such as damage to goods due to the buildup of goods in warehouses and complaints from consumers who are disappointed because the goods sent not received at the promised time.

Based on statements above, it is proposed to make an application optimization process for loading goods that can facilitate controlling cargo, optimize (maximize) the number of products shipped with containers, and have a visualization of the arrangement of goods through the application.

We propose the genetic algorithms are used for the optimization process. This algorithm is a method for solving optimization problems based on the theory of evolution in biology. This algorithm works on a population of candidate solutions called chromosomes which are initially generated randomly from the completion space of the objective function. By using the genetic operator mechanism, population crossover and mutation are evolved through the fitness function which is directed at convergence conditions so that the optimal values reached [3].

This paper is organized as follows. In Section 1, the motivation for this research is introduced. Section 2 presents the related works of this research. In section 3, our research approach explains how to implement the genetic algorithm dealing with the layout and loading work. Section 4 explains the results of the experiment and then carries out the discussions. Finally, we conclude this research in Section 5.

2 Related works

Blockchain technology gets introduced into the shipping domain. They demonstrate that the shipping industry is based upon an information infrastructure with a socio-technical kernel comprising transaction practices between shippers, freight forwarders, ports, shipping lines, and other actors in the shipping industry [4]. In another studies, the documentation of a shipment does not correctly or entirely describe the goods in transit. In an attempt to reduce the risks of document fraud, shipping companies and customs authorities typically perform random audits to check the accompanying documentation of shipments. They investigates whether intelligent fraud detection systems can improve the detection of miscoding and smuggling by analyzing large sets of historical shipment data [5]. Optimization is the science of finding decisions that satisfy a given constraint and meet a specific goal at its optimal value. In engineering, constraints may arise from physical limitations and technical specifications; in business, constraints are often related to resources, including manpower, equipment, costs, and time [3]. In optimization, its start with some kind of initial values for the variables used in the experiment. Because these values may not be the best ones to use, we should change them until getting the best ones. In some cases, these values are generated by complex functions that we cannot solve manually easily.

Optimization techniques are categorized into four main categories: constrained optimization, multimodal optimization, multi objective optimization and combinatorial optimization. Genetic Algorithm (GA) is an optimizer algorithm whose idea comes from nature which adopts Charles Darwin's natural evolutionary theory "survival of the fittest". Genetic Algorithms have characters as evolutionary algorithms, namely population based, fitness oriented and variation driven. Genetic algorithms solve optimization problems by mimicking the principles of biological evolution, repeatedly modifying individual population points using rules that are modeled on the combination of genes in biological reproduction. Because of their random nature, genetic algorithms increase opportunities to find global solutions and help to solve optimization problems that are not limited, restricted, and general, and do not require distinguishable or sustainable functions [6, 7]. The main steps in the genetic algorithm are initialization, selection (using a roulette wheel or tournament selection), cross over and mutation, so that optimal results will be obtained. Genetic algorithms can be applied to fields of human endeavor including machine learning, scheduling, signal processing, energy, robotics, manufacturing, mathematics, routing, and many more [1].

The study the optimum value of the load was calculated using the taboo search algorithm. The output of this program also can’t display visualization of payload images, only in the form of coordinate data on the results of the compilation of goods and graphs. This research takes the starting point of the research and develops it and changes the method used because it feels better and is suitable for systems that have input constraints [8]. In this research, the application of genetic algorithm research methods will be carried out to optimize the process of loading goods into container (stuffing) whereas similar studies have been carried out using taboo search algorithms. The output of this application is not yet able to display the visualization of the payload image, only in the form of coordinate data from the results of the arrangement of the goods and the graph [9].

3 System design

In this study, we designed a system based on the mathematical characteristics of the loading model, to develop a custom placement heuristic integrated with a new dynamic space division method, using a genetic algorithm to maximize the loading space. The genetic algorithm is used to solve the optimization problem above is a genetic algorithm because it is considered better and suitable for systems that have input constraints. In this research, the genetic algorithm is used to determine the exact and optimal order of item placement and implemented by using the python programming language. The optimal parameter or not is seen from the fitness value obtained on each chromosome. The fitness value is the global fitness value, which is a combination of volume fitness, weight fitness, and fitness type of each chromosome.

The coding technique implemented in making this program is permutation coding. The chromosomes are generated randomly, then the selection method used is roulette wheel selection. While the crossover method used is the crossover for permutation chromosome representation and the mutation method used is swapping mutation. The following is an overview of the work cycle of implementing a genetic algorithm in the application to be designed, as shown in Figure1.

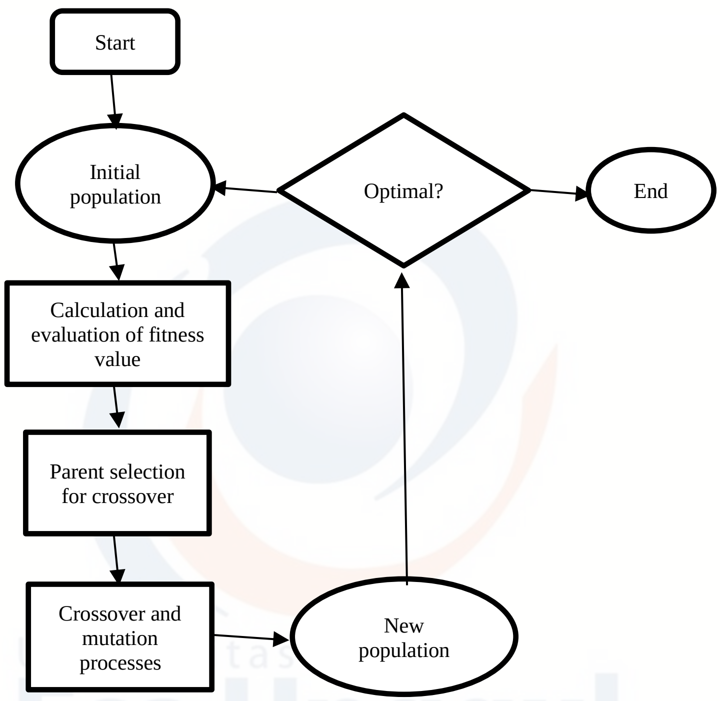

Figure 1. Cycle of Genetic Algorithm Implementation

3.1 Step 1: Initialization of Initial Population (chromosome)

The chromosome initialization is carried out to shows the sequence of goods generated randomly. The components are the serial number of items contained in an excel file and entered into an application. First take some file in the data database of goods, namely number, weight, type, destination, and dimensions of goods (length, width, height). Then each of the components is given a value and the volume of goods is calculated to make it easier to do the next process.

a. Type

Item types are weighted with integer type types. The assignment of this value is so that the goods which have the highest weight are put in the container first.

b. Destination

The data item entered consists of several purposes. These objectives are also weighted with integer type types. The assignment of value is so that goods which have the most distant destinations are first put into containers.

c. Dimensions of goods

To carry out the process of calculating the volume of data dimensions of goods (length (*l*), width (*w*), height (*h*)). Calculation of the volume of goods according to the formula:

$$v = l.w.h \tag{1}$$

3.2 Step 2: Fitness Value Calculation

Fitness value states whether or not the optimal solution is produced. The fitness value is obtained from a formula that has been determined based on the problem to be solved. In this research, the optimal parameters are based on the weight, dimensions of the goods (volume), and type, namely weight fitness, fitness volume, and overall fitness. Finding a good fitness function is of great importance to reduce the size of the search domain and to make the genetic algorithm more likely to find P in less time.

a. Weight Fitness

Determination of the weight fitness value of an arrangement of goods seen from the penalty generated. Penalty is an error or violation value. The more penalty values generated, the smaller the weight fitness value obtained, and vice versa. Here are the rules for awarding penalties for placing an item in error.

$$P = \tag{2}$$

When the item below is heavier than the item above it is not subject to a penalty. However, if the opposite occurs then a penalty of 1. That is because the goods arranged at the bottom must be heavier than the items above. If all penalty values are obtained, the weight fitness could be calculated by the equation below:

$$F_w = \frac{W_{Composed} - \sum P}{W_{Composed}} \tag{3}$$

b. Volume Fitness Function

The calculation of the ratio between the sum of total volumes of the goods and the maximum volume of one container. Next, it will produce out both the estimate value and the possible amount of goods for fulfilling the container. The volume fitness function is given below to determine the value of the fitness volume.

$$F_v = \frac{\sum V_{goods}}{V_{container}} \quad (4)$$

c. Type of Goods Fitness

Calculations of the same type of Goods fitness formula with weight fitness calculations. Only the values processed are different. The value in this fitness calculation is the result of the initialization of chromosomes in the data type, i.e. Folding chairs = 1, Director's chairs = 2, and Lecture chairs = 3. The penalty value is the opposite of how to calculate heavy fitness, i.e. if the *k*-th item value is smaller or equal to the value of goods to (k + 1), then the penalty = 0. Besides the penalty value is equal to 1.

d. The Global Fitness

The global fitness value is an objective fitness value that is used as a parameter whether a chromosome (solution) produced is optimal / not in determining the order in which items are placed in the container. This fitness value is a combination of several fitness values that have been calculated earlier, namely weight fitness, fitness volume, and fitness type. The following formula to find the value of overall fitness.

$$F = (P_v \times F_v) + (P_w \times F_w) + (P_t \times F_t) \quad (5)$$

3.3 Step 3: Parent Selection

Individuals are chosen from the population and recombined during the reproductive process, producing offspring for the next century. Parents are chosen from the community by means of a system that further develops individuals. Having chosen two parents, their chromosomes are recombined, usually using crossover and mutation processes.

Selection aims to determine which individuals will be selected for recombination (*crossover* and *mutation*) and how *offspring* is formed from these selected individuals. The selection method is *Roulette Wheel Selection*. The following selection algorithm with a roulette wheel or *roulette wheel selection*:

Selection algorithm with a roulette wheel.
Begin
Input: k=1, 2, …, popsize
initialize the number of the goods *initialize the weight of the goods* *initialize the type of the goods* *initialize the destination of the goods* $\frac{1}{\sum_{i=1}^{N} x_i(t=0)}$
calculate the total fitness :
$TotFitness = \sum F_k$
calculate the relative fitness of each individual
$P_k = \frac{F_k}{TotFitness}$

calculate cumulative fitness
$q_1 = P_1$ $q_k = q_{k-1} + P_k; k = 2,3,\ldots,popsize$
Generate a random number, r between 0 and 1 and then determine the range of it. Make the following conditions:
If $q_k \varepsilon rdanq_k + 1 > r$ then select chromosome to *k+1* as the parent candidate.
Output : *the* parent candidate
End.

3.4 Step 4: Process of Mutation and Crossover

Crossover (crossbreeding) is done on two chromosomes to produce chromosomes of children (offspring). Child chromosomes that are formed will inherit some of the characteristics of the parent chromosome. In this problem the crossover method used is the crossover for permutation chromosome representation. The following crossover algorithm:

Crossover algorithm.
Begin
Input: *The desired of Crossover Probability(usually between the range 0.75 to 0.9).*
For each population as many populations ***do :***
Generate a random number between zero and one
If *the number the crossover probability,* ***then*** *do the crossover*
If the number the crossover probability, ***then*** *parent 1 and parent 2 immediately go down to child 1 and child 2*
For *each pair chosen to do a crossover*
Generate 2 numbers between 1 and the number of genes for each individual;
The selection of genes to be crossovered on this problem is determined based on the purpose of the goods. So the random numbers that are generated correspond to the length of the gene for each destination on a chromosome;
Perform mutations with the; swapping mutation / exchange mutation algorithm:

```
        Generating two random numbers (between 0 to
        chromosome length) for each individual, as many
        as the number of individuals in the population
        Exchange genes with the location of the two
        numbers by
          numberLoc = gen1; gen1 = gen2;gen2 =
          numberLoc;
    Output:
    The Chromosom crossover selected
End.
```

3.5 Step 5: Data Processing

In this stage, the data to be processed is in the form of input data from the program features plus data items and raw data in the form of Excel files. The data was obtained from a company engaged in the field of furniture manufacturers of various kinds of chairs. The data obtained is more complex because the item description is explained in detail. Following are the contents of the data:

Table 1. Inventory Data

Item code	Item Dimensions	Item Weight	Remarks
In the form of a series / combination of letters and numbers	The value is a number in mm	The value is a number in units of kg	Contains the name of the product entered
	Components: Netto = l, w, h Gross = l, w, h	Component: Net weight and gross weight	

Table 2. Notations

Notation	Definition
P	Penalty
M_{below}	The mass / weight of the item under arrangement
M_{above}	The period /weight of the upper arrangement of the goods
$W_{Composed}$	The weight of all goods
V_{goods}	The volume of each of goods
$V_{Container}$	The volume of container

F	Overall Fitness
P_w	Weight fitness portion
P_v	Volume fitness portion
P_t	Type fitness portion
F_w	Weight Fitness
F_v	Volume Fitness
F_t	Volume Fitness
TotFitness	The total Fitness of All parameters
F_k	The Fitness value of each of parameters/ individual
P_k	The relative fitness of each parameters/Individual
q_k	The cumulative fitness of each parameters/Individual

4 Simulation Results and Discussions

The test is carried out with two scenarios involving, the weight of goods, volume of goods, delivery destination.

1.1 Scenario, 1st:

4.1.1 Inventory

The following is a table of data that will be tested, the data being tested includes 6 packages of college chairs, 10 packages of director chairs, 26 packages of TV racks and 1 package of folding chairs.

Table 3. **Data testing of the first scenario**

Id-Item	Code	Weight	Type	Destination	Length	Width	Height
1	KL601	30	Folding Chair	Batam	103	46	31
2	KK619NK	15	Lecture Chair	Batam	104	48	20
3	RSGB888	25	TV Rack	Batam	80	48	14
4	D916	20	Director Chair	Batam	107	50	25
5	D916	20	Director Chair	Batam	107	50	25
6	D916	20	Director Chair	Batam	107	50	25
7	D916	20	Director Chair	Batam	107	50	25
8	D916	20	Director Chair	Batam	107	50	25
9	D916	20	Director Chair	Batam	107	50	25
10	D916	20	Director Chair	Batam	107	50	25
11	D916	20	Director Chair	Batam	107	50	25
12	D916	20	Director Chair	Batam	107	50	25
13	D916	20	Director Chair	Batam	107	50	25
14	RSGB888	25	TV Rack	Batam	80	48	14
15	RSGB888	25	TV Rack	Batam	80	48	14
16	RSGB888	25	TV Rack	Batam	80	48	14
17	RSGB888	25	TV Rack	Batam	80	48	14
18	RSGB888	25	TV Rack	Batam	80	48	14
19	RSGB888	25	TV Rack	Batam	80	48	14
20	RSGB888	25	TV Rack	Batam	80	48	14
21	RSGB888	25	TV Rack	Batam	80	48	14
22	RSGB888	25	TV Rack	Batam	80	48	14
23	RSGB888	25	TV Rack	Batam	80	48	14
24	RSGB888	25	TV Rack	Batam	80	48	14
25	RSGB888	25	TV Rack	Batam	80	48	14
26	RSGB888	25	TV Rack	Batam	80	48	14
27	RSGB888	25	TV Rack	Batam	80	48	14
28	RSGB888	25	TV Rack	Batam	80	48	14
29	RSGB888	25	TV Rack	Batam	80	48	14
30	RSGB888	25	TV Rack	Batam	80	48	14
31	RSGB888	25	TV Rack	Batam	80	48	14
32	RSGB888	25	TV Rack	Batam	80	48	14
33	RSGB888	25	TV Rack	Batam	80	48	14
34	RSGB888	25	TV Rack	Batam	80	48	14
35	RSGB888	25	TV Rack	Batam	80	48	14
36	RSGB888	25	TV Rack	Batam	80	48	14
37	RSGB888	25	TV Rack	Batam	80	48	14
38	RSGB888	25	TV Rack	Batam	80	48	14
39	KK619NK	15	Lecture Chair	Batam	104	48	20
40	KK619NK	15	Lecture Chair	Batam	104	48	20
41	KK619NK	15	Lecture Chair	Batam	104	48	20
42	KK619NK	15	Lecture Chair	Batam	104	48	20
43	KK619NK	15	Lecture Chair	Batam	104	48	20

4.1.2 Container Type

In the first experiment, the trial was carried out with a 20 feet container type.

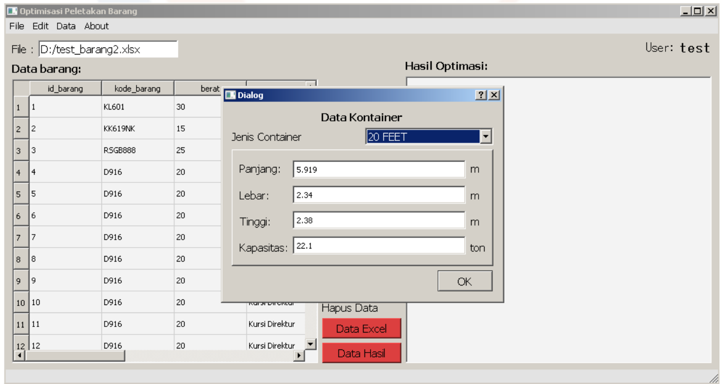

Figure 1. Display of Container Type

4.1.3 The List of Destination City

Because this application is only for sending 1 destination, the initiation of the city list is only for 1 city and give the value of the farthest city with a value of 1 because there is only 1 destination.

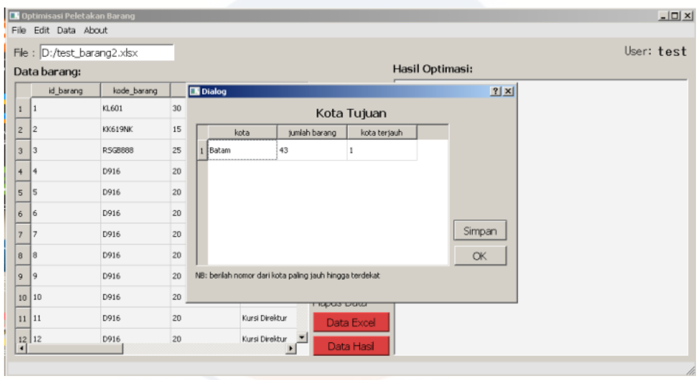

Figure 2. Display of the list of destination city

4.1.4 Initialize item value

Rate the goods based on the heaviest item to the lightest, in this experiment the heaviest item is a folding chair and the lightest is the TV rack.

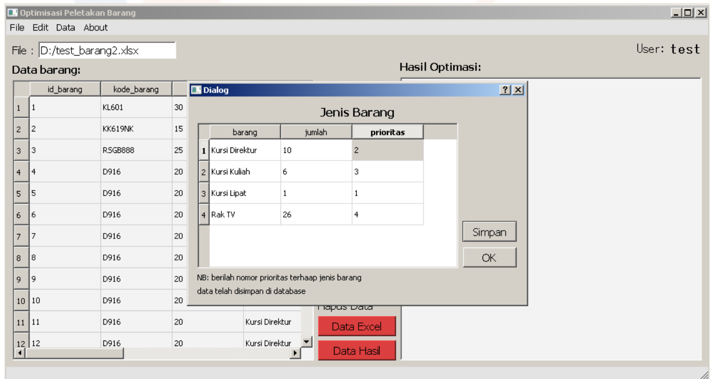

Figure 3. Display of the item value of the goods

4.1.5 Genetic Algorithm Parameters

In the population size column, fill in accordance with the total number of items, then the number of parent fill fields as desired with a record should not exceed the total population / number of items, the mutation rate and crossover rate columns are filled as desired with a range of values between 0 to 1, then the iteration column is filled with a value of 20 in the first trial, later the system will operate to find the best fitness value 20 times.

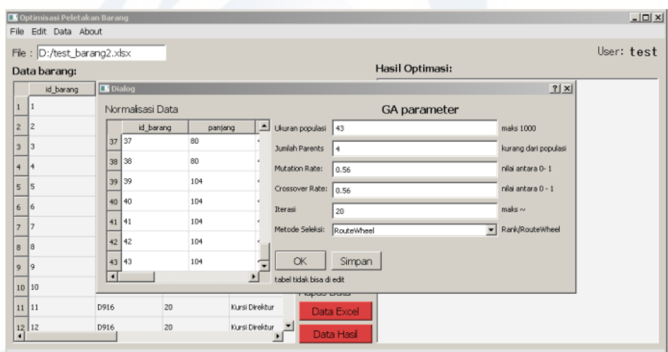

Figure 4. Display of the genetic algorithm parameters

Then in this test the selected selection method is roullette wheel selection. After that press save and on the home screen press the start button.

4.1.6 Fitness Value Optimization Results

Below this is the result of calculating the fitness value in the first testing.

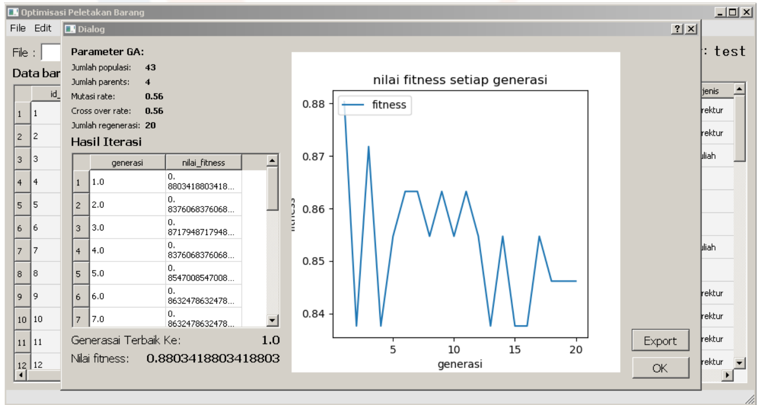

Figure 5. Display of the Fitness Value Optimization Results

4.1.7 Result of Goods Position and Visualization

The result of calculating the fitness value and visualization of goods position on container in the first testing can be seen below. The arrangement of goods in containers is depicted in figure 6 and the position of goods can be seen in table 6.

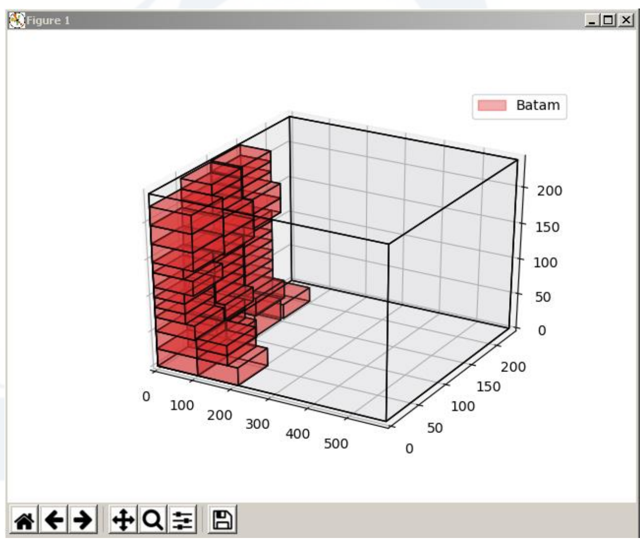

Figure 6. Display of vizualization of the arrangement of goods in containers

Table 4. Data on the position of goods on the first testing

Id-Item	Type	Destination	Length	Width	Height	Weight	Position_X	Position_Y	Position_Z
10	Director Chair	Batam	107	50	25	20	0	0	0
7	Director Chair	Batam	107	50	25	20	0	0	25
39	Director Chair	Batam	104	48	20	15	0	0	50
25	TV Rack	Batam	80	48	14	25	0	0	70
38	TV Rack	Batam	80	48	14	25	0	0	84
15	TV Rack	Batam	80	48	14	25	0	0	98
40	Lecture Chair	Batam	104	48	20	15	0	0	112
20	TV Rack	Batam	80	48	14	25	0	0	132
12	Director Chair	Batam	107	50	25	20	0	0	146
13	Director Chair	Batam	107	50	25	20	0	0	171
8	Director Chair	Batam	107	50	25	20	0	0	196
4	Director Chair	Batam	107	50	25	20	0	50	0
31	TV Rack	Batam	80	48	14	25	0	50	25
37	TV Rack	Batam	80	48	14	25	0	50	39
42	Lecture Chair	Batam	104	48	20	15	0	50	53
14	TV Rack	Batam	80	48	14	25	0	50	73
33	TV Rack	Batam	80	48	14	25	0	50	87
28	TV Rack	Batam	80	48	14	25	0	50	101
17	TV Rack	Batam	80	48	14	25	0	50	115
32	TV Rack	Batam	80	48	14	25	0	50	129
6	Director Chair	Batam	107	50	25	20	0	50	143
11	Director Chair	Batam	107	50	25	20	0	50	168
34	TV Rack	Batam	80	48	14	25	0	50	193
21	TV Rack	Batam	80	48	14	25	0	50	207
24	TV Rack	Batam	80	48	14	25	0	50	221
1	Folding Chair	Batam	103	46	31	30	0	100	0
5	Director Chair	Batam	107	50	25	20	0	100	31
29	TV Rack	Batam	80	48	14	25	0	100	56
36	TV Rack	Batam	80	48	14	25	0	100	70
35	TV Rack	Batam	80	48	14	25	0	100	84
26	TV Rack	Batam	80	48	14	25	0	100	98
30	TV Rack	Batam	80	48	14	25	0	100	112
16	TV Rack	Batam	80	48	14	25	0	100	126
19	TV Rack	Batam	80	48	14	25	0	100	140
2	Lecture Chair	Batam	104	48	20	15	0	100	154
43	Lecture Chair	Batam	104	48	20	15	0	100	174
23	TV Rack	Batam	80	48	14	25	0	100	194
27	TV Rack	Batam	80	48	14	25	0	100	208
22	TV Rack	Batam	80	48	14	25	0	100	222
41	Lecture Chair	Batam	104	48	20	15	0	150	0
9	Director Chair	Batam	107	50	25	20	0	0	0
3	TV Rack	Batam	80	48	14	25	0	0	25

1.2 Scenario, 2nd:

4.2.1 Inventory

The following is a table of data that will be tested in the second experiment, the data being tested consist of 17 college chairs, 8 TV racks and 23 folding chairs.

Table 5. Data Testing of Scenario 2nd

Id-Item	Code	Weight	Type	Destination	Length	Width	Height
1	RSGB888	25	TV Rack	Sorong	80	48	14
2	RSGB888	25	TV Rack	Sorong	80	48	14
3	RSGB888	25	TV Rack	Sorong	80	48	14
4	RSGB888	25	TV Rack	Sorong	80	48	14
5	RSGB888	25	TV Rack	Sorong	80	48	14
6	RSGB888	25	TV Rack	Sorong	80	48	14
7	RSGB888	25	TV Rack	Sorong	80	48	14
8	RSGB912	25	TV Rack	Sorong	80	48	14
9	KK619NK	15	Lecture Chair	Sorong	104	48	20
10	KK619NK	15	Lecture Chair	Sorong	104	48	20
11	KK619NK	15	Lecture Chair	Sorong	104	48	20
12	KK619NK	15	Lecture Chair	Sorong	104	48	20
13	KK619NK	15	Lecture Chair	Sorong	104	48	20
14	KK619NK	15	Lecture Chair	Sorong	104	48	20
15	KK619NK	15	Lecture Chair	Sorong	104	48	20
16	KK619NK	15	Lecture Chair	Sorong	104	48	20
17	KK619NK	15	Lecture Chair	Sorong	104	48	20
18	KK619NK	15	Lecture Chair	Sorong	104	48	20
19	KK619NK	15	Lecture Chair	Sorong	104	48	20
20	KK619NK	15	Lecture Chair	Sorong	104	48	20
21	KK619NK	15	Lecture Chair	Sorong	104	48	20
22	KK619NK	15	Lecture Chair	Sorong	104	48	20
23	KK619NK	15	Lecture Chair	Sorong	104	48	20
24	KK619NK	15	Lecture Chair	Sorong	104	48	20
25	KK619NK	15	Lecture Chair	Sorong	104	48	20
26	KL601	30	Folding Chair	Sorong	103	46	31
27	KL601	30	Folding Chair	Sorong	103	46	31
28	KL601	30	Folding Chair	Sorong	103	46	31
29	KL601	30	Folding Chair	Sorong	103	46	31
30	KL601	30	Folding Chair	Sorong	103	46	31
31	KL601	30	Folding Chair	Sorong	103	46	31
32	KL601	30	Folding Chair	Sorong	103	46	31
33	KL601	30	Folding Chair	Sorong	103	46	31
34	KL601	30	Folding Chair	Sorong	103	46	31
35	KL601	30	Folding Chair	Sorong	103	46	31
36	KL601	30	Folding Chair	Sorong	103	46	31
37	KL601	30	Folding Chair	Sorong	103	46	31
38	KL601	30	Folding Chair	Sorong	103	46	31
39	KL601	30	Folding Chair	Sorong	103	46	31
40	KL601	30	Folding Chair	Sorong	103	46	31
41	KL601	30	Folding Chair	Sorong	103	46	31
42	KL601	30	Folding Chair	Sorong	103	46	31
43	KL601	30	Folding Chair	Sorong	103	46	31
44	KL601	30	Folding Chair	Sorong	103	46	31
45	KL601	30	Folding Chair	Sorong	103	46	31
46	KL601	30	Folding Chair	Sorong	103	46	31
47	KL601	30	Folding Chair	Sorong	103	46	31
48	KL601	30	Folding Chair	Sorong	103	46	31

4.2.2 Container Type

In the second experiment, the trial was carried out with a 40 feet container type.

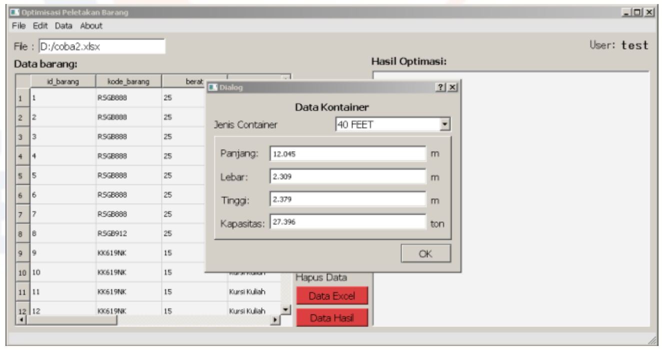

Figure 7. Display of container type menu

4.2.3 The List of destination City

Initialize the value of the city list from the farthest distance to the closest, because this application is only limited to 1 destination so fill the farthest city with a value of 1.

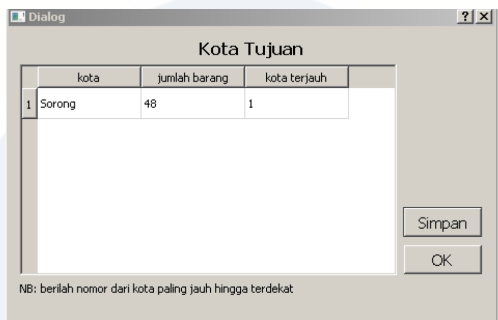

Figure 8. Display of the list of destination city menu

4.2.4 Initialize item value

In this second experiment, the heaviest and lightest items are still the same, the heaviest item is a folding chair and the lightest is a TV rack.

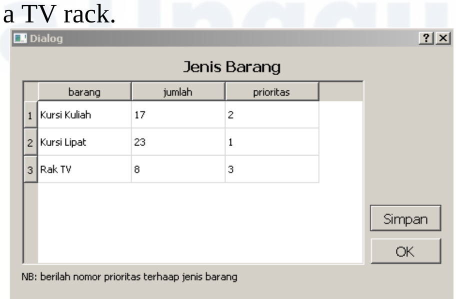

Figure 9. Display of the item value of the goods dialog menu

4.2.5 Genetic Algorithm Parameters

In the second experiment, the population size was filled according to the number of items ie 48, the number of parents 4, mutation and crossover rate were filled in at 0.56, iteration was 30, and the selection method was ranked.

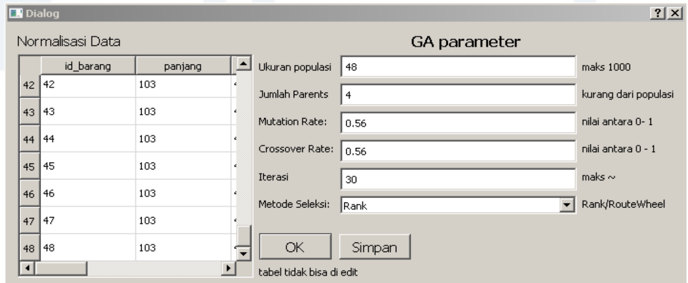

Figure 10. Display of the genetic algorithm parameters menu

4.2.6 Fitness Value Optimization Results

Below this is the result of calculating the fitness value in the second trial.

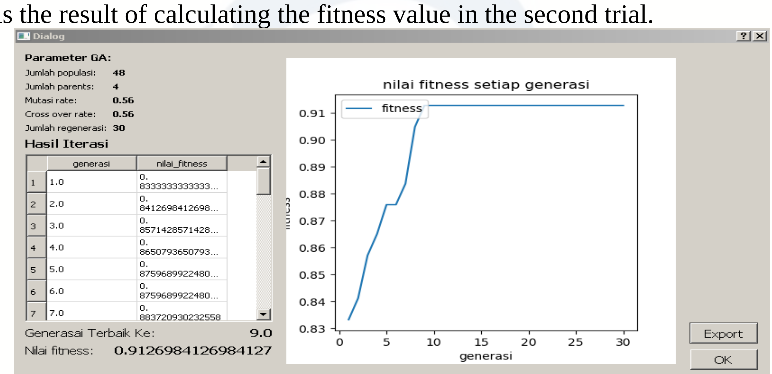

Figure 11. Display of the fitness value optimization results

4.2.7 Result of Goods Position and Visualization

Below this is the result of calculating the fitness value in the second trial.

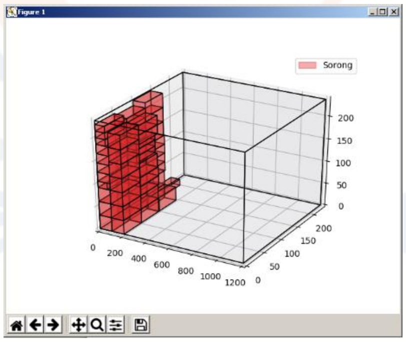

Figure 12. Display of vizualization of the arrangement of goods in containers

Table 6. Data on the position of goods on the second testing

Id-Item	Type	Destination	Length	Width	Height	Weight	Position_X	Position_Y	Position_Z
34	Folding Chair	Sorong	103	46	31	30	0	0	0
28	Folding Chair	Sorong	103	46	31	30	0	0	31
26	Folding Chair	Sorong	103	46	31	30	0	0	62
1	TV Rack	Sorong	80	48	14	25	0	0	93
11	Lecture Chair	Sorong	104	48	20	15	0	0	107
22	Lecture Chair	Sorong	104	48	20	15	0	0	127
21	Lecture Chair	Sorong	104	48	20	15	0	0	147
14	Lecture Chair	Sorong	104	48	20	15	0	0	167
4	TV Rack	Sorong	80	48	14	25	0	0	187
8	TV Rack	Sorong	80	48	14	25	0	0	201
5	TV Rack	Sorong	80	48	14	25	0	0	215
16	Lecture Chair	Sorong	104	48	20	15	0	48	0
38	Folding Chair	Sorong	103	46	31	30	0	48	20
30	Folding Chair	Sorong	103	46	31	30	0	48	51
35	Folding Chair	Sorong	103	46	31	30	0	48	82
39	Folding Chair	Sorong	103	46	31	30	0	48	113
24	Lecture Chair	Sorong	104	48	20	15	0	48	144
9	Lecture Chair	Sorong	104	48	20	15	0	48	164
15	Lecture Chair	Sorong	104	48	20	15	0	48	184
25	Lecture Chair	Sorong	104	48	20	15	0	48	204
43	Folding Chair	Sorong	103	46	31	30	0	96	0
32	Folding Chair	Sorong	103	46	31	30	0	96	31
19	Lecture Chair	Sorong	104	48	20	15	0	96	62
13	Lecture Chair	Sorong	104	48	20	15	0	96	82
23	Lecture Chair	Sorong	104	48	20	15	0	96	102
10	Lecture Chair	Sorong	104	48	20	15	0	96	122
41	Folding Chair	Sorong	103	46	31	30	0	96	142
33	Folding Chair	Sorong	103	46	31	30	0	96	173
27	Folding Chair	Sorong	103	46	31	30	0	96	204
3	TV Rack	Sorong	80	48	14	25	0	144	0
44	Folding Chair	Sorong	103	46	31	30	104	0	0
31	Folding Chair	Sorong	103	46	31	30	104	0	31
40	Folding Chair	Sorong	103	46	31	30	104	0	62
46	Folding Chair	Sorong	103	46	31	30	104	0	93
47	Folding Chair	Sorong	103	46	31	30	104	0	124
48	Folding Chair	Sorong	103	46	31	30	104	0	155
42	Folding Chair	Sorong	103	46	31	30	104	0	186
45	Folding Chair	Sorong	103	46	31	30	104	46	0
29	Folding Chair	Sorong	103	46	31	30	104	46	31
18	Lecture Chair	Sorong	104	48	20	15	104	46	62
12	Lecture Chair	Sorong	104	48	20	15	104	46	82
37	Folding Chair	Sorong	103	46	31	30	104	46	102
7	TV Rack	Sorong	80	48	14	25	104	46	133
2	TV Rack	Sorong	80	48	14	25	104	46	147
6	TV Rack	Sorong	80	48	14	25	104	46	161
17	Lecture Chair	Sorong	104	48	20	15	104	46	175
20	Lecture Chair	Sorong	104	48	20	15	104	46	195
36	Folding Chair	Sorong	103	46	31	30	104	94	0

The resulting fitness value is influenced by genetic algorithm parameter values. The results show that the greater the crossover rate and the number of iterations, the greater the fitness value generated. It also shows that the resulting solution is more optimal. In addition, the arrangement of goods produced is also influenced by the value of fitness produced.

In the second trial, the number of items entering was more than the number of items entering the first trial with a different type of container, which was 40 feet. For the 20 feet and 40 feet containers, the number of items entered in trials 1 and 2 is 43 cargos and 48 cargos, ut the difference in fitness values resulting from experiments 1 and 2 are quite different, The difference in the arrangement of goods arrangement is also increasingly apparent from the 3D visualization produced. This can be described as follows:

a. **Container 20 feet**. Container dimensions [9]:

Length = 5,919, Width = 2,34, Height = 2,38

It is obtained, fitness value = 0.8803418803418803

b. **Container 40 feet**. Container dimensions [9]:

Length = 12,045, Width = 2,309, Height = 2,379

It is obtained, fitness value = 0.9126984126984127

From the two results it can be seen that there are differences in the arrangement of goods towards the high. In the first trial picture, the larger item is still on the top stack. Whereas in the second trial picture the stacks of goods was more orderly and neat, larger and heavier items are below followed by items that are smaller in size. This is due to differences in the value of fitness resulting from the calculation process with genetic algorithm parameters. The better the fitness value, the better the visualization arrangement.

5 Conclusion

Based on the research that has been done, it can be concluded, that by making a spatial optimization system in containers, the results of planning the arrangement of goods in the container space are in accordance with the weight and container and the number of items that must be loaded. Determination of the optimal parameters of whether or not the laying of goods is not reviewed in terms of the objectives, weight and volume of goods. From testing the application that a high fitness value is needed so that the arrangement of the items can be arranged well in the container cabin. This application is made on a desktop basis. for that in the future mobile-based applications can be developed by adding features that represent other parameters, in order to obtain more optimal results.

References

Haq, E.-u., et al., *A Novel Selection Approach for Genetic Algorithms for Global Optimization of Multimodal Continuous Functions.* Computational Intelligence and Neuroscience, 2019. **2019**: p. 8640218.

Neungmatcha, W., et al., *Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem.* Computers and Electronics in Agriculture, 2013. **98**: p. 85-99.

Gad, A. *Introduction to Optimization with Genetic Algorithm*. 2018 [cited 2020 20200512]; Available from: https://www.kdnuggets.com/2018/03/introduction-optimization-with-genetic-algorithm.html.

Jabbar, K. and P. Bjørn. *Infrastructural grind: Introducing blockchain technology in the shipping domain*. in *Proceedings of the 2018 ACM Conference on Supporting Groupwork*. 2018.

Triepels, R., H. Daniels, and A. Feelders, *Data-driven fraud detection in international shipping.* Expert Systems with Applications, 2018. **99**: p. 193-202.

Bajpai, P. and M. Kumar, *Genetic Algorithm - an Approach to Solve Global Optimization Problems.* Indian Journal of Computer Science and Engineering, 2010. **1**: p. 199-206.

Jiang, D. and Z. Fan, *The Algorithm for Algorithms: An Evolutionary Algorithm Based on Automatic Designing of Genetic Operators.* Mathematical Problems in Engineering, 2015. **2015**: p. 474805.

Rahman, R.K., *Optimasi Proses Muat Barang dalam Kontainer Studi Kasus PT Antess Menggunakan Algoritma Tabu Search*, in *Information Engineering*. 2014, Universitas Islam Negeri Maulana Malik Ibrahim Malang. p. 116.

Ernawati, E., *Proses Pemuatan Barang ke Dalam Container (Stuffing) pada CV.Manggala Java Art di Klaten*, in *Fakultas Ekonomi*. 2011, Universitas Negeri Sebelas Maret: Surakarta, Indonesia.