

Transport and Telecommunication, 2016, volume 17, no. 4, 371–383
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.1515/tjt-2016-0033

STORAGE MANAGEMENT OF HAZARDOUS CONTAINERS USING THE GENETIC ALGORITHM

Khadija Yachba¹, Shahin Gelareh², Karim Bouamrane¹

*¹Department of Computer Sciences, University of Oran1 Ahmed Benbella,
BP 1524, EL MNaouer, 31000, Algeria, (LIO Laboratory),*

E-mails: yachbakhadija@yahoo.fr, kbouamrane@gmail.com

²Artois University, Béthune, France

E-mail: shahin.gelareh@gmail.com

This work discusses the problem of dangerous containers storage in a container terminal. Container terminal represents an essential intermodal interfaces for global transportation network. Several materials handling possible to move containers at the port to better meet the needs of ships awaiting loading or unloading. Have a good organization of the terminal's container storage area requires a special effort. Containers search times can be considerable and lead to delays causing financial penalties for terminal management operators. An optimal location for a container in a terminal is very important for companies because this operation reduces transportation costs. In this article, we propose an approach for the placement problem of hazardous containers through the description of an optimization model to solve and optimize available storage space. The interest of this work is to develop a computer tool to enable the identification of the best location of a hazardous container using the Genetic Algorithm.

Keywords: maritime transport, container, optimization, storage area, genetic algorithm, hazardous materials, optimum location

1. Introduction

Maritime transport is the moving operation of goods from one place to another by sea. This mode of transport may consider loading and unloading of containers at ports. Compared to another mode of transport (road, water) maritime transport remains a very important means of transport. Hazardous materials are substances, preparations, and articles (solid, liquid, or gaseous) that exhibit one or more hazardous properties. Thus, they can endanger human's health. In addition; they can pollute the environment and cause property damage as well. The maritime transport of dangerous goods is subject to specific safety regulations setting the conditions of packing, loading, unloading, handling and custody of these goods.

The objective of our study is to manage the storage of hazardous containers at a port by considering some conditions for better storage based on a meta-heuristic algorithm, which is the genetic algorithm while optimizing storage space available to handle departures and arrivals of hazardous containers at seaports.

The interest of this work is to find the best location for hazardous containers in stacks. Which is based on genetic algorithms that have a high potential for practical applications. In this article, we are interested in developing a model that allows the identification of best location for a dangerous container at a seaport. In Section 2, we present a state of art for the various works in the field of containerization and genetic algorithm while positioning our contribution. In section 3, we obviously demonstrate and explain the model. In section 4, we show the implementation part of our approach. In section 5, we focus on some obtained results. In section 6, we concentrate on the identification of the convergence criteria. In section 7, 8, we mention some advantages and disadvantages of genetic algorithm. In the section 9, we conclude with some perspectives.

2. Related Work and Contribution

The container storage problem that we are concerned about is a decision problem, which represents the container storage activity predominant component of all port management process. Therefore, we propose a part of the work in this field.

In the work of Kim and Kim (1997) a planning of container loading sequences to be exported in a seaport was made by using an optimal routing algorithm. Korbaa and Yim (2014) use a dynamic programming algorithm based on a stochastic arrival of law barges to solve the real-time allocation problem of containers unloaded to storage areas while minimizing the number of unwanted movements.

In the work of Murty *et al.* (2005): a decision support system for minimizing the time allocation of ships to berths has been proposed. Dubreuil (2008) uses an intelligent transportation system to handle the containers transition problem in port. Kefi (2008) uses the greedy heuristics in a multi-agent architecture to optimize the storage containers.

Chebli (2011) proposed in her work a heuristic approach in order to solve the planning problem of container loading operations.

Ndeye (2014) propose a branch-and-cut algorithm for solving the problem of storage containers and taking into account additional constraints to prevent alterations to the docks and in the order in which the containers are unloaded by ships.

Several engineering problems require the use of optimization methods to approach or to find the optimal solution. Deterministic methods seem to have filled that need, but the problem of local optima is a real trap to reach the global optimum. Hence, the use of stochastic methods called metaheuristics, which usually avoid this problem, and have a great ability to find the global optimum. Among these techniques, those based on collective intelligence are inspired by nature. The most common are: genetic algorithms (GA).

Genetic algorithms were invented by John Holland which was developed by his students and colleagues (Schwartz, 2005). He introduced the first formal model of genetic algorithms in his book "Adoption in natural and artificial System" published in 1975.

In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called it "genetic programming" method (Schwartz, 2005).

An Event Recap of these works is presented in Table 1.

Table 1. Summary of works using genetic algorithm

Authors	Problem Treaty and its characteristics	Resolution methods	Keys contributions
Bazzazi <i>et al.</i> , 2009	Considers several types of containers (full and empty).	Propose a genetic algorithm to solve a extended storage space allocation problem in a container terminal	A mathematical model that takes into consideration the different types of containers.
Ayachi <i>et al.</i> , 2010	Considers several types of containers (full and empty). Respect delivery wait time reduction of ships.	Genetic algorithm	Comparison with FIFO algorithm. Store all containers whose type is different.
Sharma <i>et al.</i> , 2014	Considers several types of containers (full and empty).	Genetic algorithm and Genetic algorithm Hybrid.	Improve the performance of the genetic algorithm to remedy the disadvantage of difficulty of the initial values.

The peculiarity of the proposed work comparing to those cited in Table 1 is that we treat in this article full hazardous containers respecting two constraints: the distance constraint and the constraint types. During storage of hazardous containers, taking into consideration following conditions:

- Containers of different classes should be separated by a horizontal distance of 2.4 m (width of container).
- Containers that transport radioactive goods should be removed 5 meters of any other hazardous container.
- Stacks can store the containers of their types.

The contribution is to propose a model for the hazardous container placement optimization in a port using genetic algorithm. The challenge is to determine the storage plan for hazardous containers to minimize the total processing time (loading or unloading) of this latter.

The processing time depends on various parameters such as the distance between the hazardous containers and type of hazardous material. Consequently, our job is to propose a model that simulates the in a container terminal so that the process of storage of hazardous containers in the storage area is optimal. Besides, to find the optimal location of a container, we propose a plan loading of containers in stacks.

2.1. Positioning the Contribution

This work forms a part of the research work in the field of decision support system and simulation (Yachba and Bouamrane, 2015). This article address the problem related to the placement of hazardous full containers in the storage area, where a model that simulates the operation of allocating hazardous containers at seaports is proposed. The model allows us to make analyzes of the marine transportation system and help us later to make predictions on the container storage management in a seaport.

3. The Proposed Model

This study concerns the problem of storage of hazardous containers (Container Stacking Problem) designated by the acronym CSP, which is a process that decides almost in real time the exact location from the empty slots of a container import or export, in order to make an efficient loading onto a ship, truck, or rail. In general, the determination of a location must be made to minimize the number of parasites or unproductive movement, that can take place when we want to bring a remote container extraction or a container located underneath another at the time of starting to be loading on the ship, train or truck involved (Kefi, 2008). During the storage of hazardous containers, some conditions must be taken into consideration (United Nations, 2009):

- The containers carrying radioactive goods must be 5 meters away from any other hazardous container,
- Stacks can store the containers according to their types.

The proposed model has the following objectives:

- Optimize the number of locations of hazardous containers,
- Reduce the loading and unloading time by the best choice of storage locations of hazardous containers,
- Reduce the total time spent by ships at the dock of a port (berthing time or throughout time),
- Simulate the allocation of containers in the associated stacks,
- Increase the productivity of dangerous goods transport companies.

The model is fundamentally structured on genetic algorithm that is Meta heuristics and can provide the optimal location of hazardous containers (Fig.1).

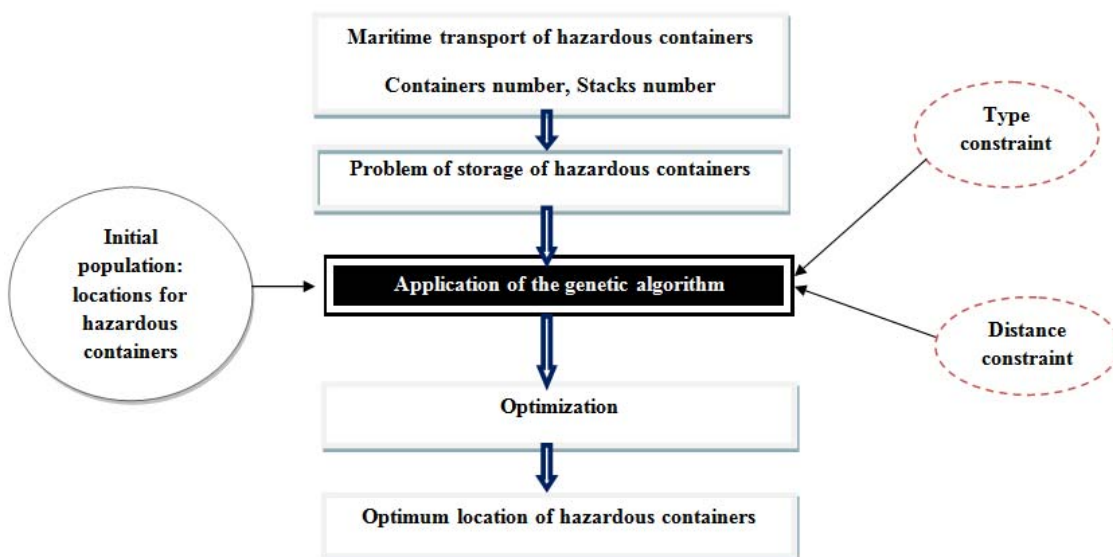


Figure 1. The Proposed Model

The use of class diagrams is recommended to present the classes and interfaces of dangerous container placement system. The class diagram in this model consists of five titled classes: Ship, Dock, Container, Stack, Area that are connected by composition links, such as the link between the class Stack and the class Container (a stack may contain a or more containers, as it may be empty). When the ship

carrying hazardous containers arrives at the port, it remains stranded if other ships occupy the docks and in case it finds it, unoccupied parked dock for dangerous containers unloaded it. Hazardous containers will be carried to the storage area, which is composed of the temporary area; loading area and unloading area to be store in stacks (Fig. 2).

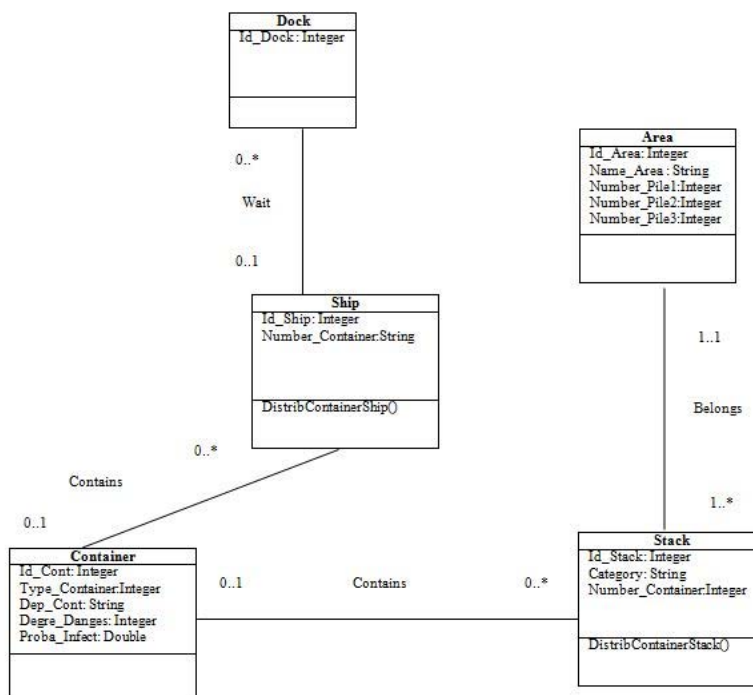


Figure 2. The Class Diagram of this Study

4. Proposed Approach

In this section, a detail of the proposal for solving the container storage problem using genetic algorithm is presented.

Before detailing the proposal, it is necessary to explain the state of the storage area.

4.1. Configuring the state of the storage area

This approach consider a storage area that has multiple stacks, including assumptions and storage strategies are:

- The containers are stored in the stack storage device of non-segregation method (Kim and Ma, 2012) (the containers are superposed on each other in the stack, regardless of their locations and with a distance separation).
- The stack has a capacity of up to 80 containers with 20 containers per floors using the group method of storage (Dekker *et al.*, 2007) (depending on the type). Once a floor is filled to the maximum (20 containers), we go upstairs following.
- The method of storage containers is indirect (the placement of containers in stacks is not originally planned).
- This approach take into account the different size of containers, and in each stack, we store only the same size container (of the same type).
- The dates of arrival and unloading container orders are known before the start. It is assumed that the containers are numbered in ascending order of their arrival times, and the containers that are unloaded in the same transportation are numbered according to their order of discharge.

Among the optimization methods, the genetic algorithm is implemented in this work with respect to the studied issues and for optimizing the location of hazardous containers in the stacks with the consideration of two constraints:

- Distance constraint: the distance between two containers must be equal to five meters to minimize risks, such as, fire or the risk of inflammation.
- Type of constraint (class): each type of container should be stored in the same type of stack.

4.2. Overview of the genetic algorithm

A genetic algorithm (GA) is a metaheuristic manipulating a population of potential solutions at once. The mode of operation of a GA is modelled on biological principles of natural selection and survival of the best adapted to the environment individuals (Darwin's theory). Natural selection is based on the idea that changes in successive generations, which are guided by external pressures and are subject to the spaces, eg limitation of resources, environmental changes, predators, and parasites, etc.). As a result, the best adapted to the environment people tend to survive longer and reproduce more frequently. Inspired and this mechanism, Holland led the foundation of the optimization technique called "genetic algorithm".

However, it was Goldberg, who subsequently became involved in the study of GA where he developed the present form we know (Dipama, 2010).

The vocabulary used by the GA is directly modelled in the theory of evolution and genetics .Thus terms such as: individual, population, gene, chromosome, parent, child, reproduction, crossover and mutation are used (Fig. 3). GA in the vocabulary of the environment relates to the search space that defines all the possible configurations of the function parameters to optimize. An individual in this environment is a possible configuration settings .A set of individuals form a population. Each individual can be represented by a chromosome that comprises a gene string containing the genetic characteristics of the individual. The gene is the basic part of a chromosome, which has a particular function. The adaptability of an individual to the environment is evidenced by measuring the performance of the individual through the function to optimize (or function adaptation or fitness) (Dipama, 2010).

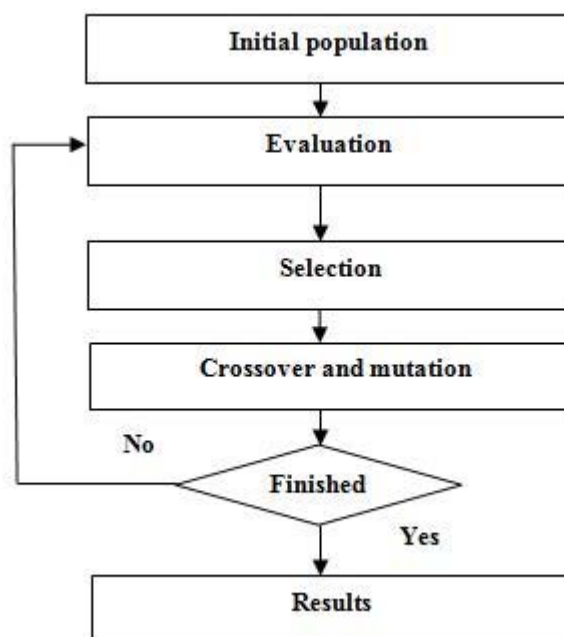


Figure 3. The Steps of the Genetic Algorithm (Dipama, 2010)

4.3. The operation of the genetic algorithm

Genetic algorithms have the feature to draw on the evolution of species in their natural environment. The species adapt to their environment with evolution, individuals of each species reproduce and creating new individuals. Some undergo changes in their DNA and other may disappear.

A genetic algorithm will replicate this model of evolution in order to find solutions to a given problem (Schwartz, 2005).

Before starting, we will make an analogy between the terms related to the genetic algorithm and example:

- In our case, a population is a set of individuals.
- An individual will be a solution to a problem: the locations of hazardous containers in stacks.
- A gene will be a part of a solution, thus an individual: a location of a container in a stack.
- A generation is an iteration of our algorithm.

Figure 4 illustrates the operation of the genetic algorithm:

```

{
Generate Population
For i=1 to Nb_max
{
For all individuals
{
Evaluate (Individual)
}
}
If BestIndividual= expected result
{
Stop
}
}
Select (Population)
Cross (Population)
Mutate (Population)
}
Solution= BestIndividual
}
We cannot demand equality and settle for approximate value

```

Figure 4. The Principle of the Genetic Algorithm (Boelle and Tourret, 2000)

4.4. Description of the genetic algorithm

The genetic algorithm consists of six steps (Fig. 3):

a) Generation of the initial population

This step is to generate the initial population of the genetic algorithm. The initial population should contain chromosomes, which are well distributed in space solutions... The easiest way is to randomly generate chromosomes (Jourdan, 2005).

However, the initial population in our approach is the locations of hazardous containers in stacks.

b) Individuals Evaluation

Once the initial population is created, we will get out the most promising individuals, who will participate in the improvement of our population. Therefore, we will assign a 'note' or a quality index for each of our individual. The individual assessment method is left to the programmer, depending on the problem he has to optimize or to solve.

The individual evaluation step may be performed before and / or after the steps of crossing and mutation. Once again, the programmer is free to implement this method (Schwartz, 2005).

This method is based on our approach as follows:

The fitness: the evaluation function will allow you to select or reject an individual to keep only individuals with the best cost based on the current population. It is calculated in case containers, which are well placed.

The Meta heuristics we have chosen to implement the genetic algorithm with respect to the problematic of optimizing the location of hazardous containers in stacks by considering two constraints:

$$Fit = \sum_{c=1}^{c=n} P_e = \frac{\text{Degrees of dangerousness} * \text{Infection rate}}{100} \quad (1)$$

P: The probability.
 L: The first container stored in stacks.
 n: The last container stored in stacks.

The Figure 5 illustrate how calculate the Fitness Function in Java code.

```

Public static double Fitness (int idIndividu, DefaultTableModel Tab, DefaultTableModel C)
{
    Double fit=0.0, prb, pena=0.0;
    String cont, pile;
    for(int i=1; I < Tab.getColumnCount()-2; i++)
    {
        cont=Tab.getColumnname(i).toString();
        Pile=Tab.getValueAt(idIndividu, i).toString();
        Prb=Prob(cont, pile, C);
        if (prb==0) pena=pena
        md.nextDouble();
        Fit=fit+prb;
    }
    Return fit;
}
    
```

Figure 5. The Fitness Function

This function is calculated in case the container is placed (the location generated in the initial population for each container that is stored in the stacks, is equivalent to the location stated in the beginning) if we calculate the penalty (in the opposite case).

c) The selection of individuals

The selection tends to increase the importance of good solutions from the poor ones. It is a heuristic used by the genetic algorithm: good solutions are assumed to be the most promising for the generation of descendants. There are several methods of selection. The best known is the method of the "biased lottery" (roulette wheel) of Goldberg's "elitist" and the method "tournament".

In our approach, we used elitist method of selecting the n individuals needed for the new generation P 'by taking the n best individuals of the population P after sorted decreasingly according to the fitness of its individuals.

d) The crossing of individuals

The crossing is to copy and recombine genes of two relative individuals to form two new individuals' sons with characteristics from both parents.

There are different types of crossing. We will briefly introduce the three main ones: The crossing point randomly determines a cut-off point and exchange, the second part of the parents. Crossing the two points (which can be extended two points) has 2 points (or) cuts, which are determined randomly. Finally the uniform crossover exchange each bit with a probability attached to 1/2 (Fig. 6).

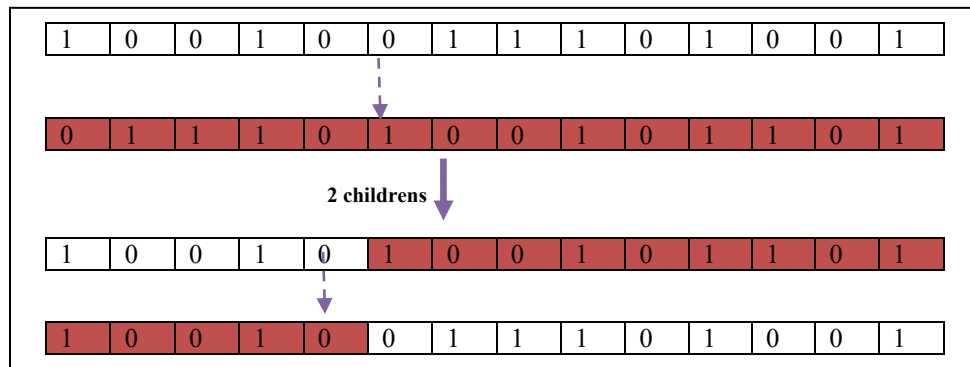


Figure 6. Crossing with a crossover point (Bokhabrine, 2006)

In this approach, we use the crossover with a crossover point of choosing randomly same cut off point on both individuals and exchange fragments, which are located after the break point in order to give the two new individuals.

e) Mutation

Traditionally, the mutation is applied to children with low probability between 0.01 and 0.1 to prevent local convergence of the genetic algorithm. The mutation is to draw lots of probability denoted P mut for each gene that constitutes the chromosome (Fig. 7).

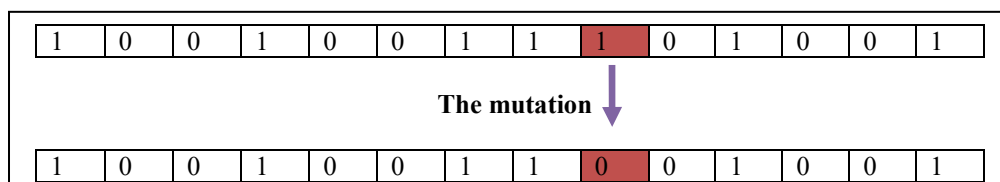


Figure 7. Mutation Step (Bokhabrine, 2006)

In the proposed approach, we change two locations of hazardous containers that belong to the same individual.

f) Reiteration process

The programmer has the opportunity to evaluate the individuals of the population before and / or after the phases of individual’s creation. Indeed, it may be appropriate to evaluate before inserting them in the future population, in addition it can be useful to reassess at the start of the next generation, if for example the valuation method depends on the population size (which very well could change). Thus, it may be necessary to evaluate twice each generation of individuals.

The number of generation is also at the discretion of the programmer. Generally, it is not possible to find suitable solutions within 10 generations and after 500 generations, the solutions will evolve more. However, this is only an order of magnitude, which depends on the problem to solve.

Once the maximum number of generations reached, we get a population of solutions. However, nothing says that the theoretical optimal solution is found. The solutions are similar to good solutions. This is not an exact method.

5. Results

Before launching the genetic algorithm, we select the latter parameters that must be fixed before the execution of the algorithm because they have a significant importance in solving the problem (Table 2).

Table 2. Genetic Algorithm Parameters

Genetic Parameters	
Selection Rate	0.8
Crossing Rate	0.7
Mutation Rate	0.5
Individual Number	100
Iterations Number	10

a) Selection Rate

Selection rate indicates the individual rate selected from the initial population. If the selection rate is 100, the set of population is selected. If the selection rate is 0 the new generation is exactly a copy of the ancient population: in this work, the selection rate is 0.8 because the optimal rate varies between 0.8 and 0.9.

b) Crossing Rate

Crossing rate indicates the rate of participation in reproduction; it represents the proportion of the population that breeds by crossing. If the crossing rate is 100 then the entire population participates in crossover. On the other hand, if the crossing rate is 0 the new generation is exactly a copy of the old population. In our example, the crossing rate is 0.7 because the optimal rate varies between 0.25 and 0.7.

c) Mutation Rate

Mutation rate indicates the rate that each gene of each individual mutating upon reproduction phase. If the mutation rate is 0 percent, individuals that are produced just after the crossing do not involve any change. On the other hand, if the mutation probability is 100 while the individual’s chromosome is changed: In our example it is fixed at 0.1 because the rate varies from 0.01 to 0.1 to prevent the algorithm converges to a local minimum.

d) Number of Individuals in the Population

This parameter is to set the number of individuals in the population. It represents the duration of the execution of the algorithm. The size of the population should not be too large because after a certain limit the performance of the algorithm decreases. In fact, a number of individuals too high affect the speed of solving the problem: In our example, we define Number of individuals to 100 because the optimal size is between 25 and 100.

e) Number of Iterations

This is the number that limits the number of changes in the population generated by the algorithm. The search is thus stopped after a number of generations: In our example, we fixed the 10 iterations because the optimal number of iterations is between 10 and 500 iterations. The creation of our initial population is the location of hazardous containers in stacks; this population is shown in Table 3.

Table 3. The Generation of the Initial Population

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Ind6	P1	P3	P2	P2	P6	P4	P1	86.67110	83.67199
Ind9	P6	P4	P3	P3	P3	P1	P6	69.11800	86.77339
Ind18	P4	P4	P4	P1	P3	P3	P5	65.3277	91.60024
Ind16	P1	P4	P1	P4	P6	P1	P1	65.1919	85.47758
Ind8	P3	P5	P2	P3	P6	P1	P2	63.8072	86.68133
Ind1	P1	P5	P3	P2	P5	P1	P1	62.9639	88.35805
Ind7	P1	P3	P5	P2	P5	P1	P2	61.083100	98.53669
Ind5	P4	P1	P6	P5	P4	P5	P3	59.353900	93.81263
Ind25	P4	P3	P2	P1	P4	P3	P6	58.4046	99.86061
Ind23	P2	P3	P3	P1	P1	P2	P1	56.682799	89.61870
Ind24	P3	P6	P5	P5	P5	P4	P5	50.841300	97.00208
Ind17	P3	P2	P1	P5	P1	P5	P4	50.6666	100.0458
Ind10	P1	P4	P6	P6	P6	P2	P4	50.436099	89.10490
Ind20	P3	P6	P1	P2	P5	P2	P2	50.41800	95.77820
Ind11	P6	P4	P5	P1	P1	P3	P3	48.948800	93.35825
Ind19	P2	P1	P3	P5	P6	P2	P1	48.894200	92.13174
Ind2	P6	P2	P1	P4	P6	P3	P3	46.643700	92.00500
Ind15	P2	P4	P3	P6	P2	P6	P6	45.393699	89.94982
Ind21	P1	P4	P6	P2	P2	P1	P4	44.502100	96.62837
Ind14	P5	P4	P4	P4	P2	P2	P5	43.984400	96.96271
Ind13	P2	P4	P4	P4	P1	P5	P3	41.484700	91.86583
Ind 3	P6	P4	P1	P4	P3	P1	P1	40.963199	95.57258
Ind12	P2	P6	P1	P3	P4	P2	P3	40.524899	91.91123
Ind22	P5	P2	P6	P1	P1	P3	P6	39.937500	89.26686
Ind4	P2	P2	P6	P6	P3	P3	P2	34.815800	98.60399

The Table 3 illustrate:

Horizontally:

- All individuals who form an initial population that represents the location of hazardous containers in stacks.

And vertically:

- All the containers which will be stored in the stacks.

f) Selected Population

Selected Population represents 80 percent of the initial population (in the example, individuals are sorted in descending order according to fitness) (see Table 4).

Table 4. The Selected Population

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Ind6	P1	P3	P2	P2	P6	P4	P1	86.67110	83.67199
Ind9	P6	P4	P3	P3	P3	P1	P6	69.11800	86.77339
Ind18	P4	P4	P4	P1	P3	P3	P5	65.3277	91.60024
Ind16	P1	P4	P1	P4	P6	P1	P1	65.1919	85.47758
Ind8	P3	P5	P2	P3	P6	P1	P2	63.8072	86.68133
Ind1	P1	P5	P3	P2	P5	P1	P1	62.9639	88.35805
Ind7	P1	P3	P5	P2	P5	P1	P2	61.083100	98.53669
Ind5	P4	P1	P6	P5	P4	P5	P3	59.353900	93.81263
Ind25	P4	P3	P2	P1	P4	P3	P6	58.4046	99.86061
Ind23	P2	P3	P3	P1	P1	P2	P1	56.682799	89.61870
Ind24	P3	P6	P5	P5	P5	P4	P5	50.841300	97.00208
Ind17	P3	P2	P1	P5	P1	P5	P4	50.6666	100.0458
Ind10	P1	P4	P6	P6	P6	P2	P4	50.436099	89.10490
Ind20	P3	P6	P1	P2	P5	P2	P2	50.41800	95.77820
Ind11	P6	P4	P5	P1	P1	P3	P3	48.948800	93.35825
Ind19	P2	P1	P3	P5	P6	P2	P1	48.894200	92.13174
Ind2	P6	P2	P1	P4	P6	P3	P3	46.643700	92.00500
Ind15	P2	P4	P3	P6	P2	P6	P6	45.393699	89.94982
Ind21	P1	P4	P6	P2	P2	P1	P4	44.502100	96.62837
Ind14	P5	P4	P4	P4	P2	P2	P5	43.984400	96.96271

g) Cross Population

Cross Population represents 70 percent of the selected population (in our example individuals sorted in descending order according to fitness). Cross population is shown (see Table 5).

Table 5. The Cross Population

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Ind6	P1	P3	P2	P2	P6	P4	P1	86.671100	89.51754
Ind9	P6	P4	P3	P3	P3	P1	P6	69.118000	89.45776
Ind18	P4	P4	P4	P1	P3	P3	P2	67.666200	89.40378
Ind16	P1	P4	P1	P4	P6	P1	P1	65.191900	92.26573
Ind8	P3	P5	P2	P3	P6	P1	P5	61.468700	99.29624

h) Mutated Population

In this example, the mutation rate is 5 percent. The mutation is made between locations of the same individual (see Table 6).

Table 6. The Mutated Population

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Ind6	P1	P3	P2	P2	P6	P4	P1	86.671100	92.11608
Ind9	P6	P4	P3	P3	P3	P1	P6	69.118000	85.62624
Ind18	P4	P4	P4	P1	P3	P3	P2	67.666200	81.74348
Ind16	P1	P4	P1	P4	P6	P1	P1	65.191900	94.98558
Ind8	P3	P5	P2	P3	P6	P1	P5	61.468700	96.75926

i) Best Individuals

Best individuals are a collection of the best individuals that have almost the same locations (see Table 7).

Table 7. Best Individuals

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Ind4	P1	P3	P2	P2	P6	P4	P1	86.671100	85.39999
Ind6	P1	P4	P2	P2	P6	P4	P1	86.67110	97.91863
Ind8	P1	P4	P2	P2	P6	P4	P1	86.671100	85.69832
Ind0	P1	P4	P2	P2	P6	P4	P2	86.671100	86.65557

j) Optimal Solution

In this stage, we get the optimal individual that contains the optimal locations of containers in the stacks (see Table 8). For example, the container (Ct449) should be stored in the stack (P1), the container (Ct450) should be stored in the stack (P3) and the container (Ct451) should be stored in the stack (P2) and so on (Table 8).

Table 8. Optimal Solution

Individuals	Ct449	Ct450	Ct451	Ct452	Ct453	Ct454	Ct669	Fitness	Pen
Optimal Solution	P1	P3	P2	P2	P6	P4	P1	86.671100	85.39999

k) Assignment of containers according to their location in the storage area

The Figure 8 illustrates the state of the storage area after allocating containers according to the result of the genetic algorithm. To the right of this figure (Fig. 8) there is an event log (the identifier of each container, the optimal location of each container).

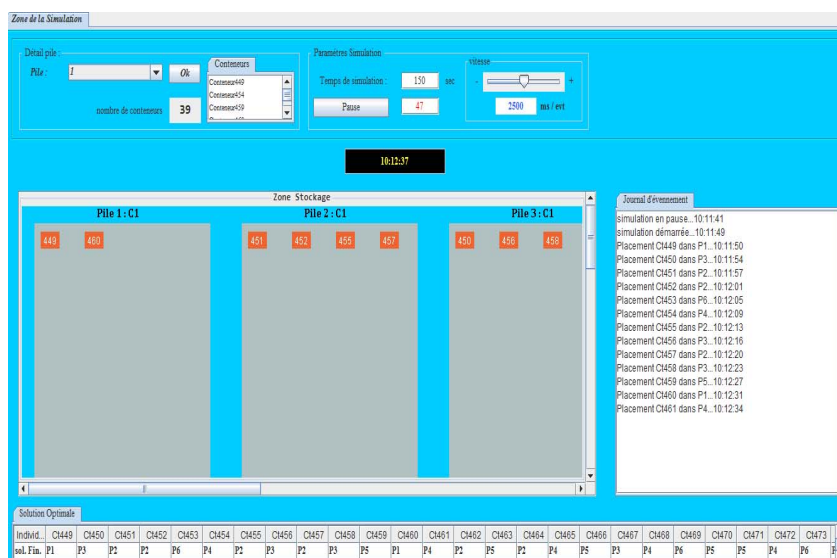


Figure 8. The Assignment Containers in Their Optimal Places

l) Discussion

The solution given by the genetic algorithm is better by providing a random solution where the dangerous containers are classified according to their arrival at the storage areas without taking into account the constraints treated in this article. The solution offered by the genetic algorithm makes it

possible to identify the best available location in the storage area (stack) instead of randomly assigning the container.

6. Convergence Criteria

Unfortunately, there is no deterministic type and stopping criteria for stochastic methods.

Some authors often use criteria based on their expertise (maximum number of iterations). Besides, this type of test is purely fictitious and can give very bad results depending on the original population and the speed. The genetic algorithm will explore space.

The criteria proposed in the literature (Bhandari *et al.*, 2012) is:

- Maximum number of generations (iteration).
- Elapsed time.
- The best individual of the last generation reached a critical level (best).

In this work, the algorithm converges after a maximum number of generations (number of iterations, when the generated optimal solutions are stable.

7. The advantages of the genetic algorithm

- Obtaining an approximate solution in proper time, to an optimization problem, when it does not exist or we do not know exact method to solve it in a reasonable time.
- A genetic algorithm gives us great freedom in setting and implementing different treatments. Free to change them if the resulting solutions do not suit us.

8. The disadvantages of the genetic algorithm

- The difficulty of setting parameters (population size, % change).
- Choosing of delicate evaluation function.
- Doubting in the found solution and accepting approximate ones.

9. Conclusions and Perspectives

To conclude, this article masterfully presented the steps for the realization of the approach when it proposed an optimization model. Indeed, the result is the fruit of an initial configuration duly identified in relation to the environment of a marine transportation network in its storage part, launching the genetic algorithm to give the final optimal locations of dangerous containers with a least cost.

The solution proposed in this article is dedicated to the handling operator in a seaport in order to facilitate them the way of storing dangerous containers instead of using a random storage way.

We aim in the future of this work to implement other algorithms and to do a comparative study of results obtained with a list of methods. Also, a hybridization with other algorithms to increase the number of factors taken into account and the use of simulation software to simulate the different movements in a seaport.

References

1. Ayachi, I., Kammarti, K., Ksouri, M. and Borne, P. (2010) A Genetic algorithm to solve the container storage space allocation problem, *International conference on Computational Intelligence and Vehicular System (CIVS)*.
2. Bazzazi, M., Safaei, N. and Javadian, N. (2009) A genetic algorithm to solve the storage space allocation problem in a container terminal, *Computers & Industrial Engineering*, 56 (2009) 44–52.
3. Bhandari, D., Murthy, C.A. and Pal, S.K. (2012) *Variance as a Stopping Criterion for Genetic Algorithms with Elitist Model*. Center for Soft Computing Research. Indian Statistical Institute. *Fundamenta Informaticae*, 120 (2012), 164–145 DOI 10.3233/FI-2012-754
4. Boelle, I. and Tourret, J. (2000) *Algorithme génétique*, Ingénieurs 2000 – IR3, Exposé. Available from: http://www-igm.univ-mlv.fr/~dr/XPOSE2007/iboelle_jtourret_algos_genetiques/resources/Pr%C3%A9sentation.pdf

5. Bokhabrine, Y. (2006) *Etude et comparaison d'algorithmes d'optimisation pour la reconstruction 3D par supershapes et R-fonctions*, Stage Master recherche. Available from: http://le2i.cnrs.fr/IMG/publications/2444_rapport%20master.pdf
6. Chebli, K. (2011) *Optimisation des mouvements des conteneurs dans un terminal maritime*, Mémoire présenté en vue de l'obtention du diplôme de maîtrise des sciences appliquées (Génie Industriel), École polytechnique de Montréal, Canada. Available from: https://publications.polymtl.ca/737/1/2011_KhaoulaChebil.pdf
7. Dekker, R., Voogd, P. and Asperen, E.V. (2007) Advanced methods for container stacking, *Container Terminals and Cargo Systems*, pp. 131-154, 2007. Available from: http://link.springer.com/chapter/10.1007%2F978-3-540-49550-5_7#page-1
8. Dubreuil, J. (2008) *La logistique des terminaux portuaires de conteneurs*, Centre interuniversitaire de recherche sur les réseaux d'entreprise, Mémoire, Centre Universitaire de Recherche sur les Réseaux d'Entreprise la Logistique et Transport, Montréal, Canada. Available from: <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2008-38.pdf>
9. Dipama, J. (2010) *Optimisation multi objectif des systèmes énergétiques*, Thèse de doctorat, Université de Montréal, Canada.
10. Jourdan, L. (2005) *Métaheuristiques pour l'extraction de connaissances: Application à la génomique*, Université des Sciences et Technologie de Lille - Lille I, France. Available from: <https://tel.archives-ouvertes.fr/tel-00007983/document>
11. Kefi, M.G. (2008) *Optimisation Heuristique Distribuée du Problème de Stockage de Conteneurs dans un Port*. Thèse du doctorat, l'Ecole Centrale de Lille, France, Available from: <https://tel.archives-ouvertes.fr/tel-00366467/document>.
12. Kim, K.H. and Ma, Y. (2012) A comparative Analysis: Various storage rules in container yards and their performances, *Industrial engineering & Management systems*, vol. 11, pp. 276-287, 2012.
13. Kim, K. and Kim, H. (1997) A routing algorithm for a single transfer crane to load export containers onto a containership, *Computers and Industrial Engineering*, 33(3-4), pp. 673-676, doi:10.1016/S0360-8352(97)00219-2
14. Korbaa, S. and Yim, O. (2004) Assignment to stock in a fluvial port. In: *International Conference on Systems, Man and Cybernetics*, The Netherlands.
15. Murty, K.G., Liu, J., Wan, Y.W. and Linn, R. (2005) A decision support system for operations in a container shipping terminal. *Decision Support Systems*. 2005, 39, pp. 309-332.
16. Ndeye, F.N. (2014) A Branch-and-Cut Algorithm to Solve the Container Storage Problem, *The Ninth International Conference on Systems*, University of Le Havre, France.
17. Schwartz, P. (2005) *Les algorithmes génétiques*. Developpez.com. Available from: <http://khayyam.developpez.com/articles/algo/genetic/>
18. Sharma, M., Sindhwani, P. and Maheshwari, V. (2014) Genetic Algorithm Optimal approach for Scheduling Processes in Operating System, *IJCSNS International Journal of Computer Science and Network Security*, vol.14, No.5, May 2014.
19. United Nations (2009) *Recommendations on the transport of dangerous goods, Model Regulations*, Vol. II, Sixteenth revised edition. Available from: <https://www.unece.org/fileadmin/DAM/tkrans/danger/publi/unrec/rev16/English/Volume2.pdf>
20. Yachba, K. and Bouamrane, K. (2015) Containers storage optimization in a container terminal using a multimethod multi-level approach. In: *The International Conference on Computers & Industrial Engineering (CIE45)*, 28-30 October 2015, Metz, France. ISSN 2164-8689.