Container vessel stowage plan using genetic, hill-climbing and simulated annealing algorithms

M.A. Yurtseven, E. Boulougouris, O. Turan
Department of Naval Architecture, Ocean and Marine Engineering (NAOME)
The University of Strathclyde, Glasgow, The UK

ABSTRACT: Container vessel stowage is vital for reducing extra cost and the vessel's turn-around time in ports. Several studies have been conducted to be able to solve this problem. However, this problem has not been dealt with completely due to container stowage problem is complex problem which is literally considered as a NP-hard problem. Hence, the principal objective of this study is to find optimal container stowage plan for container vessel calling at multiple ports.

Algorithms are the most widely-used methods for finding optimal solution in container vessel stowage problem. In this study, Genetic, Hill-Climbing and Simulated Annealing algorithm are implemented considering same size and two different types of containers (refrigerated and standard containers). After finding the numerical test results from these three algorithms, the effectiveness of the algorithms for the problem are evaluated.

1.INTRODUCTION

Seaborne trade is of the main component in the international trading since it holds the over 80 per cent and more than 70 per cent of world merchandise trade by volume and its value respectively (UNCTAD, 2017).

In the history of maritime industry development, containerization is fast becoming a key instrument in liner shipping. The most of the world's seaborne containerized shipping is operated in liner shipping. This liner shipping is carried by variety capacity of special designed seagoing vessels that can carry over tens of thousands of containers on regular advertised schedules between ports (Tierney,2015).

For last few decades, the size of container vessels has increased continuously exceeding over 21,000 Twenty Foot Equivalent Units (TEUs). This increase provides the advantage in reducing the vessel running costs. Along with this growth in vessel capacity, however, some serious challenges have been encountered in operating these vessels.

Container ship stowage planning(CSSP) in shipping industry generally known as Master Bay Plan Problem can be counted the one of the serious challenges needed to cope with because it is literally seen a NP-hard problem (non-deterministic polynomial-time). Hence, to generate well-conceived stowage plan for container vessel is not straightforward and carried out by human planners.

Stowing containers on the vessel is restricted in a last-in first-out method. Considering the huge number of varied containers carried, the sequence of ports visited, stability and the other restrictions, several containers in subsequent ports are temporarily unloaded and reloaded on a multiport voyage. This is called re-handle which causes the increase in vessel turnaround time and port expenses.

The last few decades, there has been an increasing amount of literature on this subject. Different methodologies exist in the literature regarding optimization of container ship stowage plan. Up to now, a number of studies have attempted to deal with the CSSP implementing different algorithms.

Avriel & Penn (1993) suggest a 0-1 binary linear programming formulation that aim to minimize the number of shiftings from stowage planning. For this problem, while the vessel stability constraint is not considered, the number of ports to be visited and the number of containers to be shipped are determined in advance. The GAMS software is used to implement the model.

An analysis and discussion on the subject of stowage plan for small vessel running in short sea shipping is presented by Martins et. al. (2009) using and comparing two different approaches that are Microsoft Excel Solver and GA.

Liang et. al. (2016) present a new optimization approach named Social Network-based Swarm Optimization Algorithm(SNSO) in order to able to deal with the slot planning problem of container vessel bays.

Cohen et al. (2017), in their paper, divide the CSSP in two phases which are master bay planning phase and slot planning phase respectively. Firstly, master bay plan phase deals with container distribution to each bay. Secondly, slot planning phase arrange specific slot for each container assigned to each bay in first phase. The authors utilise a GA approach solve the problem in this study.

In Nikos's thesis (2017), it is adopted a GA approach to produce a feasible CSP strategy in order to provide accuracy result and low computational time. The problem covers some constraints such as, the moment between form bow and stern sections, the maximum weight of stack, heel righting moment and also the vessel sails in full loading condition during the voyage.

The aim of this study is to find the optimal appropriate solution in terms of container location assignment on board container vessel. Herein, three different algorithms are deployed by MatLab software to assign containers into cells on container vessel.

The overall structure of the paper takes the form of four sections. The second section explains the problem statement. The third section gives briefly three different algorithms to be deployed in the problem. The final section compares the results and the performance obtained from the algorithms and concludes with authors' final remarks.

2 PROBLEM DESCRIPTION

In this section, we describe not only the representative problem with variables, constraints and but also the methodology which is followed for the problem solution.

The present study was designed to reduce the containers moves throughout the vessel sailing. So that, unnecessary expenses will be able to drop from undesired containers moves.

For this problem, we will introduce the formula generated by ourselves to calculate the cost value from the number of moves occurred at every port. The formula is as follows,

$$C = \sum_{j=1}^{Ncol} \sum_{k=1}^{Nrow} 2k \left(abs \left(sign \left(\prod_{i=1}^{Nrow-k} \left(X(k,j) - X(Nrow\text{-}i+1,j) \right) \right) \right) \right)$$

Where;

C is cost of total containers movement,

$Ncol$ is the number of column

$Nrow$ is the number of rows

x_{kj} is 1 if the ; 0 otherwise

Let us give a simple example to understand the calculation of the cost by using different container stowing figurations, looking the given the data in the table 1 and assume that refrigerated containers (reefers) can be placed only across last two bottom rows (one of the constraints) where are near power plugs.

Ports of Destination (n)	1	2	3	4
Refrigerated Containers	2	2	2	2
Total Containers Quantity	4	4	4	4

Table 1. container distribution for each port with reefers

R= Refrigerated Containers (Reefer)

n= Port of destination

The given containers data on the table 1 with their own destination are placed to slots as in the Table 2 below. As seen on the table 2, the containers are so distributed that no need the shifting throughout its journey. In other words, there would not be additional cost. Hence, the optimum cost function value is 32.

1	2	3	4
1	2	3	4
R1	R2	R3	R4
R1	R2	R3	R4

Table 2. possible stowage plan as an example

On the other hand, if the containers in table 2 are allocated as in Table 3, the new optimum cost function value increases to 40 because compulsory shifting has to be executed in table 3.

1	2	3	4
2	1	3	4
R1	R2	R3	R4
R1	R2	R3	R4

Table 3. possible stowage plan in different condition

Similarly, if the containers in Table 2 are allocated to slots as the table 4 the updated cost function value will be 44 because more compulsory shifting has to be executed in table 4

1	2	3	4
1	2	3	4
R1	R2	R3	R4
R2	R1	R3	R4

Table 4. possible stowage plan to data in table1

The reason for this cost difference between tables above is the sequence of the containers into the cells.

Even if solving this problem is straightforward manually because of small number of containers carried, to find the optimal solution for the huge number of container stowage problem cannot be easy like the examples above. As it seen, in the example above the numbers of the variables (containers) are 16. However, in our problem the number of variables (containers) to be used is much larger. Therefore, Hill-climbing, Genetic, Simulated Annealing algorithms techniques are implemented to be able to solve our problem.

This problem model is built under these assumptions;

- The number of containers carried and their destination port are known

• The bay layout is known including rows and columns, also considering that the whole vessel consisting of one bay.
• Each discharging container at a container terminal must be replaced by other container in the same slot at the same container terminal. That is, container vessel must be operated full capacity during throughout its journey
• Each container loading and discharging movement total cost 2 units
• Slots can be occupied by only one container.
• Two types and same size (40' long) containers can be loaded and unloaded
• Refrigerated containers cannot be assigned in non-reefer slots.
• Stability, hatch cover and containers' weight are ignored

For our problem, a container vessel consists of a single bay with fifteen columns of ten rows (15x10) and 150 containers (variables) and refrigerated containers (reefers) can be placed only across the last 4 bottom rows where power plugs exist. Also, the number of vessel calling ports are (P) =5.

Every slot from S1 to S150 has to be filled and the slots assigned for refrigerated containers are must be filled only by refrigerated containers.

Ports of Destination	1	2	3	4	5
Refrigerated Containers (Reefer)	11	13	7	15	14
Total Containers Quantity	23	31	37	26	33

Table 5: container distribution according to the ports

The data in table 5 was generated and distributed randomly to the 5 destination ports.

For the problem solution, it will be benefit from three different algorithms. Our objective function is represented by the cost function which helps to calculate total container moves for whole journey

3 THE OPTIMIZATION METHODS FOR THE PROBLEM SOLUTION

Optimization is a way of finding optimal solution of the problem. The optimal solution is sought for in the whole solution space. For the problem in this study, three different algorithms, GA, SA, Hill-climbing Algorithm are proposed to minimize the number of shifting) operation.

3.1 Hill Climbing Algorithm

Hill climbing, our first optimization method implementing in this problem, carries out a loop where the currently known best solution state p* is employed to generate one new state p_{new}. Then, these two states are compared. If the new state is better than previous one, the new state replaces it and becomes the best solution. Afterwards, this process stars all over again. On the other hand, it can be faced the problem which happens premature convergence which results in stopping the algorithm process while it runs. In other words, the algorithm reaches an impasse easily on a local optimum and gets stuck.(Weis, 2009.)

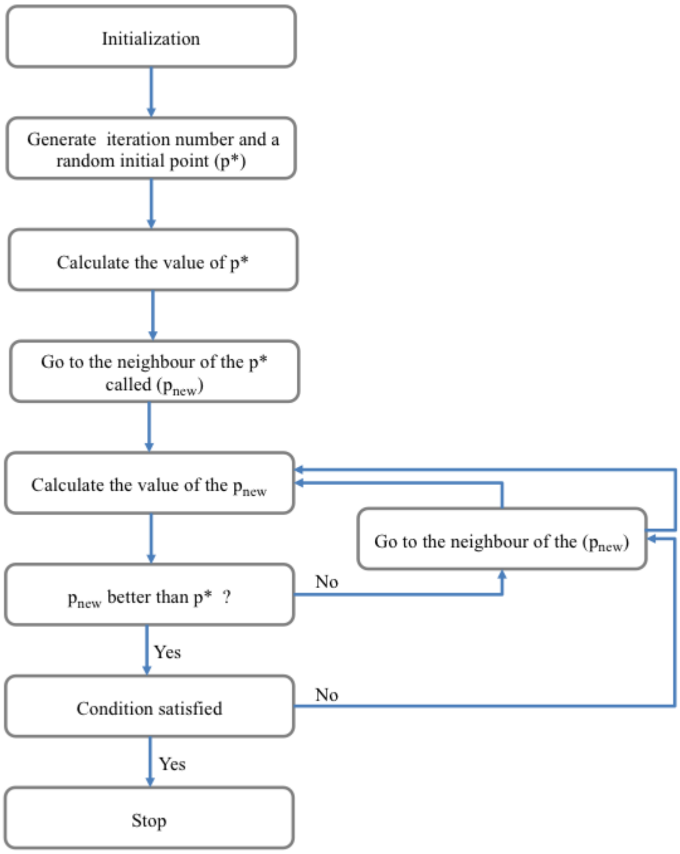

Figure1. Hill climbing algorithm flow chart

Variable	Iteration	Best Cost Value	Time(min)
150	10	398	5.2
150	50	374	26.4
150	100	374	54.6
150	200	370	113.5
150	300	372	169.0
150	400	378	216.8
150	500	374	281.1
150	1000	370	526.1
150	2000	370	1063.6

Table 6. results from the use of different parameter in Hill-climbing

As shown in table 7, the different parameters were employed in Hill Climbing. With increasing iteration number, the computational time increases and so the best solution is also obtained.

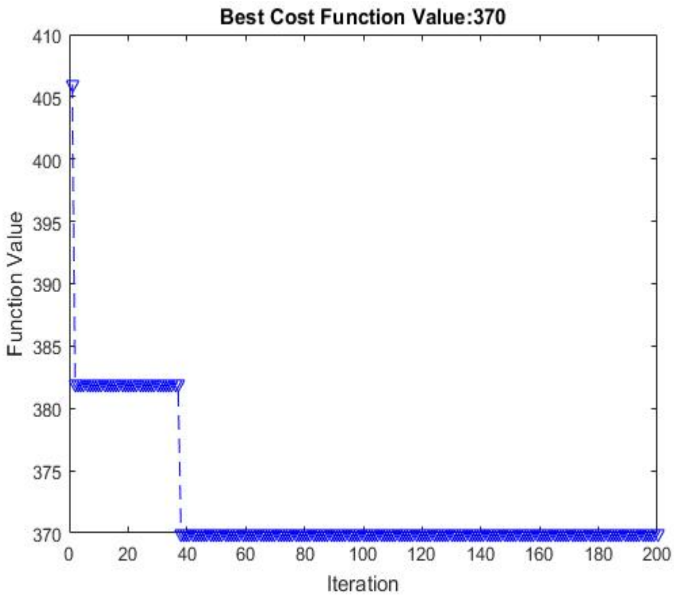

Figure 2. Simulation result of the Hill Climbing method. Total processing time is 113.5 minute and best cost function value is 370.

3	3	5	1	3	5	4	1	5	5	3	2	3	2	3
4	3	5	1	3	2	4	1	5	5	3	2	3	2	3
4	3	5	1	3	2	4	1	5	5	3	2	3	2	3
4	3	5	2	3	2	4	1	5	5	3	2	3	2	3
4	3	5	1	3	2	4	1	5	5	3	2	3	2	3
4	3	5	1	3	2	4	1	5	5	3	2	1	2	3
14	15	15	12	14	12	14	11	15	15	13	12	11	12	13
14	15	15	11	14	12	14	11	15	15	13	12	11	12	14
14	13	15	11	14	12	14	11	15	15	13	12	11	12	14
14	13	15	11	14	12	14	11	15	15	13	12	11	12	14

Table7. outline of container distribution in a bay after using Hill Climbing

It can be seen in Figure 2 that Hill Climbing calculated the best cost function value as 370 in 113.5 minute. As shown in Table 7, the numbers in the slots represent destination port for that slot's container after Hill Climbing method implementation.

3.2 Problem Solution using Genetic Algorithm

Gas applied as a successful computational method in optimization of mathematical complex problems are biologically inspired by the basic principles of natural selection and evolution.

In the point of theoretical steps of GAs;

1) A random initial population(chromosomes) is generated Calculate the fitness function of the generated population

2) Compere the fitness function with the existing criteria in the problem

a) If criteria are met, stop it

b) If criteria are not met, go next step

3) Choose elite member looking the best fitness value

4) Produce offsprings applying crossover and then mutation

5) Replace the new generation with the current generation

6) Go to step 2 and 3 respectively

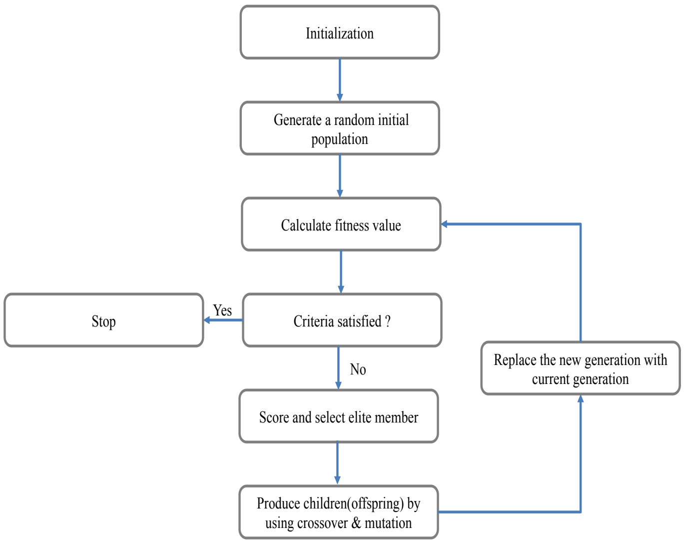

Figure 3. Genetic algorithm flow chart

Initial inputs (chromosome encoding) required for the population in this problem are considered as same size of the number of containers which represents our variables. Each generation has a certain amount of population size. The proper solution space searching is directly proportional with the number of population however; the computational time is inversely proportional with the number of population.

The number of repetition crucial parameter for finding the optimal solution is defined as the number of generations based on the information in Matlab options. This repetition number is, in practice, 100*Number of variables. However, in this problem the number of generation is taken 10000 because of increasing the computational time. The problem result to Genetic Algorithm is as follows

Variable	Iteration	Population Size	Elite Count	Best Cost Value	Time(Min)
150	1000	500	25	404	1.8
150	1000	1000	50	396	3.1
150	2000	500	25	406	4.4
150	2000	1000	50	416	7
150	3000	500	25	402	7.4
150	3000	1000	50	398	10.9
150	5000	500	25	390	15.3
150	5000	1000	50	420	21.5
150	10000	500	25	406	48.2
150	10000	1000	50	378	60.5

Table 8. results from the use of different parameter in GA

As shown in table 8, the different parameters were used in GA. With increasing iteration number, the computational time increases and so the best solution is also obtained.

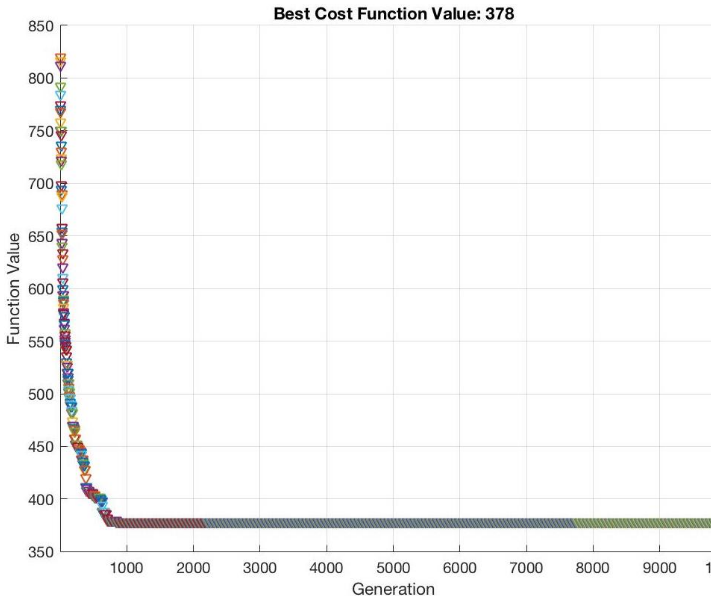

Figure 4. Simulation result of the GA method. Total processing time is 60.5 minute and best cost function value is 378.

5	3	3	3	2	3	2	3	3	1	5	3	4	3	2
5	5	3	3	2	3	2	4	3	1	5	3	4	3	2
5	5	3	3	2	3	2	4	3	1	5	1	4	3	2
5	5	3	3	2	3	1	4	3	1	5	1	4	3	2
5	5	2	3	2	3	2	4	5	1	5	1	4	3	2
5	5	2	3	2	3	1	4	5	1	5	1	4	3	2
15	15	12	15	12	14	12	14	15	11	15	11	14	13	12
14	15	12	13	12	14	11	14	15	11	15	11	14	13	12
14	15	12	13	12	14	11	14	15	11	15	11	14	13	12
14	15	12	13	12	14	11	14	15	11	15	11	14	13	12

Table 9. outline of container distribution in a bay after using GA

It can be seen in Figure 4 that GA calculated the best cost function value as 378 in 60.5 minute. As shown in Table 9, the numbers in the slots represent destination port for that slot's container after GA method implementation.

3.3 *Problem Solution using Simulated Annealing Algorithm*

For this problem, the last possible experimental way of finding optimal solution is SA. The study of SA was first carried out by Kirkpatric et. al. (1983).

To begin SA process, a random trial point generates and then it calculates. In the meantime, the initial value of temperature which is crucial parameter is assigned for the problem to be optimised. Afterwards, according to a probability distribution with a scale based on the current temperature, the distance of the trial point is chosen from the current point. The trial point distance distribution is set as a function with the AnnealingFcn option. The trial point can be changed via SA, if needed, to be able to stay within bounds. Then, the algorithm compares the new point with the current point which one is better or worse. If the new point is better, the new point is accepted and used as a next point. If not, again the next point is generated using the worse point depending on an acceptance function

If the new point is better than the current point, the new point is used as a next point, otherwise the SA tries to create the next point using the worse point depending on an acceptance function (probability base). Then, the temperature is dropped systematically via the SA algorithm, recording the best point obtained so far. The algorithm used the specified the function by TemperatureFcn option to update the temperature. The annealing parameter remain stable same as the iteration number till reannealing.

After Simulannealbnd accepts ReannealInterval points, it reanneals. To reduce values than the iteration number Reannealing arranges the annealing parameters, hence the temperature increases in each dimension. Also, the estimated gradients of the objective function values in each dimension describes the annealing parameters.

Once the average change in the objective function is small relative to FunctionTolerance, the SA algorithm terminates or when it satisfies any other stopping conditions (MathWorks, 2017).

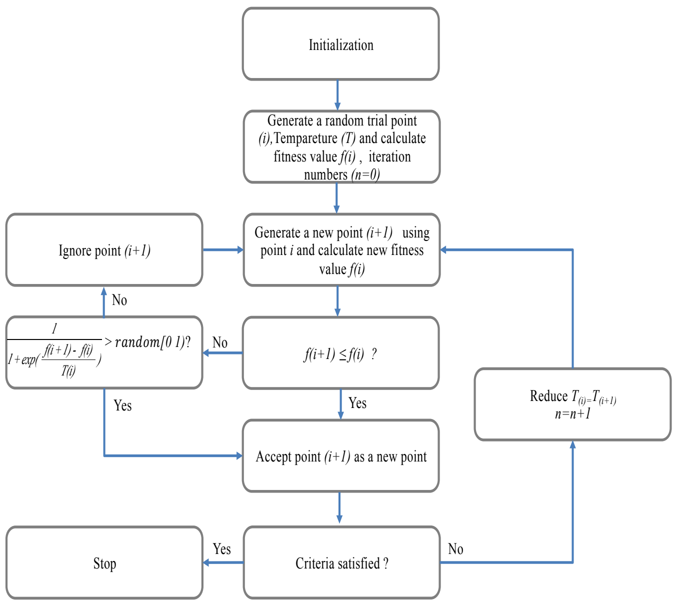

Figure 5. Simulated annealing algorithm flow chart

The problem results to Simulated Annealing Algorithm is as follows;

Variable	Iteration	ReannealInterval	Best Cost Value	Time(Min)
150	10000	200	380	1.09
150	10000	300	380	1.08
150	20000	100	380	2.24
150	20000	300	384	2.49
150	30000	300	394	4.10
150	30000	400	398	3.89
150	40000	100	386	4.75
150	40000	500	400	5.17
150	50000	200	370	6.92
150	50000	400	376	5.66
150	100000	100	382	13.66
150	100000	500	370	14.29

Table 10. results from the use of different parameter in SA

As shown in table 5, the different parameters were used in SA. With increasing iteration number the computational time increases and so the best solution is also obtained.

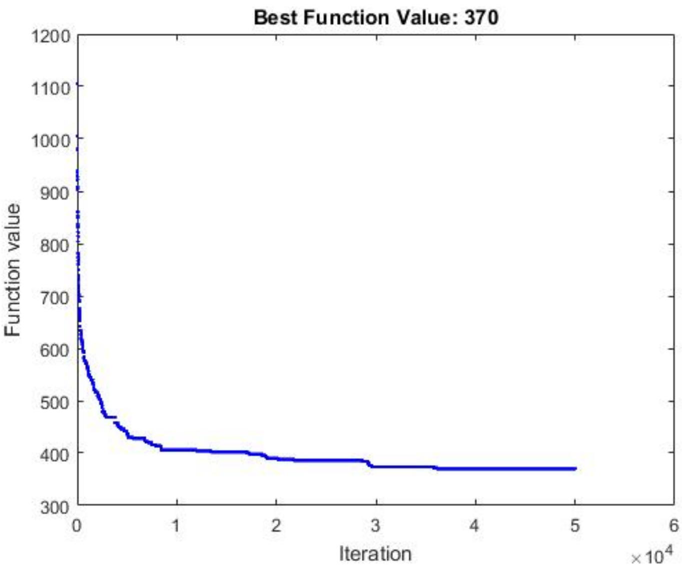

Figure 6. Simulation result of the SA method. Total processing time is around 7 minutes and best cost function value is 370.

3	2	2	3	5	3	3	1	5	1	5	3	5	3	3
3	2	2	3	5	4	3	1	5	1	5	3	4	2	3
3	2	2	3	5	4	3	1	5	1	5	3	2	2	3
3	2	2	3	5	4	3	1	5	1	5	3	4	2	3
3	2	2	3	5	4	3	1	5	1	5	3	4	2	3
4	2	2	3	5	4	4	1	5	1	5	3	4	2	3
14	12	12	15	15	14	14	11	15	11	15	13	12	12	13
14	12	12	15	15	14	14	11	15	11	15	11	14	12	13
14	12	12	13	15	14	14	11	15	11	15	11	14	12	13
14	12	12	13	15	14	14	11	15	11	15	11	14	12	13

Table 11. outline of container distribution in a bay after using SA Algorithm

It can be seen in Figure 6 that SA calculated the best cost function value as 370 in around 7 minutes. As shown in Table 11, the numbers in the slots represent destination port for that slot's container after SA method implementation.

Importantly, the best scenario to our problem data, the cost value would be 300 if whole container was travelled to the same destination. So, the results gained from three algorithms show that there is no huge differences between them on the other hand this situation is not seen for the computational time.

4 CONCLUSION

The aim of the present research was to find the optimal solution for container stowing onboard a container vessel. To solve this problem, the study set out to implement three different algorithms that are Genetic, Simulated Annealing, Hill Climbing. The obtained results from these methodologies indicate that there is no significant difference between these algorithms' results. However, a comparison of the three algorithm results reveals that each computational time are significantly different form each other to cover the best solution. SA, GA and Hill Climbing covered their best solutions in almost 7, around 60 minutes and around 449 minutes respectively. Overall, these results indicate that SA algorithm is better and faster than the Hill Climbing and GA because of their working principles even if the results gained from them are quite close to each other.

5.REFERENCES:

S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi,"Optimization by Simulated Annealing," Science, new series, 220.4598, 1983, pp. 671 – 680

Avriel, M. & Penn, M.1993. Exact and approximate solutions of the container ship stowage problem[J]. Computers and and Industrial Engineering, 25(1-4): 271-274.

Liang,X. and Li,B. and Li,W. and Zhang,Y. and Yang,L. 2016. A Method Based on SNSO for Solving Slot Planning Problem of Container Vessel Bays. 9th International Conference, IDCS 2016, Wuhan, China, September 28-30, 2016, Proceedings

Martins, P. T. and Lobo, V. J.A.S and Vairinhos V. 2009. Container stowage problem solution for short sea shipping. Proceedings of 14th Congress of the Portuguese Association of Operational Analysis

Cohen M.W., Coelho V.N., Dahan A., Kaspi I. 2017. Container Vessel Stowage Planning System Using Genetic Algorithm. In: Squillero G., Sim K. (eds) Applications of Evolutionary Computation. EvoApplications 2017. Lecture Notes in Computer Science, vol 10199. Springer, Cham

MathWorks 2017. How simulated annealing works [online] Available at: http://uk.mathworks.com/help/gads/how-simulated-annealing-works.html [Accessed 22 Oct. 2017].

UNCTAD (2017). Review of Maritime Transport 2017. [online] Available at: http://unctad.org/en/PublicationsLibrary/rmt2017_en.pdf [Accessed 10 Oct. 2017]

Tierney K (2015) Optimizing liner shipping fleet repositioning plans, operations research/Computer science interfaces series, vol 57. p.7 Springer International Publishing, New York

Nikolas P.2017.Optimization of Container Loading. Ph.D. thesis. Glasgow: The University of Strathclyde

Weise,T. 2009. *Global Optimization Algorithms-Theory and Application*. [online] Available at: http://www.it-weise.de/projects/book.pdf [Accessed 10 Dec. 2017]