

# Decision Support System for Discrete Robust Berth Allocation

Reda Adel Samkari

Information System Department, Faculty of Computing and Information Technology, Jeddah, Kingdom of Saudi Arabia

**Abstract:** This paper discusses algorithm to reduce total service time for set of vessels arriving at port. The objective is to reduce the total stay or delay times of vessels at a port i.e. the waiting, loading and discharging of containers should be done as quickly as possible, in order to save on terminal costs. Mentioned algorithm will also provide the coordinate along the quay at which vessel can be berthed, start of handling time after arrival, waiting time for each vessel. This research provides a decision-making system of berth ships in all available berths on the port based on the required handling time of the ship after a meeting between several departments such as operations management, maritime transport and representative companies discharge. All departments cooperate with each other to work out the schedule for the allocation of berth suitable for all vessel requirements.

**Keywords:** Berth Allocation Problem, minimum service time, waiting time.

## I. INTRODUCTION

Berth allocation problem deals with assigning of vessels to the berth in such a way that minimum service time will be required to handle all vessels. Basically, this type of problem is divided in three types:

1. Discrete berth allocation:

In discrete berth allocation model vessel can be berthed at finite section of pre-determined lengths along the quay.

2. Continuous berth allocation:

In case of continuous berth allocation, vessel can be berthed anywhere along the quay. This case is generally used as it gives more flexibility in assigning length along the quay.

3. Hybrid berth allocation:

In this case, Lengths of sections are pre-determined but more than one vessel can be berthed in one section if sufficient length is available

Following figure will give idea about these three types:

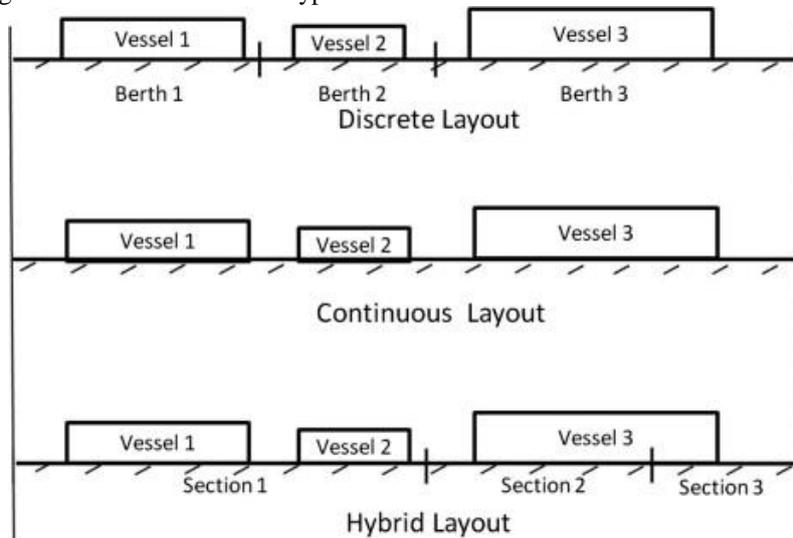


Figure 1: Different types of berthing

As shown in figure, for discrete berth allocation vessel are berthed in pre-determined section lengths. In continuous berth allocation there are no sections used. Vessels can be berthed at any length. While in hybrid layout vessel can occupy more than one section or one section can handle more than one vessel.

This research paper deals with algorithm used for continuous berth allocation model. Allotted section for the vessel will be given as output in form of coordinate along the quay. Handling time for vessel can be calculated directly by

contacting port. Handling time for each vessel depends on efficiency on type of material it is carrying, amount of material, sources available at port, etc. If for arrived vessel, berth is not available it will have to wait in anchorage till sufficient length is available along the quay. System is modelled using MATLAB.

Assumptions:

1. Vessel will arrive on scheduled arrival time. Any change in arrival time will cause change in output. Though we can consider different probabilities in delay of vessel, it will increase complexity of program. Also, it will affect accuracy of output i.e. minimum service time and all other outputs
2. Facilities required to empty the vessel are not fixed on quay. They can move along the quay as per requirement. Generally, equipment such as load shovel, wheel loader, mobile cranes are used to unload vessel. If quay contain different facilities such as conveyor, pipeline, etc. it will be treated as completely another case for that specified length.
3. Any time required for documentation and entrance procedure is considered in required handling time for vessel
4. Any clearance length required for vessel is compensated in length of vessel.

## II. LITERATURE REVIEW

In the literature, Berth Allocation Problem has been studied since the 1990s. Brown et al. used an integer programming model for assigning one possible berthing position to a vessel on various practical constraints. They considered a berth as a collection of discrete berthing positions and berth shifting of moored vessels was allowed. Port authorities always attempt to provide cost optimization and service enhancements. A seaport successes with short berthing duration for vessels and low prices for loading and discharging. Furthermore, minimizing the berthing duration for vessels becomes an overall objective with respect to terminal operations [5]. The terminal operations are complex. Generally, the terminal operation system consists of three subsystems: vessel planning, storage and stacking, and transportation. Based on the monthly planned or on-line schedule of represents its overall-length and the horizontal side represents its operation duration. Hence, Berth Allocation Problem becomes a constraint two-dimensional unrotation bin packing problem.[6]

This problem is commonly referred to as the berth allocation problem (BAP). The transshipment of containers between a vessel and the quay is generally performed by specialized cranes, which are mounted on rail tracks alongside the quay. The assignment of these quay cranes (QCs) to vessels and the determination of work plans for the cranes addresses two further problems, namely the quay crane assignment problem (QCAP) and the quay crane scheduling problem (QCSP).[7]

In the BAP we are given the berth layout of a CT together with a set of vessels that have to be served within the planning horizon For each vessel additional data like the vessel's length including clearance, its draft, the expected time of arrival, and the projected handling time can be given.[8] To show similarities and differences in the existing models for berth allocation, a classification scheme is developed in the following. Studies that concentrate on quay crane assignment either presuppose a particular type of BAP or integrate quay crane assignment decisions in the berth planning process. For this reason, QCAP approaches are captured by the classification scheme as well. Problems are classified according to four attributes. The spatial attribute concerns the berth layout and water depth restrictions. The temporal attribute describes the temporal constraints for the service process of vessels.[9]

The decisions of the berthing of ships to different berths are usually based on first in first out (FIFO) or priority rules set by port owner, which do not satisfy the customer demand. Some customers are more important to the ports based due to contracts. A priority is maintained for each of the customer expected to come at the port. As a port manager, it is required to allocate the berth to the customer based on this set priority. The priority can be set based on different criteria like the business contract, quantity of product or frequency of visit.. Due to this nature of setting priorities for the customers, a new objective of minimizing deviation from customer priority to actual berthing order is developed. Saurabh Pratap [10]

Most of the researchers focus their interest in strategic and tactical issues facing the container handling terminal port. A very few studies have been carried out for berth allocation as a ship scheduling problem for bulk material handling port. Saurabh Pratap [11]

Imai et. al. (2001) proposed a model for dynamic berth allocation and solved by heuristic based on Lagrangian relaxation. Imai et. al. (2010) introduced a model for berth allocation at Asian port and minimizes the ships staying time and minimizes dissatisfaction of the ships in terms of the berthing order. Brown et. al. (1994) introduced a berth allocation problem and to determine and optimize the berthing time to particular berth such that at a time only one vessel can occupy for a time horizon. Ting et. al. (2014) developed a model for discrete and dynamic berth allocation and minimizes the waiting and handling time of ships.[12]

Xu et.al. (2012) presented a model based on robust berth scheduling to encounter the uncertainties due to arrival time, handling time of ship and yard space allocation. Ursavas (2014) developed a decision support system to determine berthing and quay crane allocation decisions. Babu et. al. (2014) suggested a model to schedule ships and rake

scheduling to minimize the ship delay at the port terminal. In this paper, we foreground the port management issue that researchers usually consider, but in a holistic view; specifically, the scenario focus on ship sequencing and customer priority simultaneously.[13]

Akio Imai, Etsuko Nishimura and Stratos Papadimitriou Desired public berthing (1999) generally, ships are serviced in a port on a First-Come-First-Served (FCFS) basis, which However, does not necessarily minimize the total staying time of ships. If the ship arrival order is ignored, the total cumulative staying time for all ships can be reduced. This, however, may yield longer waiting time for some ships. In a previous research (Nishimura, 1998), a simple simulation for the public berthing system was carried out in order to examine the average time a ship spends for service, with the number of berths ranging from three to seven. [14]

The public berth system may yield inequitable waiting time for each ship. This may not make the port attractive for carriers even if the port charges are low. However, the issue of equity is less important than port charges as ships had inequitable waiting times at every private berth in Kobe (Nishimura, 1998).[15]

In Guan and Cheung [2004] weights in the objective reflect the relative importance of vessels. Two mathematical models (a relative position formulation and position assignment formulation) are given, and an LR approach is used for the latter. A tree search is used to guide the search. Due to large running times, vessels are clustered and the tree search algorithm is then run on each cluster. Note that Guan et al. [2002] also describe a similar, but simpler, solution method for the continuous SBAP. It should be mentioned that Cordeau et al. [2005] also describe a continuous version of their tabu search metaheuristic. A heuristic for the continuous problem is also given in Imai et al. [2005]. Katja Buhrkal [16] For the berth allocation problem, Zhou and Kang (2008) have used the genetic algorithm to search through the solution space and compared it with the greedy algorithm. The procedure of the algorithm is designed so as to reduce the search space based on the characteristics of the optimal solution. Constraints of the stochastic model are considered in the processes of individuals coding, initializing, crossover and mutation operations and also within the fitness function by employing sufficient punishment value. Zhou, P., & Kang [17]

Lai and Shih (1992) proposed a heuristic algorithm for berth allocation, which is motivated by more efficient terminal (actually berth) usage in the HIT terminal of Hong Kong. Their problem considers a First-Come-First-Served (FCFS) allocation strategy, which is not the case in our problem. Lai and Shih [18].

In the berth scheduling problem, vessels arrive over time at a port and the terminal operator assigns them to berths for unloading and loading of containers based on several factors and considerations (Theofanis et al, 2009). Three broad classification schemes of the BSP may be specified: (a) the discrete versus continuous berthing space, (b) the static versus dynamic vessel arrivals, and (c) the static versus dynamic vessel handling time.[19]

The discrete problem (Imai et al, 1997, 2001, 2003, 2007; Hansen et al, 2008) considers the quay as a finite set of berths. In the continuous problem (Kim and Moon, 2003; Park and Kim, 2003; Guan and Cheung, 2004; Imai et al, 2005; Moorthy and Teo, 2006) vessels can berth anywhere along the quay.[20].

**III. ALGORITHM AND METHODOLOGY FOR BERTH ALLOCATION PROBLEM**

- Input data for the code is given using excel file. Excel file is imported in MATAB

1	22	35	3	23	16	20	40	89	166
2	15	0	2	36	17	9	0	190	174
3	2	0	5	10	18	11	50	123	174.01
4	6	5	48	11	19	13	40	147	179
5	5	5	6	7	20	11	50	150	180
6	0	20	21	20	21	13	30	156	183
7	3	20	56	9	22	2	15	142	183.12
8	23	20	20	102.2	23	19	0	100	190.02
9	16	30	10	10	24	10	40	36	199.9
10	9	30	36	12	25	7	20	41	200
11	9	30	12	121.3	26	9	0	90	200
12	1	25	75	12	27	9	0	45	227
13	9	0	25	13	28	1	50	48	270
14	11	35	14	138.8	29	11	10	45	300
15	10	20	140	14	30	13	40	25	179

Figure 2: Input Data

Above data shows input used for program. Data for 30 vessels is used.

In above data

- Column 1: Index of vessel
- Column 2: Arrival Time(Hours)
- Column 3: Arrival Time(Minutes)
- Column 4: Handling Time(Hours)
- Column 5: Length of vessel(Meter)



2. Input data for each vessel is sorted in increasing order according to arrival time and required handling time and also cumulative sum is taken so as to check whether total length of all vessels exceeds length of quay. If in sorted data cumulative length of vessel is greater than length of quay then vessel has to wait till required length will be free to handle the vessel.
  3. Length along the quay is considered as continuous and coordinates will be allotted along this length. For SAQR Port this length is 1600m and for Al Jeddah Islamic port it is 11600m. To measure coordinated of length along the quay, it is divided in equal parts of 1 meter each. Specified length will be allotted for each vessel having cumulative length less than length of quay and handling of that vessel will start immediately after arrival.
  4. This handling time will be assigned for each section of 1 mtr. Please take note that matrix containing information on handling load for each section and coordinate of vessel updates at every iteration as it proceeds to next vessel in loop. (d1 be the name of updating matrix)
  5. When cumulative length of vessel exceeds than length of quay it will search for length with sections of minimum handling load. For this procedure it will use following algorithm.
    - a. First it will determine number of sections required for that length. Please note that, those sections must be consecutive.
    - b. Then program will scan the matrix d1 for handling loads on sections. It will scan matrix d1 in the window of elements equal to required number of sections. It will note maximum from each window.
    - c. The minimum from these maximums will be handling for vessel.
    - d. This handling time will be added in load on each section in window. Hence this matrix will get updated again.
    - e. Hence section length and start of handling time will be allotted for each vessel
  6. Coordinates of starting section and ending section of vessel will get stored in matrix named as 'lstart' and 'lend'. These will give coordinates of berth where the vessel is to be berthed. Matrix d1 is updating at every iteration in loop in terms of load on each section hence there will be no overlapping in terms of space and time. This eliminates the need of decision variable required to constrain overlapping.
  7. Graph of section load and vessel is plotted. In graph if there will be no overlapping then output will be considered as correct.
  8. Objective function that is total service time for vessels is calculated.
- Following figure will explain this algorithm:

Table 1: Step One

Load on section	10	10	10	10	10	15	15	15	15	15	15	25	25	25	25
Section number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

In above matrix, some sections are allotted for vessels having cumulative length less than length of quay. Load on section each section indicates the time required for section to complete handling vessel of vessel at that section.

Now, if next vessel is arrived before any of the section becomes free, program will search for. Let's assume next vessel will require 5 sections then it will search with window size of 5. It will first search for maximums from every 5 consecutive sections. Then it will choose minimum from from all those maximums i.e. minimax. Following table will illustrate this selection:

Table 2: Decision of minimax

Section Number	Section load	Maximums	Maximin
1	10	0	<b>10</b>
2	10	0	
3	10	10	
4	10	15	
5	10	15	
6	15	15	
7	15	15	
8	15	15	
9	15	15	
10	15	25	
11	15	25	
12	25	25	
13	25	25	
14	25	25	
15	25	25	

From table it is clear that new vessel will be berthed between section 1 and section 5. Those will start handling after 10 hours, hence new vessel will have to wait for minimum 10 hours. Now after allocation of this sections, new load will be added respective sections. (Let's assume new vessel will require 20 hours of handling). New load on each section are shown in next table:

Table 3: Step Two

Load on section	30	30	30	30	30	15	15	15	15	15	15	25	25	25	25
Section number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This process for berth allocation will proceed repeatedly for each vessel and for each iteration this table will be updated.

### IV. PROGRAM FOR OBJECTIVE FUNCTION

MATLAB is used to code above algorithm. Screenshots of program are shown below:

Input for program is imported by using “xlsread(‘ArrivalTime’)”. All the data regarding arrival time, length of vessel, required handling time is acquired from ‘ArrivalTime.xls’ Data is sorted according to arrival time and required handling time using “sortrows”

```

1   %Input data is given using .xls file named as "Arrivaltime"
2   v=xlsread('ArrivalTime');
3   %Arrival Timing are converted to decimal form
4   v1=[v(:,1) v(:,2)+v(:,3)/60 v(:,4) v(:,5)];
5   v2=v1(:,4);
6   Data_2=[v1(:,1) v1(:,2) v1(:,3) (v1(:,2)+v1(:,3)) v1(:,4)];
7   Data_3=sortrows(Data_2,4);
8   %Matrix named as data is created with all the input data in required form
9   %First Column: index number of vessels
10  %Second Column: Arrival Time
11  %Third Column: Handling Time for each vessel
12  %Fourth Column: Addition of Arrival time and handling time
13  %Fifth Column: Length of vessel
14  %Sixth Column: Cumulative length if each vessel
15  Data=[Data_3;0 0 0 0 0;0 0 0 0 0];
16  Data(:,6)=cumsum(Data(:,5));
17  %Length of the quay for which model is to be simulated
18  L=1600;
19  %Length of each section
20  %It is kept 1 meter to get accurate coordinate for berthing of vessel
21  r=1;
22  r2=2*r;
23  N=max(Data(:,1))+3;
24  %Defining arrival time and handling times for each vessel
25  for i=4:N;
26      A(i,1)=Data(i,2);
27      c(i,1)=Data(i,3);
28  end
29  %Defining matrix b with each section coordinate
30  b=[1:r:L]';
31  %Initially there will be no load on section hence column with zeros is
32  %added

```

Figure 3: Matrix ‘b’ Description

Matrix ‘b’ is used to store information regarding coordinates of each section. Initially section have zero handling hours because no sections are allotted for vessel yet.

Matrix ‘b1’ is used to allot sections for vessels having cumulative lengths less than length of quay and b1 updates matrix b with new handling times for those vessels.

```

33  b1=[b zeros(size(b,1),1)];
34  %Defining number of sections required by each vessel
35  for t=4:N
36      b2(t)=ceil(Data(t,6)/r);
37  end
38  %Allotting sections for each vessel having cumulative length less than
39  %length of quay
40  for t=1:N,
41      for t=5:N
42          for i=b2(t-1):b2(t)
43              b1(i,2)=Data(t,4);
44              if Data(t,6)>max(b1(:,1))
45                  b1(i,2)=0;
46              end
47          end
48      end
49  for t=4
50      for i=1:b2(4);
51          b1(i,2)=Data(t,4);
52      end
53  end
54  end
55  b1(1+L/r:end,:)=[];
56  %Above line ensures that only vessels having cumulative length less than
57  %length of quay will have sections

```

Figure 4: Matrix ‘b1’ Description

Now for vessels having cumulative lengths greater than length of quay program proceeds to find new handling time. All maximums for window length equal to length of vessel are evaluated and stored in matrix 'd' using "movmax". It may happen, more than one equal maximum occur. To avoid repeated calculations, repeated maximums are converted to high constant. Thus there are no repeated maximums. 'o' is variable used to evaluate whether number of required sections are odd or even.

```

59 - %Code below will calculate maximums from window of required section length
60 - %from matrix b1
61 - for q=4:(max(Data(:,1))+3) %For all the vessels in input data
62 - %d will calculate movmax with required window length
63 - d=[(movmax(b1(:,2),ceil(Data(q,5)/r)))]);
64 - %Program will have more tha 1 similar maximums. To avoid repetitions
65 - %in further process all repeated values will be converted to high
66 - %constatnt value
67 - y = zeros(size(d));
68 - [~,i] = unique(d,'first');
69 - y(i) = d(i);
70 - y(y==0)=5000000;
71 - d1=[b1 y];
72 - s1=size(d1);
73 - %o is variable for each vessel which calculate wheather required number
74 - %of vessel is odd or even
75 - o=rem((ceil(Data(q,5)/25)),2);
76 - end

```

Figure 5: stored in matrix 'd' using "movmax"

```

77 - for q=4:(max(Data(:,1))+3)
78 - %To update matrix at every iteration it is required to define this
79 - %matrix again in loop
80 - d=(movmax(d1(:,2),ceil(Data(q,5)/r)));
81 - y = zeros(size(d));
82 - [~,i] = unique(d,'first');
83 - y(i) = d(i);
84 - y(y==0)=5000000;
85 - d1=[b1 y];
86 - s1=size(d1);
87 - %The code shown below will choose minimums from all the maximums.
88 - if Data(q,6)>max(b1(:,1));
89 - m(q)=min(d1((ceil(Data(q,5)/r2):(s1(1,1)-ceil(Data(q,5)/r2)),3));
90 - %If for specific vessel arrival time is grater trhan handling time
91 - %then it will take arrival time as handling time of vessel
92 - if A(q)>m(q)
93 - m(q)=A(q);
94 - end
95 - %P will find index of minimax from matrix d1
96 - p=find(d1(:,3)==min(d1((ceil(Data(q,5)/r2):(s1(1,1)-ceil(Data(q,5)/r2)),3)));
97 - %If o is odd then to add new handling load in all section load in window
98 - %and to find coordinate of vessel it will follow loop shown below.
99 - if o==1;
100 - %If o is odd then window will be centered exactly and it will add
101 - %new handling time in all elements equally in section loads above
102 - %and below of centered value
103 - g=m(q)+c(q);
104 - b1((p-(round(Data(q,5)/r2))-1):(p+(round(Data(q,5)/r2))-1),2)=[g];
105 - d1((p-(round(Data(q,5)/r2))-1):(p+(round(Data(q,5)/r2))-1),2)=[g];
106 - lstart(q)=d1((p-(round(Data(q,5)/r2))-1),1);
107 - lend(q)=d1((p+(round(Data(q,5)/r2))-1),1);
108 - else

```

Figure 6: stored in matrix 'd' using "movmax"

Now minimum value from all maximums is selected and added to section loads from other sections loads in window as discussed earlier. If o is 1 then number required section is odd, then new handling time will be added in equal sections above and below of centred section or section with minimax. If o is zero then number required sections is even then new handling time is added in one more section above centred section. Thus, sections are allotted for each vessel. It is required to define matrix 'd' again as it updates after every new allocation. If new handling for vessel is less than arrival time, program will set handling time at arrival time.

```

109 - %If o is even it will add handling load in one more section above centered value
110 - g=m(q)+c(q);
111 - b1((p-(round(Data(q,5)/r2))-1):(p+(round(Data(q,5)/r2))),2)=[g];
112 - d1((p-(round(Data(q,5)/r2))-1):(p+(round(Data(q,5)/r2))),2)=[g];
113 - lstart(q)=d1((p-(round(Data(q,5)/r2))-1),1);
114 - lend(q)=d1((p+(round(Data(q,5)/r2))),1);
115 - end
116 - %lstart and lend will store section number or coordinate of section
117 - %at start and end of wndoe thereby giving coordinate at which
118 - %vessel should be berthed
119 - else
120 - m(q)=A(q);
121 - lstart(q)=Data(q,6);
122 - lend(q)=Data(q+1,6);
123 - end
124 - end

```

Figure 7: minimum value from all maximums

```

125 %Loop shown below will plot rectangle showing handling of vessel and length
126 %of vessel.
127 %If these ractangle do not intersect each other, solution is correct
128 for q=4:max(Data(:,1))+3
129     axis([0 L 0 inf])
130     grid minor
131     hold on
132     %Lower left coordinate of rectangle are
133     %x-coordinate=Starting section of each vessel
134     %y-coordinate=start of handling time of vessel
135     %Length along x-axis=length oif vessel
136     %height along y-axis=total handling time
137     rectangle('position',[lstart(q) m(q) lend(q)-lstart(q) c(q)], 'FaceColor',[0.13 0.22 0.67], 'EdgeColor','k')
138 end

```

Figure 8: minimum value from all maximums

Now, graph is plotted showing load on each section. Each vessel is represented in rectangular form. Lower left coordinate of rectangle is [starting section of vessel, start of handling time]. Height of rectangle along y-axis is required handling time for vessel and length along x-axis is length of vessel. If these rectangles are not intersecting each other then there is no overlapping in terms of space and time.

```

139 %Result matrix will give information of each vessel
140 %Description of each column
141 %First column=Index of vessel
142 %Second Column=Arival time for vessel
143 %Third Column=Start of handling time
144 %Fourth Column=Handling time required
145 %Fifth Column=Difference between start of handling time and arrival time
146 %which is waiting time
147 %Sixth Column=Starting coordinate along quay for vessel
148 %Seventh Column=Ending coordinate along quay.
149 Result=[Data(1:max(Data(:,1))+3,1) A m' c m'-A lstart' lend'];
150 Result=Result(4:end,:);
151 Result=sortrows(Result,1);
152 %Service time is objective function for our code
153 %It is calculated by adding waiting time and handling times for each vessel
154 Service_Time=sum(m)-sum(A)+sum(c);
155 %To get information of any vessel index number should be given as input and
156 %it will show required information
157 fprintf('Minimum service time for set of vessel is: %d',Service_Time);
158 h=input('\n Enter index of vessel: ');
159 fprintf('Handling will start on %d and it will be berthed at %f',Result(h,3),Result(h,6))
160 %This code eliminated requirement of decision variable and gives the output
161 %in minimum time

```

Figure 9: Result is obtained in matrix named as “result”

Result is obtained in matrix named as “result”. It gives start of handling times for vessels, starting section for each vessel where it should be berthed, waiting time for each vessel. Also, this information can be retrieved by entering index number of vessel for which information is needed. Objective function that is minimum service time for set vessels is calculated by adding waiting time for all vessels and handling time for all vessels.

V. PROGRAM OUTPUT

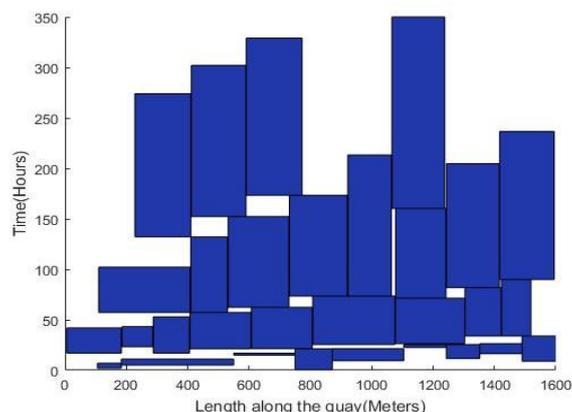


Figure 10: Graphical representation shown is output for program

Graphical representation shown is output for program. It shows all vessels in form of rectangle. As we can see there is no overlapping any rectangle which indicates that, at a time only one vessel can be arrived at section.



```

C:\Program Files\MATLAB\R2016a\bin
Command Window
>> test_5
Minimum service time for set of vessel is: 3.088583e+03
Enter index of vessel: 15
fx Handling will start on 7.358333e+01 and it will be berthed at 922.000000>>
    
```

Figure 11: output for vessel having index number 15

Above screenshot shows output for vessel having index number 15. Also, it shows total service time for given set of vessels.

1	1	22.5833	22.5833	3	0	1.1044e+03	1.2432e+03
2	2	15	15	2	0	550	750
3	3	2	2	5	0	106	184
4	4	6.0833	34	48	27.9167	1304	1422
5	5	5.0833	5.0833	6	0	184	550
6	6	0.3333	0.3333	21	0	750	871.3900
7	7	3.3333	34	56	30.6667	1423	1519
8	8	23.3333	23.3333	20	0	186	286
9	9	16.5000	16.5000	10	0	1.3522e+03	1.4902e+03
10	10	9.5000	17	36	7.5000	288	406
11	11	9.5000	9.5000	12	0	871.3900	1.1044e+03
12	12	1.4167	57.3333	75	55.9167	410	530
13	13	9	9	25	0	1.4902e+03	1.6692e+03
14	14	11.5833	11.5833	14	0	1.2432e+03	1.3522e+03
15	15	10.3333	73.5833	140	63.2500	922	1064

Figure 12: shows "Result" matrix with all the information about vessels

16	16	20.6667	71.5000	89	50.8333	1078	1242
17	17	9	160.5000	190	151.5000	1066	1238
18	18	11.8333	82	123	70.1667	1244	1416
19	19	13.6667	90	147	76.3333	1417	1595
20	20	11.8333	152.5000	150	140.6667	412	590
21	21	13.5000	173.5833	156	160.0833	591	773
22	22	2.2500	132.3333	142	130.0833	228	410
23	23	19	73.5833	100	54.5833	732	920
24	24	10.6667	21.3333	36	10.6667	408	606
25	25	7.3333	21.5000	41	14.1667	608	806
26	26	9	62.5000	90	53.5000	532	730
27	27	9	26.5000	45	17.5000	1077	1303
28	28	1.8333	25.5833	48	23.7500	808	1076
29	29	11.1667	57.3333	45	46.1667	110	408
30	30	13.6667	17	25	3.3333	6	184

Figure 13: shows "Result" matrix with all the information about vessels

Above screenshot shows "Result" matrix with all the information about vessels. Description for each column,  
 1<sup>st</sup> column: Index of vessel  
 2<sup>nd</sup> column: Arrival time for each vessel  
 3<sup>rd</sup> column: Start of handling time for each vessel  
 4<sup>th</sup> column: Required handling time each vessel  
 5<sup>th</sup> column: waiting time for each vessel  
 6<sup>th</sup> column: Starting coordinate along the quay at which vessel is to be berthed  
 7<sup>th</sup> column: Ending coordinate along the quay for each vessel

**VII. CONCLUSION**

In this research we have concluded many different results in the use of berth allocation problem. We hope that these results will contribute to increasing the use of modern technology and scientific methods in berth allocation problem. There are many methods that can be used by owners of dockyard to help them in berth allocation problem. Finally, the algorithm that is used to achieve the greatest goals of achieving the minimum service time, according to resources available such as mobile crane, wheel loader, load shovel etc. We believe that the waiting times and delays at the berth and yard are significant, which calls for better coordination between berthing activities and yard management at the port. We presented a model that can be solved using general purpose solver.

**REFERENCES**

- [1] Akio Imai a, Etsuko Nishimura a, Stratos Papadimitriou. (1999, 10 4). The dynamic berth allocation problem for a container port. p. 17.
- [2] Christian Bierwirth , Frank Meisel. (2009, May 28). A survey of berth allocation and quay crane scheduling problems in container terminals.
- [3] Gerald G. Brown. Siriphong Lawphongpanich, and Katie Podolak Thurman. (1994, 1 15). Optimizing Ship Berthing. p. 16.
- [4] <http://www.ports.gov.sa/Arabic/Aboutus/Pages/Introduction.aspx>. (n.d.). Retrieved from ports: www.ports.gov
- [5] <http://www.ports.gov.sa/Arabic/Aboutus/Pages/Introduction.aspx>. (n.d.). Retrieved from ports: www.ports.gov
- [6] <http://www.ports.gov.sa/Arabic/Aboutus/Pages/Introduction.aspx>. (n.d.). Retrieved from ports: www.ports.gov
- [7] <http://www.ports.gov.sa/Arabic/Aboutus/Pages/Introduction.aspx>. (n.d.). Retrieved from ports: www.ports.gov
- [8] Imai, E. Nishimura, and S. Papadimitriou. (2003). Berth allocation with service. Transportation Research Part B, 37:437, p. 37.
- [9] Imai, E. Nishimura, and S. Papadimitriou. (2003). Berth allocation with service. Transportation Research Part B, 37:437, p. 37.
- [10] Imai, E. Nishimura, and S. Papadimitriou. (2003). Berth allocation with service. Transportation Research Part B, 37:437, p. 37.
- [11] Imai, K. Nagaiwa, and C. W. Tat. (1997). Journal of Advanced Transportation., 31.
- [12] Katja Buhkara, Sara Zuglianb, Stefan Ropkeb, Jesper Larsenc, Richard Lusbyc. (2009, 14). Models for the Discrete Berth Allocation Problem: A. p. 27.
- [13] Lai, KK and Shih. (1992). A study of container berth allocation. Berth Allocation Problem with Service Time and Delay Time Objectives, p. 26.
- [14] Maria Boile, Sotirios Theofanis. (2009). The berth allocation problem: Optimizing vessel arrival time. Optimizing vessel arrival time, p. 20.
- [15] Mihalis, Georgios. (2009). The berth allocation problem: Optimizing. berth allocation problem, p. 20.
- [16] Saurabh Pratap, Ashutosh Nayak, Naoufel Cheikhrouhou, Manoj Kumar Tiwari. (2015). Decision Support System for Discrete Robust Berth Allocation. ScienceDirect, p. 16.
- [17] Wang, F., & Lim, A. (2006, 07 20). A stochastic beam search for the berth allocation problem. A stochastic beam search for the berth allocation problem, p. www.elsevier.com/locate/dss.
- [18] Zhou, P., & Kang. (2008). Berth and Quay-crane allocation under stochastic environments in. Priority control of berth allocation problem in container, p. 20.