Journal of Sustainable Development of Transport and Logistics
journal home page: https://jsdtl.sciview.net

Amuji, H. O., Onwuegbuchunam, D. E., Aponjolosun, M. O., Okeke, K. O., Mbachu, J. C., & Ojutalayo, J. F. (2022). The dynamic programming model for optimal allocation of laden containers to Nigerian seaports. *Journal of Sustainable Development of Transport and Logistics, 7*(2), 69-79. doi:10.14254/jsdtl.2022.7-2.5.

Scientific Platform
ISSN 2520-2979

The dynamic programming model for optimal allocation of laden shipping containers to Nigerian seaports

Harrison O. Amuji *, Donatus E. Onwuegbuchunam **, Moses O. Aponjolosun **, Kenneth O. Okeke **, Justice C. Mbachu **, John F. Ojutalayo ***

** Department of Statistics, Department of Maritime Management Technology, Federal University of Technology Owerri, Nigeria*
Harrison.amuji@futo.edu.ng
*** Department of Maritime Management Technology, Federal University of Technology, Owerri, Nigeria*
don@futo.edu.ng; aponjolosunmoses@gmail.com
**** Department of Nautical Science, Federal College of Fisheries and Marine Technology, Victoria Island, Lagos, Nigeria*

OPEN ACCESS

Article history:

Received: September 06, 2022
1st Revision: September 29, 2022
Accepted: November 20, 2022

DOI:
10.14254/jsdtl.2022.7-2.5

Abstract: In highly competitive shipping market environment, container network operators-Freight forwarders, shipping companies etc. are concerned about design, development and deployment of optimized allocation model to achieve cost savings through improved container storage yard operations, crane productivity, outbound container allocation/distribution to seaport terminals and hence reduction in ships' waiting times. In this paper, we developed two models, the Dynamic programming model and optimal allocation policy (model), for the optimal allocation of units of outbound laden cargo containers of sizes: 20ft and 40ft to six (6) major seaports in Nigeria. The distributions of the laden containers were allocated as follows: Port-Harcourt, Tincan Island, Onne, and Calabar seaports were allocated with 1,064 units of stuffed containers each. Apapa seaport was allocated with 2,128 units of laden containers, and zero allocation was made to Warri seaport. These results were arrived at through the implementation of the optimal allocation policy. The zero units allocation made to Warri seaport could be attributed to poor shipper patronage and hence the low frequency of ship visits. Apapa seaport was allocated double the number of containers moved to the remaining ports because it attracted more shipper patronage and hence more ship visits. Hence, freight forwarding companies will be assured of cargo spaces and make more profit by allocating more containers. Policy implications of the developed models were discussed.

Keywords: dynamic programming, ships' waiting times, optimal

Corresponding author: *Donatus E. Onwuegbuchunam*
E-mail: don@futo.edu.ng

allocation policy, Nigerian seaports, loaded shipping containers

1. Introduction

Globalization of production centres has engendered the upsurge in deployment of cargo shipping containers for improved handling and security cargoes at nodal points. Between years 2000 and 2014, the worldwide container traffic has grown from 225 million of Twenty-foot Equivalent Units (TEU) to 680 million TEU, with an increase of about 200% and this value is expected to nearly double by 2020 (Fancesco et. al, 2018). On the other hand, container network operators have been inundated with increasing demand and pressure for optimized management of container yard operations, reduction of uncertainties leading to over capacity or underutilization of container units. Other expectations include improved crane productivity, cost effective outbound distribution/allocation to seaport terminals and hence reduction in ships' waiting times. In Nigeria's maritime sector, container operators, shippers and terminal operators e.g., are faced with challenges emanating from low investment in container yard facilities and multimodal transport infrastructures despite the terminal concession reforms (Onwuegbuchunam, 2018). It has therefore become critical to employ cost effective optimization models to reduce cargo container dwell times and ships' waiting times at the port. Dynamic programming is one of the family models that have been applied in a container network setting involving shipping companies, freight forwarders, road haulage, bonded warehouse operators and port terminal operators to tackle shipping container allocation problem.

Dynamic programming is a class of non-linear programming that applies the principle of optimality and recursive relationship to attain an optimal decision. The optimal decision at the last stage of the problem depends on the preceding decisions at the initial stages. Though non-linear, its computation has a linear relationship. Dynamic programming has variant forms, depending on the problem to be solved; hence, the name. Its optimal solution always goes beyond the immediate presentation to provide much more meaning. It focuses on the principles of decision theory. It provides a solution to various problems that may be too complex to solve using other methods by breaking the solution into different stages so that optimal solutions can be obtained at each stage, and the combination of these stages gives a complete solution to the problem. Dynamic programming is a mathematical technique for solving sequential decision problems. The sequential decision problem is when a sequence of decisions must be made, with each decision affecting future decisions. Dynamic programming is associated with stage variables and state variables. State variables are the variables whose values specify the instantaneous situation of the process, while the stage variables represent the stages. Dynamic programming could be serial or non-serial (Ronald, 1966). The dynamic programming technique decomposes a multistage decision problem as a sequence of single-stage decision problems. A multistage decision process is one in which a number of single-stage processes are connected in series so that the output of one stage is the input of the succeeding stage. This is serial dynamic programming since the individual stages are connected head to tail with no recycle (Amuji, Uguanyim, Ogbonna, Iwu & Okechukwu 2017).

In this paper, we developed dynamic programming models for the optimal allocation of 20ft & 40ft containers (laden with cargo) to six (6) major seaports in Nigeria. To achieve this, we applied the developed models to determine the distribution of the cargo-stuffed containers presented in Table 4.1. We then developed a dynamic programming model suitable for allocating the loaded container units to the six seaports and finally developed the optimal allocation policy. This policy proffered a solution for the number of containers allocated to each seaport to optimize profit. Though we built on the existing literature in this area of study, no such work has been previously done. In this paper, stuffed, laden and loaded containers are used interchangeably. This paper is divided into five sections. Section one gives a general introduction to the work. Section two treats the literature review. Section three treats the material and method. Section four treats the data presentation and analysis, and section five treats the results and discussions. A calculation on how tables 4.6 to table 4.10 were obtained was presented briefly in the appendix.

2. Literature review

Augustine & Barry (1974) trace the development of non-serial dynamic programming from the basic theory underlying dynamic programming to the latest applications of non-serial dynamic programming. They thought that the applications of non-serial dynamic programming could be extended from chemical engineering, natural gas pipeline, and water resource systems to more complex systems. The non-serial structure is where at least one stage in the system receives inputs from more than one stage or sends outputs to more than one stage. This situation exists whenever a system combines serial and parallel processes. Such systems are found in the study of chemical processing systems, natural gas transmission pipelines, water resource systems, and various other processes. In this paper, we are concerned with the serial structure, where one stage receives input from the previous stage and sends its output to precisely one preceding stage.

Adelson, Norman & Laporte (1976) demonstrate the various applications of dynamic programming modelling in solving real-life problems. Such applications include archaeological findings, where they observed that it might be required to arrange in sequence a number of archaeological sites based on the various types of pottery found there. This objective is based on the "archaeological view that different kinds of objects, by and large, tend to replace one another rather than be in use simultaneously. Many authors on dynamic programming for example, Peters (1989) complained about the lack of practical applications of the technique. However, the increasingly powerful computing facilities now available mean that the solution to many earlier unresolved problems is becoming a reality. The dynamic program used by the authors in their paper was presented as a financial control model, and its optimization allows the user to incorporate new information as it becomes available. However, dynamic programming has not been generally used in solving significant scale problems because of the sizeable high-speed memory and excessive computational time requirements (Peter, 1970). The author presents a new decomposition procedure that reduces both the high-speed memory requirement and the computational time by introducing interpolations in the dynamic programming algorithm.

In developing dynamic programming recursive formulae, the problem is decomposed into different stages, which are evaluated independently, given a set of environmental conditions (states). By combining the solutions to the smaller problems, we obtain the solution to the whole problem. Since the new state gives rise to a new decision, it is possible to link together the sub-problems. The key to the process is the principle of optimality. Brian (1986) used a dynamic programming approach to design a transformer. The author observed that while dynamic programming might be an intellectually appealing way of formulating problems, people believe that it does not help solve them; but admitted that dynamic programming could be used in electrical engineering to handle some of the tasks which appeared to be both time consuming and exceedingly dull such as the design of the transformer. Finally, Sophie, Laetitia & El-Ghazali (2016) used multi-objective dynamic programming to improve their design and operational strategies. The researchers aimed to adapt a dynamic programming-based meta-heuristic to solve optimization problems and apply it to the multi-objective unit commitment problem (MO-UCP). They thought their model overcame standard evolutionary operators' poor performance on such heavily-constrained problems.

Recent papers have focused on container yard storage space allocation. Dhahri, Mezghani, & Rekik, (2020) were more concerned with storage space optimization of port terminals. A multi-objective programming approach- Weighted goal programming was applied to balance loading and unloading container storage spaces and minimize storage costs. Guo, Atasoy, Van-Blokland, and Negenborn (2020) however, applied stochastic programming to investigate dynamic and stochastic shipment matching problems in multimodal transport terminals. Incorporating stochastic information in the optimization model enabled the authors to solve barge and train capacity challenges in static and real time scenarios. Consistent with Guo et al., (2020), Chena, Lua, Xinb, Yangc, Zhud and Xue (2022), addressed stochastic demand in inland container stations. Their allocation model significantly reduced own-empty container storage level, provided liner companies decision support mechanism to deal with fluctuations in empty container demand, and management. From energy efficiency perspective Cobo (2016) used discrete event simulation to analyse container yard operations. The model proposed new stacking algorithms to reduce energy costs, improve crane productivity and dimensions of perpendicular yard layouts and distribution of containers in yards leading to reduced operational costs.

Xu, Wang, Lai and Ram (2022) addressed stacking space allocation of export containers using hybrid storage model. The model combined class-dedicated and sharing strategies to construct stochastic programming model using the concept of recourse. This approach yielded policy for allocating space for containers. While Facchini, Boenzi, Digiesi, & Mummolo, (2018) provided a model based decision support system for multiple container terminal hub management, De Armas, Valdes, Morell, Bello (2019), applied two methods- integer programming and metaheuristic methods to optimize storage space allocation to imported containers based on which one returns optimization value and computational times. It should be noted that most of these papers utilised stochastic and integer programming combining discrete event simulation and other heuristics. They were also concerned with finding efficient storage space allocation for yard optimization, better link operations and cost savings. However, there is little application of dynamic programming involving allocation of outbound containers involving profit maximization. In this paper, our interest is in the serial dynamic programming system where one stage output forms the input for the preceding stage, and at the end, the independent decisions from each stage form the optimal decision for the entire process.

3. Materials and methods

3.1. Introduction

Two cargo container sizes were involved: 20ft and 40ft, and were to be stuffed with general cargoes by three Freight Forwarding (FF) companies owned by Global Shipping Inc. The stuffed containers would then be allocated to six major seaports. The first phase covers six months (January to June 2021) of the stuffing of 6,200 units of 20ft shipping containers with general cargoes, and the second phase covers six months (July to December) of the stuffing (or production) of 7,500 units of 40ft containers. The monthly weighted capacities for each month were given to each of the forwarding companies (table 3.1). These weights depend on their past performance in cargo stuffing and forwarding efficiency at the ports. This paper adopts the production weight [1] given in equation (1) to obtain the proportion of containers due to each of the three FF companies, see table 4.1. Our interest in this paper is to allocate the total units of stuffed containers to the six seaports in the study to optimize profit. In trying to achieve this, we develop a dynamic programming model that will suit the stuffed containers' distribution to the different seaports. Secondly, we develop the optimal allocation policy to determine the number of containers allocated to each seaport to maximize profit.

$$p(N \mid n) = \frac{1/k_n}{\sum_{i=1}^{n} 1/k_i} N, \quad n = 1, 2, \ldots \tag{1}$$

Equation (1) is the scheduled production weight, where K_i is the assigned weight.

3.2. Development of dynamic programming model

Assumptions of the model:

The following are the assumptions of dynamic programming;

1. Optimal decision at the future stage is independent of the optimal decision at the previous stage.
2. Optimal solution contains optimal sub-solution.
3. The problem has overlapping sub-problems.

By recursive relationship, we can find an optimal solution if a problem has an optimal substructure. If a problem has an overlapping sub-problem, we can, by recursive implementation, compute each sub-problem only once. The above three conditions are necessary and sufficient conditions for the application of dynamic programming.

For us to develop the dynamic programming model, we define the following variables; let

S_n = state variable

n = stage variables
y_n = decision variables
R_n = returns on investment
f_n = function of state and stage variables
y_n^* = optimal decision variables
f_{n+1}^* = optimal function of stage and state variables at the current time.
Therefore,

$$\sum_{i=1}^{n} y_n = S_n \tag{2}$$

$$\sum_{i=1}^{n} y_n \mathrm{n} \ = \ \text{objective function} \tag{3}$$

Where $y_n > 0$, $S_n > 0$ and $n > 0$

Equation (2) is the constraint equation. We need to maximize the objective function subject to the constraint equation to have

$$Max \sum y_n n + (S_n - \sum y_n) \tag{4}$$

Applying the principle of optimality, which states that the total profit from this investment is the same as the sum of the return on the investment at the current time + the sum of returns at the previous time, that is

$$f_n(S_n, y_n) = Max[\sum y_n n + (S_n - \sum y_n)] \tag{5}$$

Let

$$R_n(y_n) = \sum y_n n\,; \ \ \mathrm{f}_{\text{n-1}}(S - y_n) = (S_n - \sum y_n)$$

Where $R_n(y_n)$ is the sum of maximum returns from investing the amount y on the variable n at the current time, and $\mathrm{f}_{\text{n-1}}(S_n - y_n)$ is the sum of maximum returns from investing the amount (S - y_n) on the variable (n – 1) at the previous time. Substituting into equation (5), we have

$$f_n(S_n, y_n) = R_n(y_n) + f_{n-1}^*(S_n - y_n) \tag{6}$$

Equation (6) can be written as equation (7) without altering the model; hence,

$$f_n(S_n, y_n) = R_n(y_n) + f_{n+1}^*(S_n - y_n) \tag{7}$$

Equation (7) is the dynamic programming model for this work. From the model, we observed that the final state is fixed, and the initial state is free.

3.3. Optimal allocation policy

To develop the optimal allocation policy, we define the following parameters; let
S = state variables
S* = optimal state variables
y = decision variables
y* = optimal decision variables
k = specific value assumed by state and decision variables.
Hence, we have;

$$S_i = S_i^* \ ; \ \mathrm{y_i}^* = k_i \ ; \ \mathrm{i} \ = \ 1, 2, \ldots, \mathrm{n},$$

where S_i^* is the optimal value of the state variable at the last stage i; i = 1, 2, . . . , n, but since we have a recursive relationship where the solution starts from the last stage to the first stage, we start from i = n, the last stage and proceed back to the first stage. Again, y_i^* is the optimal value of the decision variable corresponding to S_i^*, then we have

$$S_i = \mathrm{k_i} \ ; \ \mathrm{y_i}^* = k_i$$

$$S_{i-1} = (S_i^{*} - k_i)^{*} = k_{i-1}\ ;\ y_{i-1}^{*} = k_{i-1}$$
$$S_{i-2} = (S_i^{*} - k_{i-1})^{*} = k_{i-2}\ ;\ y_{i-2}^{*} = k_{i-2}$$
$$\vdots \ = \ \vdots \ = \ \vdots \ \ \vdots = \vdots$$
$$S_{i-i+1} = (S_i^{*} - k_{i-i+1})^{*} = k_{i-i+1}\ ;\ (y_{i-i+1})^{*} = k_{i-i+1} \tag{8}$$

Allocation of the containers in this order (k_i, k_{i-1}, k_{i-2}, . . . , k_{i-i+1}) will yield an optimal allocation of the 20ft & 40ft loaded containers to each of the seaports where they are most needed to maximize profit. This optimal allocation policy helps us to determine where more products, in this case, loaded containers, should be allocated.

3.4. Schedule of number of units of 20ft & 40ft shipping containers to be delivered

The FF companies were given a target to stuff and deliver 6,200 units of 20ft containers within the first six months (January – June) of 2021 and 7,500 units of 40ft stuffed containers in the second six months (July – December) of 2021. The weights assigned to each company per month are presented in Table 1 below.

Table 1: Specified Weight of 20ft & 40ft Containers to be stuffed and delivered per month

Months	1	2	3	4	5	6	
Company			Weights K_i (%)				% Total weight
1	20	15	12	25	18	10	100
2	14	20	17	20	10	19	100
3	18	16	19	15	17	15	100

Source: Global Shipping Inc., 2021

4. Data generation, presentation and analysis

4.1. Data generation/ Presentation

From Table 1, we derived the company's stuffing and delivery ratios which helped us determine the number of containers to be stuffed and delivered per company.

Table 2: Company stuffing and delivery ratios for each month

Month	1	2	3	N_1	N_2
		Company ratios			
$\sum_{i=1}^{n} \frac{1}{k_i}$	0.3956	0.3829	0.3628	6200	7500

Using the company's delivery ratio in Table 1 and the specified monthly delivery weight of table 2 and applying equation (1) to them, we obtain the number of 20ft & 40ft containers for each company for the two separate stuffing and delivery periods as presented in Table 3.

Table 3: Stuffing and Delivery for 12 months by three Companies (1, 2, 3)

Month	1	2	3	4	5	6	Total	Total
			Company Deliveries				N_1	N_2
20ft (N_1)							6,200	
(1)	784	1,045	1,306	627	871	1,567		
(2)	1,156	810	953	810	1,619	852		
(3)	949	1,068	899	1,139	1,005	1,139		
40ft (N_2)								7,500
(4)	948	1,264	1,580	758	1,054	1,896		
(5)	1,399	979	1,152	979	1,959	1,032		
(6)	1,149	1,292	1,088	1,378	1,216	1,377		

The companies cover six (6) major seaports in Nigeria, namely: (1) Port Harcourt seaport, located in River State, (2) Tincan Island seaport, Lagos State, (3) Onne seaport, Rivers State, (4) Apapa seaport, Lagos State, (5) Warri seaport, Delta State and (6) Calabar seaport, Cross River State and want to determine how these stuffed containers could be distributed to maximize profit. Note that each of the seaports is represented by their respective serial numbers.

4.2. Analysis

Table 4: 20ft & 40ft Stuffed Shipping Containers available for distribution

Sea Ports	1	2	3	4	5	6
1	784	1,045	1,306	627	871	1,567
2	1,156	810	953	810	1,619	852
3	949	1,068	899	1,139	1,005	1,139
4	948	1,264	1,580	758	1,054	1,896
5	1,399	979	1,152	979	1,959	1,032
6	1,149	1,292	1,088	1378	1,216	1,377

From equation (7), let S_n be the number of seaports available for allocation to 20ft & 40ft stuffed containers (n = 1, 2, . . . , 6). The resulting dynamic programming calculations are given below, beginning from the last stage n = 6 and proceeding back to the first stage, n = 1.

Table 5: 20ft & 40ft Containers available for distribution

Distribution Centers	1	2	3	4	5	6
0	0	0	0	0	0	0
1	784	1,045	1,306	627	871	1,567
2	1,156	810	953	810	1,619	852
3	949	1,068	899	1,139	1,005	1,139
4	948	1,264	1,580	758	1,054	1,896
5	1,399	979	1,152	979	1,959	1,032
6	1,149	1,292	1,088	1,378	1,216	1,377

Table 6: for n = 6

S6	f*6	y*6
0	0	0
1	1,567	1
2	852	2
3	1,139	3
4	1,896	4
5	1,032	5
6	1,377	6

Applying equation (7), we obtain Tables 7-11.

$$f_5(S_5, y_5) = R_5(y_5) + f_6^*(S_5 - y_5)$$

Table 7: for n = 5

$S_5 \backslash y_5$	0	1	2	3	4	5	6	f^*_5	y^*_5
0	0							0	0
1	1,567	871						1,567	0
2	852	2,438	1,619					2,438	1
3	1,139	1,723	3,186	1,005				3,186	2
4	1,896	2,010	2,471	2,572	1,054			2,572	3
5	1,032	2,767	3,515	1,857	2,621	1,959		3,515	2
6	1,377	1,903	3,515	2,144	1,906	3,526	1,216	3,526	5

$$f_4(S_4, y_4) = R_4(y_4) + f_5^*(S_4 - y_4)$$

Table 8: for n = 4

$S_4\backslash y_4$	0	1	2	3	4	5	6	f^*_4	y^*_4
0	0							0	0
1	1567	627						1567	0
2	2438	2194	810					2438	0
3	3186	3065	3248	1139				3248	2
4	2572	3813	3248	2706	758			3813	1
5	3515	3199	3996	4325	2325	979		4325	3
6	3526	4142	3382	4325	3196	2546	1378	4325	3

$$f_3(S_3, y_3) = R_3(y_3) + f_4^*(S_3 - y_3)$$

Table 9: for n =3

$S_3\backslash y_3$	0	1	2	3	4	5	6	f^*_3	y^*_3
0	0							0	0
1	1567	1306						1567	0
2	2438	2873	953					2873	1
3	3248	3744	2520	899				3744	1
4	3813	4554	3391	2466	1580			4554	1
5	4325	5119	4147	3337	3147	1152		5119	1
6	4325	5631	4766	4147	4018	2719	1088	5631	1

$$f_2(S_2, y_2) = R_2(y_{21}) + f_3^*(S_2 - y_2)$$

Table 10: for n =2

$S_2\backslash y_2$	0	1	2	3	4	5	6	f^*_2	y^*_2
0	0							0	0
1	1567	1045						1567	0
2	2873	2612	810					2873	0
3	3744	3918	2377	1068				3918	1
4	4554	4789	3683	2635	1264			4789	1
5	5119	5599	4554	3941	2831	979		5599	1
6	5631	6164	5364	4812	4137	2546	1292	6164	1

$$f_1(S_1, y_1) = R_1(y_1) + f_2^*(S_1 - y_1)$$

Table 11: for n=1

S1\y1	0	1	2	3	4	5	6	f^*_1	y^*_1
0	0							0	0
1	1567	784						1567	0
2	2873	2351	1156					2873	0
3	3918	3657	2723	949				3918	0
4	4789	4702	4029	2516	948			4789	0
5	5599	5573	5074	3822	2515	1399		5599	0
6	6164	6383	5945	4867	3821	2966	1149	6383	1

4.3. Optimal allocation policy

Applying equation (8) for the optimal solution, we proceed as follows:

$$S_1 = 6, \quad y^*{}_1 = 1$$
$$S_2 = 6 - 1 = 5, \quad y^*{}_2 = 1$$
$$S_3 = 5 - 1 = 4, \quad y^*{}_3 = 1$$
$$S_4 = 4 - 1 = 3, \quad y^*{}_4 = 2$$

$$S_5 = 3 - 2 = 1, \quad y^*_5 = 0$$
$$S_6 = 1 - 0 = 1, \quad y^*_6 = 1$$

Thus (1, 1, 1, 2, 0, 1) distribution of the available stuffed containers to the seaports will generate 6,383 containerized cargo throughputs, thereby improving port terminal productivity.

The policy (1, 1, 1, 2, 0, 1) will result in the distribution of the 20ft, and 40ft stuffed containers as shown in Table 12 below.

Table 12: Optimal allocation policy

	Seaports					
Quantity	Port-Harcourt	Tincan	Onne	Apapa	Warri	Calabar
Allocation of stuffed Containers	1,064	1,064	1,064	2,128	0	1,064

5. Results and conclusions

In this paper, we developed two models: the dynamic programming model presented in equation (7) and the optimal allocation policy presented in section 3.3. We implemented these models under data presentation/generation and analysis in section 4. The data analysis gave us an intelligent formula for efficiently allocating the 20ft & 40ft laden containers. We observed that: Port Harcourt seaport (1), Tincan Island seaport (2), Onne seaport (3) and Calabar seaport (6) will be allocated 1,064 units of 20ft & 40ft laden containers each. Apapa seaport (4) will be allocated 2,128 units of 20ft & 40ft laden containers, and no allocation should be made to Warri seaport (5). These results were arrived at through the implementation of the optimal allocation policy. The zero allocation to Warri seaport implies that port user patronage in that port could be very low; hence, it may not be economical to route loaded containers to such ports where ship visits are also expected to be low. We also observed that the Apapa seaport was allocated twice the number of laden container units compared to other seaports. This may be attributed to the port's hinterland market size, which generates more cargo and hence attracts more ship visits to Apapa port. Therefore, the freight forwarding companies will be guaranteed -cargo-booking spaces in the visiting vessels and make more profit by allocating more containers. The outbound shipping container allocation model developed in this paper forms a veritable decision support tool for container network operators involving shipping lines, Freight Forwarders, port terminal operators etc. Port authorities and policy makers could apply the model in formulating strategies for improving efficiency of transport network operations and hence minimize generalized cost of transport associated with inland container freight stations-to-seaport terminal operations.

Authors' contributions

All authors contributed to the work's conception, design, and writing.

Funding

The study did not receive any financial support

Declaration statements

No potential conflict of interest is reported by the authors.

Citation information

Amuji, H. O., Onwuegbuchunam, D. E., Aponjolosun, M. O., Okeke, K. O., Mbachu, J. C., & Ojutalayo, J. F. (2022). The dynamic programming model for optimal allocation of laden containers to Nigerian seaports. *Journal of Sustainable Development of Transport and Logistics, 7*(2), 69-79. doi:10.14254/jsdtl.2022.7-2.5.

References

Adelson, R. M., Norman, J. M., & Laporte, G. (1976). A dynamic programming formulation with diverse applications. *Operational Research Quarterly*, 119-121.

Amuji, H. O., Ugwuanyim, G. U., Ogbonna, C. J., Iwu, H. C., & Okechukwu, B. N. (2017). The usefulness of dynamic programming in course allocation in the Nigerian Universities. *Open Journal of Optimization*, *6*(4), 176-186. https://doi.org/10.4236/ ojop.2017.64012

Augustine, O. E., & Barry, R.M. (1974). Non-Serial Dynamic Programming: A Survey. *Palgrave Macmillan Journals,* 25, 253-265.

Chen, K., Lu, Q., Xin, X., Yang, Z., Zhu, L., & Xu, Q. (2022). Optimization of empty container allocation for inland freight stations considering stochastic demand. *Ocean & Coastal Management, 230*(1), 1-10. https://doi.org/10.1016/j.ocecoaman.2022.106366

Cobo, P.T (2016) Optimization of yard operations in container terminals from an energy efficiency approach. Unpublished PhD Thesis. Industrial PhD Pilot program in Civil Engineering, Universidad Politécnica de Cataluña – Barcelona Tech.

Armas, L. D., Valdes, D., Morell, C., & Bello, R. (2019). Solutions to storage spaces allocation problem for import containers by exact and heuristic methods. *Computación y Sistemas*, *23*(1), 197-211. https://doi.org/10.13053/CyS-23-1-2916

Dhahri, M., Mezghani, M., & Rekik, I. (2020). A Weighted Goal Programming model for Storage Space Allocation problem in a container terminal. *Journal of Sustainable Development of Transport and Logistics, 5*(2), 6-21. https://doi.org/10.14254/jsdtl.2020.5-2.1

Facchini, F., Boenzi, F., Digiesi, S., & Mummolo, G. (2018). A model-based Decision Support System for multiple container terminals hub management. *Production*, *28*, e20170074. https://doi.org/10.1590/0103- 6513.20170074.

Guo, W., Atasoy, B., Beelaerts van Blokland, W., & Negenborn, R. R. (2020). Dynamic and stochastic shipment matching problem in multimodal transportation. *Transportation Research Record*, *2674*(2), 262-273.

Howard, R. A. (1966). Dynamic programming. *Management Science*, *12*(5), 317-348.

Jacquin, S., Jourdan, L., & Talbi, E. G. (2016). A multi-objective dynamic programming-based metaheuristic to solve a bi-objective unit commitment problem using a multi-objective decoder. *International Journal of Metaheuristics*, *5*(1), 3-30.

Moores, B. (1986). Dynamic programming in transformer design. *Journal of the Operational Research Society*, *37*, 967-969.

Onwuegbuchunam, D. E. (2018). Assessing port governance, devolution and terminal performance in Nigeria. *Logistics*, *2*(1), 6. https://doi.org/10.3390/logistics2010006

Peter, S. (1989). Dynamic Programming in Action. *Palgrave Macmillan Journals*, *40*, 779-787.

Wong, P. J. (1970). A new decomposition procedure for dynamic programming. *Operations Research*, *18*(1), 119-131.

Xu, Y., Wang, M., Lai, K. K., & Ram, B. (2022). A stochastic model for shipping container terminal storage management. *Journal of Marine Science and Engineering*, *10*(10), 1429. https://doi.org/10.3390/jmse10101429

Appendix

$$f_5(S_5, y_5) = R_5(y_5) + f_6^*(S_5 - y_5)$$

$$f_5(0,0) = R_5(0) + f_6^*(0) = 0$$

$$f_5(1,0) = R_5(0) + f_6^*(1) = 1567$$

$$f_5(1,1) = R_5(1) + f_6^*(0) = 871$$

$$f_5(3,2) = R_5(2) + f_6^*(1) = 3186$$

$$f_5(6,4) = R_5(4) + f_6^*(2) = 1906 \text{, etc.}$$

$$f_4(S_4, y_4) = R_4(y_4) + f_5^*(S_4 - y_4)$$

$$f_4(1,0) = R_4(0) + f_5^*(1) = 1567$$

$$f_4(3,1) = R_4(1) + f_5^*(2) = 3186$$

$$f_4(5,3) = R_4(3) + f_5^*(2) = 4325 \text{, etc.}$$

$$f_1(S_1, y_1) = R_1(y_1) + f_2^*(S_1 - y_1)$$

$$f_1(4,3) = R_1(3) + f_2^*(1) = 4702$$

$$f_1(5,5) = R_1(5) + f_2^*(0) = 1399$$

$$f_1(6,5) = R_1(5) + f_2^*(1) = 2966$$