AN ABSTRACT OF THE THESIS OF

Young-Gyo Chung for the degree of Master of Science in Industiral Engineering presented on June 9, 1986

Title: An Animated Simulation Model for a Transtainer Based Container Handling Facility

Abstract Approved: Redacted for Privacy
Sabah U. Randhawa

The efficiency of a marine cargo terminal depends primarily upon a smooth and orderly flow of material during the container loading process. It has been observed that the transtainer operation is the bottleneck in the loading process due to frequent rehandles of containers and the speed of the transtainer. Past attempts at improving the port operation have concentrated on the computerization of the load planning function and the efficient warehousing of the outbound containers. These attempts have been successful to some extent, but have not been able to significantly eliminate the unproductive movements of the material handling equipment during the loading operation.

This study proposes a methodology of utilizing a buffer space as a method to increase the utilization of the material handling equipment, and reduce the total container loading time. A simulation model using a

graphics simulation system is developed to compare the proposed methodology with the current practice at the Port of Portland.

The results from 96 simulation runs show that the use of a buffer space significantly improves the flow of material during the loading operation resulting in an average of 4% reduction in the total loading time. The methodology proposed in this study can be evaluated and implemented for any transtainer-based container port operation, and is anticipated to make considerable contribution to the planning of future container port design.

An Animated Simulation Model for A Transtainer Based Container Handling Facility

by

Young-Gyo Chung

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed on June 9, 1986

Commencement June 1987

Approved:

Redacted for Privacy

Assistant Professor of Industrial Engineering
in charge of major

Redacted for Privacy

Head of Department of Industrial and General Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented June 9, 1986

Typed by Young-Gyo Chung

ACKNOWLEDGEMENTS

I wish to express my appreciation to all of the people who helped me, both directly and indirectly, during my study at Oregon State University.

My special thanks go to Dr. Edward D. McDowell, whose insightful suggestions and constant encouragement have always been a source of inspiration to me.

I am especially indebted to my major professor, Dr. Sabah U. Randhawa. Without his valuable comments and friendly support, I would have never completed this study. My heart-felt appreciation to him is beyond description.

Finally, I must thank my parents and wife for their devoted support and tolerance.

I dedicate this thesis to my son, Won-Yong, with a wish for a life full of happiness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF TABLES

AN ANIMATED SIMULATION MODEL FOR A TRANSTAINER BASED CONTAINER HANDLING FACILITY

I. INTRODUCTION

Background

The traditional method of handling cargo was to use the muscle power with simple tools. The rapid increase of intercontinental trade in the twentieth century created a need for more efficient cargo handling method. As a respond to that, the concept of consolidating cargo was introduced to the maritime industries about thirty years ago. The continuous growth of cargo volume has been a burden to the existing ports, which in turn necessitated the development of more efficient cargo handling equipments including the container vessels.

The rapid growth rate of containerized cargo transportation can be primarily attributed to the cost effectiveness resulting from the consolidation of cargo into sealed standard size unit, a container. The benefit from the consolidation of cargo became more apparent with the recent development of specialized cargo handling equipments, such as a transtainer, a fork-lift, a ship crane, and so on. Today, the cost of handling containerized cargo is less than fifty percent of that for conventional cargoes [McDowell et al, 1985], and containerized cargo has became a predominant method of transporting cargo in the modern oversea trade.

The need for more specialized and efficient container handling facilities involves a large amount of capital investment, and this in return has made the container port operation very highly capital intensive. Presently, the cost of a container port facilities is approaching $15,000,000 per berth and the standby cost of a modern container vessel easily exceeds $20,000 per day [Port of Portland Technical Seminar, 1979].

As a consequence, the nature of a modernized container port operation requires an efficient and effective use of the port facilities to achieve an acceptable economic rate of return as well as to provide for enhanced customer satisfaction.

Terminal 6 at the Port of Portland is one of the highly modernized container handling facilities on the west cost of The United States. It offers a 59 acre container storage yard, 6 mobile container yard cranes, 21 tractors, five ship cranes, 64 chassis, two 40-ton top-loaders and side-loaders [Marine, Port of Portland, 1986]. The total container volume processed through the Terminal 6 of the PoP (Port of Portland) has grown from about 47,500 TEUs (twenty foot equivalent units) during its first year of operation to more than 100,000 TEUs in 1984. The increase of the container volume rapidly increased the complexity of the Terminal 6's operation, creating the necessity for more elaborate operation planning.

The container loading activity consists of three stages: arrival of the outbound containers, load planning, and the execution of the load plan. The complexity of container port operation arises from the conflicts between the random arrival order of outbound containers and the bay plan from the ship companies, as well as the amount of information to be processed. The major concern of the port authority is to develop a load plan that best compromises these conflicts. However, since the port has no control over these two conflicting factors, the actual execution of the loading plan inevitably includes a certain amount of unproductive movements of material handling equipment.

Past attempts to improve the loading operation can be categorized into two methods (development of new material handling equipments is not included). One method to reduce the complexity of the container loading operation is to computerize the load planning procedure. Another is to develop a strategy for the efficient warehousing of outbound containers, or for improvement in the actual execution of the loading plan.

Purpose of the Study

It has been observed that the efficiency of a marine cargo terminal depends primarily upon a smooth and orderly flow of material to and from the dock area, and that a disfunctional material flow pattern results in congestion and ultimately in increased material handling costs because of increased move distance and the frequency of container rehandles. The decreased efficiency caused by disfunctional container handling results in longer ship turnaround times which further adds to the cost of operation making the service less cost-competitive. Also, past studies have identified the transtainer's operation as the bottleneck in the material flow during the loading operation.

Therefore, the primary purpose of this study is to develop and test strategies that can reduce the unproductive movements of material handling equipment, especially the transtainer, during the actual loading process. The proposed methodology will be evaluated based on the current practice of Terminal 6 at the PoP.

Approach

A certain number of rehandles and unnecessary material handling equipment movements are inevitably involved in a loading plan. Based on this fact, this research proposes the utilization of a buffer area as a method to reduce the unnecessary movements of material handling equipments during the actual loading operation. The simulation model developed in this research evaluates the use of buffer space under different system parameters. The system parameters varied in this study include the number of trucks and the number of chassis used in the buffer space available for storing containers temporarily. Further, the idea of utilizing buffer space is compared with the current practice at PoP.

To keep the complexity of the system to a manageable size, assumptions regarding material handling equipments and container port were made in the development of the simulation model. It was assumed that the material handling equipment would be of the transtainer yard type, that is, a combination of transtainer, trucks, and ship cranes would be used to transport containers from the dock to the ship. It was also assumed that all containers for a voyage have arrived and their yard positions are known. Most ports set a cutoff time for container arriving for a voyage; hence this assumption is not unrealistic.

Direct application of the proposed methodology on the Terminal 6 at PoP involves a high experimental cost as well as a risk of interrupting the port operations. This justifies the use of simulation as an experimental tool for this study. As a special feature of the study, it was decided to use a graphics simulation system in developing the simulation program. The advantage of using a graphics simulation system is that the visual presentation of the system behavior facilitates the analysis of material flow during the simulation, and is easy for the user to understand.

Another characteristic of the model developed here is its ability to conduct sensitivity analysis. By running the model under different scenarios, it may be possible to spot the bottlenecks in the operation for a given set of parameters. The results from the model are a valuable aid for the design of port operations, and to the management of a facility for resource allocation.

II. PROBLEM ANALYSIS

Container Port Operation

Container port operation can be viewed as consisting of two separate but interrelated components, yard side operation and dock side operation. These two components are shown in Figure 2.1.

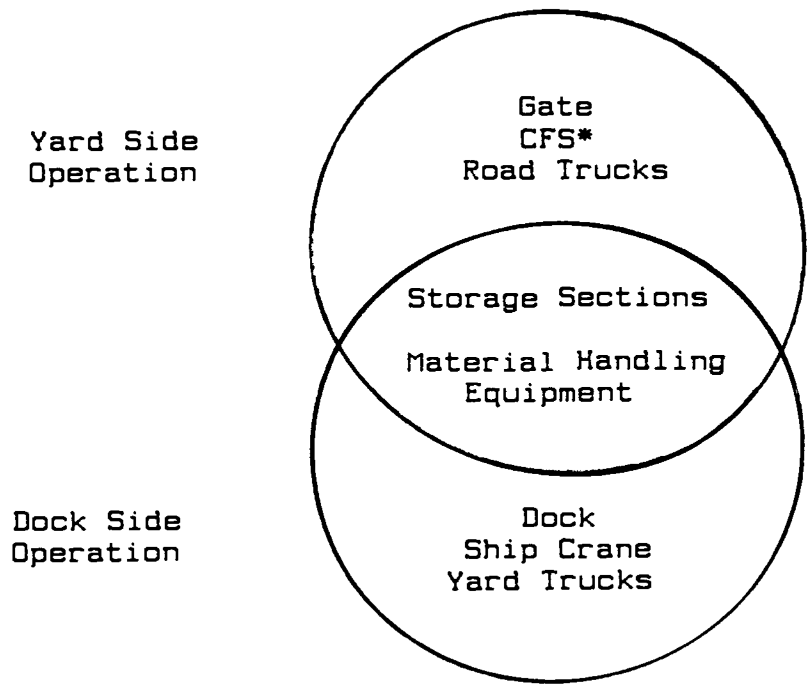


Figure 2.1 Yard Side and Dock Side Operations

---

*CFS = Container Freight Station where small amount of loads are collected and packed in one container.

The activities performed by these two operations are as follows :

* Yard Side Operation
  - Storing outbound containers
  - Distributing inbound containers

* Dock Side Operation
  - Loading outbound containers
  - Unloading inbound containers

Since the loading operation is the primary emphasis of this research, this chapter will concentrate on the port facilities and the operations involved in the loading operation.

Yard Side Operation Yard side operation associated with the loading operation begins with the arrival of outbound containers. Road trucks with outbound containers arrive to a check point (referred to as a gate). The outbound containers may be full or empty. The truck goes to an inspection station with a container load plan from the truck company. The load plan contains information on the steamship company, the container ship, and the voyage. After all information about the container is filed, the container is assigned a location in the yard that will offer the fastest loading of the container onto the ship. The truck then proceeds to this location in the yard, where the container is unloaded by either a fork-lift truck or a transtainer. If the container is empty, it is unloaded from the truck by a fork-lift, else a transtainer is used. The traffic flow of yard side operation is roughly circular from the check point through the storage sections and back to the check point as shown in Figure 2.2 (a).

Dock Side Operation   Dock side operation related with shipping begins with the arrival of a ship. The ship crane, yard trucks, and the transtainer start the loading operation based on the work sequence sheet. Participating facilities are equipped with radio communication devices, so that the movements of each facility is monitored and controlled to make the loading operation smooth and consistent with the loading plan. The traffic flow of dock side operation is also roughly circular starting from the ship crane to the storage sections and back to the ship crane as shown in Figure 2.2 (b). Based on the operating instructions, an empty yard truck travels to the location to pick up the container to be shipped. The container is loaded onto the truck using a transtainer. The yard truck then returns to the ship crane, where the container is loaded onto the ship.

Load Planning   Figure 2.3 describes the role of load planning in a shipping activity of container port operation. Load planning is an intermediate step where information from the truck company and the ship company is transformed into a load plan, and the load plan is then

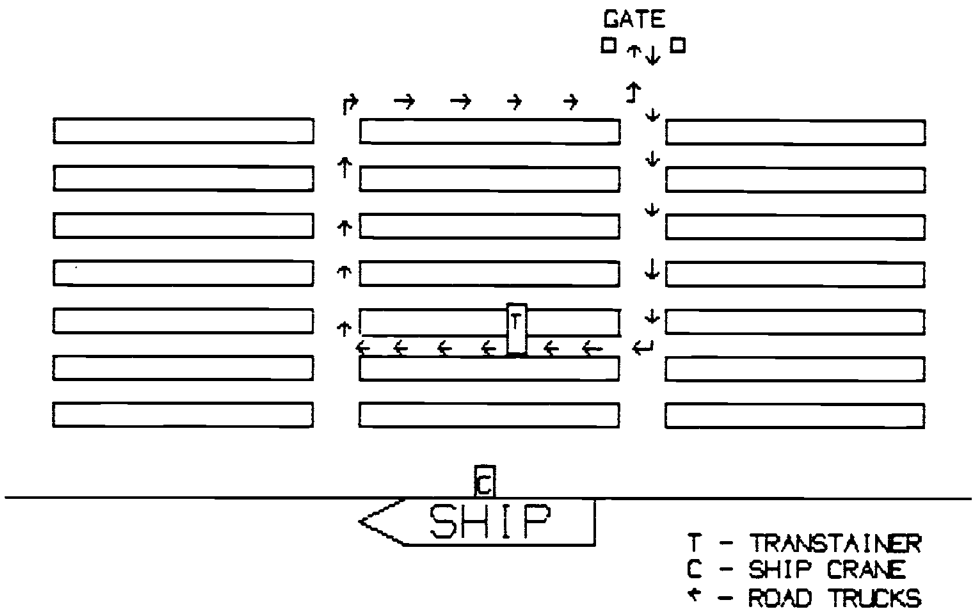


Figure 2.2(a) Traffic Flow -- Yard Side Operation

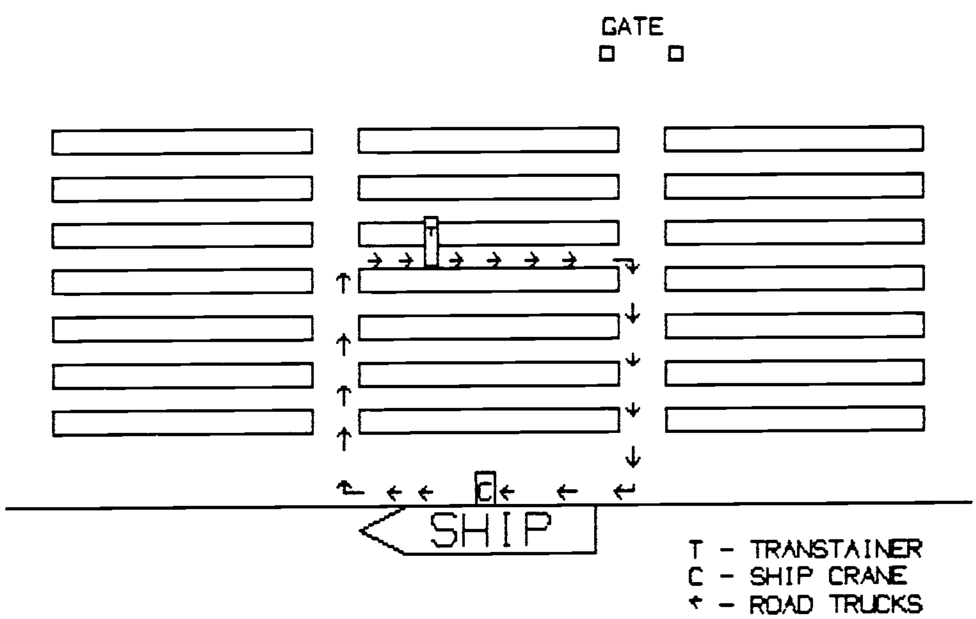


Figure 2.2(b) Traffic Flow -- Dock Side Operation

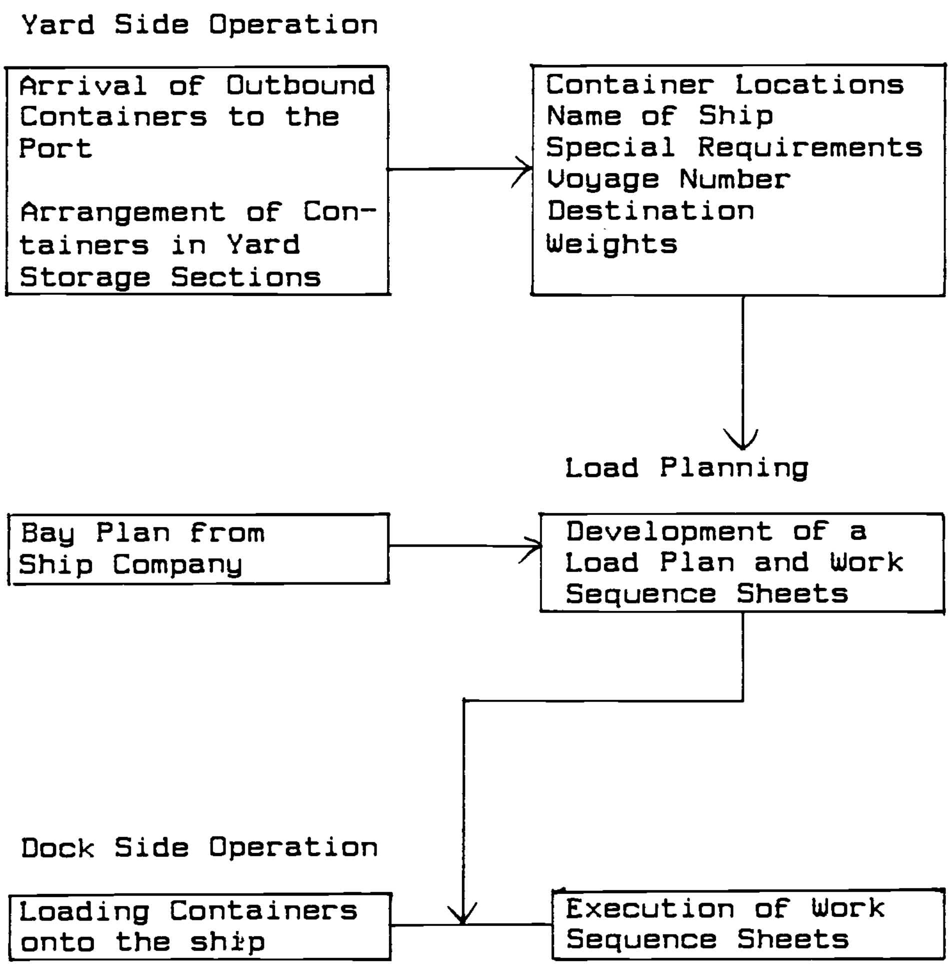


Figure 2.3 Role of Load Planning

converted into the sequential movement orders of the ship crane and the transtainer for their dock side operation. This movement order (referred to as a work sequence) consists of container location, identification number, and length of container as shown in the Figure 2.4.

The following factors are taken into consideration in developing a load plan :

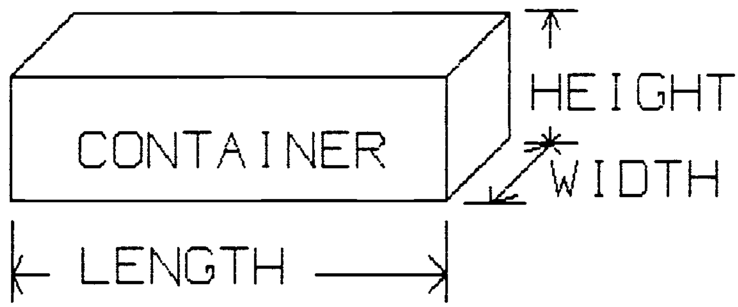
1. Under deck bays are built to accept only one length of container, either 20 ft or 40 ft.
2. It is desirable to have cargo for a single port consolidated in a single place within a bay. However, they should not be concentrated at either end of the ship to allow two or more transtainers to work at the same time and prevent crane interference.
3. Cargo for different ports are loaded based on the ship's route.
4. Effect of weight distribution on the ship's stability should be within an acceptable range.
5. Containers with special requirements should be placed in the designated areas.

<u>Containers</u> Containers can be divided into two categories depending on the purpose they are built for. These are standard containers and special containers. Both of these type of containers are currently built based on the specifications recommended by ISO (International

BAY	ACTION		20'	40'
20 O/D	Disch.	Uncover port & starboard		12 E
21 U/D	-"-		34	
24 O/D	Disch.	Uncover port & starboard		23 E
24 U/D	Disch			18 E
-"-	-"-			15
28 O/D	Disch.			8, 9 R/H
-"-	Disch.			1 R/H
-"-	Load	Mach Fwd		10 RFR
-"-	-"-			7
-"-	-"-			1 R/H
33 O/D	-"-			1 R/H
-"-	Disch.		2 R/H	
35 O/D	Disch.		4 R/H	
34 O/D	Load	Mach Aft		8 RFR
31 O/D	-"-	Mach Fwd	3 RFR	
-"-	-"-		2	
35 O/D	-"-		1	
33 O/D	-"-		2	
34 O/D	Load			6
24 U/D	Load	Cover Stb & Port		17
24 O/D	-"-	Mach Fwd		14 RFR
-"-	-"-			7
-"-	-"-			3 R/H
21 U/D	Load		32	
19 U/D	-"-		28	
-"-	-"-	Cover Stb & Port	6 R/H	
20 O/D	Load	Mach Aft		9 RFR
-"-	-"-			12

Figure 2.4 Example of Work Sequence Sheet

Standardization Organization). The dimensions of the most common containers are as shown in Figure 2.5.



Length	-	20 feet or 40 feet
Height	-	8 feet or 8.5 feet
Width	-	8 feet

Figure 2.5 Container Dimensions

Standard containers are the most common type of containers currently being used in the container transportation systems. These containers are designed to carry dry materials, and are sealed and waterproof so that they can be stored in an open storage yard.

Special containers include all containers that are built to meet special purposes, such as :

1. Refrigerated Containers - These containers are built to carry materials that should be kept at a certain temperature. These containers should be

be stored in places where suitable electrical outlets are available both in the storage yard and on the ship. Special considerations are given during storage planning for this reason.

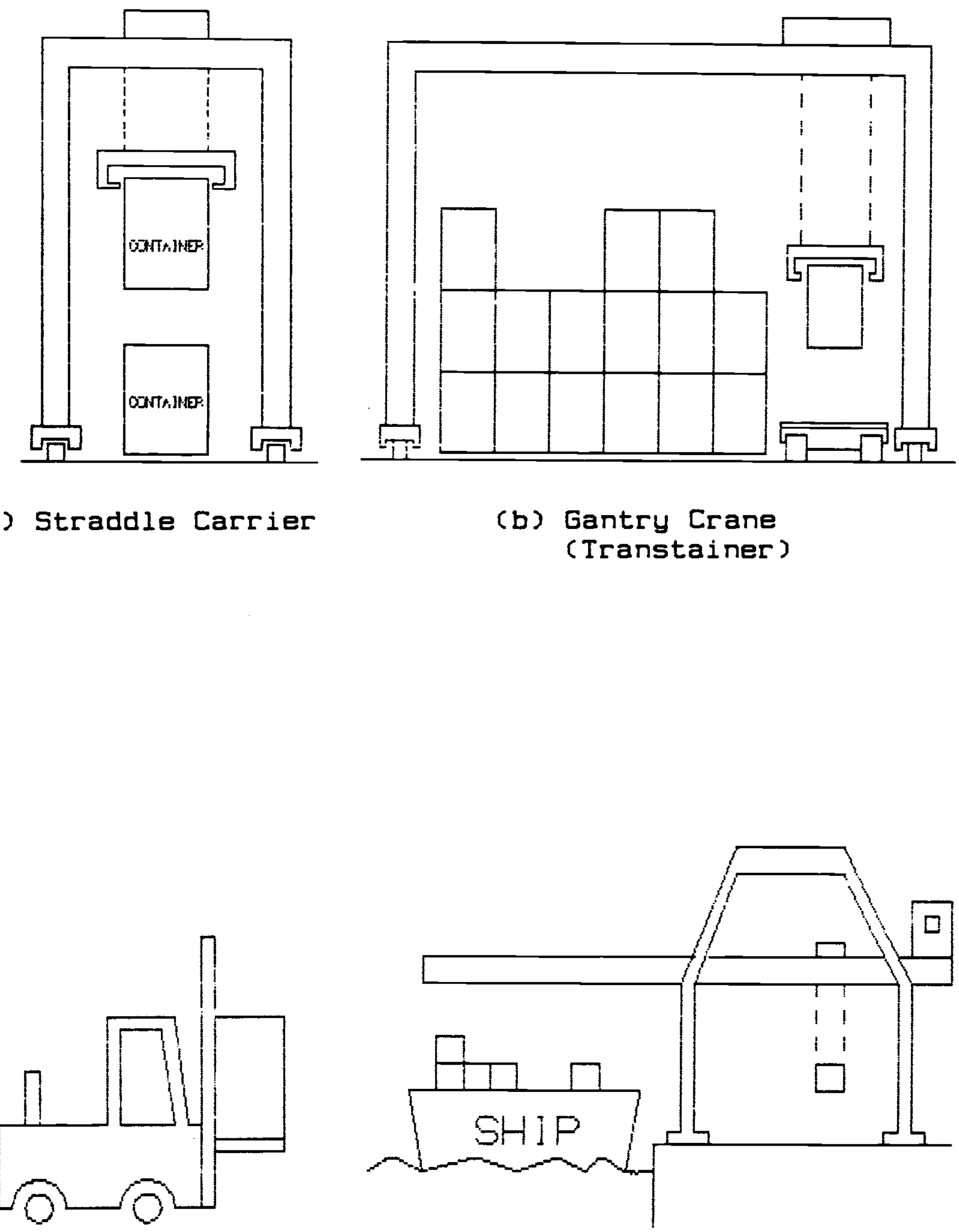
2. Open Top Containers - No other containers can be stacked on the top of these containers. These containers are placed on the bottom alone or on the top of other container in the storage yard. Containers of this type require special hooks to be loaded to the ship.

3. Half Size Containers - These containers have an open top with sides only half as high as normal container. They are used for very heavy loads.

Both standard and special containers have specially built corner fittings which enables the material handling equipment to pick them up from the top or the side.

Material Handling Equipment A variety of material handling equipment is used in container port operation (Figure 2.6). Each of these equipments is assigned to a specific operation based on their unique features and the operational situations. Following is a description of the material handling equipment currently being used at the Port of Portland :

1. Straddle Carrier - This equipment has wheels so that it can move over a container from one end of a container with the wheels on both sides as



(a) Straddle Carrier

(b) Gantry Crane
(Transtainer)

(c) Fork-lift

(d) Crane

Figure 2.6 Material Handling Equipment

shown in Figure 2.6 (a). Also, it can move over a truck chassis to unload the container on the chassis. It uses spreader bars to pick up a container. It can stack containers up to three high, which means the bottom container cannot be reached until the upper containers are removed. This equipment requires high maintenance cost.

2. Gantry Crane or Transtainer (Figure 2.6(b)) - This equipment has 50 ton capacity and rubber wheels. It runs on a special concrete paths due to its heavy weight. Wheels will turn 90 degree to change direction during section changes. Normally, transtainer is scheduled not to move between sections except when absolutely necessary. The beam spans over several stacks and the spreader bar moves down to reach a specific container. It can stack containers four high or more. It cannot reach the bottom container until the upper containers have been removed. Figure 2.7 shows a transtainer movement in a container section.

3. Modified Fork-lift (Figure 2.6 (c)) - This equipment is not as efficient as the other two introduced above.It is used to handle small number of containers. Its capital cost is relatively low in comparison with the others. This equipment cannot reach inside the row and

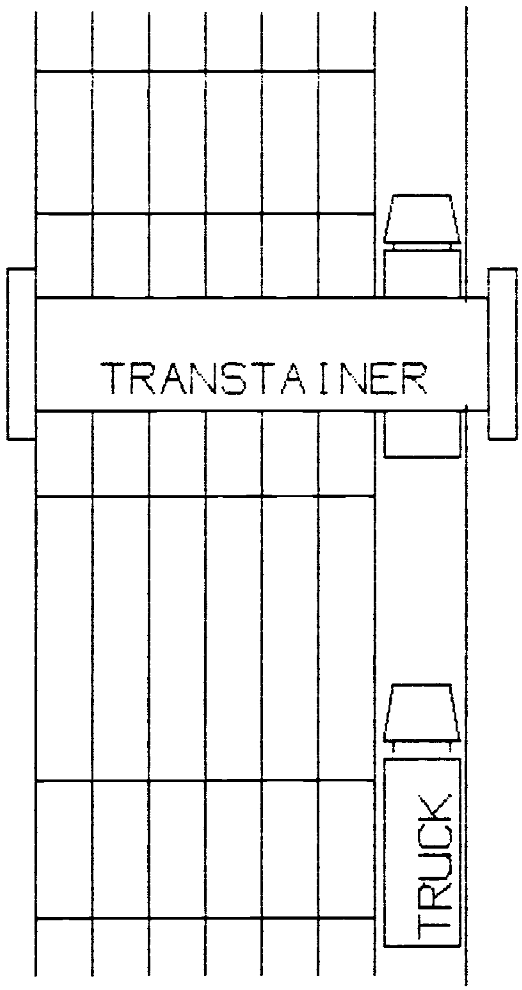


Figure 2.7 Transtainer in a Section

pick out the middle containers.

4. Ship Crane (Figure 2.6 (d)) - Ship crane is electric-powered and has 50 ton capacity. It can run on rails along the length of the dock. Ship crane should be kept at least 50 feet apart when more than one cranes are working at the same time in order to prevent crane interference.
5. Trucks and Chassis - These are used to carry containers from the storage locations to ship crane. They are specially designed for container port operations. Chassis have adjustments for two 20 ft containers, one 24 ft container, or one 40 ft container. The trucks have special connections for chassis ; they raise the chassis and travel with them without having to adjust the landing wheels. In general, 3 to 4 trucks are assigned to one transtainer-crane operation.

<u>Container Yard</u> The container yard, located next to the dock, stores the outbound containers while waiting for a ship, and the inbound containers until they are picked up by road trucks. Containers are stored in the yard in a block formation, called a section. Figure 2.8 describes the terms used to define a specific container location. A stack is a group of containers on top of each other. A row is a group of 6 containers side by side. A tier (or level) is a layer of containers.

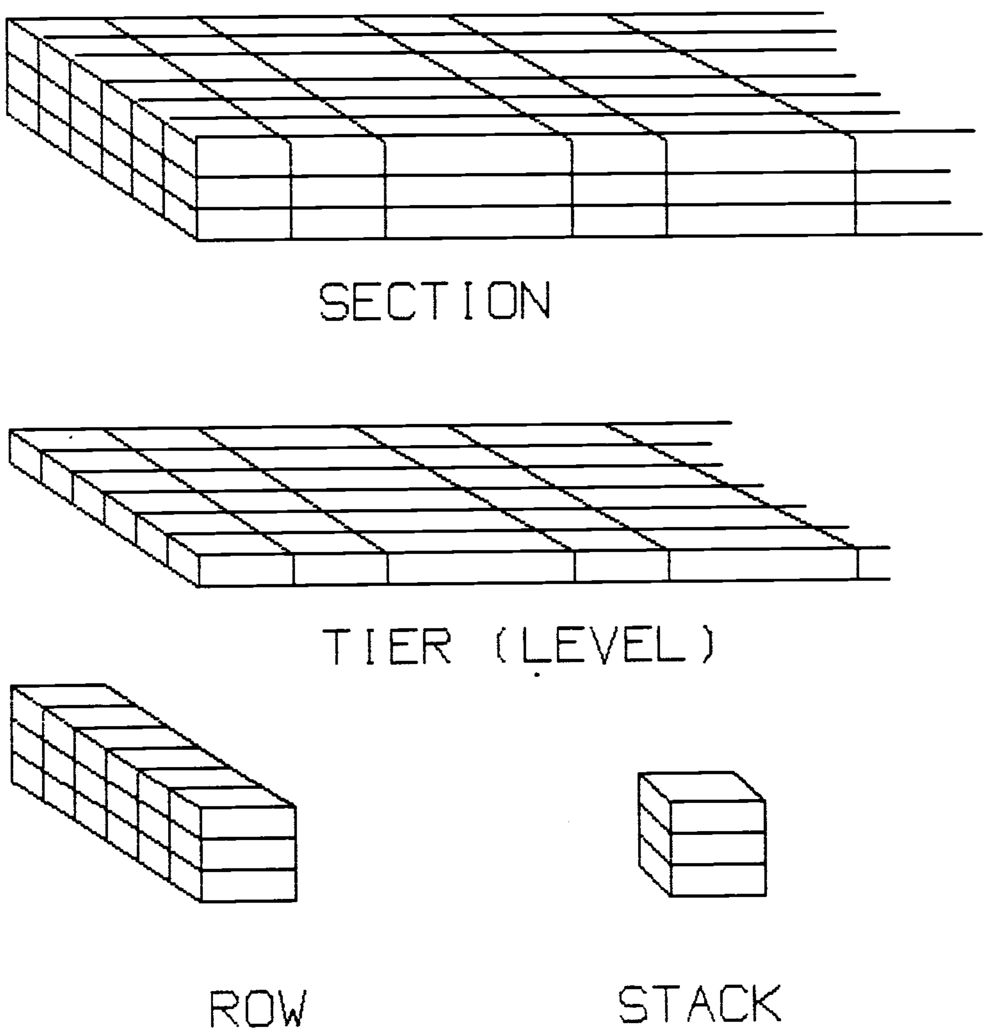


Figure 2.8 Yard Storage Section

Current practice at Port of Portland's Terminal 6 is to store containers up to three high so that a container in inside stack can be lifted and moved over other stacks to the truck lane. Each section can accommodate 37 rows of 20 feet containers, and each row can have up to 6 stacks. Containers of different lengths are not stored in the same stack or in the same row. Currently, Terminal 6 has 21 storage sections. There are a number of electric outlets for refrigerated containers. These locations can be used for the regular containers when they are not occupied by refrigerated containers.

In the yard, the cargo is arranged by ship and voyage number. Furthermore, cargo is segregated by weight and commodities, if possible. If there are a large number of containers for the same voyage, an attempt is made not to place cargo for the same voyage in one spot. This is to ensure that two or more transtainers can work on these containers at the same time without interference.

Literature Review

In a broad sense, a loading activity can be viewed as three interrelated stages -- pre-load planning, load planning, and post-load planning or the actual loading process. A literature review revealed that most of the past studies have devoted their efforts to the development of new strategies for yard side operation or to the computerization of load planning function. Both these activities correspond to the first two stages. The objective of these studies was to reduce the total loading time, and to identify factors for improvement in the port operation. However, no serious consideration has been given to the post-load planning operation. Significant improvements in post-loading operation can be achieved through improvement of the actual loading process, such as the utilization of a buffer area while performing the loading operation.

Cho (1981) developed a simulation program using GASP IV to study the merits of following operational parameters and policies of a container port :

1. Maximum stack heights.
2. Loading from a randomly stacked yard sections versus loading from a pre-ordered sections.
3. Implementation of any initial yard stowage pattern through input data to provide the flexibility of being able to handle various policies of partial preload yard stowage.
4. Number of yard trucks to be used.
5. Loading from a single section versus loading from two sections.

The most interesting result of the study was that a significant amount of improvement could be made by pre-ordering containers before actual loading operation begins. However, realization of this idea is possible only in the circumstance where the bay loading plan necessary to develop a loading plan can be secured by the port before ship's arrival at the port. This is generally difficult to accomplish.

Martin (1981) did a study of computer assisted load planning based on the operation of Port of Portland's container handling terminal. The study evaluated the possible use of integer programming and dynamic programming in the load planning problem. However, these alternatives did not yield a pragmatic solution due to the difficulties in formulating a proper objective function and the problem size getting too big. The study developed a heuristic computer program for load planning based on the decision criteria used by PoP's load planner. The results from this heuristic were compared with the PoP's manually generated load plans in terms of several material handling factors. Results of the study showed that total loading time could be reduced by 0.6% by using the computer assisted loading plan.

Beliech (1974) proposed a heuristic solution method for load planning with emphasis on overstow minimization and hazardous cargo handling. This study was designed primarily for military applications, and is of limited use due to the assumptions of a single container length and random access in the yard, and the omission of vessel constraints such as deck strength and location of refrigerated containers.

Hydronautics, Inc. (Cojeen and Van Dyke, 1976) developed an interactive computer load planning system. The program takes into consideration minimization of overstows, different lengths of containers, ship stability, and restrictions due to refrigerated containers. The objective of the program was to minimize the ballast needed and to prevent crane interference. The load planning system was applied to Matson Terminals. Matson terminal reported that interactiveness of the Hydronautics' system was excellent, but erroneous assumptions about the distribution of container weights could cause stability problem.

Problem Formulation

The transportation of goods using containers has increased tremendously, both in volume and value, over the past decade or so. The cost of container port facilities is now approaching $15,000,000 per berth, and the standby cost of a modern container vessel can easily exceed $20,000 per day [Port of Portland Technical Seminar, 1979]. Hence, the highly capital intensive nature of a container port operation demands better utilization of the port facilities and enhanced satisfaction of port customers.

These goals can be achieved through the development of better strategies for container port operation, improved port design, and more efficient material handling facilities. This research is concerned with development of a better strategy for port operation.

The container port operation associated with the loading activity can be viewed as three phases as described in the previous section. These three phases are:

1. Arrangement of outbound containers in the storage yard.
2. Load planning
3. Actual loading operation

Based on past statistic on utilization of port facilities, about 1.3 transtainers are required to meet one ship crane's capacity during the loading operation [McDowell et al, 1985]. This implies that the operation of the transtainer could be a bottleneck in a loading operation. In fact, the ideal operation of the transtainer is possible only in the case where all outbound containers are arranged in the yard in the perfect order of the future loading sequence, so that unnecessary movements of the transtainer and rehandling situations can be avoided. However, this ideal situation is never accomplished in reality due to the random order of container arrivals. Furthermore, a loading plan cannot be developed until the ship arrives at the port.

Efforts have been made to reduce the unnecessary movements of the transtainer by :

1. developing a strategy for arranging containers in the yard so that the unnecessary movements of the transtainer are reduced during the load planning
2. developing a computer based load planning program that can provide better loading plan.

It is possible to reduce the unnecessary movements of transtainer to some extent using these approaches. However, these approaches are not able to eliminate unproductive time completely from the loading plan.

A certain number of rehandles and unnecessary transtainer movements are inevitably involved in a loading plan. Based on this fact, this research proposes the idea of utilizing a buffer area as a method to reduce unnecessary transtainer movements during the actual loading operation. This idea requires the assumption that the container port has a buffer area where a number of empty chassis are available at the beginning of loading operation and loaded containers can be temporarily stored while waiting to be loaded onto the ship. In this study, the buffer area is assumed to be located in between the storage sections and the dock area, as shown in Appendix E, and it has 30 spaces available for either empty chassis or loaded chassis. The objective is to investigate the effect of buffer area on the port operations; the specifics assumed here should not effect the generality of the result.

The proposed strategy of utilizing buffer area is composed of two basic components. The first component can be illustrated through the following example. Consider a situation as shown in Figure 2.9 where container 1 is scheduled to be loaded onto the truck before container 2, necessitating the rehandling of container 2. According to the POP's current practice, the transtainer picks up container 2 and places it at any empty stack in the same row. If there is no empty stack in the same row, any empty stack in the nearest row is chosen. Then transtainer loads the container 1 onto the truck. The container 2 remains at the new location until work sequence requires it to be loaded onto the ship.

If the buffer area concept was to be utilized, the transtainer will load container 2 onto an empty truck instead of placing it at some other location. The truck with container 2 travels to the buffer area, leaves container 2 there loaded on the chassis, picks up an empty chassis from the buffer area and joins the loading operation. After the removal of container 2 from the top of container 1, container 1 is transported to the ship for loading. When the loading sequence calls for container 2 to be loaded onto the ship, it is directly transported from the buffer area to the ship. The movements of the transtainer for handling container 1 or container 2 can be broken down into the following components.

1. Time for transtainer to pick up a container or put it down, (X),

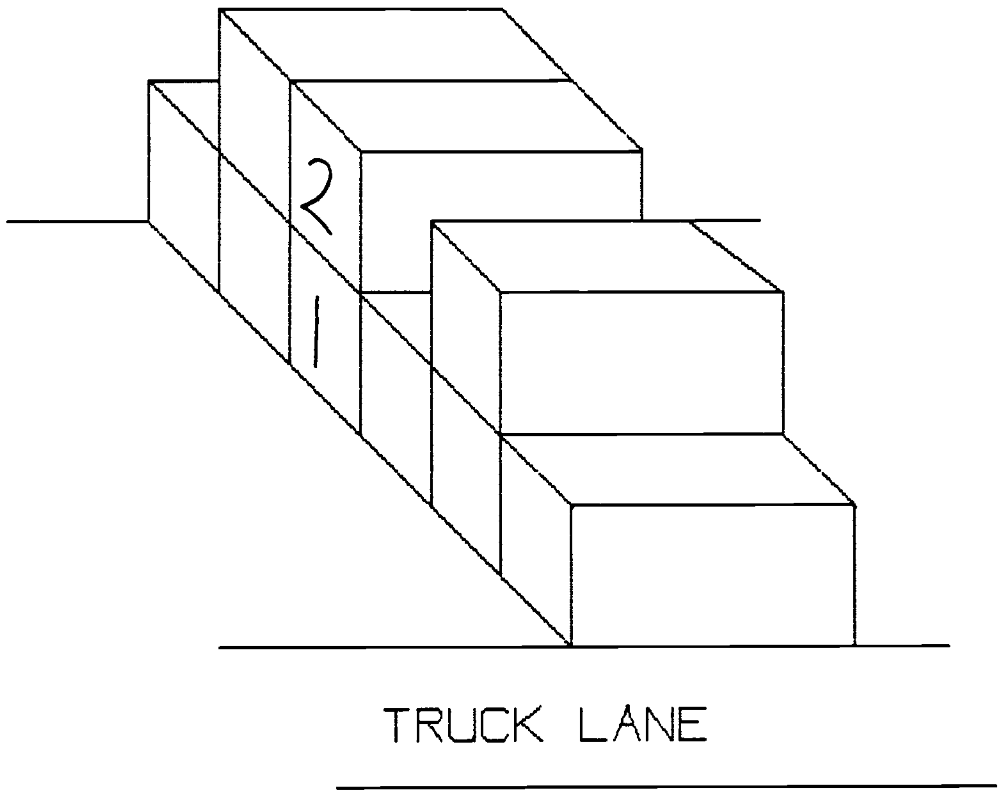


Figure 2.9 Container Rehandling

2. Time for the transtainer to move over stacks,(Y), and

3. Time for the transtainer to travel to an adjacent row, (Z).

Based on these three components, the total transtainer time for the current PoP's practice and for the proposed method is compared in Table 2.1.

Table 2.1 Comparison of Transtainer Operation Time

	PoP's practice	(time)	Proposed method	(time)
1	pick up cont. 2	(X)	pick up cont. 2	(X)
2	row change	(Z)	move over stacks	(Y)
3	put down cont. 2	(X)	put down cont. 1	(X)
4	row change	(Z)	move over stacks	(Y)
5	pick up cont. 1	(X)	pick up cont. 1	(X)
6	move over stacks	(Y)	move over stacks	(Y)
7	put down cont. 1	(X)	put down cont. 1	(X)
8	move over stacks	(Y)		
9	pick up cont. 2	(X)		
10	move over stacks	(Y)		
11	put down cont. 2	(X)		

Thus, the total time using the current POP's practice is

$$6X + 3Y + 2Z, \text{ and}$$

the total time by the proposed method is

$$4X + 3Y,$$

resulting in a possible reduction of

$$2X + 2Z.$$

According to Martin (1981), the transtainer operation times at Terminal 6 are estimated as follows :

loading operation $(2X + Y) = 158$ seconds adjacent row change (Z) = 21 seconds

Unfortunately, no data is available to estimate the value of Y (movement over stacks), separately. However, if it is assumed that Y is negligible, X becomes approximately 79 seconds. The amount of possible reduction using the proposed method is 2*(79 + 21) or 200 seconds.

The second component involved in the use of the buffer space is concerned with the movements of transtainer. One of the most time consuming operation of a transtainer is section changes. In particular, a non-adjacent section change takes about 162 seconds [Martin, 1982]. Thus, the operation of a transtainer becomes the bottleneck in the loading plan.

Consider a situation where the transtainer is currently working in a section and the transtainer is scheduled to work in another section after processing the current container. Further, assume that 2 containers will remain in the current section when the transtainer makes a section change. This situation implies that the transtainer will return to the current section later in order to process the remaining 2 containers. Transtainer section changes involved in this situation can be avoided by what is referred to here as a 'sweeping operation'. A sweeping operation consists of sending all remaining containers to the buffer area when a transtainer section change is scheduled and whenever the total number of remaining containers in the current section is less than or equal to a user specified number (referred to as the sweeping criterion).

Assume that the sweeping criterion is set to 5 at the beginning of a loading operation. Then, in the situation described previously, the transtainer starts the sweeping operation before it makes a section change since there are only two containers remaining in the current section. These two containers are thus sent to the buffer area before the transtainer makes the section change. These containers stay in buffer area until they are to be loaded onto the ship. By applying the sweeping operation, some of the time involved in the transtainer making section changes can be avoided.

The two basic components of utilizing the buffer area described above will be referred to as rehandling for buffer and sweeping operation throughout the rest of this study. This research is primarily concerned with investigating the effect of these two components on the loading operation at a container handling terminal.

At this point, considerations should be given to some other factors that are associated with the application of these two strategies in the actual loading operation. One of the factors to be considered is the number of empty chassis in the buffer area that are available at the beginning of loading operation. Both rehandling for buffer and sweeping operation will be executed only when there are enough number of empty chassis at the time of execution. This is because all the trucks participating in the execution will require empty chassis for subsequent operations. Therefore, the executions of the two strategies will be scheduled according to the loading plan but decisions concerning their actual executions will be dynamically made depending on the status of buffer area at the time of execution. Another factor to be considered is the total number of yard trucks participating in the loading operation. The effect of this factor will be examined in this study.

The advantages that could result from application of these two strategies to an actual loading operation are:

1. Reduction in time spent on rehandling by the transtainer,
2. Reduction in section changes made by the transtainer,
3. As a result of 1 and 2 above, reduction in operation bottleneck due to the transtainer speed and resulting in smoother material flow,
4. Reduction in loading time,
5. Increase in ship crane utilization, and
6. Increased customer satisfaction.

The biggest disadvantage in implementing these strategies is that they would require capital investment for construction of the buffer area and purchase of extra chassis.

Simplified Problem

Container port operation involved with shipping activities is viewed as a linkage between land transportation and maritime transportation, consisting of three phases: arrangement of outbound containers, load planning, and actual loading operations. The complexity of a container port operation arises from the conflict between the random arriving order of outbound containers and the bay stowage plan. The container port has no control over these two conflicting factors. Therefore, the goal of port operation is to find a loading plan that provides the best result within the constraints.

Past attempts to accomplish this goal have concentrated on the first two phases of the shipping operation. Some of these were successful in reducing total loading time and increasing facilities utilization, and in identifying important factors in port operation. It was found that transtainer operation is a bottleneck during the loading operation, and rehandling containers and section changes are the most unproductive movements of transtainer.

As described earlier, unproductive movements can be avoided by utilizing buffer area for rehandling for buffer and sweeping operation. The objective of this study is to develop a simulation model for a typical container handling port, and evaluate the following strategies:

1. Use of rehandling for buffer and the sweeping operation.
2. Effects on the port operation resulting from variations in number of yard trucks, number of empty chassis in the buffer area, and criterion for the sweeping operation.

The following assumptions will be made in developing the simulation program:

1. There are six storage section in the model, as shown in Appendix E.
2. One transtainer and one ship crane are available for the loading operation.
3. The buffer area is located as shown in Appendix E , and has 30 buffer spaces.
4. The location of ship crane is fixed during the simulation.

The following are considered to be basic requirements that the developed simulation program should be able to handle:

1. Generation of yard data.
2. Generation of a loading plan.
3. Execution of the loading plan.
4. Execution of rehandling for buffer and sweeping operation.
5. Collection of statistics.

Detailed description about each of these requirements is given in the next chapter.

III. MODELING APPROACH

This chapter describes the simulation model, YARD, designed to analyze the port operations. The simulation program was written and executed in PCModel.

Programming Language

The more commonly used computer languages in simulation studies are GASP IV, SLAM , and SIMAN. In this study, it was decided to use PCModel, a graphic simulation system, for the following reasons :

1. A graphic simulation system will allow the visual analysis of material flow of the yard and behavior of the port facilities.
2. The behavior of the system resulting from analyzing different stages can be "physically" observed during the simulation.
3. The results of the simulation can be more easily understood by the decision maker.

Also there are some specific features of PCModel that make the use of PCModel appropriate in this study. These are :

1. It has provision for defining up to 21 statistic generating sites. Data is automatically collected for these sites, giving the percent utilization of these sites. Furthermore, this data can be stored in a user defined file for later usage.

2. Equipment failures can be simulated through placement of a blocking character on the screen location of the equipment that is expected to fail.
3. The pace of the simulation can be adjusted by the user, thus allowing the user to concentrate on the operations of interest and consequences.
4. Simulation can be paused at any time during execution, and other system functions in the menu screen can be utilized to see the data regarding the system status or other system parameters.

Appendix A provides a brief review of the capability of PCModel. Also, some of the difficulties encountered in the application of PCModel to this study are described in Appendix A. The rest of this section outlines the terminology of PCModel used in this study.

1. Logical Screen - Logical screen is a memory-mapped display on which the movement of movable objects of the model are displayed along user defined paths. The dimension of a logical screen is set by the user, and it may consist of a maximum of 32767 row-column character cells.

2. Overlay File - Overlay screen is a background reference field on which movement of objects are calibrated. Overlay screen is created by the user with the screen editor of PCModel, and is stored in an overlay file (filename.OLY). The overlay file of this program is

shown in Appendix E.

3. Routing - Routes are the paths that objects in the model follow. The routing consists of a sequence of movement instructions used to define the various paths that objects of each job are to follow. Thus, a routing defines routes for a group of objects following the same paths. The program YARD consists of four routings.

4. User Symbols - Users can use three types of symbols to store values. The three types of values stored are constant values, 16 bit variable values, and 32 bit clock values. The constant value symbol starts with the character '#', and the value cannot be changed once it has been set at the beginning of the simulation. This type of symbol is used, for example, to define random number seeds. Symbols for variable values are named starting with '@'. These values can be modified during the simulation. Clock variables also can be modified during the simulation, and its name starts with '%'. Clock values are set and displayed using the HOUR:MIN:SEC format and can be modified during the simulation. Examples of these three types of values are :

#SEED=(2351) -- #SEED has a constant value of 2351:
This value cannot be changed during the simulation.

@LOADTIME=(10) -- @LOADTIME is set to 10 currently, and its value can be modified during the simulation.

%STARTIME=(0009:30:00) -- %STARTIME, a clock variable, is set to 9:30:00, and its clock value can be modified during the simulation.

5. Location Offsets - A location offset is used to define a symbolic representation for a screen location. Names of location offsets start with '*'. For example, *GATE=(XY(10,30)) defines that the variable *GATE refers to the screen location, X=10 and Y=30, during the simulation.

6. Array - The concept of array in PCModel is same as in FORTRAN. The only difference is that in PCModel the array is always dimensioned starting at zero through the given index value whereas in FORTRAN it is dimensioned starting at one. Thus, a 10 by 10 array will actually have 11 * 11 = 121 entries specified. An array name starts with the set of characters '@@'.

7. Posting and Clearing a Location Offset - These are used to post a screen location as occupied by an object and to clear a screen location. These can also be used as a logic control flags. For example,

```
*RHD=(XY(100,110))   --  define a flag location

PO(*RHD)             --  post the flag *RHD

JB(1,*RHD,:STOP)     --  if *RHD is posted, jump to
                         the label, STOP.
```

Program Structure

The simulation program, YARD, consists of five main components. These are :

1. System Configuration and Initialization.
2. Generation of Input Data and Loading Plan (Routing 1).
3. Transtainer and Yard Truck Operation (Routings 2 and 3).
4. Link Definitions
5. Generation of Echo Report (Routing 4).

The conceptual organization of the program YARD is shown in Figure 3.1. The term 'routing' in Figure 3.1 is specific to PCModel and refers to the different program segments for different jobs. Each of the segments are outlined below. A detailed description of each phase follows.

1. System Configuration and Initialization : This part of the program is used to define considerations such as :

a. Maximum number of objects that can be displayed at any moment during a simulation run.
b. Maximum number of user symbols and labels.
c. Dimension of the logical screen (width and height).
d. Overlay file designation.

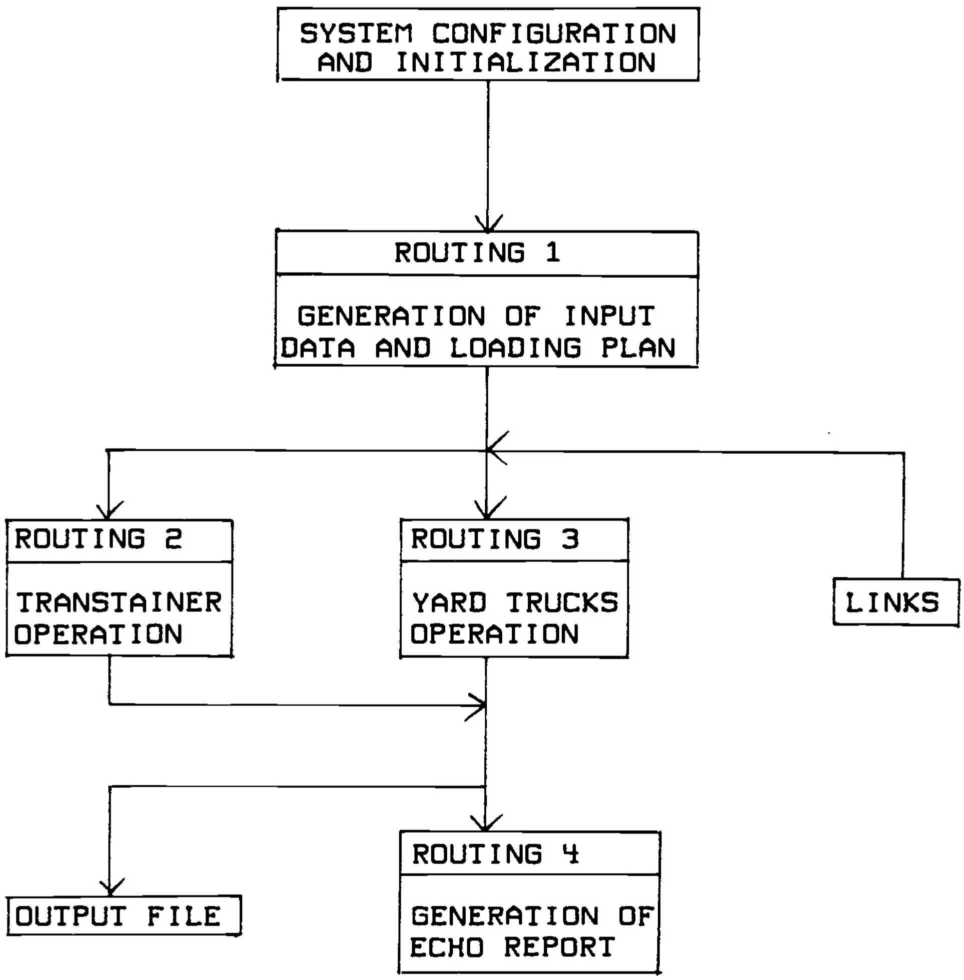


Figure 3.1 Overall Structure of Program YARD

e. Random number seed.

f. Name of numeric variables and their initial values.

g. Name of location offsets and their X and Y coordinates.

h. Job descriptions.

i. Collection points for utilization statistics.

2. Generation of Input Data and Loading Plan : In port operations, the first stage is the arrival of outbound containers and their storage in the storage yard in an efficient way. The next stage is the development of a loading plan that minimizes total loading time on the ship and maximizes utilization of yard facilities. The execution of the loading plan is the last stage of the loading process.

In order to execute and evaluate a load plan, the yard data and a load plan on the ship is required. As mentioned previously, this data is provided by the port and ship authorities. In this study, this data is generated randomly using some of the parameters currently applicable to the Port of Portland with some adjustments.

3. Routing Definitions for Transtainer and Trucks : Routing 2 and routing 3 describe the operations of the transtainer and yard trucks during the simulation. This segments of the program describe routes for every possible movements of the transtainer and yard trucks for the simulation.

4. Link Definitions : Link is conceptually equivalent to a subroutine in FORTRAN. Link names are preceded by the character '!'.

5. Generation of Echo Report : This portion of the program generates an echo report at the end of the simulation. This echo report contains a load plan and complete information about the actual operating sequence of the transtainer and the trucks during the loading process. This report is used mainly for debugging purposes and may also be used for more detailed analysis of the system status change during the loading operation. This portion of the program is designed to be activated optionally by a user. If the user sets the value of the variable, @SWITCH, to 1 at the beginning of the simulation, the user will be requested to enter a report file name, and the echo report generated will be stored in this file. Otherwise, this portion of the program will not be activated for the simulation.

System Configuration and Initialization

System configuration of the program includes the following specifications :

1. A maximum of 3,000 objects can be displayed on the screen, simultaneously.

2. A maximum of 60,000 numeric variables and location offsets can be used in the program.

3. The dimension of the logical screen for the program YARD is 150 columns and 200 rows. The overlay of the logical screen is shown in Appendix E. This overlay is stored in an overlay file named YARD.OLY, which is called from the program file during the loading process.

4. A random seed is specified.

5. All the numeric variables are defined and given initial values. This includes definitions of parameters that are used in the generation of yard data and the loading plan, and moving speed of the transtainer and yard truck. The more important among these parameters are:

@PRCHG -- probability that the next container in the loading plan requires the transtainer to change rows.

@PADCHG -- probability that the next container in the loading plan requires the transtainer to make an adjacent section change.

@PNADCHG -- probability that the next container in the loading plan requires the transtainer to make a non-adjacent section change.

@PRHD -- probability that the next container in the loading plan requires the transtainer to do a rehandle.

As mentioned previously, evaluation of the proposed strategy requires a randomly generated loading plan. The generated loading plans will be different from each other due to different values of the random seeds, and different values assigned to the four parameters defining the probabilities. To generate an applicable loading plan that has a reasonable number of row changes, section changes and rehandles, appropriate values of the four probabilities must be chosen. An initial estimates of these values were obtained from the operations at PoP [Martin, 1981], and are as follows :

* Probability of row change per container (@PRCHG) = 0.223
* Probability of adjacent section change per container (@PADCHG) = 0.039735
* Probability of non-adjacent section change per container (@PNADCHG) = 0.03784
* Probability of rehandling per container (@PRHD) = 0.01608

Using these initial estimates, the values of these parameters are determined such that the generated loading plan constitutes a reasonable experimental environment.

The moving speeds of the transtainer and yard trucks are important for the simulation model to closely mimic the real system. The movements on the screen consist of vertical movements and horizontal movements, and the moving speeds should be specified as the times to pass one column or one row of the overlay screen in order to be used in PCModel.

A transtainer requires 162.94 seconds for a non-adjacent section change and 15.09 seconds for an adjacent section change [Martin, 1981]. A non-adjacent section change can be divided into two components, vertical movements to cross the center aisle, and horizontal movement to pass one side of a storage section. The movement across the center aisle corresponds to the adjacent section change. The time to pass a side of a storage section is 147.85 seconds (162.94 - 15.09). Since the side of a storage section consists of 14 columns on the overlay screen, the horizontal movement speed is 10.56 seconds per column (147.85 / 14). The center aisle is 104.3 feet wide or the equivalent of 9.66 row numbers and thus the time for passing one row is 1.562 seconds (15.09 / 9.66) [Martin, 1981]. Since the center aisle of the overlay is represented as consisting of three screen rows in the overlay screen of the program YARD, the time for passing one screen row becomes 4.686 seconds (1.562 * 3). Thus, one column of the overlay screen is equivalent to about 20 feet and one row is approximately the same as 64.4 feet (20 * 9.66 / 3). The variables in PCModel always assume integer values. Therefore, the time for the transtainer to pass one screen column (@TTMRL) is set to 10 seconds, and the time to pass one screen row (@TTMOV) is set to 5 seconds.

The moving speed of a truck is assumed to be 25 miles per hour or 36.66 feet per second. Since one column of the overlay screen is equivalent to 20 feet, the time for a truck to pass one screen column is 0.55 seconds.

Similarly, the time for the truck to pass one screen row is 1.76 seconds (64.4 / 36.66). Again, due to the inability of PCModel to handle real variables, the horizontal moving speed (@TRMOV) of trucks is set to 1 second per screen column, and the vertical moving speed (@TRMOVV) is set to 2 seconds per screen row. The other major numeric variables are described in Appendix B.

6. Creating a location offset. This involves giving a name to a specific point on the overlay, and this point is then referred to by the location offset name during the simulation. The definitions of location offsets are given in Appendix B.

7. Specification of job descriptions. Job description describes the manner in which groups of objects, following the same segment logic, are to be released into the simulation. A job description consists of specifying seven parameters. To clarify, consider the job specification, J=(1,I,1,0,0,0,1). The first parameter, 1, indicates that this description is for job number 1. The second parameter, I, implies that any object doing this job will be displayed as a character 'I' on the screen during the simulation. The third index, 1, means that objects doing this job will follow routing description 1. The fourth and the fifth parameters are not utilized in this program YARD, but are for conditional job release. The sixth parameter, 0, indicates the priority of this job. The lower this value, the higher the priority.

Therefore, job 1, which in this program is for generation of yard data and a loading plan, will be done before any other job is initiated. The last parameter, 1, indicates the number of objects that will be released for this job. Thus, only one object will be released to generate yard data and a load plan, as described in routing description 1. Similarly, job descriptions 2 and 3 in this program describe the transtainer operation and yard trucks operation, respectively. Finally, job description 4 is used to generate an echo report at the end of the simulation.

9. Utilization statistics about the ship crane and the transtainer are collected during the simulation. The collection of statistics involves specifying a collection number, a label, and a location of collection. For example, U=(1,CRANE,*CRANE) means that the utilization statistics 1 is for crane operation and the collected statistics will show what percent of total simulation time the location offset, *CRANE, was occupied by objects. Utilization for the transtainer operation can be interpreted in a similar manner.

An important consideration concerning utilization statistics is the definition of busy and idle status for the ship crane and the transtainer. A transtainer is considered to be in a idle status only when it is waiting for a truck. Thus, all the other activities such as row change, section change, loading a container onto a truck, and rehandling consider the transtainer to be busy. Similarly, a ship crane is considered to be busy only when it is loading a container onto the ship.

Utilization statistics are stored in a user defined file. The user is required to enter a file name on the screen menu at the beginning of simulation.

Generation of Input Data and a Loading Plan

Syntax of PCModel requires arrays to be specified with array names and dimensions before they are utilized. This is usually done at the beginning of first routing. In program YARD, 12 arrays are defined at the beginning of routing 1. A brief description of each array follows. The detailed structure for each arrays is given in Appendix B.

1 - 5. Array @@STATUS1 is used to store the number of containers in each stack of storage section 1. Also it contains the total number of containers in each row of section 1. Likewise, @@STATUS2 through @@STATUS5 specify the total number of containers in each stack and the total number of containers in each row for sections 2 through 5, respectively. The model assumes a total of six sections. However, only five of these six sections are used to store containers in this program.

6. Array @@CLIST is a collection of all six @@STATUS arrays in one array as shown in Figure 3.2. This array is used as a basis for developing the loading plan.

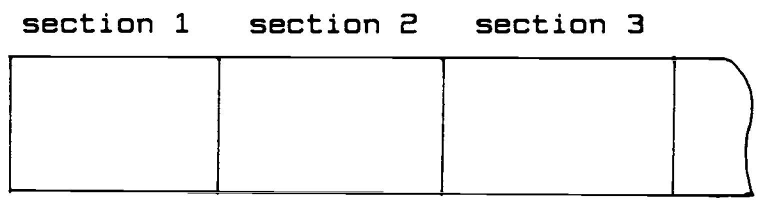


Figure 3.2   Structure of @@CLIST

7. Array @@PLAN contains an initial loading plan generated from the yard data stored in @@CLIST at the beginning of the simulation. This array basically provides information to trucks about their next operation, such as the container to be worked on next, location of the container, rehandles and the sweeping operation.

8. Array @@TTJOB contains information about the working sequence of transtainer. Entries of this array are made by yard trucks as the loading operation proceeds.

9. Array @@RHD contains lists of containers that will be transferred to buffer area as a result of rehandling or sweeping operation.

10. Array @@BUFFER represents the status of buffer space. Each cell of this array is assigned a value according to its status. The possible values are :

0 - the space is empty,

1 - the space is occupied by an empty chassis,

n - the space is occupied by a container whose load plan job number of the load plan is n.

There could be a confusion when n is equal to 1. However, this situation never occurs, since the first container in a loading plan would not be transferred to the buffer space as a result of a rehandling for buffer or sweeping operation.

11. Array @@BFJOB is used to list containers that are to be included in a special standby list so that the container in the buffer can be taken out to the crane at the appropriate time. When there is a container in this list, each empty yard truck first checks for the container. This is described in detail in Yard Truck Operation section.

12. Array @@ROWTOT is used to store the total number of containers remaining in each row for all sections. The values of this array are dynamically updated by transtainer operation, and are used to display the current row total during the simulation.

The rest of routing 1 is devoted to the generation of yard data and an initial loading plan. The first step is to generate the number of containers in each stack for all the sections. This requires the user to specify the following two parameters :

1. rows of each section that will have containers stored for the current voyage, and
2. the length of container in each row specified above.

Based on these specifications, the number of containers in each stack are generated in a random order as shown in Figure 3.3.

1. The row index, I, is incremented to indicate current row. If this index is greater than total number of row in a section (which is set at 34, corresponding to PoP's practice), then processing for current section is completed and the procedure is repeated for the next section.
2. If the Ith row is less than the maximum number of rows in a section, then a check is made if this row was specified by the user to contain any containers for loading. If the row was not specified by the user, the program moves onto the next row.
3. Otherwise, a random number between 0 and 3 is generated to assign a spot to the container. A stack can only store containers 3 level high. If the container length in current row is 40 ft, the next row will not be used to store containers since the 40 ft containers require two rows to be stored.
4. If all the stacks in the current row are already assigned, the procedure moves to next row, and repeats for all rows in the section, and for all five sections.

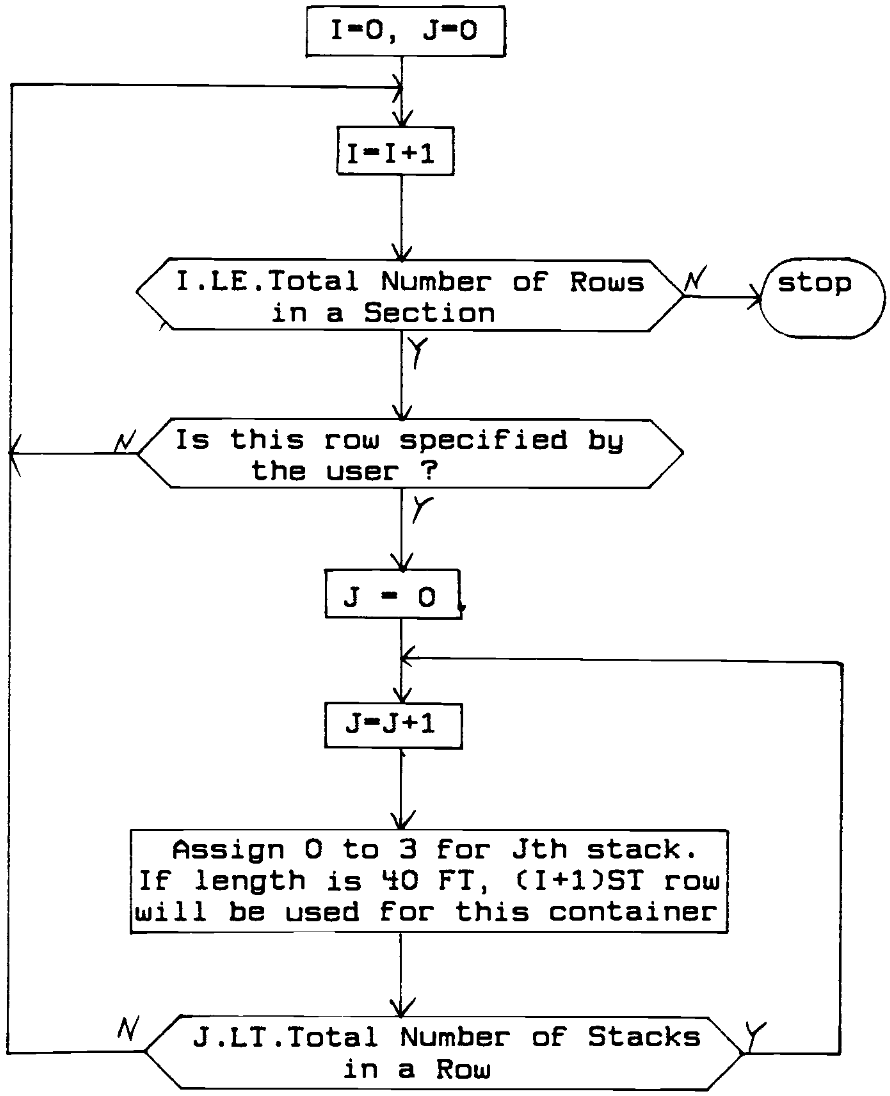


Figure 3.3 Procedure for Generation of Containers in each Stack of a Section

The number of containers in each stack and the total number of containers in each row of each sections are stored in arrays @@STATUS1 through @@STATUS5.

The total numbers of containers for each rows are displayed on the screen using a link (or subroutine), !NUMBER. As mentioned earlier, the data stored in @@STATUS1 through @@STATUS5 is stored collectively in @@CLIST for generating the loading plan. Section totals are computed and stored in numeric variables, @S1TOT through @S5TOT.

At this stage, an initial load plan is generated based on the yard data generated previously, and row and section changes, and rehandle probabilities (@PRCHG, @PADCHG, @PNADCHG, and @PRHD) specified in the initialization stage.

The overall procedure for generating an initial loading plan (@@PLAN) is shown in Figure 3.4. Based on the comparison of a random number with the values of @PADCHG, @PNADCHG, and @PRCHG, a selection is made on section change. If the selection involves a section change, a random number is generated to select the next section. Once a section is decided upon, a similar procedure is used for row selection. If the section or the row selected is empty, the appropriate portion of the procedure is repeated until an acceptable selection is reached.

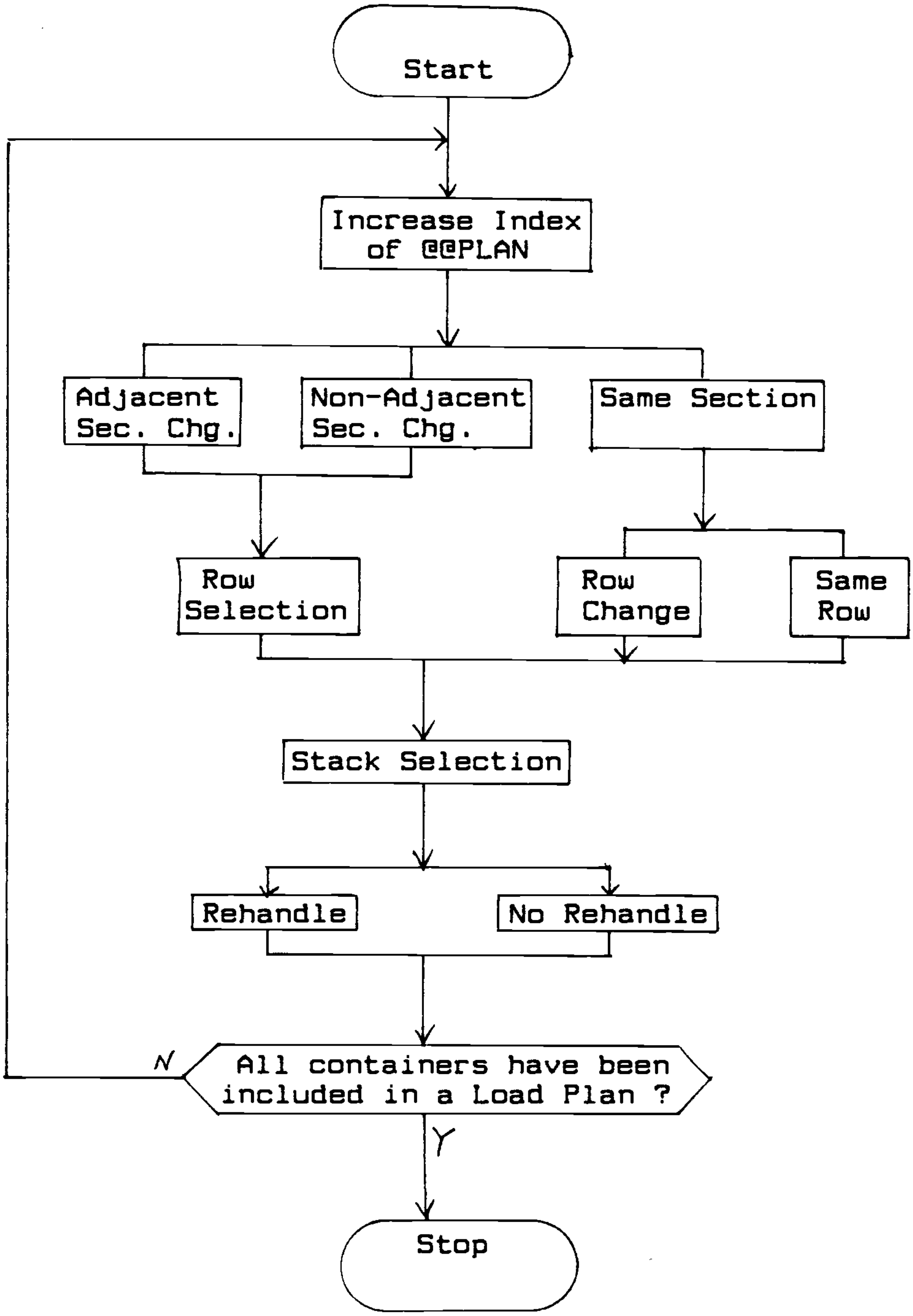


Figure 3.4 Generation of an Initial Loading Plan

Next, a stack selection is made using another set of random numbers. Again, if the selected stack is empty, the stack selection procedure is repeated until an acceptable selection is made. Finally, level selection in a stack is made as follows: if there is only one container in the stack, this container is selected as a next container to be loaded; if there is more than 1 container in the stack, a random number is generated and compared with @PRHD to decide if a rehandle is involved or not. If a rehandle is not involved in this container selection, the container on the top of the stackis selected as a next container to be loaded. If a rehandle is required in this selection, any container not on the top of the stack is selected, and the number of containers that should be rehandled for this container is determined accordingly, and the value is stored in the array @@PLAN(0,@JOBCOUNT). This value can assume one of the following three values :

0 - Ith container does not cause rehandle,

1 - Ith container requires one container to be rehandled, or

2 - Ith container requires two containers to be rehandled.

Also, the total number of containers available for loading in the selected row and selected section are adjusted accordingly, and the loading plan @@PLAN is updated. The row index of @@PLAN is incremented and the procedure is repeated until all the containers in the yard are included in the loading plan.

One strategy to be tested with this program is the effect of empty chassis on the loading operation. This requires user specification of the number of empty chassis in the initialization of variables. After the initial load plan @@PLAN is completed, each of the initial empty chassis in buffer area are displayed on the screen as '*' using the link !PUTCHA, so that the transtainer and yard trucks can start loading operation with buffer spaces.

Yard Truck Operation

This section of the program describes the operation of yard trucks during the simulation. The possible paths of a yard truck movements can be summarized as follows :

1. From crane to a section (Figure 3.5).
2. From a section to crane (Figure 3.6).
3. From crane to buffer (Figure 3.7).
4. From a section to buffer (Figure 3.8).
5. From buffer to a section (Figure 3.9).
6. From buffer to crane (Figure 3.10).

As shown in the Figures 3.5 through 3.10, the traffic flow of yard trucks is roughly circular from the ship crane to a section or the buffer area, then back to the crane.

To simulate the yard truck operation, a couple of locations need to be specified. These locations are

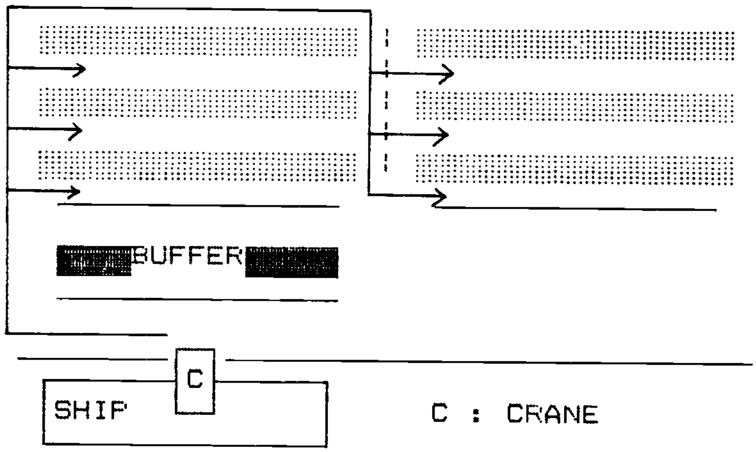


Figure 3.5 Movements of Yard Trucks From Crane to Section

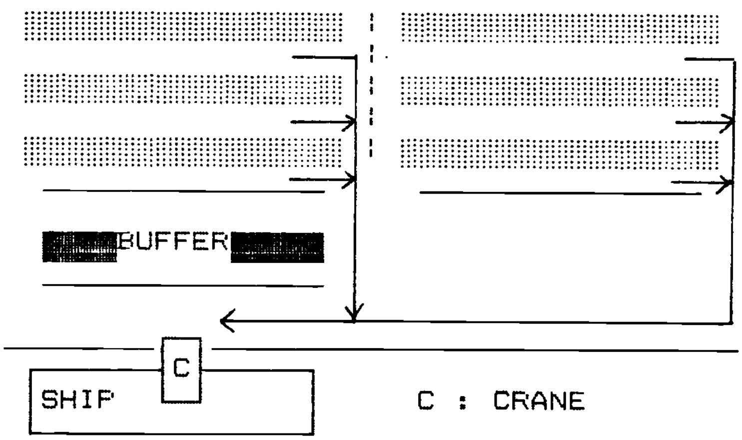


Figure 3.6 Movements of Yard Trucks from Section to Crane

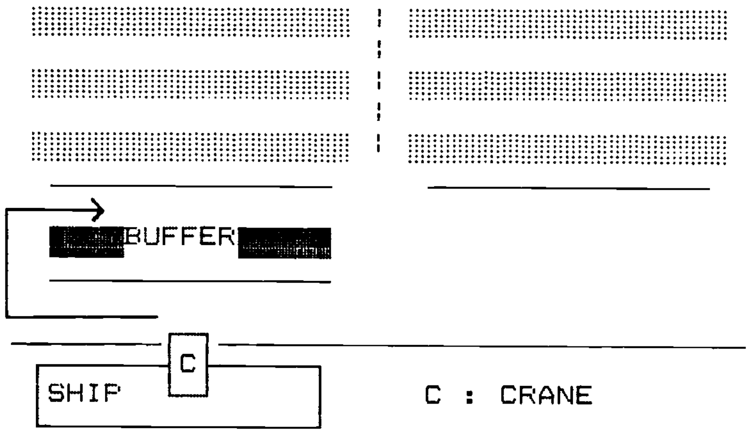


Figure 3.7 Movements of Yard Trucks from Crane to Buffer

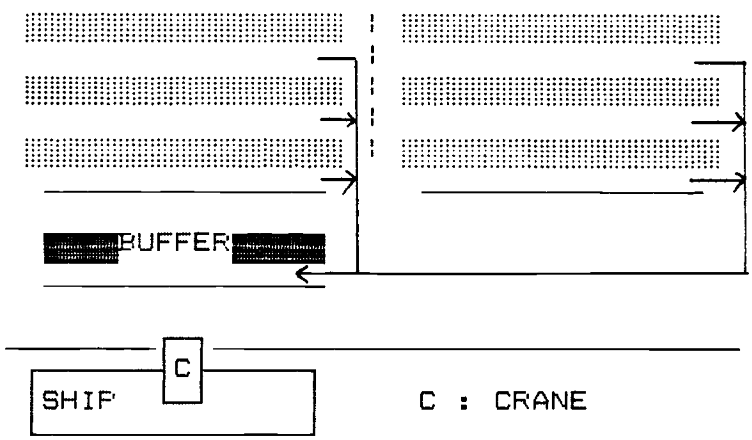


Figure 3.8 Movements of Yard Trucks from Section to Buffer

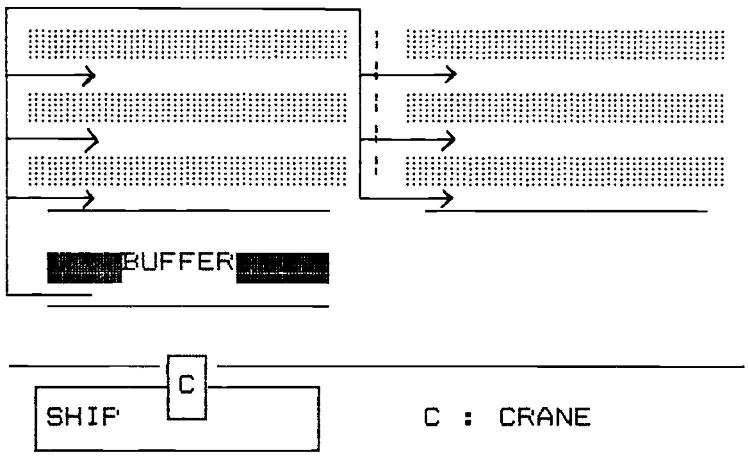


Figure 3.9 Movements of Yard Trucks from Buffer to Section

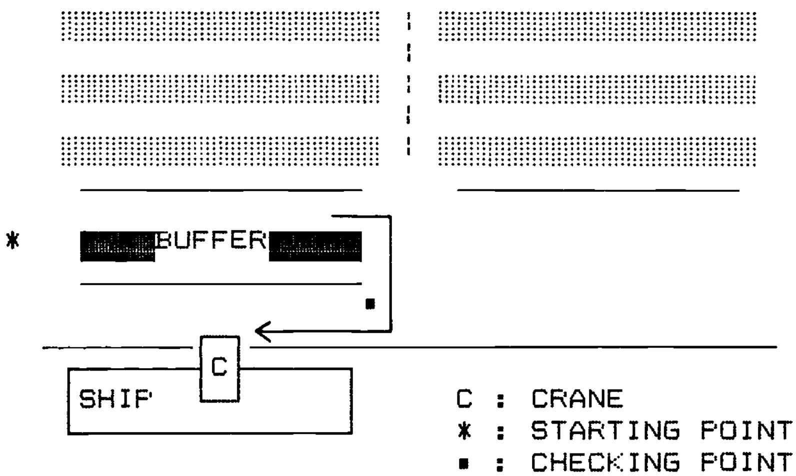


Figure 3.10 Movements of Yard Trucks from Buffer to Crane

referred to as the 'starting point' and the 'checking point', respectively, as shown in Figure 3.10. An empty truck coming from either the crane or the buffer area passes the starting point. At the starting point, the algorithm assigns the truck the container that is to be next loaded onto the truck. This assignment may consist of a regular container in a storage section to be transported to the ship, a container from a storage section to be rehandled and be moved to the buffer, a container from a storage section included in a sweeping operation or a container to be picked up from the buffer area and be transported to the ship.

Similarly, every container coming to the crane passes the checking point and the job number of each container passing this location is recorded and updated in a variable, @ORDER. A truck arriving at the starting point from either the buffer area or the ship crane uses the value of @ORDER to check if it is the appropriate time for removing a container stored in the buffer area, if any.

The transference of containers from the storage sections to buffer area as a result of a sweeping operation or rehandling for buffer is indicated through the use of the dummy location offset, *RHD. Thus, if *RHD is posted, a truck arriving at the starting point will transfer container from the storage sections to the buffer area.

The array, @@RHD, stores the job numbers of containers to be sent to the buffer area. Every yard truck arriving at the starting point checks whether or not the assigned container involves rehandling or sweeping.If it involves rehandling for buffer or the sweeping operation, it then checks the availability of empty chassis in buffer area. If there are enough chassis, the job number of the containers to be sent to buffer area are entered in the array @@RHD, so that the next arriving trucks will be assigned a container from @@RHD instead of from @@PLAN.

If the available chassis are less than the number of containers to be handled for rehandling for buffer, the transtainer follows the current PoP's rehandling practice. If the available chassis is less than the number of containers to be handled as a sweeping operation, this sweeping operation is suspended, and this section will be later checked for a sweeping operation with a reduced number of containers by future trucks in the same manner.

The program assumes that the buffer area has space for 30 empty chassis. The status of each buffer space is represented by the value of the corresponding cell of array @@BUFFER, as described in the beginning of this chapter. A yard truck entering the buffer area will be either from a storage section or from the ship crane. If it is coming from a storage section, it first discharges the container it is carrying, and then hooks onto an empty chassis to continue the loading operation. If it is coming from the crane to pick up a container from the buffer area, its operation consists of disconnecting its empty chassis, and picking up the target container. In either case, the search for an empty space, the target container, or an empty chassis starts from the first cell through the last cell as shown in Figure 3.11.

The number of initial empty chassis in buffer area is one of the parameters that effects the performance of loading operation, and is specified by the user at the beginning of the simulation. The number of empty chassis available in the buffer area is dynamically updated during the simulation, and it is stored in the variable @CHANUM. @CHANUM is decremented whenever a yard truck arriving at the starting point is assigned a container that will be placed in the buffer space. It is incremented whenever a yard truck at the starting point is assigned a container to be transferred from the buffer space to the crane. The reason for updating @CHANUM at the starting point rather than at the time of actual discharge or actual pick up in the buffer area is because decisions regarding sweeping operations or rehandling operations are to be made at the starting point, and this requires an assessment of @CHANUM when the current truck arrives at the buffer area. The main idea behind this scheme of updating @CHANUM is to take into consideration both containers currently in

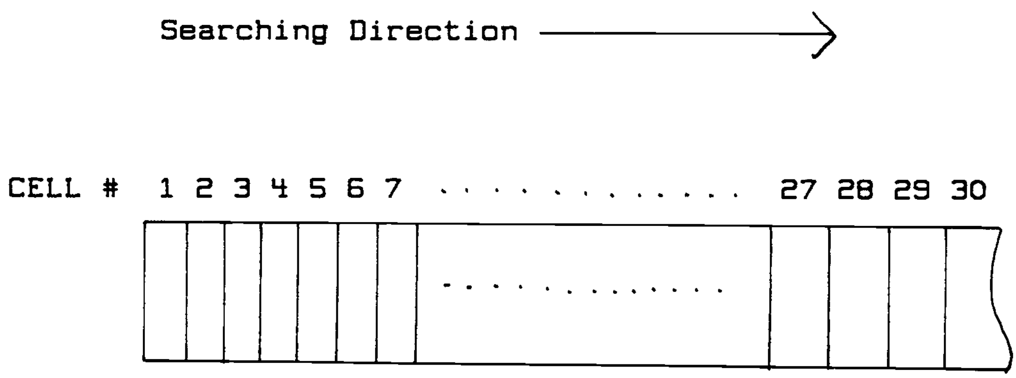


Figure 3.11 Search for Containers in the Buffer Area

buffer area and potential containers on their way to buffer area.

Figure 3.12 describes how a truck is assigned a container at the starting point. As shown in the figure, a yard truck arriving at this point first checks if there is any container in buffer area. If there is a container in buffer area, it then checks if it is the appropriate time to transfer the container to crane. This is done by checking a special standby list, @@BFJOB, and the value of a variable, @ORDER. If the value of @ORDER is greater than the job number of the target container in the buffer area, the container in the buffer space is assigned to the truck and transferred to the crane for shipping. For example, suppose that @ORDER is 40 and the job number of the container in the buffer space is 45. This means the 40th container of the loading plan has passed the checking point for shipping. Thus, the 45th container will not be removed from the buffer space to the crane at this time.

If the buffer space is empty or no container in the buffer space is to be transferred to the crane at this time, the truck then checks if *RHD has been posted. If *RHD is posted, the truck is assigned a container from @@RHD. It then moves to the container location, and transfers this container to the buffer area. If all the containers for that sweeping or rehandling operation have already been assigned yard trucks, this truck clears the *RHD, so that no more trucks will participate in this

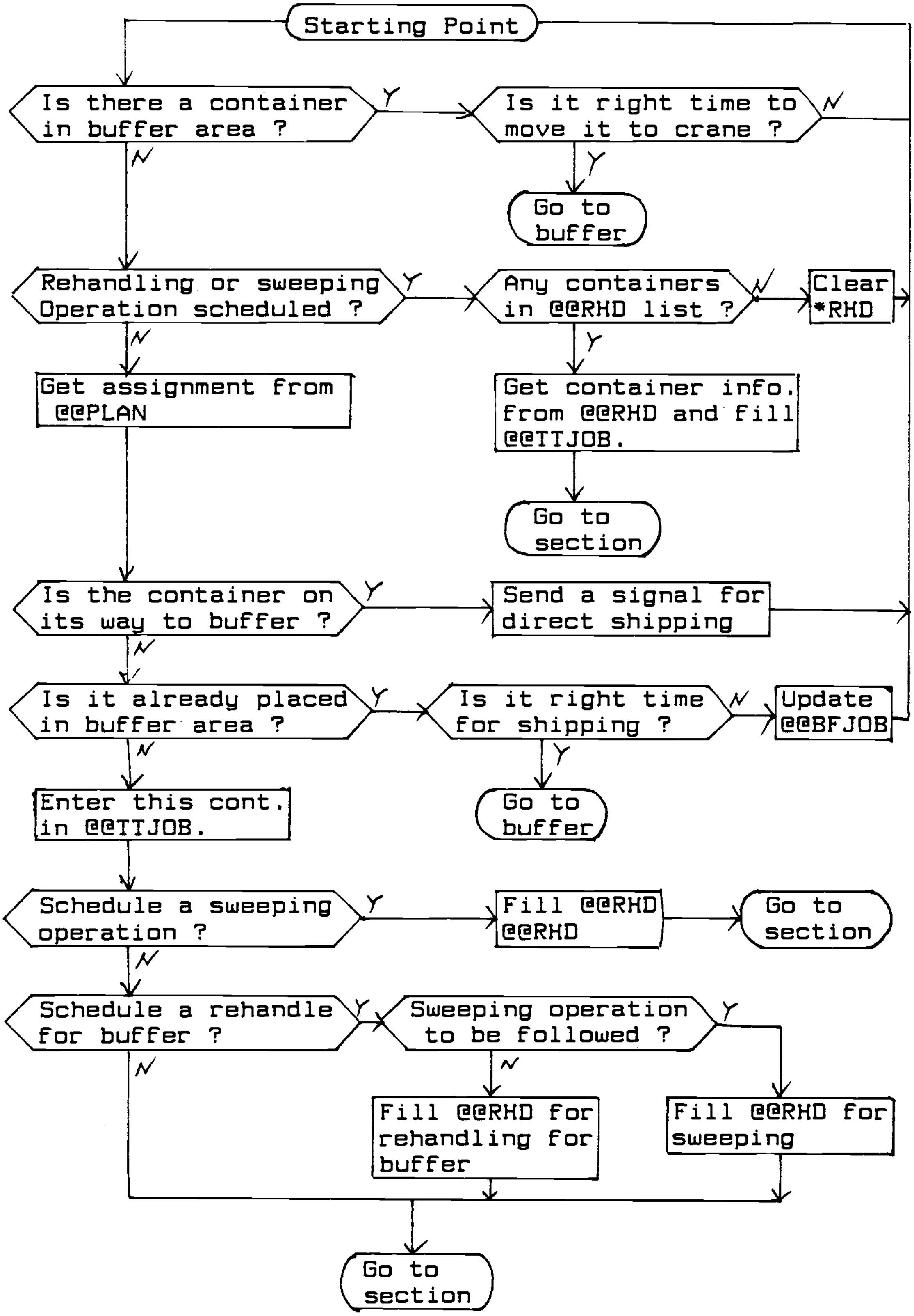
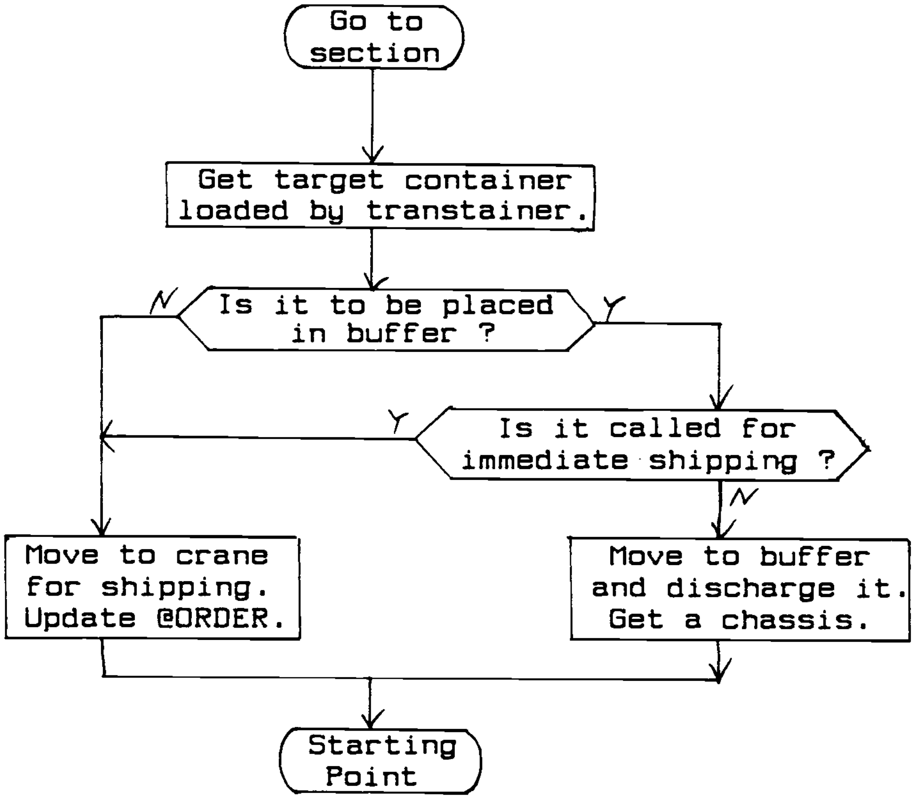


Figure 3.12 Yard Truck Operation



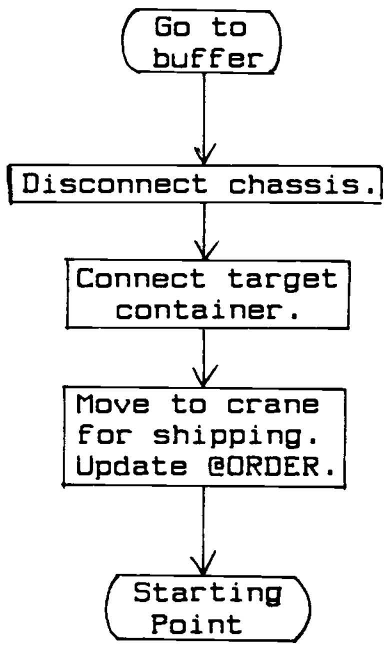


Figure 3.12 Yard Truck Operation (continued)

buffer operation.

If neither of the above works, the truck is assigned a container from @@PLAN. In this case, there could be three possible alternatives for the current container. The first alternative is that this container is located in the buffer area as a result of a previous rehandling or sweeping operation. In that case, it checks the value of @ORDER to see if it is the appropriate time for the container to be sent to the crane. If it is, the truck moves to the buffer area and transfers the target container to the crane. If it is not the appropriate time for removing the container, this container is listed in a special standby list, @@BFJOB. Then the job index of @@PLAN is incremented and same procedure is repeated.

The second alternative is that the assigned container is being transferred to the buffer area as a result of a previous rehandling or sweeping operation. In this case, this container would be called for shipping immediately after it is discharged in the buffer area. Thus, a signal is sent to the coming truck to directly transfer the container to the crane without discharging the container in the buffer area. The current yard truck at the starting point increments the job index of @@PLAN, and repeats same procedure until a container is assigned.

The third alternative is that the assigned container is still located in its initial location of the storage section. However, before the truck travels to the container location, a check is made regarding sweeping and rehandling for buffer operation. First, the algorithm checks the total number of containers that would remain in the section after processing this container. If it is less than or equal to the sweeping criterion, it then checks the value of @CHANUM to see if there would be enough empty chassis in the buffer area. If enough chassis will be available for this sweeping, the job numbers of the containers to be swept out are stored in @@RHD, and *RHD is posted by the current yard truck so that next arriving trucks will be assigned containers from @@RHD instead of from @@PLAN. Then the current truck enters information about the assigned container into @@TTJOB and moves to the container location.

If the sweeping criterion is not satisfied, then the algorithm checks if the assigned container is initially scheduled to require a rehandling. If it requires a rehandling, the algorithm searches for the containers to be rehandled and @CHANUM is checked for the chassis availability. If the value of @CHANUM is greater than or equal to the number of containers to be rehandled, @@RHD is updated, and *RHD is posted. Then, another check is made for the possibility that the number of the container that would remain in the section after this rehandling operation satisfies the sweeping operation. If it does, scheduling of a sweeping operation is checked as described previously.

After the truck is assigned a container and a check is made for scheduling of rehandling for buffer or sweeps at the starting point, the truck travels to the container location as described in the corresponding routing definition. After the target container is loaded onto the truck, it travels to next destination, which is either the buffer or the crane.

Transtainer Operation

The transtainer operations can be broken down into several elementary movements.

1. Loading a container : transtainer moves from the truck lane to a specific slot to pick up a container and moves back to the truck lane and loads a container onto the truck.
2. Regular rehandling : transtainer removes containers sitting on the top of the target container to reach the target container.
3. Rehandling for buffer : transtainer loads the container sitting on the top of the target container onto a truck to reach the target container. The rehandled container is transferred to the buffer area.
4. Row change : transtainer changes its row position along the truck lane.
5. Adjacent section change : transtainer moves to

an adjacent section.

6. Non-adjacent section change : transtainer moves to a non-adjacent section.

The operation of a transtainer is shown in Figure 3.13.

The first step in the transtainer operation is to retrieve information about the next container operation from @@TTJOB as shown in Figure 3.13. The value of @@PLAN(O,@JOBCOUNT), provides information about rehandling operation. A value of 0 specifies a regular loading operation, a value of 1 specifies a regular rehandle of one container, and a value of 2 specifies a regular rehandling of two containers. If the next operation is regular container loading, the transtainer will move to the location of the container with section change and row change, if necessary. It then waits until a yard truck becomes available. After loading the container onto the truck, it increases its job count based on the transtainer job list, @@TTJOB, and repeats the operation.

If the value of @@TTJOB(O,@JOBCOUNT) is 1 or 2, transtainer needs to remove 1 or 2 containers located on top of the target container before loading the target container. While the transtainer is rehandling a container, the symbol 'R' instead of the regular transtainer symbol is displayed on the screen. The program assumes that the time required for rehandling one container is 157 seconds [Martin, 1981].

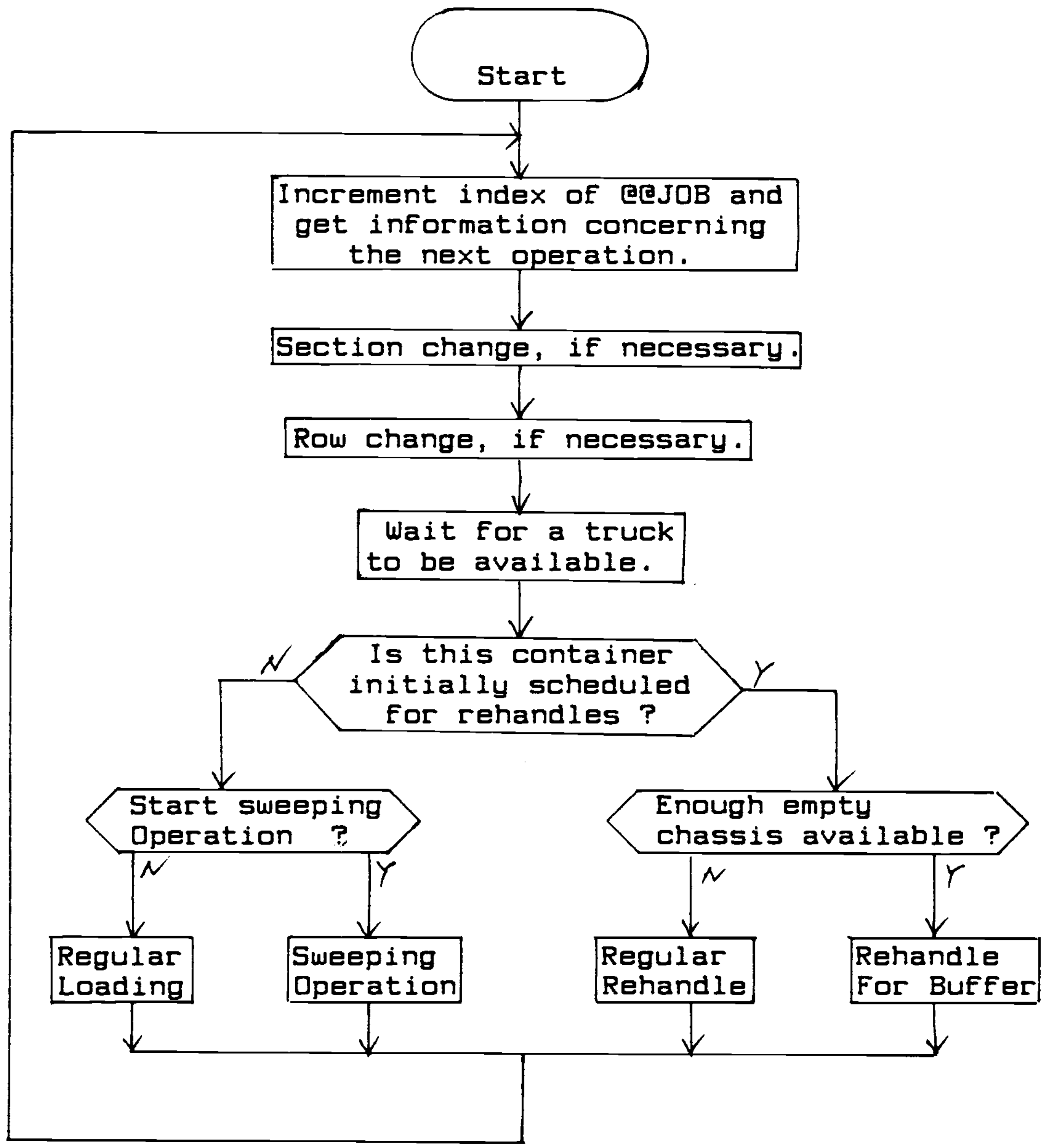


Figure 3.13 Transtainer Operation

After processing each container, the transtainer decrements the current row total, and displays new row total on the screen using the link !NUMBER. After new row total is displayed, it is stored in @@ROWTOT for later use.

Links

Links are conceptually equivalent to subroutines in FORTRAN. A link is defined in PCModel by BL(!linkname) and it ends with EL. BL implies Begin Link and EL implies End Link. Links are called from within a routing by LK(!linkname). Links are basically used to avoid the replication of same program logic in the program, as illustrated in Figure 3.14.

There are 20 links defined in the program YARD, and they are listed in Table 3.1 with a brief explanation.

Table 3.1 Links used in Program YARD

Link Name		Definition
!NUMBER	:	display number of containers in a row
!PUTCHA	:	display a chassis in buffer area
!PUTCON	:	display a container in buffer area
!CLEAR	:	erase a chassis or a container in buffer area on the screen
!TTSCHG	:	transtainer section change
!TTH	:	horizontal movement of the transtainer on the screen
!TTV	:	vertical movement of the transtainer on the screen
!TT15	:	transtainer movement from section 1 to 5
!TT51	:	transtainer movement from section 5 to 1
!TT2435	:	transtainer movement from section 2 to 4 or from section 3 to 5
!TT4253	:	transtainer movement from section 4 to 2

Table 3.1 Links used in Program Yard (continued)

Link Name		Definition
		or from section 5 to 3
!TT34	:	transtainer movement from section 3 to 4
!TT43	:	transtainer movement from section 4 to 3
!TTMOVU	:	upward movement of transtainer on the screen (this link is called from !TTV)
!TTMOVD	:	downward movement of transtainer on the screen (this link is called from !TTV)
!TTMOVL	:	left movement of transtainer on the screen (this link is called from !TTH)
!TTMOVR	:	right movement of transtainer on the screen (this link is called from !TTH)
!GOBF	:	truck movement to and in buffer area
!TRMOVD	:	downward movement of trucks on the screen
!TRMOVU	:	upward movement of trucks on the screen

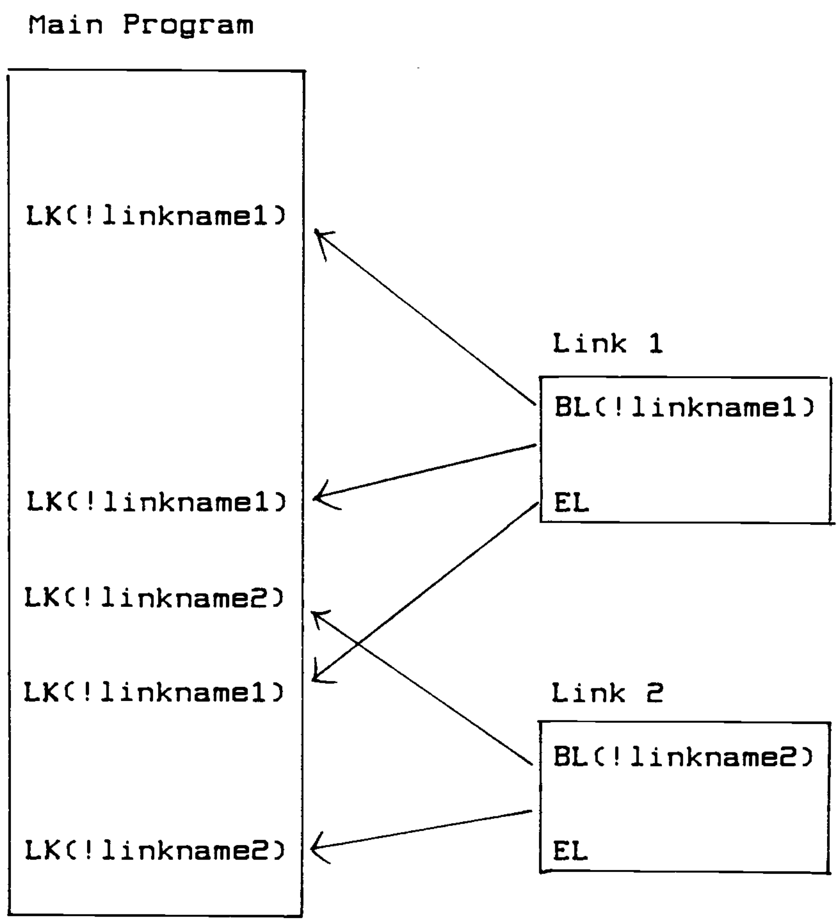


Figure 3.14 Role of Links in PCModel

Echo Report and Output

Routing 4 generates a echo report at the end of the simulation. As explained earlier, this routing is activated only when the value of the variable @SWITCH is set to 1 at the beginning of the simulation. As shown in Appendix H, the echo report contains two parameters and one array. The two parameters are the number of empty chassis and the sweeping criterion used for the loading operation; the array is @@PLAN. Array @@PLAN is used to store the initial load plan generated in routing 1 at the start of the simulation, and its contents are updated with time during the simulation. As described previously, the movements of the trucks and the transtainer are dynamically controlled by the information stored in this array. Thus, the echo report can be used for debugging purposes.

The output file contains information on the hourly utilizations of the crane and the transtainer during the simulation time, as shown in Appendix I. To get the output file, the user is required to enter a name for output file on the menu screen of PCModel after loading the program. The output files always have an extension of 'STS' after the user file name. Generation of this statistics file is a built-in feature of PCModel.

IV. SELECTED APPLICATIONS AND ANALYSIS OF RESULTS

This chapter presents two examples to analyze the effectiveness of the proposed methodology. Two different yard data and load plans were developed as described in the Chapter III. Then, each of these plans were simulated using different combinations of the following variables -- number of trucks(T), number of empty chassis in the buffer(C), and the sweeping criterion(S). The levels of each of these factors were determined using the PoP's current practice as the basis. Since the current practice of POP is to assign 3 or 4 yard trucks to one transtainer-crane operation, it was decided to use three levels for the number of trucks, 2, 4, and 6. The levels for the number of the empty chassis used in this study were 0, 5, 10, and 15, and for the sweeping criteria 0, 4, 8, and 12. A zero value for the chassis corresponds to the current PoP's practice. A total of 48 simulation runs were made with each load plan corresponding to the 48 combinations of the three factors. These combinations of the factor levels are shown in Table 4.1.

A difficulty involved in the analysis of the results is the deterministic nature of the results produced from the simulation. This is due to the fact that PCModel does not have the built-in functions to handle statistical distributions. Since the model parameters , such as the truck moving speed and the crane operation time are fixed

Table 4.1 Factor Level Combinations for Simulation

Truck Levels (T) - 2, 4, and 6

Chassis Levels (C) - 0, 5, 10, and 15

Sweeping Criterion Levels (S) - 0, 4, 8, and 12

T	C	S	T	C	S	T	C	S
2	0	0	4	0	0	6	0	0
2	0	4	4	0	4	6	0	4
2	0	8	4	0	8	6	0	8
2	0	12	4	0	12	6	0	12
2	5	0	4	5	0	6	5	0
2	5	4	4	5	4	6	5	4
2	5	8	4	5	8	6	5	8
2	5	12	4	5	12	6	5	12
2	10	0	4	10	0	6	10	0
2	10	4	4	10	4	6	10	4
2	10	8	4	10	8	6	10	8
2	10	12	4	10	12	6	10	12
2	15	0	4	15	0	6	15	0
2	15	4	4	15	4	6	15	4
2	15	8	4	15	8	6	15	8
2	15	12	4	15	12	6	15	12

during the simulation, each combination of the factor levels always results in one deterministic result. (This shortcoming of PCModel is further discussed in Appendix A). It was therefore decided to analyze the results graphically instead of performing statistical analysis. One disadvantage of using graphics as an analysis tool is that the interactions between factors cannot be explicitly ascertained, and the contributions of each factor to the dependent variable cannot be quantified.

Applications

Two load plans, referred to as Plan A and Plan B, were generated using different random number seeds. The parameters used in developing plan A and B are shown in Table 4.2. The main characteristic of the load plans developed based on the data in Table 4.2 and different random number seeds are given in Table 4.3. The load plans A and B, and the corresponding user input are listed in Appendix F, and G, respectively. The results from 48 simulation runs for both the plans are summarized in Appendix J.

Analysis of Results

<u>Loading Time Analysis</u> Figure 4.1 shows the total

Table 4.2 Values of Parameters used in Generation of Load Plans A and B

Parameter	Plan A	Plan B
@PNADCHG	0.045	0.045
@PADCHG	0.04	0.044
@PRCHG	0.45	0.45
@PRHD	0.05	0.051

@PNADCHG	:	probability of a non-adjacent section change per container
@PADCHG	:	probability of a adjacent section change per container
@PRCHG	:	probability of a row change per container
@PRHD	:	probability of a rehandle per container

Table 4.3 Characteristic of Load Plans A and B

Item (total)	Plan A	Plan B
Containers	333	342
Rehandles	11	13
Row Changes	1658	1757
Adjacent Section Changes	13	14
Non-adjacent Section Changes	14	10

loading times for Plans A and B when 2, 4, and 6 yard trucks are used. The abscissa of each graph represents

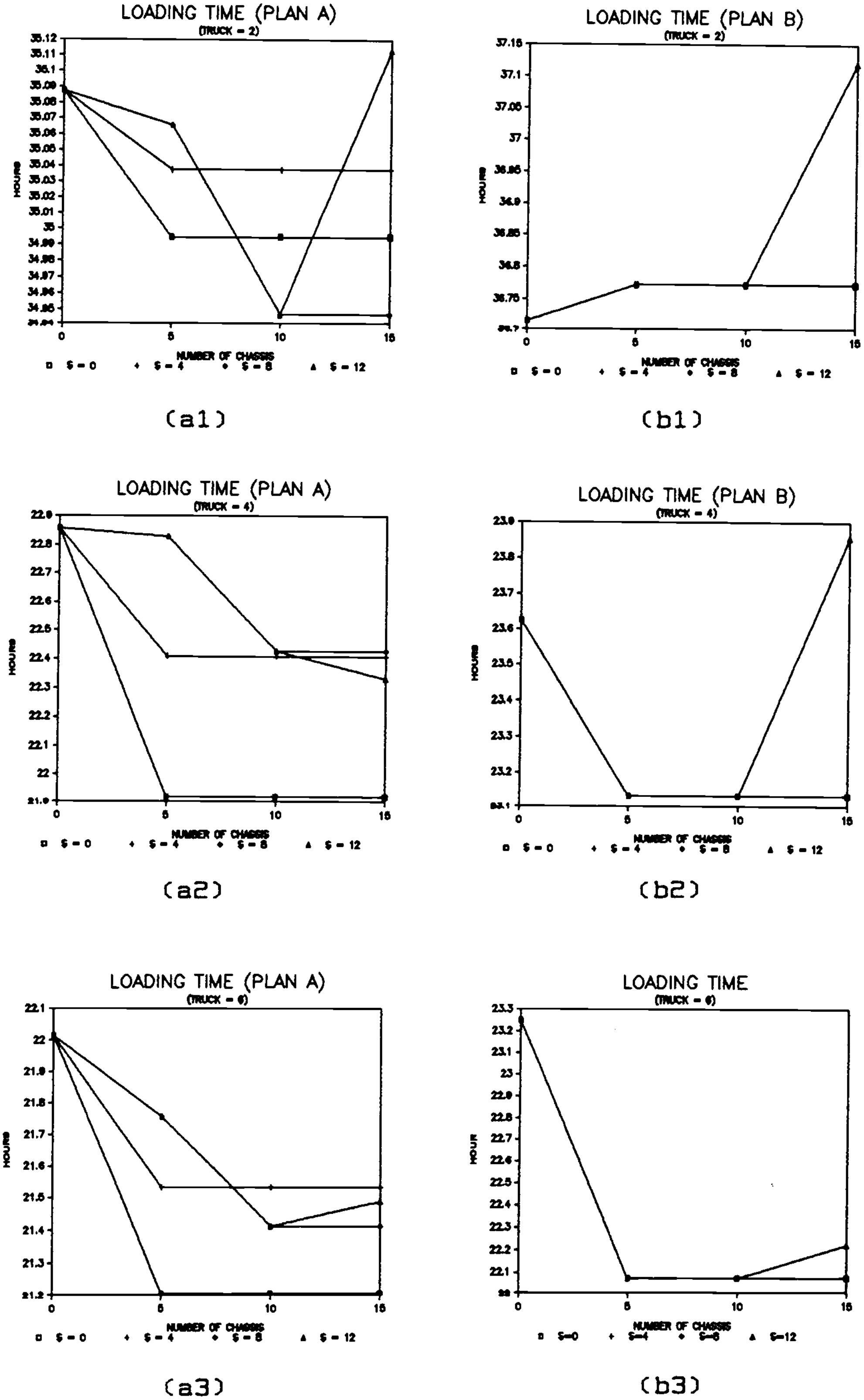


Figure 4.1 Load Times for Plans A and B

the different levels of chassis used, the ordinate represents the loading time in hours and each curve of the graphs represents different values of the sweeping criterion. The zero chassis level (C=0) represents the PoP's current practice since no buffer operation will occur. The zero sweeping criterion (S=0) implies that rehandling for buffer is attempted but no sweeping operation are to be scheduled during the loading operation. Thus, the amount of reduction made by zero sweeping criterion, when the chassis level is greater than zero, represents the time reduction resulting exclusively from rehandling for buffer. When both the sweeping criterion and the chassis level are greater than zero, both the sweeping operation and rehandling for buffer are attempted during the loading operation.

If any of the graphs in Figure 4.1 show identical results for different chassis levels, it indicates that the load time for that sweeping criterion is stabilized, and further increase of chassis is redundant. For example, Figures 4.1 (a1) and (b1) show that the lines for zero sweeping criterion (S=0) do not change when more than five chassis are used. This implies that five empty chassis are enough to handle 11 and 13 rehandles, initially scheduled in plans A and B. Similarly, it shows that load time for the sweeping criterion of eight (S=8) in Figure 4.1 (a1) is stabilized after 10 chassis. This implies that all possible sweeping operations and rehandles for the buffer could not be fully actualized due to lack of the empty chassis at a chassis level of five, but ten chassis are enough to handle the sweeping criterion of eight, and use of more than 10 chassis becomes redundant at this sweeping level.

When two trucks are used, 35.09 and 36.72 hours are required to execute load plan A and load plan B according to the PoP's current practice (Figures 4.1 (a1) and (b1)), corresponding to C=0 and S=0. When only rehandling for buffer is applied to plans A and B with five chassis, (a combination represented by T = 2, C = 5, and S = 0), a 0.2% reduction of load time for plan A and a 0.15% increase of the loading time for B is achieved. These results are summarized in Appendix J.3. These results show that the application of rehandling for buffer did not improve the total loading time at T=2, and C=5. This lack of improvement is not due to the lack of chassis, because the line for S=0 is stabilized after C=5. The reason for this is the low number of yard trucks. This conclusion can be explained as follows:

1. Suppose that ten containers have been sent to the buffer area as a result of rehandling for buffer. Whenever one of these containers is called for loading onto the ship, one of the two yard truck will be assigned to the buffer operation to remove the container from the buffer to the crane. Then, the transtainer will have to operate with only one yard truck until the other truck is

released from the buffer operation. During this time, the transtainer idle time is increased, and the overall loading operation is delayed to a certain extent.

2. Another factor to be considered is the location of transtainer when the buffer operation occurs. If the transtainer is working in the section farthest from the buffer location at the time of buffer operation, the transtainer undergoes longer idle status, since the one yard truck assigned to the transtainer must travel longer distance to be loaded. The location of transtainer at the time of buffer operation depends on the structure of each load plan. However, the location of transtainer would not be a limiting factor if enough number of yard trucks were available, as some of the yard trucks can still keep the transtainer busy.

Therefore, the effect by rehandling for buffer depends both on the number of yard trucks and on the structure of each load plan. The structure of each load plan becomes more critical when a small number of yard trucks are used. The difference in loading times for the two plans is also due to the difference in structures of the plans.

When the sweeping criterion is set to 4 for plan A, the performance is inferior to that for S=0, as shown in Figure 4.1 (a1). The line for S=4 stabilizes at C=5, which implies that 5 chassis are enough to allow all possible sweeping operations at S=4, and rehandling for buffer can be processed without the chassis being a bottleneck. The fact that the line for S=0 and S=4 do not coincide indicates that some containers were processed through the sweeping operation. The inferior performance for plan A with S=4 as compared with plan A when S=0 may be summarized as follows:

1. For S=4, more containers need to be removed from the buffer area to the crane during the loading operation, than when the sweeping criterion was zero. Therefore, the transtainer is idle for a longer period of time as one of the two trucks is being used for transferring containers to the crane from the buffer.

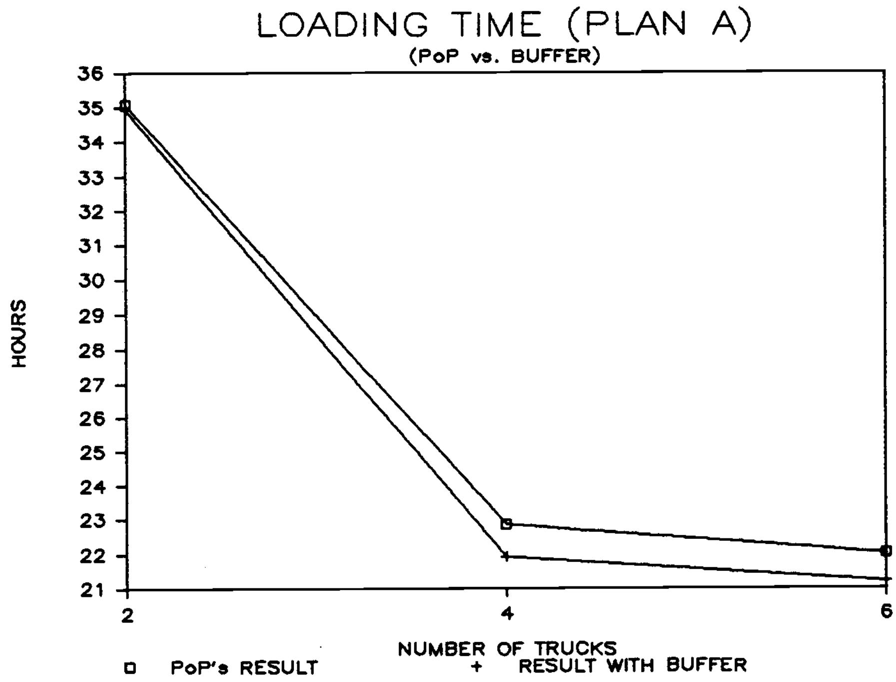
2. Another reason for relatively poor performance is the nature of the sweeping operation. The objective of the sweeping operation is to send all remaining containers in a section to the buffer area, when the sweeping criterion is met and enough empty chassis are available in the buffer area. Suppose that 4 containers are placed in the buffer as a result of a sweeping operation. The chance that these 4 containers will be called for loading onto the ship consecutively is very high. Therefore, when these 4 containers are called for loading, both of the two yard trucks are assigned to the buffer operation. As a result of this, transtainer will be kept completely idle while these 4 containers are being transferred to the crane.

3. The location of transtainer at the time of buffer

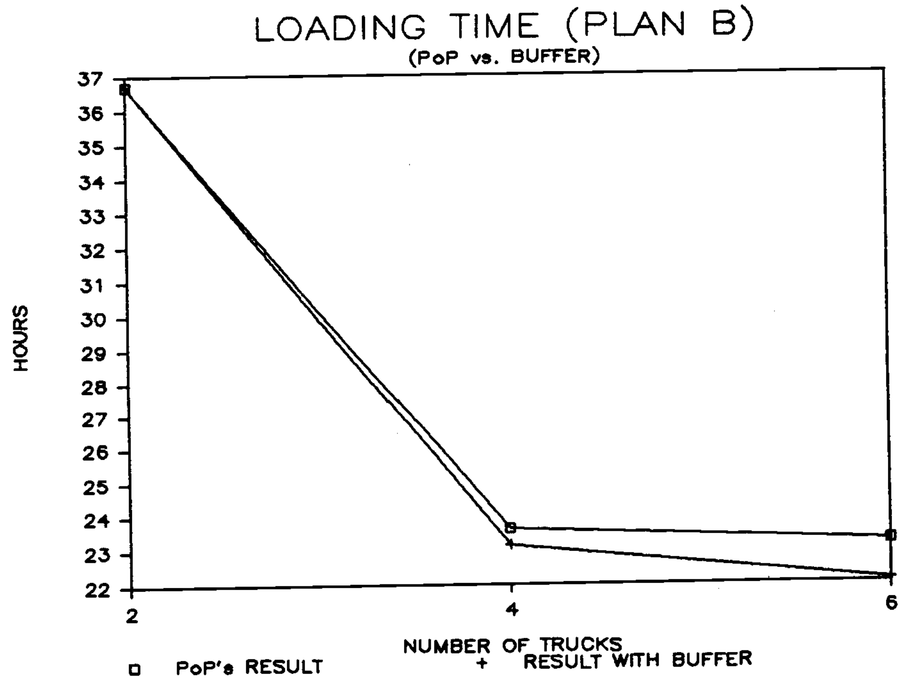
operation affects the duration of transtainer's idle status, as explained previously.

When the sweeping criterion is set to 8, the sweeping operation is stabilized at the chassis level of ten, as shown in Figure 4.1 (a1). When sweeping criterion is set to 12, the loading time is same as S=8 at C=5 and C=10. This is because total number of chassis is less than sweeping criterion. Thus, the number of empty chassis constrains the actual size of the sweeping operation during loading. This is further shown by the line for S=12 changing when the number of chassis is increased from 10 to 15, while the line for S=8 is stabilized at C=10.

It can be concluded for Figures 4.1 (a1) and (b1) that the results for different sweeping criteria when two trucks are used do not provide meaningful information on their use in the load planning operation, and the main differences in the performance of the load plans are due to the structure of each load plan which represents a random factor in this study. This is more clearly illustrated by Figures 4.2 (a) and (b). Figure 4.2 shows the relationship between the number of yard trucks and the advantages associated with the use of the buffer area. The line for PoP's results corresponds to C=0; the other line represents the best results obtained from different strategies for the buffer area, as indicated in Figure 4.1. The difference between the two lines in Figure 4.2 indicates the savings in the load time that can be



(a)



(b)

Figure 4.2 Effect of Truck Level on Total Load Time realized from the use of buffer area, and the variation of a line itself indicates the effect of different levels of yard trucks on the load time. The figure shows that the gap between the two lines increases as the number of trucks is increased for both plans. This means that the benefit from the use of buffer area becomes more significant as the number of trucks used for loading is increased. The use of buffer area will significantly improve the total loading time only when enough number of trucks are available. However, the increase of yard trucks would not improve the performance of the system due to the use of buffer area beyond a certain level. This is the level beyond which the idle time of the transtainer, caused by the buffer operation, cannot be reduced any more, and the increase of yard trucks becomes redundant.

When 4 trucks are used, 22.86 and 23.66 hours are required for plan A and plan B to be executed according to PoP's current practice (C=0), as shown in Figures 4.1 (a2) and 4.1 (b2). As mentioned previously, the benefits from the use of buffer become significant as the number of available trucks increases (Figures 4.2 (a) and 4.2 (b)). Appendix J.3 shows that 4.09% and 2.08% of total loading time were reduced for plan A and plan B, respectively, when the number of trucks available was increased from 2 to 4.

When 6 trucks are used during the loading operation, PoP's results are 22.02 and 23.25 hours for plan A and plan B, as shown in Figures 4.1 (a3) and 4.1 (b3). Also, total loading times for plan A and plan B are reduced by 3.67% and 5.07% through the use of buffer area when truck level is increased from 4 to 6 as shown in Appendix J.3. Figures 4.2 (a) and 4.2 (b) show that the increase of truck level from 4 to 6 does not significantly improve the loading time. Furthermore, the benefits from the use of buffer area become more significant, since the gab between the two lines is much greater for 6 trucks than for 4 trucks. Specifically :

1. The random effect of the structure of each plan does not significantly affect the performance of each sweeping criterion at T=4 or T=6, whereas when only two trucks were used, the pace of the transtainer operation rapidly slowed down at the time of buffer operation and the random effect resulting from the structure of each load plan affected the outcome of buffer usage.

2. The transtainer's idle time during the buffer operation is reduced significantly as the truck level is increased.

3. As a result of 1 and 2, benefits from the use of buffer area will be increased compared to T=2, and the comparisons of sweeping criterion will provide meaningful and more reliable information.

Figures 4.1 (a2) and 4.1 (b2) also show that the zero sweeping criterion dominated other sweeping criteria for both plans. However, the difference between the results obtained for S=0 and S>0 become smaller as the truck level is increased from 4 to 6. Based on these observations, it may be concluded that an increase in truck level reduces the transtainer idle time caused by the buffer operations. Further, the increase in truck level will continue to improve the performance of the system until the transtainer's idle time cannot be reduced any more. Also, the performance when the sweeping operation (S>0) is used will be superior to a plan with no sweeping operation, S=0. Therefore, the number of yard trucks utilized in port operations should correspond to the point where the cost of additional trucks and transtainer's idle time are minimized.

Material Handling Results Figures 4.3 (a) and 4.3 (b) show the total number of transtainer row changes for plans A and B, respectively. When the buffer area is used, the total number of row changes is always reduced because both rehandling for buffer and sweeping operations eliminate the unproductive movements of transtainer from one row to another. The number of row changes, as well as the number of adjacent and non-adjacent section changes (Figure 4.3 through Figure 4.6) are independent of the truck levels. The truck level affects the speed of material flow, but has no effect on the movements of the transtainer which is a function of a load plan.

Figure 4.4 and Figure 4.5 show the effects of the buffer area on adjacent and non-adjacent section changes. Non-adjacent section change is one of the most unproductive movement of transtainer operation. The primary objective of the sweeping operation (S>0) is to improve loading operation by the elimination of non-adjacent section changes. However, this results in an increase of adjacent section changes. For example, suppose that the initial load plan requires the transtainer to travel from section 1 to section 5 to section 4 at some time during the loading operation. Based on the layout in Appendix E, this traveling path involves two non-adjacent section changes. If a sweeping operation is applied to section 5, all the containers in section 5 will be transferred to the buffer area and section 5 would become empty. Then the actual travel path of the transtainer will become section 1 to section 4. This implies eliminating two non-adjacent section changes and adding one adjacent section change. Even for this simple situation, the total transtainer operation time will be reduced very significantly because an adjacent section change takes about 15.09 seconds compared to 253.46 seconds for a non-adjacent section change [Martin, 1981]. Figures 4.4 and 4.5 show that adjacent section changes are sometimes increased but non-adjacent section changes are always reduced when the sweeping operation is applied.

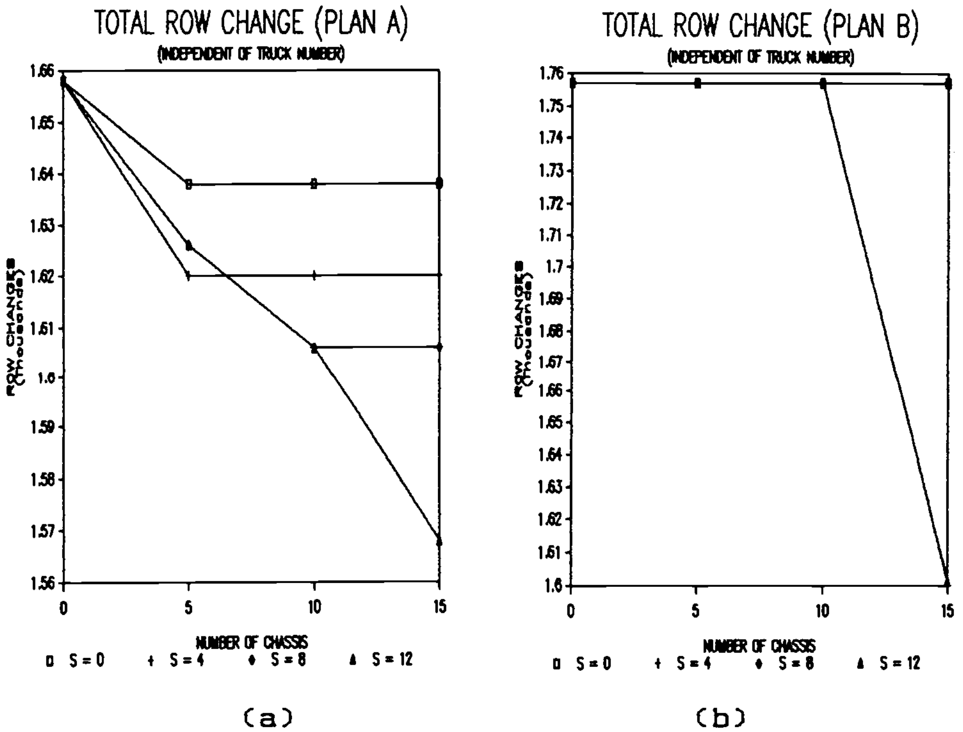


Figure 4.3 Total Row Changes

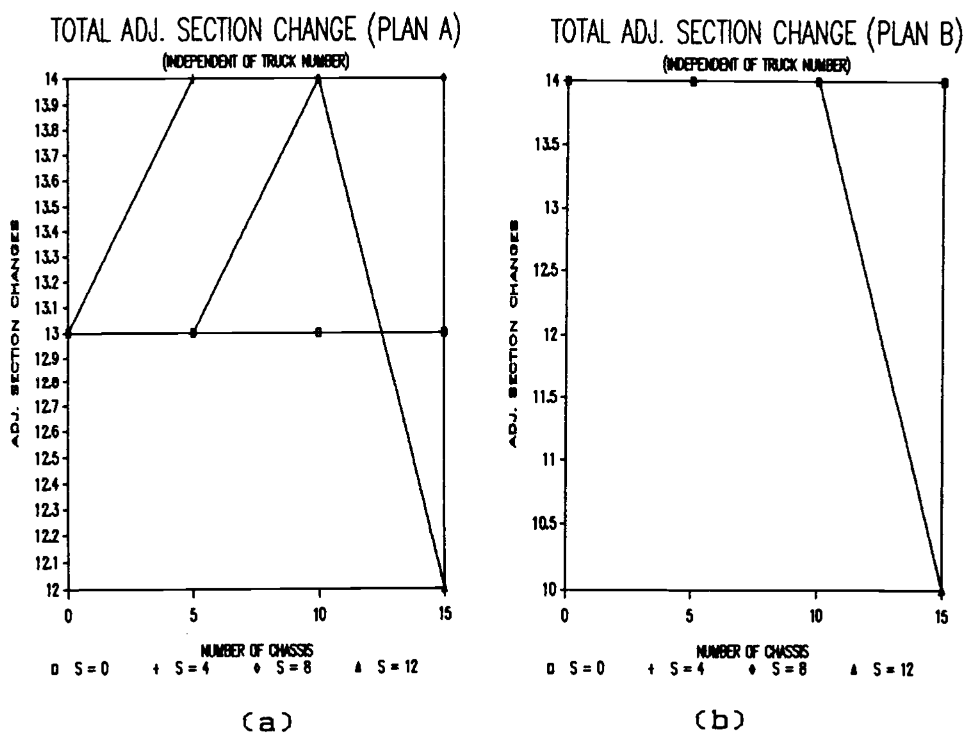
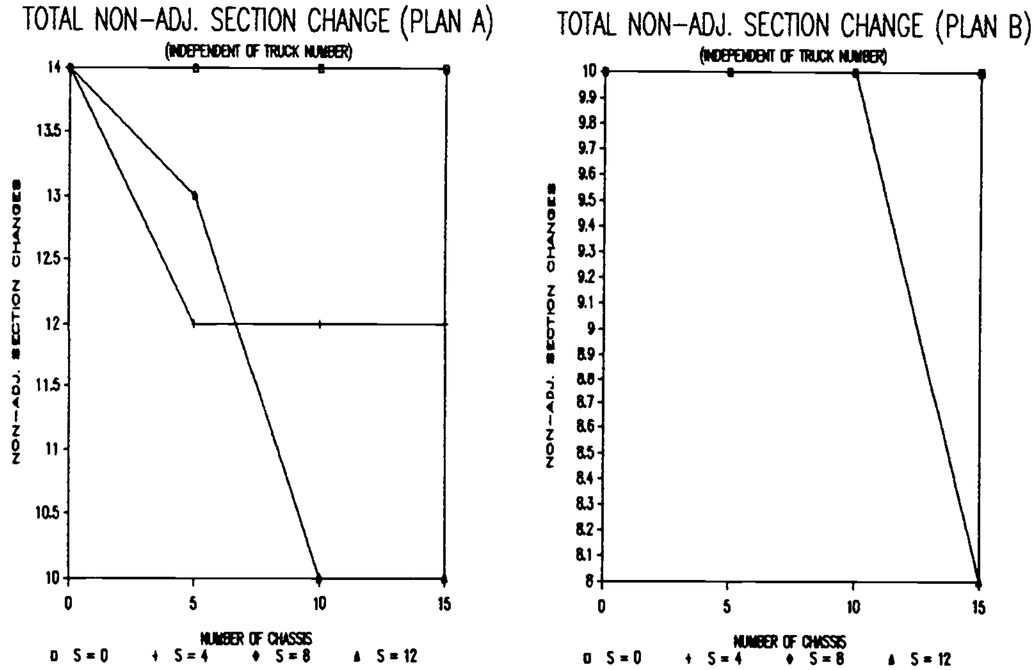


Figure 4.4 Total Adjacent Section Changes

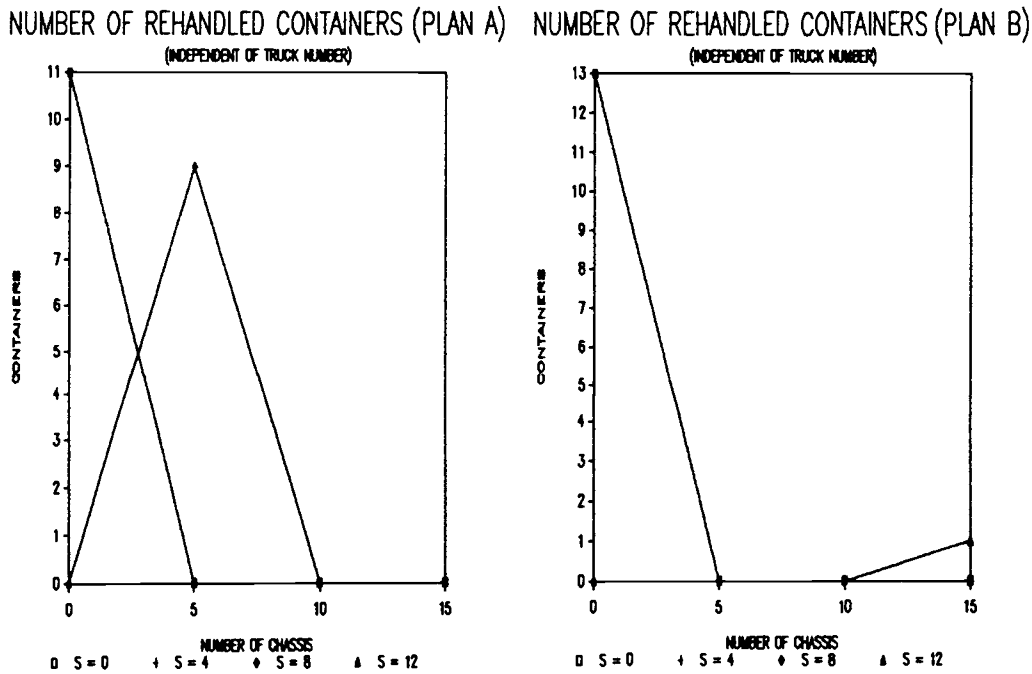
The number of rehandled containers for plans A and B is shown in Figures 4.6 (a) and 4.6 (b). The reduction of rehandled containers is one of the primary objectives of rehandling for buffer. Figure 4.6 shows that the number of containers rehandled is always reduced when rehandling for the buffer occurs. However, when both the sweeping criterion and the chassis level are greater than 0, there could be a situation where rehandling for buffer cannot be fully actualized because the sweeping operation reduces the availability of the empty chassis during the loading operation. This situation is indicated by the non-zero points in the figures. For example, one rehandle was observed when fifteen chassis were used for sweeping criterion of twelve for plan B (Figure 4.6 (b)). This implies that one of the 13 initially scheduled rehandles for plan B could not be processed for rehandling for buffer because not enough empty chassis were available for the rehandle at the time of the request. This is because the sweeping operation for the sweeping criterion reduced the availability of the empty chassis during the loading operation.

Figure 4.7 shows the utilization of crane at different truck and chassis levels when sweeping criterion is set to 0. Due to the inavailability of distribution functions in PCModel, the crane operation time per container is set to a constant value of 175 seconds during the simulation. Thus, the total operation time of the



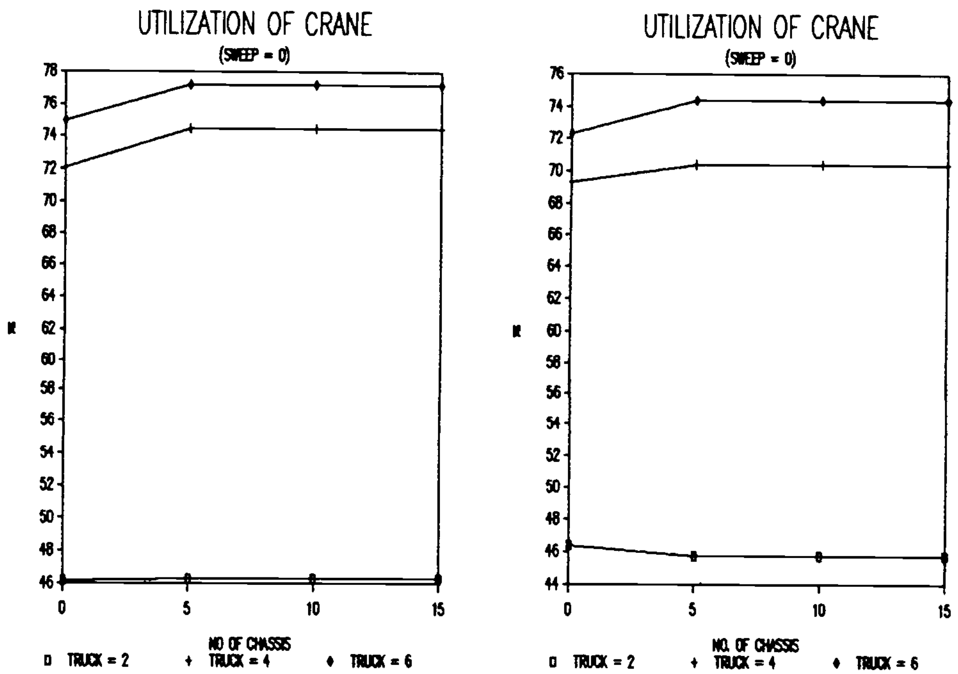
(a) (b)

Figure 4.5 Total Non-adjacent Section Changes



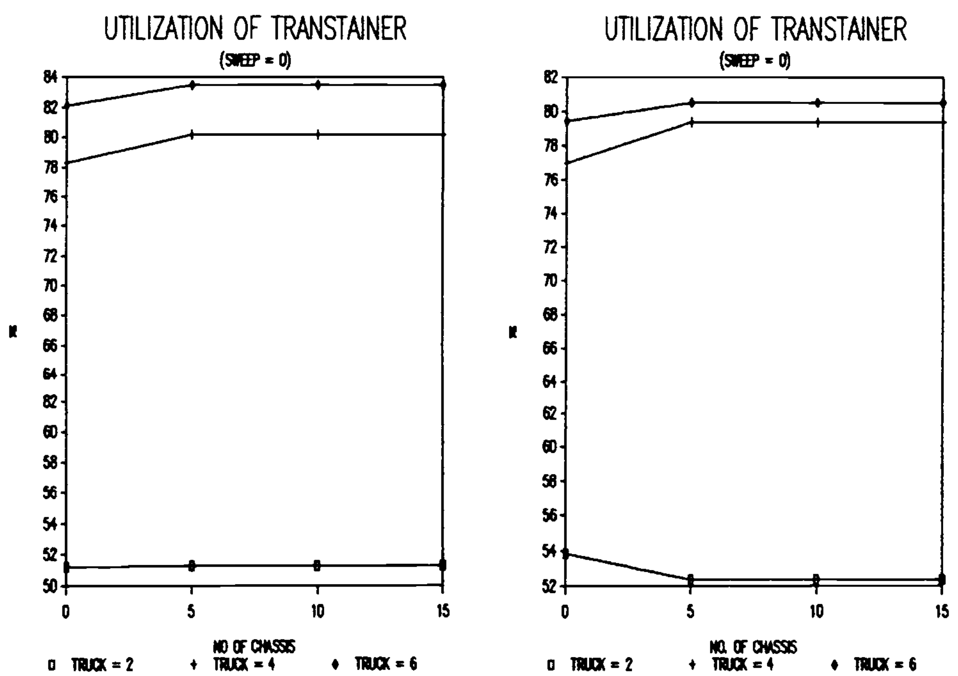
(a) (b)

Figure 4.6 Number of Rehandled Containers



(a) (b)

Figure 4.7 Utilization of Crane



(a) (b)

Figure 4.8 Utilization of Transtainer

crane can be computed from the total number of containers loaded, which can be used to determine the crane utilization. A comparison of total load times (Figure 4.2) and the crane utilization (Figure 4.7) shows that the utilization of the crane is inversely proportional tothe loading time. As the total loading time decreases, the utilization of the crane increases proportionally.

Figure 4.8 describes the utilization of transtainer at different levels for trucks and chassis. The transtainer's utilization is determined by two elements: the first is the rehandling for buffer and the sweeping operation, and the second is the pace of material flow or the number of trucks available. In general, as the number of containers processed by rehandling for buffer and sweeping operation increases, the unproductive transtainer time decreases. However, if a small number of trucks are used for the loading operation, transtainer will stay idle while the trucks are transferring the containers from the buffer area to the crane. In this case, the unproductive movements of transtainer are eliminated with the use of buffer area, but this elimination does not improve the total loading time. Thus, the use of buffer area improves both transtainer operation and the total loading time only when a sufficient number of yard trucks are used.

The simulation program YARD was run on an IBM PC/XT; the average run time was about 7 minutes. The minimum hardware requirements for the application of this simulation model is 128 KB memory with IBM PC Monochrome Adaptor and Display.

Summary Two load plans were generated to evaluate the simulation model developed in this study. Both the load plans were analyzed for the current PoP practice, rehandling for buffer and the sweeping criterion. The plans were evaluated on two factors: load performance and material handling efficiency.

As is evident from Figures 4.1 through 4.8, one of the factors that effects the loading operation is the structure of the load plan. Generally, the structure of the load plan -- number and type of containers to be transported -- is beyond the control of port management.

Of the controllable factors, the number of trucks appears to be the most significant factor. In general, the use of both rehandling for buffer and the sweeping criterion result in increased crane and transtainer utilization, and reduced loading time. However, these improvements are only significant if sufficient yard trucks are available to carry on operation without causing the transtainer to wait for trucks.

V. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

Conclusions

The primary objective of this study was to develop and evaluate strategies that can reduce the unproductive movements of material handling equipment during the actual loading process, and thus increase the efficiency of container port operations.

A methodology of utilizing a buffer area was proposed for improvement of the container loading operation. The proposed methodology was tested on two randomly developed load plans. The results from the two applications were analyzed. The following conclusions are presented as the primary results of this research:

1. The factors that affect the performance of proposed methodology are identified as the number of yard trucks, the number of empty chassis, sweeping criterion, and structure of the load plan.

2. The objective of the proposed methodology was to investigate procedures for minimizing bottlenecks resulting due to the transtainer operation. The use of buffer area always eliminates the unproductive movements of transtainer. However, the operation of yard trucks become a bottleneck in the loading operation when relatively small number of yard trucks are used.

3. The primary objective of rehandling for buffer is

to reduce the number of containers to be rehandled. The benefits resulting from rehandling for buffer depend on the number of yard trucks, the structure of each load plan, and the number of empty chassis. The structure of each load plan becomes more of a constraining factor when small number of yard trucks are used.

4. The primary objective of the sweeping operation is to reduce transtainer's non-adjacent section changes. In general, the sweeping operation accomplishes this objective. However, when the sweeping criterion is relatively high or the level of empty chassis is relatively low, the application of sweeping criterion may cause undesirable effects on the overall buffer operation by reducing the availability of empty chassis or by increasing transtainer idle time. This is especially true when relatively small number of yard trucks is used.

5. As the number of yard trucks is increased, the random effect due to the structure of load plan on the sweeping operation is decreased. As a result, the transtainer idle time during the buffer operation is decreased. However, the increase of truck level will not improve the use of buffer area beyond a certain level. At this truck level, the transtainer idle time due to buffer operation cannot be reduced any more.

6. The total number of transtainer row changes is always reduced through the use of buffer area. When the sweeping operation occurs, the non-adjacent section

changes of transtainer is always reduced. However, this may result in an increase in adjacent section changes.

7. The utilization of crane is significantly improved when the buffer area is properly used. However, the utilization of transtainer is determined by other factors, such as transtainer's idle time during the buffer operation and the total number of containers processed through the buffer area. The unproductive movements of transtainer are eliminated and the total transtainer operation time is directly reduced, as more containers are processed through sweeping operations or rehandling for buffer. If enough number of trucks are used, both the total transtainer operation time and the total loading time will be reduced simultaneously, resulting in a proportional increase in utilization of the transtainer.

To summarize, even though some of the parameters specifications were based on PoP's current practice, the model is general enough such that its conclusions are applicable to any container handling port operations. The results obtained from the model show that the proper use of the buffer area during the container loading operation will provide significant improvement in the crane and transtainer utilization, and substantial reduction in the total container loading time.

Recommendations for Future Research

Extension of this study leads into several areas, some of which are extension of this research while others go considerably beyond it. Some of the proposed extensions are:

1. The simulation program, YARD, developed in this study can be extended to handle statistical distributions. This may require enhancement of PCModel, or interacting PCModel with an external environment built to handle random variables.

2. Further evaluation of the methodology proposed in this study by testing other load plans on the simulation model.

3. When load planning begins, it is assumed that all containers for the voyage have arrived and their locations are known. Current practice is to arrange the containers according to voyage, destination, type, and weight, if possible. When a bay is filled, the heavy containers are likely to be placed on the bottom of the bay in order to maintain good stability. However, these containers will be loaded onto the ship earlier than the light ones, again because of stability reasons. Thus, an alternative to the current strategy is to set weight ranges for heavy, medium, and light containers, and assign the location of each outbound container according to this weight range, so that the heavier containers can be placed nearer the truck

lane. Such a strategy can then be evaluated using a simulation model as the one developed in this study.

4. Finally, the possibility of modeling the load planning problem using artificial intelligence should be investigated. Such an approach ideally would include ways to look ahead in the load planning process to find improved solutions.

BIBLIOGRAPHY

Beliech, D.E., Jr., A Proposed Method for Efficient Pre-Load Planning for Containerized Cargo Ships, Master's Thesis for Naval Post-graduate School, Monterey, California, June, 1974.

Cho, D.W., Development of a Methodology for Containership Load Planning, Ph.D. Thesis, Oregon State University, Corvallis, July, 1981.

Cojeen, P.H., Van Dyke, P., "The Automated Planning and Sequencing of Containers for Containership Loading and Unloading", Ship Operation Automation, eds. Pitkin, Roche, and Williams, North-Holland Publishing Co., 1976.

Gordon, G., System Simulation, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

Martin, G.L.,Jr., A Containership Load Planning Heuristic for a Transtainer Based Container Port, Master's Thesis, Oregon State University, Corvallis, Oregon, August, 1981.

McDowell, E.M., Martin, G.L., Cho, D.W., West, T.M., A Study of Maritime Container Handling, Sea Grant College Program, AdS 402, Oregon State University, Corvallis, Oregon, 1985.

"Port of Portland Technical Seminar", Port of Portland, May, 1979.

Scott, D.K., Chen, D.S., A Loading Model for A Container Ship, Matson Navigation Company and University of Alabama, November, 1978.

White, D.A, Personal Computer Screen Graphics Modeling System, User's Guide, Simulation Software Systems, California, 1985.

APPENDICES

APPENDIX A

PCModel

1. Application of PCModel is considered to be best suitable for a small to medium size model, especially for an assembly line operation. When the model to be studied is very big, some difficulties are encountered with PCModel's memory limits.

The first limit is the text editor limit. The text editor built in PCModel is used to create a MDL file, which is the program file of the simulation. The maximum size of a MDL file that the text editor can create is 64K bytes.

The second limit of PCModel is 64K byte capacity for internal presentation. The MDL file written in PCModel is converted to assembler language for internal presentation during a loading process. This limit is imposed on the internal memory space that the converted MDL file requires. However, when a MDL file is converted to assembler language, the memory space required by the converted MDL file is reduced to some extent depending on the contents of the MDL file. From experience of this research, approximately 240K bytes is considered as the maximum MDL file size that can fit into 64K byte limit for internal presentation.

The only problem then in creating a 240K byte long MDL file is to overcome the limit of the text editor, which is 64K bytes. This problem can be resolved by creating four seperate small files with the text editor, each of which is about 64K byte long, and combining these four files as one big final file using a DOS command, COPY. In actual application, a MDL file of 240K bytes will cover most of the general applications.

2. There are two types of arrays in PCModel. These are value type array and clock value type array. One inconvenience involved with the use of arrays is that an array value must be stored into @ or % type variable every time its value is to be used in other instructions.

Suppose that we want to multiply the value of array, X(1,2), by 10 and store the result in a variable A. As shown below, programming becomes very long due to the specific array execution and storage in PCModel:

FORTRAN

```
A = X(1,2) * 10
```

PCMODEL

```
SV(@A,@@X(1,2))        ; STORE VALUE OF X(1,2) IN @A
AO(@A,*,10)            ; @A = @A X 10
```

Another computational shortcoming of PCModel is that the values of variables should always be integers. This implies that values are always rounded up, and consequently, the accuracy of the simulation is degraded.

3. The nature of graphic simulation requires many location variables to be used in the program. As

mentioned previously, array can be defined for value type and clock value type. Another problem related with array is that arrays cannot be used for location offsets, particularly, when many location offsets are repeatedly used in the program. This shortcoming of PCModel makes the program very long and tedious.

4. Simulation study frequently involves probability distributions. Other simulation systems, like SLAM or GASPIV has these distributions built-in the system. PCModel provides only one built-in distribution, the uniform distribution. Other important distributions can be programmed as a subroutine inside the program, but programming becomes very long and inefficient. This is considered to be the biggest shortcoming of PCModel as a simulation language.

5. Simulating speed is one of the user's concerns about a simulation language. The simulation speed of PCModel is about 500 times faster than real time. However, the pace of simulation gets slowed down in some specific situation and becomes a very significant problem. This occurs when there are more than 150 objects in the system and their movements on the screen are stopped by IF statements simultaneously.

APPENDIX B

Arrays / Variables / Location Offsets Definitions

Arrays

1. STATUS1 = (7,34)

		COLUMN j
	0	container type, 20ft or 40ft.
R	1	number of container in 1st stack of jth row
O	2	.. 2nd ..
W	3	.. 3rd ..
	4	.. 4th ..
i	5	.. 5th ..
	6	.. 6th ..
	7	total number of containers in jth row

@@STATUS2, @@STATUS3, @@STATUS4, and @@STATUS5 have the same structure as @@STATUS1.

2. @@CLIST = (7,204)

Description about structure of @@CLIST is made in Figure 3.2.

3. @@TTJOB = (4,500)

		COLUMN j
R	0	transtainer operation mode
O	1	section number for jth transtainer operation
W	2	row number .. .. ..
	3	stack number .. .. ..
i	4	index of @@PLAN for jth container of @@TTJOB

4. @@PLAN = (13,500)

		COLUMN j
	0	number of containers to be rehandled for jth con.
R	1	section number of jth container
O	2	row number .. ..
W	3	stack number ..
	4	level of .. ..
i	5	
	6	truck arrival at jth container location
	7	transtainer completes loading jth container
	8	
	9	departure of jth container for buffer area
	10	arrival of jth container at buffer space
	11	direct shipping signnal w/o discharge in buffer
	12	
	13	whether or not jth container is included in @@RHD

5. @@RHD = (30,100)

		COLUMN j
	0	
R	1	job number of 1st cont. for jth buffer operation
O	2	.. 2nd .. ..
W	3	.. 3rd .. ..
	4	.. 4th .. ..
i		
	100	.. 100th .. ..

6. @@BUFFER = (1,34)

		COLUMN j
R	0	
O	1	status of the jth space of buffer area
W		
i		

7. @@BFJOB = (1,100)

		COLUMN j
R	0	
O	1	job number of a container to be checked for on time removing from buffer
W		
i		

8. @@ROWTOT = (6,34)

		COLUMN j
	0	
R	1	tot. number of containers in jth row of section 1
O	2	.. .. section 2
W	3	.. .. section 3
	4	.. .. section 4
i	5	.. .. section 5
	6	

Variables

Most of the variables used in YARD program are dummy variables for computations. The following list consists of the more important variables :

Variable	Description
@AS	total number of containers scheduled in a loading plan
@PADCHG	P(an adjacent section change per container)
@PNADCHG	P(a nonadjacent section change per cont.)
@PRCHG	P(a row changing event per container)
@PRHD	P(a rehandling event per container)
@RANDOM	random number
@SWITCH	flag variable for generation of output file
@S1TOT	total nuber of containers in section 1
@S2TOT	total nuber of containers in section 2
@S3TOT	total nuber of containers in section 3
@S4TOT	total nuber of containers in section 4
@S5TOT	total nuber of containers in section 5
@SECTOT	dummy variable for section totale
@RTOT	dummy variable for row totale
@STOT	dummy variable for stack total
@SEC	dummy variable for transtainer section
@ROW	row number (1 - 34)
@STK	stack number (1 - 6)
@LVL	level number (1 - 3)
@TRMOVV	Y axis movingg speed of a yard truck
@TRMOV	X axis moving speed of a yard truck
@TTMOV	Y axis moving speed of transtainer
@TTMRL	X axis moving speed of transtainer
@TTSEC	section of current transtainer operation
@TTROW	row of current transtainer operation
@NRHD	column index of @@RHD
@TROWCHG	total number of transtainer row changes
@TADCHG	total number of transtainer adjacent section change
@TNADCHG	total number of transtainer non adjacent section change
@RHDCNT	total number of containers processed for regular rehandlings
@ORDER	job number of the last container that passed the checking point
@CNT1	job index of transtainer job list, @@TTJOB
@CNT3	job index of load plan for truck, @@PLAN
@CNT4	current total of containers loaded onto ship
@DCNT	dummy index for transfering informations to @@TTJOB

There are six variables automatically assigned to each moving object by the PCMODEL. They are OBJ@1, OBJ@2, OBJ@3, OBJ@4, OBJ@5, and OBJ@6. These variables are used to store informations directly related with the object, such as location of container, job number of the container, etc.

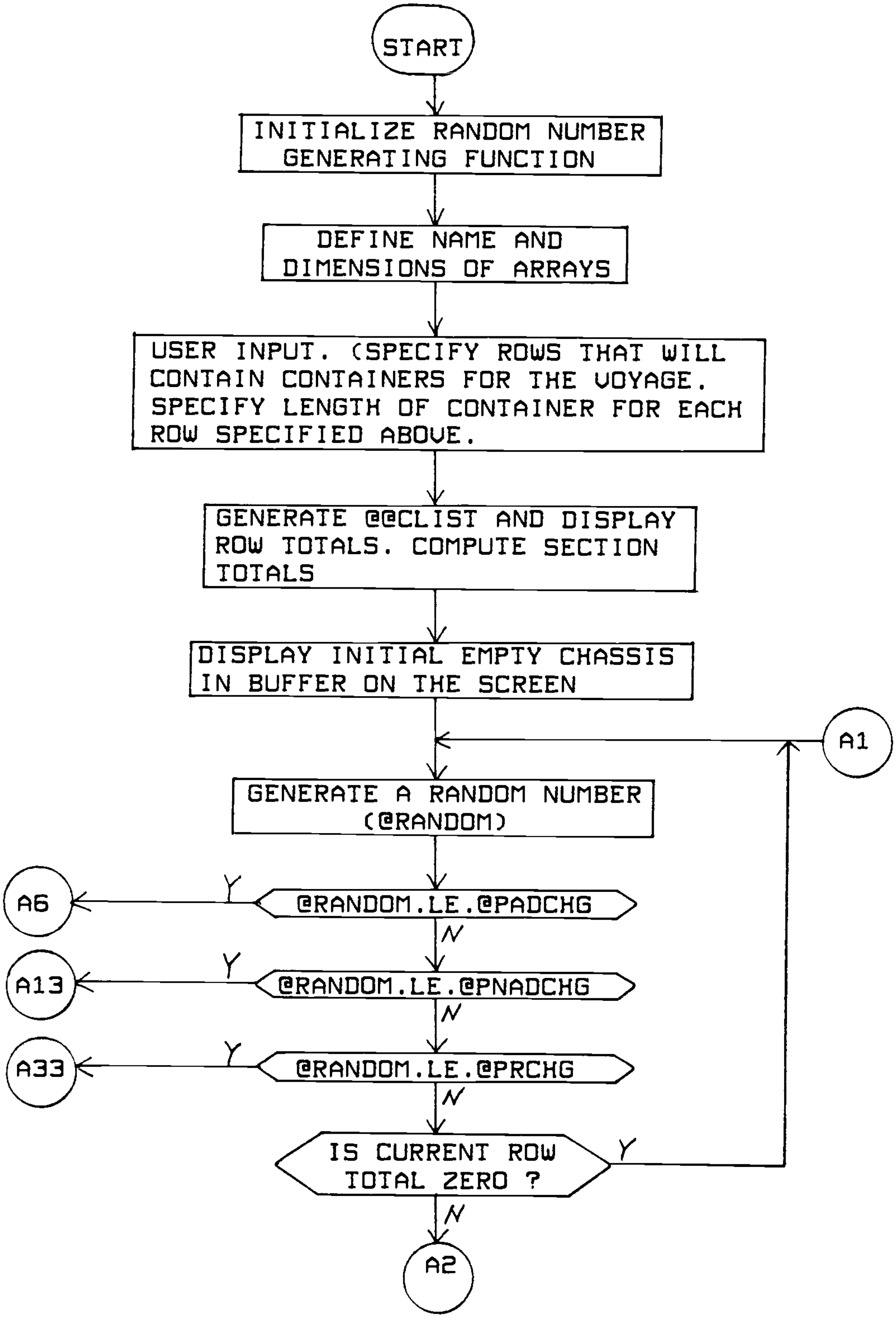
Location Offsets

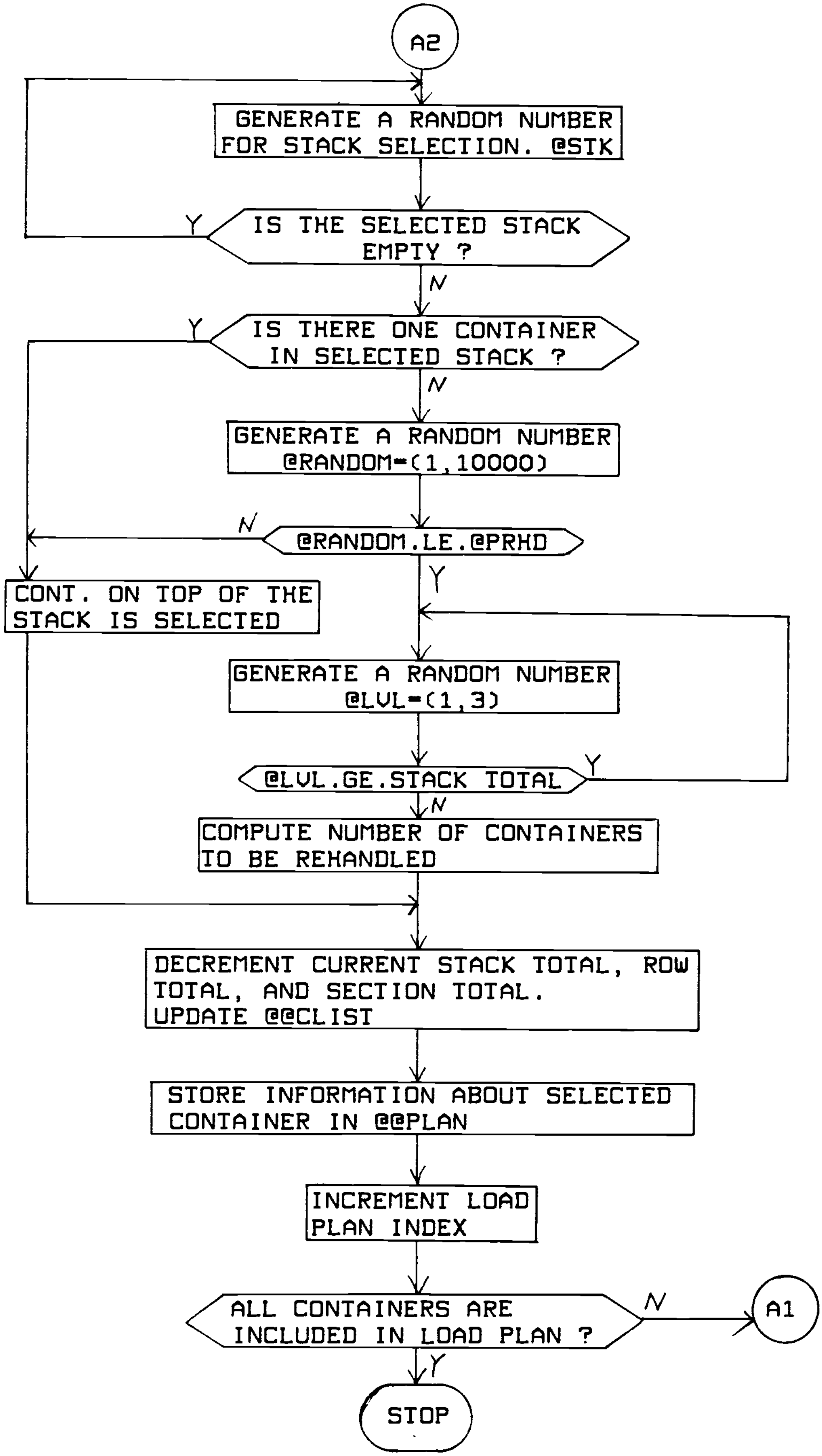
Name	Descriptions
*N11 *N534	*N11 is a location offset used to display the current number of containers in 1st row of section 1. Likewise, *Nij is for jth row of ith section.
*BF1 *BF34	*BF1 is a location offset used to display the status of 1st cell of buffer spaces. In general, *BFn is used for nth cell of buffer spaces.
*NUMINIT	starting location for routing 1
*TTINIT	starting location for transtainer routing
*YTRINIT	starting location for yard trucks routing
*DATA	starting location for routing 4
*CRANE	location for collection of crane utilization
*TTUT	.. .. transtainer utilization
*RHD	signal for sending containers to buffer area
*MSG1	location to display the total number of containers scheduled for shipping
*MSG2	location to display current total number of containers loaded onto ship
*MSG3	location to display current total numbers of containers processed for regular rehandles
*MSG4	location to display @TROWCHG
*MSG5	location to display @TADCHG
*MSG6	location to display @TNADCHG

APPENDIX C

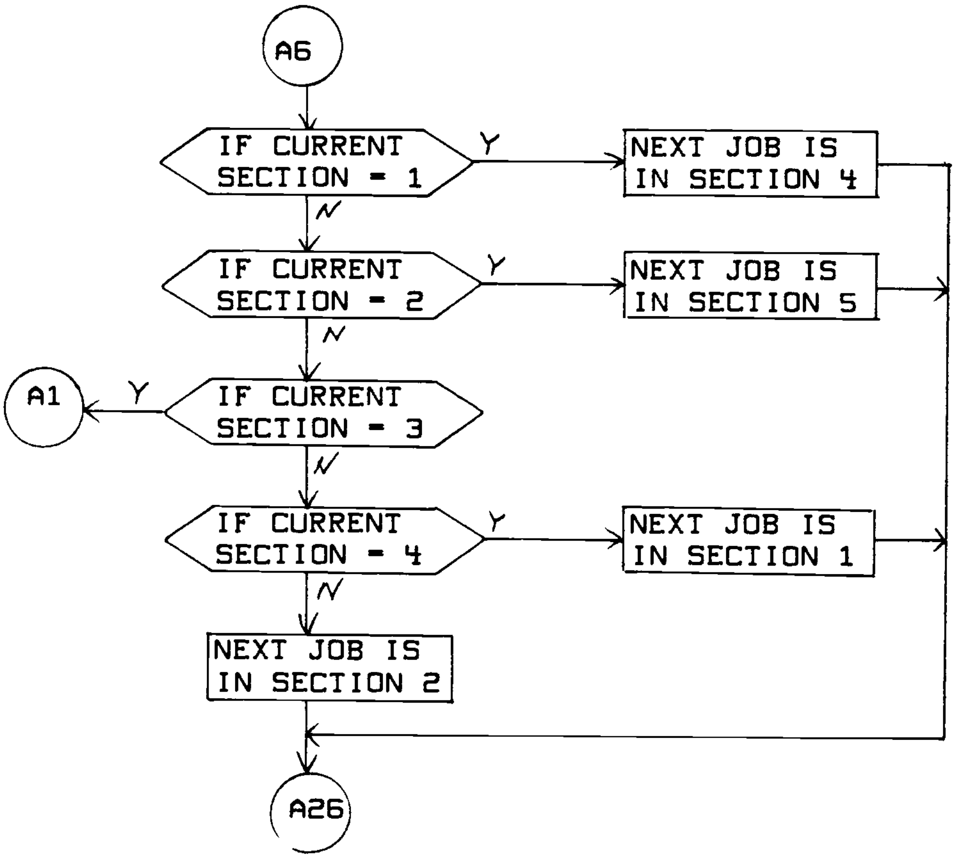
Flow Charts of Program YARD

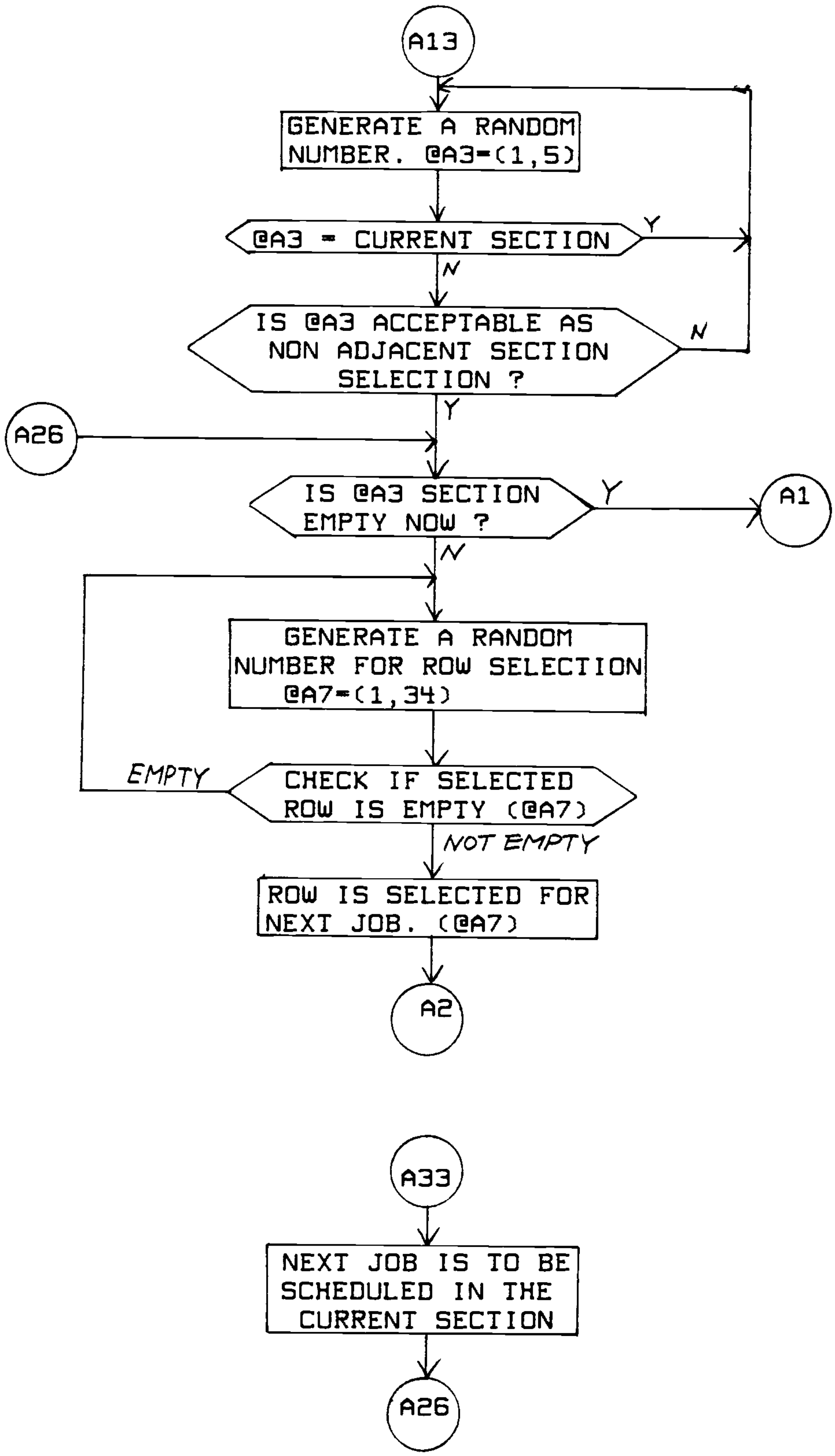
ROUTING 1 - GENERATION OF YARD DATA AND LOAD PLAN





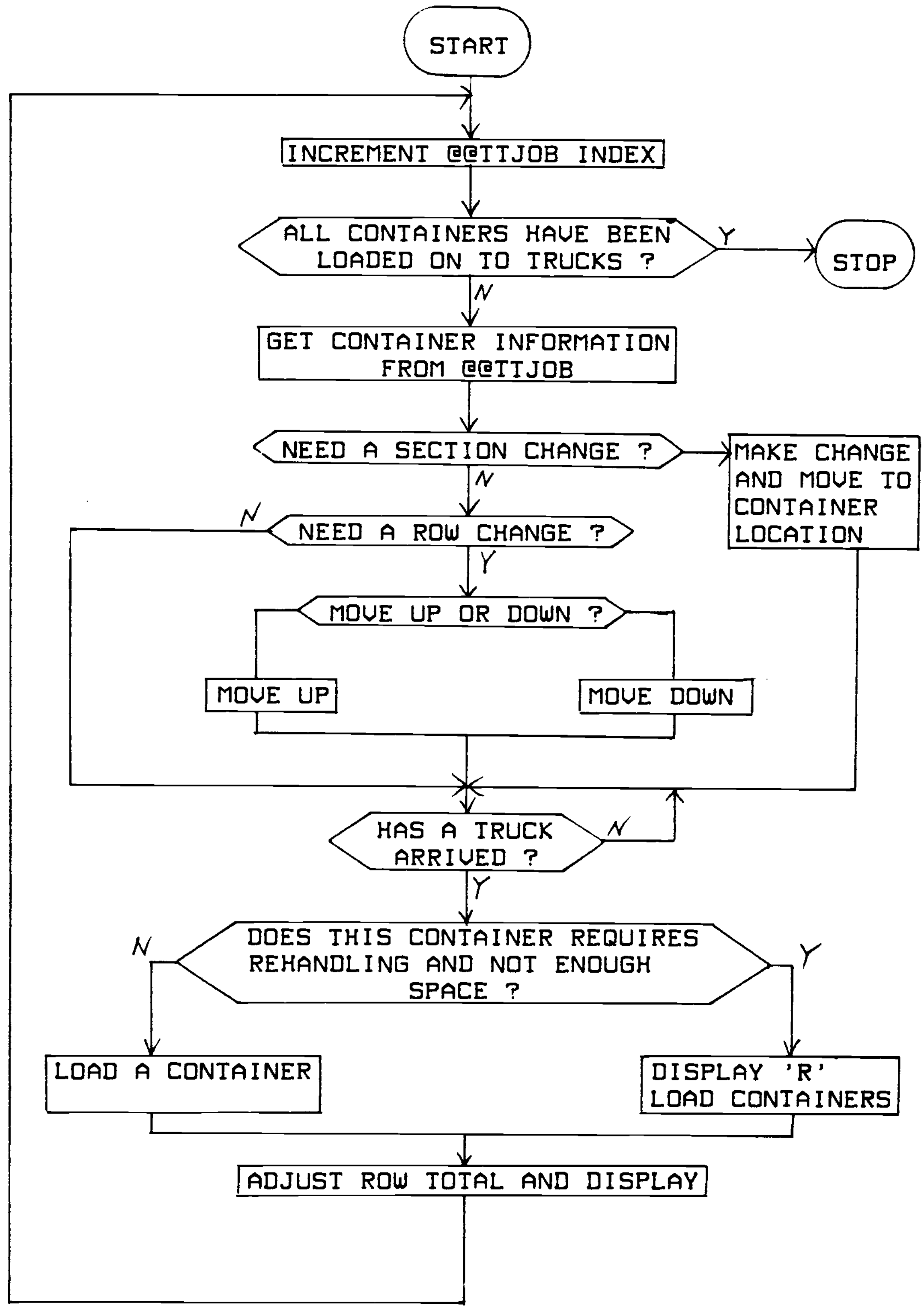
A2
GENERATE A RANDOM NUMBER FOR STACK SELECTION. @STK
IS THE SELECTED STACK EMPTY ?
Y
N
IS THERE ONE CONTAINER IN SELECTED STACK ?
Y
N
GENERATE A RANDOM NUMBER @RANDOM=(1,10000)
N
@RANDOM.LE.@PRHD
Y
CONT. ON TOP OF THE STACK IS SELECTED
GENERATE A RANDOM NUMBER @LVL=(1,3)
@LVL.GE.STACK TOTAL
Y
N
COMPUTE NUMBER OF CONTAINERS TO BE REHANDLED
DECREMENT CURRENT STACK TOTAL, ROW TOTAL, AND SECTION TOTAL. UPDATE @@CLIST
STORE INFORMATION ABOUT SELECTED CONTAINER IN @@PLAN
INCREMENT LOAD PLAN INDEX
ALL CONTAINERS ARE INCLUDED IN LOAD PLAN ?
N
A1
Y
STOP


A6
IF CURRENT SECTION = 1
Y
NEXT JOB IS IN SECTION 4
N
IF CURRENT SECTION = 2
Y
NEXT JOB IS IN SECTION 5
N
A1
Y
IF CURRENT SECTION = 3
N
IF CURRENT SECTION = 4
Y
NEXT JOB IS IN SECTION 1
N
NEXT JOB IS IN SECTION 2
A26

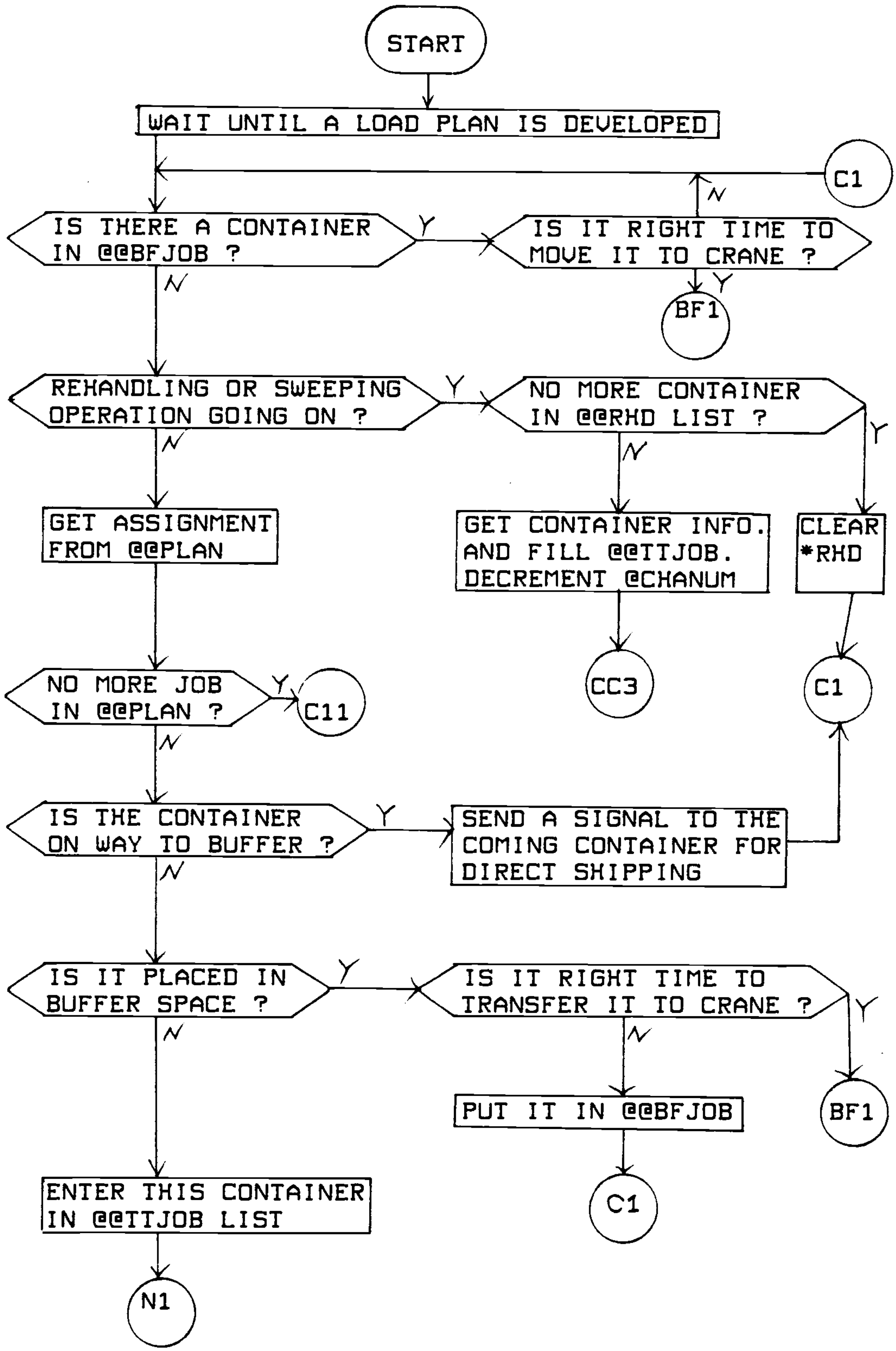


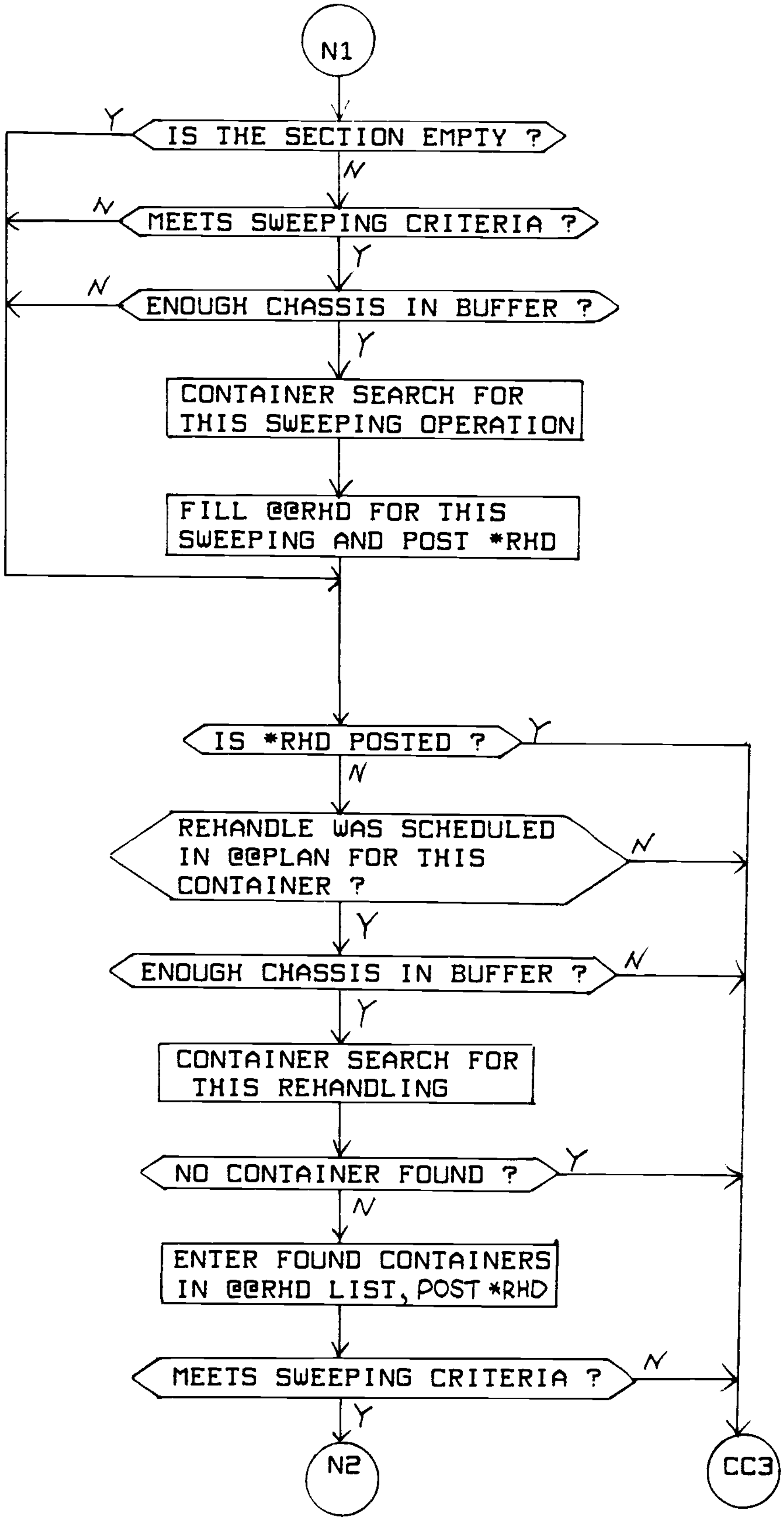
A13
GENERATE A RANDOM NUMBER. @A3=(1,5)
@A3 = CURRENT SECTION
Y
N
IS @A3 ACCEPTABLE AS NON ADJACENT SECTION SELECTION ?
N
Y
A26
IS @A3 SECTION EMPTY NOW ?
Y
A1
N
GENERATE A RANDOM NUMBER FOR ROW SELECTION @A7=(1,34)
EMPTY
CHECK IF SELECTED ROW IS EMPTY (@A7)
NOT EMPTY
ROW IS SELECTED FOR NEXT JOB. (@A7)
A2
A33
NEXT JOB IS TO BE SCHEDULED IN THE CURRENT SECTION
A26

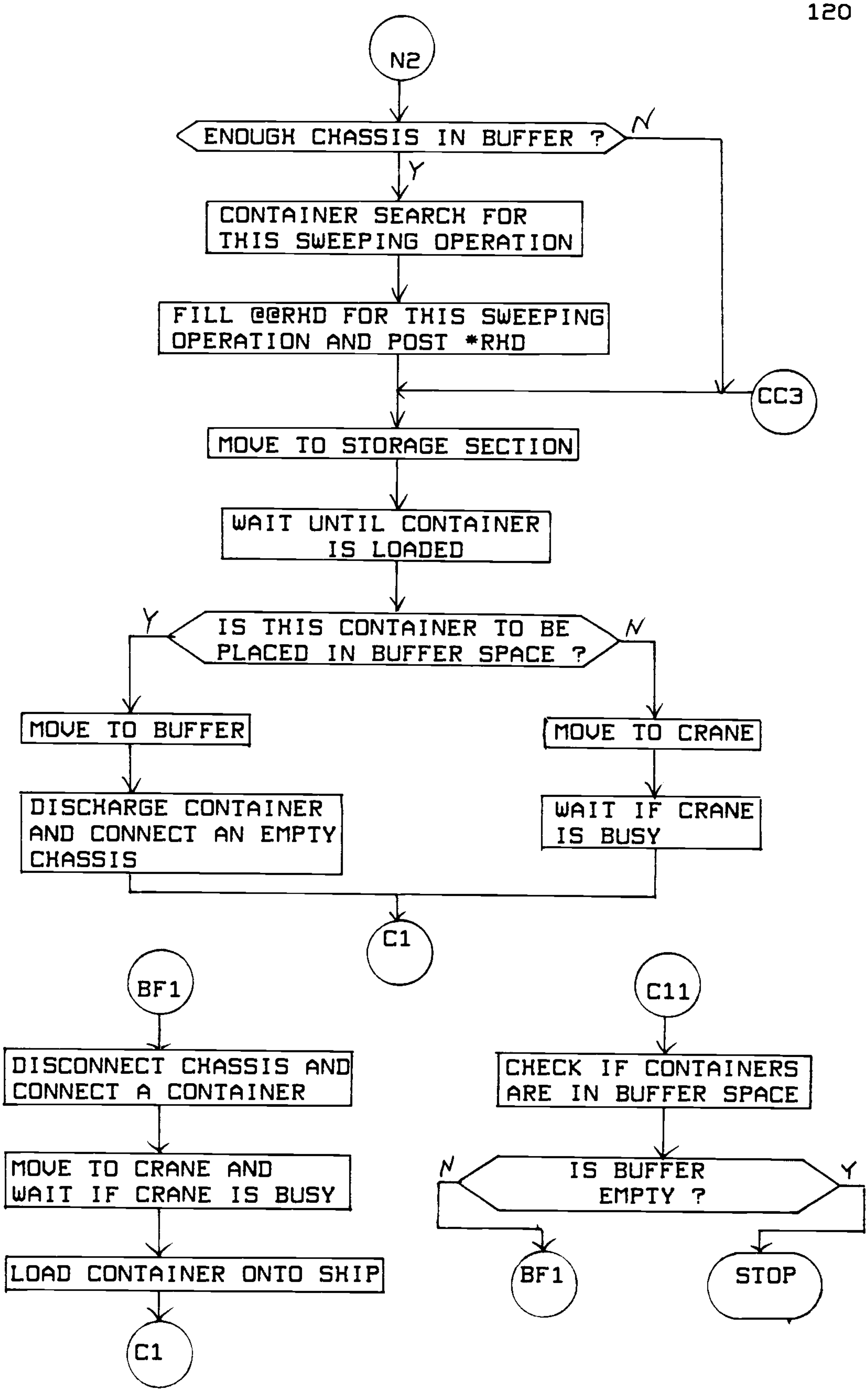
ROUTING 2 - TRANSTAINER OPERATION



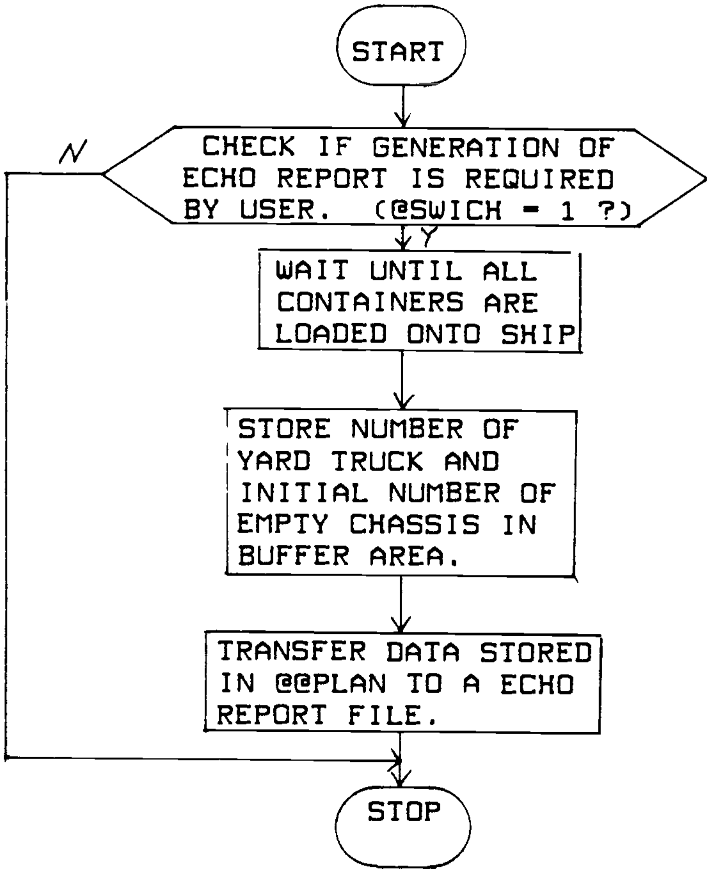
ROUTING 3 - YARD TRUCK OPERATION




N1
IS THE SECTION EMPTY ?
Y
N
MEETS SWEEPING CRITERIA ?
N
Y
ENOUGH CHASSIS IN BUFFER ?
N
Y
CONTAINER SEARCH FOR THIS SWEEPING OPERATION
FILL @@RHD FOR THIS SWEEPING AND POST *RHD
IS *RHD POSTED ?
Y
N
REHANDLE WAS SCHEDULED IN @@PLAN FOR THIS CONTAINER ?
N
Y
ENOUGH CHASSIS IN BUFFER ?
N
Y
CONTAINER SEARCH FOR THIS REHANDLING
NO CONTAINER FOUND ?
Y
N
ENTER FOUND CONTAINERS IN @@RHD LIST, POST *RHD
MEETS SWEEPING CRITERIA ?
N
Y
N2
CC3


N2
ENOUGH CHASSIS IN BUFFER ?
N
Y
CONTAINER SEARCH FOR THIS SWEEPING OPERATION
FILL @@RHD FOR THIS SWEEPING OPERATION AND POST *RHD
CC3
MOVE TO STORAGE SECTION
WAIT UNTIL CONTAINER IS LOADED
Y
IS THIS CONTAINER TO BE PLACED IN BUFFER SPACE ?
N
MOVE TO BUFFER
DISCHARGE CONTAINER AND CONNECT AN EMPTY CHASSIS
MOVE TO CRANE
WAIT IF CRANE IS BUSY
C1
BF1
DISCONNECT CHASSIS AND CONNECT A CONTAINER
MOVE TO CRANE AND WAIT IF CRANE IS BUSY
LOAD CONTAINER ONTO SHIP
C1
C11
CHECK IF CONTAINERS ARE IN BUFFER SPACE
N
IS BUFFER EMPTY ?
Y
BF1
STOP

ROUTING 4 - GENERATION OF ECHO REPORT



APPENDIX D

YARD Program Listing

```
;
;******************** SYSTEM CONFIGURATION AND INITIALIZATION *****************
;
M=(3000)            ; RESERVE SPACE FOR 3000 MOVEMENT CONTROL BLOCKS
S=(60000)           ; NUMBER OF MAXIMUM SYMBOLS
W=(3000)            ; MAXIMUM WORK IN PROGRESS
X=(150)             ; X DIMENSION OF OVERLAY
Y=(200)             ; Y DIMENSION OF OVERLAY
V=(XY(2,2))         ; VIEWING WINDOW INITIAL SETING AT UPPER LEFT CONER
;
;
;
;******************** DESCRIPTION OF THE MODEL ********************************
;
D=
        Description of this simulation model is provided in thesis.
        $
;
;
;******************** OVERLAY FILE DESIGNATIOON *******************************
;
O=(B:YARD.OLY)      ; OVERLAY FILE IS STORED IN YARD.OLY
;
;******************** RANDOM NUMBER SEED **************************************
;
#SEED=(9135)
;
;******************** VARIABLE DEFINITIONS ************************************
;
;                   ; NAMES OF VERIABLES AND THEIR INITIAL VALUES ARE
;                     SPECIFIED
;
@A1=(0)          @A2=(0)          @A3=(1)          @A4=(0)
@A5=(0)          @A6=(0)          @A7=(0)          @A8=(0)
@A9=(0)
;
@B1=(0)          @B2=(0)          @B3=(1)          @B4=(0)
@B5=(0)          @B6=(0)
@B9=(0)          @B10=(0)         @B11=(0)
;
@C1=(0)          @C2=(0)          @C3=(0)          @C4=(0)
@C5=(0)          @C6=(0)          @C7=(0)          @CCC=(0)
;
@K1=(0)          @K2=(0)          @K3=(0)          @K4=(0)
@CC1=(0)         @CC2=(0)         @CC3=(1)         @CC4=(0)
;
@N1=(0)          @N3=(0)          @N4=(0)
@N5=(0)          @N6=(0)          @N7=(0)
;
@S1=(0)          @S2=(0)          @S3=(0)          @S4=(0)
@S5=(0)          @S7=(0)
;
@D1=(0)          @D2=(0)          @D3=(0)          @D4=(0)
@D5=(0)          @D7=(0)
```

```
;
@Z1=(0)             @Z2=(0)             @Z3=(0)             @Z4=(0)
@Z5=(0)             @RRR=(0)            @R4=(0)

@KK1=(0)            @KK2=(0)            @KK3=(0)            @KK4=(0)
@CNT1=(1)           @CNT2=(1)           @CNT3=(1)           @CNT4=(0)
@DCNT=(0)           @DMCNT=(0)          @RHDCNT=(0)
;
@S1TOT=(0)          @S2TOT=(0)          @S3TOT=(0)          @S4TOT=(0)
@S5TOT=(0)          @RW=(0)             @SEC=(1)
;
@F1=(450)           @F2=(400)           @F3=(4500)          @F4=(510)
@PNADCHG=(0)        @PADCHG=(0)         @PRCHG=(0)          @PRHD=(0)
;
@RTOT=(0)           @STOT=(0)           @STK=(0)            @ROW=(0)
@RHD=(0)            @RANDOM=(0)         @LVL=(0)            @TRMOVV=(2)
@TTROW=(1)          @TTMOV=(5)          @TRMOV=(1)          @TTMRL=(10)
@TTSEC=(1)          @SWITCH=(0)         @START=(0)
@NRHD=(1)           @SECTOT=(0)
@TROWCHG=(0)        @TADCHG=(0)         @TNADCHG=(0)        @ORDER=(0)
;
@SWEEP=(0)
@CHANUM=(0)
;
;******************* LOCATION DEFINITIONS **************************************
;
;                   ; DEFINE LOCATION OFFSETS WITH NAMES AND COORDINATES
;
;                   ; THESE LOCATION OFFSETS ARE USED TO DISPLAY THE TOTAL
                    ; NUMBER OF CONTAINERS IN EACH ROWS
;
*N11=(XY(42,5))     *N21=(XY(56,5))     *N41=(XY(42,43))    *N51=(XY(56,43))
*N12=(XY(42,6))     *N22=(XY(56,6))     *N42=(XY(42,44))    *N52=(XY(56,44))
*N13=(XY(42,7))     *N23=(XY(56,7))     *N43=(XY(42,45))    *N53=(XY(56,45))
*N14=(XY(42,8))     *N24=(XY(56,8))     *N44=(XY(42,46))    *N54=(XY(56,46))
*N15=(XY(42,9))     *N25=(XY(56,9))     *N45=(XY(42,47))    *N55=(XY(56,47))
*N16=(XY(42,10))    *N26=(XY(56,10))    *N46=(XY(42,48))    *N56=(XY(56,48))
*N17=(XY(42,11))    *N27=(XY(56,11))    *N47=(XY(42,49))    *N57=(XY(56,49))
*N18=(XY(42,12))    *N28=(XY(56,12))    *N48=(XY(42,50))    *N58=(XY(56,50))
*N19=(XY(42,13))    *N29=(XY(56,13))    *N49=(XY(42,51))    *N59=(XY(56,51))
*N110=(XY(42,14))   *N210=(XY(56,14))   *N410=(XY(42,52))   *N510=(XY(56,52))
*N111=(XY(42,15))   *N211=(XY(56,15))   *N411=(XY(42,53))   *N511=(XY(56,53))
*N112=(XY(42,16))   *N212=(XY(56,16))   *N412=(XY(42,54))   *N512=(XY(56,54))
*N113=(XY(42,17))   *N213=(XY(56,17))   *N413=(XY(42,55))   *N513=(XY(56,55))
*N114=(XY(42,18))   *N214=(XY(56,18))   *N414=(XY(42,56))   *N514=(XY(56,56))
*N115=(XY(42,19))   *N215=(XY(56,19))   *N415=(XY(42,57))   *N515=(XY(56,57))
*N116=(XY(42,20))   *N216=(XY(56,20))   *N416=(XY(42,58))   *N516=(XY(56,58))
*N117=(XY(42,21))   *N217=(XY(56,21))   *N417=(XY(42,59))   *N517=(XY(56,59))
*N118=(XY(42,22))   *N218=(XY(56,22))   *N418=(XY(42,60))   *N518=(XY(56,60))
*N119=(XY(42,23))   *N219=(XY(56,23))   *N419=(XY(42,61))   *N519=(XY(56,61))
*N120=(XY(42,24))   *N220=(XY(56,24))   *N420=(XY(42,62))   *N520=(XY(56,62))
*N121=(XY(42,25))   *N221=(XY(56,25))   *N421=(XY(42,63))   *N521=(XY(56,63))
*N122=(XY(42,26))   *N222=(XY(56,26))   *N422=(XY(42,64))   *N522=(XY(56,64))
*N123=(XY(42,27))   *N223=(XY(56,27))   *N423=(XY(42,65))   *N523=(XY(56,65))
```

```
*N124=(XY(42,28))  *N224=(XY(56,28))  *N424=(XY(42,66))  *N524=(XY(56,66))
*N125=(XY(42,29))  *N225=(XY(56,29))  *N425=(XY(42,67))  *N525=(XY(56,67))
*N126=(XY(42,30))  *N226=(XY(56,30))  *N426=(XY(42,68))  *N526=(XY(56,68))
*N127=(XY(42,31))  *N227=(XY(56,31))  *N427=(XY(42,69))  *N527=(XY(56,69))
*N128=(XY(42,32))  *N228=(XY(56,32))  *N428=(XY(42,70))  *N528=(XY(56,70))
*N129=(XY(42,33))  *N229=(XY(56,33))  *N429=(XY(42,71))  *N529=(XY(56,71))
*N130=(XY(42,34))  *N230=(XY(56,34))  *N430=(XY(42,72))  *N530=(XY(56,72))
*N131=(XY(42,35))  *N231=(XY(56,35))  *N431=(XY(42,73))  *N531=(XY(56,73))
*N132=(XY(42,36))  *N232=(XY(56,36))  *N432=(XY(42,74))  *N532=(XY(56,74))
*N133=(XY(42,37))  *N233=(XY(56,37))  *N433=(XY(42,75))  *N533=(XY(56,75))
*N134=(XY(42,38))  *N234=(XY(56,38))  *N434=(XY(42,76))  *N534=(XY(56,76))
;
*N331=(XY(70,35))  *N332=(XY(70,36))  *N333=(XY(70,37))  *N334=(XY(70,38))
;
;
;                  ; THESE LOCATION OFFSETS ARE USED TO DISPLAY AN
;                  ; EMPTY CHASSIS OR A CONTAINER IN EACH BUFFER CELL
;
*BF1=(XY(28,5))    *BF2=(XY(28,6))    *BF3=(XY(28,7))    *BF4=(XY(28,8))
*BF5=(XY(28,9))    *BF6=(XY(28,10))   *BF7=(XY(28,11))   *BF8=(XY(28,12))
*BF9=(XY(28,13))   *BF10=(XY(28,14))  *BF11=(XY(28,15))  *BF12=(XY(28,16))
*BF13=(XY(28,17))  *BF14=(XY(28,18))  *BF15=(XY(28,19))
*BF16=(XY(28,20))  *BF17=(XY(28,21))  *BF18=(XY(28,22))  *BF19=(XY(28,23))
*BF20=(XY(28,24))  *BF21=(XY(28,25))  *BF22=(XY(28,26))  *BF23=(XY(28,27))
*BF24=(XY(28,28))  *BF25=(XY(28,29))  *BF26=(XY(28,30))  *BF27=(XY(28,31))
*BF28=(XY(28,32))  *BF29=(XY(28,33))  *BF30=(XY(28,34))
*BF31=(XY(28,35))  *BF32=(XY(28,36))  *BF33=(XY(28,37))  *BF34=(XY(28,38))
;
;                  ; THESE ARE DUMMY LOCATIONS FOR INTERNAL USE OR FOR
                   ; DISPLAY OF MESSAGES
;
*NUMINIT=(XY(2,90)) *CRANE=(XY(18,22)) *RHD=(XY(2,93))   *LRHD=(XY(4,93))
*MSG1=(XY(11,21))  *EXIT=(XY(2,91))   *MSG2=(XY(11,22))  *MSG3=(XY(11,23))
*MSG4=(XY(11,24))  *MSG5=(XY(11,25))  *MSG6=(XY(11,26))  *MSG7=(XY(11,17))
*TTINIT=(XY(35,5)) *YTRINIT=(XY(26,38))
*TTUT=(XY(5,93))   *DATA=(XY(7,93))   *FINAL=(XY(9,92))
;
;
;
;
;
;******************* JOB DESCRIPTIONS ******************************************
;
J=(1,I,1,0,0,0,1)   ; THIS JOB IS FOR GENERATION OF YARD DATA AND A LOAD PLAN
J=(2,T,2,0,0,1,1)   ; THIS JOB IS FOR TRANSTAINER OPERATION
J=(3,*,3,0,0,1,2)   ; THIS JOB IS FOR TRUCK OPERATION
J=(4,D,4,0,0,1,1)   ; THIS JOB IS FOR GENERATION OF OUTPUT FILE
;
;
;
;******************* UTILIZATION DEFINITIONS ***********************************
;
U=(1,CRANE,*CRANE)  ; COLLECT UTILIZATION OF CRANE AT *CRANE
U=(2,TRANS,*TTUT)   ; COLLECT UTILIZATION OF TRANSTAINER AT *TTUT
```

```
;
;******************* ROUTING 1 ***********************************************
;
;                  ; THIS ROUTING IS FOR GENERATION OF YARD DATA AND LOAD PLAN
;
BR(1,#NUMINIT,0)
  RS(#SEED)        ; INITIALLIZE RANDOM NUMBER GENERATION FUNCTION
  SV(@CNT1,0)
;
  @@STATUS1=(7,34)  ; ARRAY DEFINITIONS
  @@STATUS2=(7,34)
  @@STATUS3=(7,34)
  @@STATUS4=(7,34)
  @@STATUS5=(7,34)
  @@STATUS6=(7,34)
  @@CLIST=(7,204)
  @@PLAN=(16,500)
  @@TTJOB=(5,500)
  @@RHD=(30,100)
  @@BUFFER=(1,34)
  @@BFJOB=(1,100)
  @@ROWTOT=(6,34)
;
;******************* THIS PORTION IS REINITIALIZATION OF VARIABLES SO THAT
;                    MULTIPLE RUNS CAN BE MADE WITH ONE LOADING PROCESS
;
:AB1  IV(@CNT1)
  SV(@@BUFFER(0,@CNT1),0)  SV(@@BUFFER(1,@CNT1),0)
;
  SV(@@ROWTOT(0,@CNT1),0)  SV(@@ROWTOT(1,@CNT1),0)
  SV(@@ROWTOT(2,@CNT1),0)  SV(@@ROWTOT(3,@CNT1),0)
  SV(@@ROWTOT(4,@CNT1),0)  SV(@@ROWTOT(5,@CNT1),0)
  SV(@@ROWTOT(6,@CNT1),0)
;
  SV(@@STATUS1(0,@CNT1),0)  SV(@@STATUS2(0,@CNT1),0)
  SV(@@STATUS1(1,@CNT1),0)  SV(@@STATUS2(1,@CNT1),0)
  SV(@@STATUS1(2,@CNT1),0)  SV(@@STATUS2(2,@CNT1),0)
  SV(@@STATUS1(3,@CNT1),0)  SV(@@STATUS2(3,@CNT1),0)
  SV(@@STATUS1(4,@CNT1),0)  SV(@@STATUS2(4,@CNT1),0)
  SV(@@STATUS1(5,@CNT1),0)  SV(@@STATUS2(5,@CNT1),0)
  SV(@@STATUS1(6,@CNT1),0)  SV(@@STATUS2(6,@CNT1),0)
  SV(@@STATUS1(7,@CNT1),0)  SV(@@STATUS2(7,@CNT1),0)
;
  SV(@@STATUS3(0,@CNT1),0)  SV(@@STATUS4(0,@CNT1),0)
  SV(@@STATUS3(1,@CNT1),0)  SV(@@STATUS4(1,@CNT1),0)
  SV(@@STATUS3(2,@CNT1),0)  SV(@@STATUS4(2,@CNT1),0)
  SV(@@STATUS3(3,@CNT1),0)  SV(@@STATUS4(3,@CNT1),0)
  SV(@@STATUS3(4,@CNT1),0)  SV(@@STATUS4(4,@CNT1),0)
  SV(@@STATUS3(5,@CNT1),0)  SV(@@STATUS4(5,@CNT1),0)
  SV(@@STATUS3(6,@CNT1),0)  SV(@@STATUS4(6,@CNT1),0)
  SV(@@STATUS3(7,@CNT1),0)  SV(@@STATUS4(7,@CNT1),0)
;
  SV(@@STATUS5(0,@CNT1),0)  SV(@@STATUS6(0,@CNT1),0)
  SV(@@STATUS5(1,@CNT1),0)  SV(@@STATUS6(1,@CNT1),0)
```

```
  SV(@@STATUS5(2,@CNT1),0)  SV(@@STATUS6(2,@CNT1),0)
  SV(@@STATUS5(3,@CNT1),0)  SV(@@STATUS6(3,@CNT1),0)
  SV(@@STATUS5(4,@CNT1),0)  SV(@@STATUS6(4,@CNT1),0)
  SV(@@STATUS5(5,@CNT1),0)  SV(@@STATUS6(5,@CNT1),0)
  SV(@@STATUS5(6,@CNT1),0)  SV(@@STATUS6(6,@CNT1),0)
  SV(@@STATUS5(7,@CNT1),0)  SV(@@STATUS6(7,@CNT1),0)
  IF(@CNT1,LT,34,:AB1)
  SV(@CNT1,0)
:AB2
  IV(@CNT1)
  SV(@@CLIST(0,@CNT1),0)    SV(@@CLIST(1,@CNT1),0)
  SV(@@CLIST(2,@CNT1),0)    SV(@@CLIST(3,@CNT1),0)
  SV(@@CLIST(4,@CNT1),0)    SV(@@CLIST(5,@CNT1),0)
  SV(@@CLIST(6,@CNT1),0)    SV(@@CLIST(7,@CNT1),0)
  IF(@CNT1,LT,204,:AB2)
  SV(@CNT1,0)
:AB3
  IV(@CNT1)
  SV(@@PLAN(0,@CNT1),0)  SV(@@PLAN(1,@CNT1),0)  SV(@@PLAN(2,@CNT1),0)
  SV(@@PLAN(3,@CNT1),0)  SV(@@PLAN(4,@CNT1),0)  SV(@@PLAN(5,@CNT1),0)
  SV(@@PLAN(6,@CNT1),0)  SV(@@PLAN(7,@CNT1),0)  SV(@@PLAN(8,@CNT1),0)
  SV(@@PLAN(9,@CNT1),0)  SV(@@PLAN(10,@CNT1),0)  SV(@@PLAN(11,@CNT1),0)
  SV(@@PLAN(12,@CNT1),0)  SV(@@PLAN(13,@CNT1),0)  SV(@@PLAN(14,@CNT1),0)
  SV(@@PLAN(15,@CNT1),0)  SV(@@PLAN(16,@CNT1),0)
  SV(@@TTJOB(0,@CNT1),0)  SV(@@TTJOB(1,@CNT1),0)  SV(@@TTJOB(2,@CNT1),0)
  SV(@@TTJOB(3,@CNT1),0)  SV(@@TTJOB(4,@CNT1),0)  SV(@@TTJOB(5,@CNT1),0)
  IF(@CNT1,LT,500,:AB3)
  SV(@CNT1,0)
:AB4
  IV(@CNT1)
  SV(@@BFJOB(0,@CNT1),0)  SV(@@BFJOB(1,@CNT1),0)
  SV(@@RHD(0,@CNT1),0)  SV(@@RHD(11,@CNT1),0)  SV(@@RHD(21,@CNT1),0)
  SV(@@RHD(1,@CNT1),0)  SV(@@RHD(12,@CNT1),0)  SV(@@RHD(22,@CNT1),0)
  SV(@@RHD(2,@CNT1),0)  SV(@@RHD(13,@CNT1),0)  SV(@@RHD(23,@CNT1),0)
  SV(@@RHD(3,@CNT1),0)  SV(@@RHD(14,@CNT1),0)  SV(@@RHD(24,@CNT1),0)
  SV(@@RHD(4,@CNT1),0)  SV(@@RHD(15,@CNT1),0)  SV(@@RHD(25,@CNT1),0)
  SV(@@RHD(5,@CNT1),0)  SV(@@RHD(16,@CNT1),0)  SV(@@RHD(26,@CNT1),0)
  SV(@@RHD(6,@CNT1),0)  SV(@@RHD(17,@CNT1),0)  SV(@@RHD(27,@CNT1),0)
  SV(@@RHD(7,@CNT1),0)  SV(@@RHD(18,@CNT1),0)  SV(@@RHD(28,@CNT1),0)
  SV(@@RHD(8,@CNT1),0)  SV(@@RHD(19,@CNT1),0)  SV(@@RHD(29,@CNT1),0)
  SV(@@RHD(9,@CNT1),0)  SV(@@RHD(20,@CNT1),0)  SV(@@RHD(30,@CNT1),0)
  SV(@@RHD(10,@CNT1),0)
  IF(@CNT1,LT,100,:AB4)
  SV(@CNT1,0)
:AB5
  IV(@CNT1)
  IF(@CNT1,LT,30,:AB5)
  SV(@CNT1,1)
;
SV(@A5,0)       SV(@A3,1)       SV(@A4,0)       SV(@B1,0)
SV(@C1,0)       SV(@CC2,0)      SV(@CC3,1)
SV(@KK2,0)      SV(@KK3,0)      SV(@START,0)
SV(@N1,0)       SV(@N3,0)       SV(@N4,0)       SV(@N5,0)
SV(@N6,0)       SV(@N7,0)
```

```
SV(@R4,0)
SV(@RRR,0)
SV(@S1,0)        SV(@S2,0)        SV(@S3,0)        SV(@S4,0)       SV(@S5,0)
SV(@S7,0)
SV(@Z1,0)        SV(@Z2,0)        SV(@Z3,0)        SV(@Z4,0)       SV(@Z5,0)
SV(@CNT1,1)      SV(@CNT2,1)      SV(@CNT3,1)      SV(@CNT4,0)
SV(@DCNT,0)      SV(@DMCNT,0)     SV(@RHDCNT,0)
SV(@S1TOT,0)     SV(@S2TOT,0)     SV(@S3TOT,0)     SV(@S4TOT,0)    SV(@S5TOT,0)
SV(@RW,0)        SV(@SEC,1)
SV(@RTOT,0)      SV(@STOT,0)      SV(@STK,0)
SV(@ROW,0)       SV(@RHD,0)       SV(@LVL,0)
SV(@TTROW,1)     SV(@TTSEC,1)     SV(@NRHD,1)      SV(@SECTOT,0)
SV(@TROWCHG,0)   SV(@TADCHG,0)    SV(@TNADCHG,0)   SV(@ORDER,0)
;
SV(@A1,0)        SV(@A2,0)        SV(@A6,0)        SV(@A7,0)       SV(@A8,0)
SV(@B2,0)        SV(@B3,1)        SV(@B4,0)        SV(@B5,0)       SV(@B6,0)
SV(@B9,0)        SV(@B10,0)       SV(@B11,0)       SV(@A9,0)
;
SV(@C2,0)        SV(@C3,0)        SV(@C4,0)        SV(@C5,0)       SV(@C6,0)
SV(@K3,0)        SV(@K4,0)        SV(@C7,0)        SV(@K1,0)       SV(@K2,0)
SV(@D1,0)        SV(@D2,0)        SV(@D3,0)        SV(@D4,0)       SV(@D5,0)
SV(@KK1,0)       SV(@KK4,0)       SV(@RANDOM,0)    SV(@D7,0)       SV(@CC1,0)
;
SV(@PNADCHG,@F1) SV(@PADCHG,@F2) SV(@PRCHG,@F3) SV(@PRHD,@F4)
;
;
;                 ; USER SPECIFIES ROWS THAT WILL CONTAIN CONTAINERS
;                 ; FOR THE VOYAGE WITH CONTAINER TYPE SPECIFICATION
;
                         SV(@@STATUS1(0,5),40)  SV(@@STATUS1(0,6),41)
 SV(@@STATUS1(0,4),20)  SV(@@STATUS1(0,9),40)  SV(@@STATUS1(0,10),41)
 SV(@@STATUS1(0,8),20)  SV(@@STATUS1(0,15),40) SV(@@STATUS1(0,16),41)
                         SV(@@STATUS1(0,27),40) SV(@@STATUS1(0,28),41)
 SV(@@STATUS1(0,12),20)
 SV(@@STATUS1(0,18),20)
 SV(@@STATUS1(0,22),20)
 SV(@@STATUS1(0,25),20)
 SV(@@STATUS1(0,26),20)
 SV(@@STATUS1(0,13),20)
;
 SV(@@STATUS2(0,12),20) SV(@@STATUS2(0,13),40)  SV(@@STATUS2(0,14),41)
 SV(@@STATUS2(0,15),20) SV(@@STATUS2(0,22),40)  SV(@@STATUS2(0,23),41)
 SV(@@STATUS2(0,16),20) SV(@@STATUS2(0,25),40)  SV(@@STATUS2(0,26),41)
 SV(@@STATUS2(0,24),20)
 SV(@@STATUS2(0,18),20)
 SV(@@STATUS2(0,19),20)
;
 SV(@@STATUS4(0,6),20)  SV(@@STATUS4(0,4),40)  SV(@@STATUS4(0,5),41)
 SV(@@STATUS4(0,7),20)  SV(@@STATUS4(0,12),40) SV(@@STATUS4(0,13),41)
 SV(@@STATUS4(0,11),20) SV(@@STATUS4(0,21),40) SV(@@STATUS4(0,22),41)
 SV(@@STATUS4(0,17),20)
 SV(@@STATUS4(0,20),20)
 SV(@@STATUS4(0,24),20)
;
```

```
                         SV(@@STATUS5(0,5),40)  SV(@@STATUS5(0,6),41)
 SV(@@STATUS5(0,4),20)  SV(@@STATUS5(0,10),40) SV(@@STATUS5(0,11),41)
 SV(@@STATUS5(0,12),20)
 SV(@@STATUS5(0,13),20)
;
 SV(@@STATUS3(0,32),20) SV(@@STATUS3(0,33),40) SV(@@STATUS3(0,34),41)
;
;
;                    ; GENERATE NUMBER OF CONTAINERS FOR SECTION 1
;
:I1
  SV(@A1,@@STATUS1(0,@CNT1))
  IF(@A1,EQ,0,:I3)
  IF(@A1,EQ,41,:I3)
:I2
  RV(@A2,1,4)
  DV(@A2)
  SV(@@STATUS1(@A3,@CNT1),@A2)
  IV(@A3)
  AO(@A4,+,@A2)
  SV(@@STATUS1(7,@CNT1),@A4)
  IF(@A3,LE,6,:I2)
:I3
  IV(@CNT1)
  SV(@A3,1)
  AO(@A5,+,@A4)
  SV(@A4,0)
  IF(@CNT1,LE,34,:I1)
  SV(@CNT1,1)
;
;                    ; GENERATE NUMBER OF CONTAINERS FOR SECTION 2
:I4
  SV(@A1,@@STATUS2(0,@CNT1))
  IF(@A1,EQ,0,:I6)
  IF(@A1,EQ,41,:I6)
:I5
  RV(@A2,1,4)
  DV(@A2)
  SV(@@STATUS2(@A3,@CNT1),@A2)
  IV(@A3)
  AO(@A4,+,@A2)
  SV(@@STATUS2(7,@CNT1),@A4)
  IF(@A3,LE,6,:I5)
:I6
  IV(@CNT1)
  SV(@A3,1)
  AO(@A5,+,@A4)
  SV(@A4,0)
  IF(@CNT1,LE,34,:I4)
  SV(@CNT1,1)
;
;                    ; GENERATE NUMBER OF CONTAINERS FOR SECTION 3
:I7
  SV(@A1,@@STATUS3(0,@CNT1))
```

```
  IF(@A1,EQ,0,:I9)
  IF(@A1,EQ,41,:I9)
:I8
  RV(@A2,1,4)
  DV(@A2)
  SV(@@STATUS3(@A3,@CNT1),@A2)
  IV(@A3)
  AO(@A4,+,@A2)
  SV(@@STATUS3(7,@CNT1),@A4)
  IF(@A3,LE,6,:I8)
:I9
  IV(@CNT1)
  SV(@A3,1)
  AO(@A5,+,@A4)
  SV(@A4,0)
  IF(@CNT1,LE,34,:I7)
  SV(@CNT1,1)
;
;                    ; GENERATE NUMBER OF CONTAINERS FOR SECTION 4
:I10
  SV(@A1,@@STATUS4(0,@CNT1))
  IF(@A1,EQ,0,:I12)
  IF(@A1,EQ,41,:I12)
:I11
  RV(@A2,1,4)
  DV(@A2)
  SV(@@STATUS4(@A3,@CNT1),@A2)
  IV(@A3)
  AO(@A4,+,@A2)
  SV(@@STATUS4(7,@CNT1),@A4)
  IF(@A3,LE,6,:I11)
:I12
  IV(@CNT1)
  SV(@A3,1)
  AO(@A5,+,@A4)
  SV(@A4,0)
  IF(@CNT1,LE,34,:I10)
  SV(@CNT1,1)
;
;                    ; GENERATE NUMBER OF CONTAINERS FOR SECTION 5
:I13
  SV(@A1,@@STATUS5(0,@CNT1))
  IF(@A1,EQ,0,:I15)
  IF(@A1,EQ,41,:I15)
:I14
  RV(@A2,1,4)
  DV(@A2)
  SV(@@STATUS5(@A3,@CNT1),@A2)
  IV(@A3)
  AO(@A4,+,@A2)
  SV(@@STATUS5(7,@CNT1),@A4)
  IF(@A3,LE,6,:I14)
:I15
  IV(@CNT1)
```

```
  SV(@A3,1)
  AO(@A5,+,@A4)
  SV(@A4,0)
  IF(@CNT1,LE,34,:I13)
  SV(@CNT1,1)
;
  PV(#MSG1,@A5)
  SV(@CNT1,1)
  SV(@A1,0)
  SV(@A2,0)
  SV(@A3,0)
  SV(@A4,0)
;
;
;                  ; TRANSFER DATA IN @@STATUS1 - @@STATUS6 TO @@CLIST AND
                   ; DISPLAY THE INITIAL NUMBER OF CONTAINERS IN EACH ROW
                   ; OF ALL SECTIONS
;
:I19
  SV(OBJ@1,1)
  SV(OBJ@2,@CNT1)
  SV(OBJ@6,@@STATUS1(7,@CNT1))
  SV(OBJ@3,@@STATUS1(0,@CNT1))
  SV(@@CLIST(0,@CNT2),@@STATUS1(0,@CNT1))
  SV(@@CLIST(1,@CNT2),@@STATUS1(1,@CNT1))
  SV(@@CLIST(2,@CNT2),@@STATUS1(2,@CNT1))
  SV(@@CLIST(3,@CNT2),@@STATUS1(3,@CNT1))
  SV(@@CLIST(4,@CNT2),@@STATUS1(4,@CNT1))
  SV(@@CLIST(5,@CNT2),@@STATUS1(5,@CNT1))
  SV(@@CLIST(6,@CNT2),@@STATUS1(6,@CNT1))
  SV(@@CLIST(7,@CNT2),@@STATUS1(7,@CNT1))
  SV(@@ROWTOT(1,@CNT1),OBJ@6)
  IV(@CNT2)
  SV(@A1,@@STATUS1(7,@CNT1))
  AO(@S1TOT,+,@A1)
  IF(OBJ@3,EQ,0,:A)
  LK(!NUMBER)
:A
  IV(@CNT1)
  IF(@CNT1,LE,34,:I19)
  SV(@CNT1,1)
;
;
:I20
  SV(OBJ@1,2)
  SV(OBJ@2,@CNT1)
  SV(OBJ@6,@@STATUS2(7,@CNT1))
  SV(OBJ@3,@@STATUS2(0,@CNT1))
  SV(@@CLIST(0,@CNT2),@@STATUS2(0,@CNT1))
  SV(@@CLIST(1,@CNT2),@@STATUS2(1,@CNT1))
  SV(@@CLIST(2,@CNT2),@@STATUS2(2,@CNT1))
  SV(@@CLIST(3,@CNT2),@@STATUS2(3,@CNT1))
  SV(@@CLIST(4,@CNT2),@@STATUS2(4,@CNT1))
  SV(@@CLIST(5,@CNT2),@@STATUS2(5,@CNT1))
```

```
  SV(@@CLIST(6,@CNT2),@@STATUS2(6,@CNT1))
  SV(@@CLIST(7,@CNT2),@@STATUS2(7,@CNT1))
  SV(@@ROWTOT(2,@CNT1),OBJ@6)
  IV(@CNT2)
  SV(@A1,@@STATUS2(7,@CNT1))
  AO(@S2TOT,+,@A1)
  IF(OBJ@3,EQ,0,:B)
  LK(!NUMBER)
:B
  IV(@CNT1)
  IF(@CNT1,LE,34,:I20)
  SV(@CNT1,1)
;
;
:I21
  SV(OBJ@1,3)
  SV(OBJ@2,@CNT1)
  SV(OBJ@6,@@STATUS3(7,@CNT1))
  SV(OBJ@3,@@STATUS3(0,@CNT1))
  SV(@@CLIST(0,@CNT2),@@STATUS3(0,@CNT1))
  SV(@@CLIST(1,@CNT2),@@STATUS3(1,@CNT1))
  SV(@@CLIST(2,@CNT2),@@STATUS3(2,@CNT1))
  SV(@@CLIST(3,@CNT2),@@STATUS3(3,@CNT1))
  SV(@@CLIST(4,@CNT2),@@STATUS3(4,@CNT1))
  SV(@@CLIST(5,@CNT2),@@STATUS3(5,@CNT1))
  SV(@@CLIST(6,@CNT2),@@STATUS3(6,@CNT1))
  SV(@@CLIST(7,@CNT2),@@STATUS3(7,@CNT1))
  SV(@@ROWTOT(3,@CNT1),OBJ@6)
  IV(@CNT2)
  SV(@A1,@@STATUS3(7,@CNT1))
  AO(@S3TOT,+,@A1)
  IF(OBJ@3,EQ,0,:C)
  LK(!NUMBER)
:C
  IV(@CNT1)
  IF(@CNT1,LE,34,:I21)
  SV(@CNT1,1)
;
;
:I22
  SV(OBJ@1,4)
  SV(OBJ@2,@CNT1)
  SV(OBJ@6,@@STATUS4(7,@CNT1))
  SV(OBJ@3,@@STATUS4(0,@CNT1))
  SV(@@CLIST(0,@CNT2),@@STATUS4(0,@CNT1))
  SV(@@CLIST(1,@CNT2),@@STATUS4(1,@CNT1))
  SV(@@CLIST(2,@CNT2),@@STATUS4(2,@CNT1))
  SV(@@CLIST(3,@CNT2),@@STATUS4(3,@CNT1))
  SV(@@CLIST(4,@CNT2),@@STATUS4(4,@CNT1))
  SV(@@CLIST(5,@CNT2),@@STATUS4(5,@CNT1))
  SV(@@CLIST(6,@CNT2),@@STATUS4(6,@CNT1))
  SV(@@CLIST(7,@CNT2),@@STATUS4(7,@CNT1))
  SV(@@ROWTOT(4,@CNT1),OBJ@6)
  IV(@CNT2)
```

```
  SV(@A1,@@STATUS4(7,@CNT1))
  AO(@S4TOT,+,@A1)
  IF(OBJ@3,EQ,0,:D)
  LK(!NUMBER)
:D
  IV(@CNT1)
  IF(@CNT1,LE,34,:I22)
  SV(@CNT1,1)
;
;
:I23
  SV(OBJ@1,5)
  SV(OBJ@2,@CNT1)
  SV(OBJ@6,@@STATUS5(7,@CNT1))
  SV(OBJ@3,@@STATUS5(0,@CNT1))
  SV(@@CLIST(0,@CNT2),@@STATUS5(0,@CNT1))
  SV(@@CLIST(1,@CNT2),@@STATUS5(1,@CNT1))
  SV(@@CLIST(2,@CNT2),@@STATUS5(2,@CNT1))
  SV(@@CLIST(3,@CNT2),@@STATUS5(3,@CNT1))
  SV(@@CLIST(4,@CNT2),@@STATUS5(4,@CNT1))
  SV(@@CLIST(5,@CNT2),@@STATUS5(5,@CNT1))
  SV(@@CLIST(6,@CNT2),@@STATUS5(6,@CNT1))
  SV(@@CLIST(7,@CNT2),@@STATUS5(7,@CNT1))
  SV(@@ROWTOT(5,@CNT1),OBJ@6)
  IV(@CNT2)
  SV(@A1,@@STATUS5(7,@CNT1))
  AO(@S5TOT,+,@A1)
  IF(OBJ@3,EQ,0,:E)
  LK(!NUMBER)
:E
  IV(@CNT1)
  IF(@CNT1,LE,34,:I23)
  SV(@CNT1,1)
;
  SV(@Z1,@S1TOT)
  SV(@Z2,@S2TOT)
  SV(@Z3,@S3TOT)
  SV(@Z4,@S4TOT)
  SV(@Z5,@S5TOT)
;
;
;******************** DISPLAY EMPTY CHASSIS IN BUFFER SPACE ON SCREEN **********
;
  SV(@CNT1,4)
  AO(@CHANUM,+,4)
  IF(@CHANUM,EQ,4,:FFF1)
:F1
  IV(@CNT1)
  SV(@@BUFFER(1,@CNT1),1)
  SV(OBJ@4,@CNT1)
  LK(!PUTCHA)
  IF(@CNT1,LT,@CHANUM,:F1)
:FFF1
  AO(@CHANUM,-,4)
```

```
;
;
;******************* LOAD PLANNING *********************************************
;
  SV(@SEC,1)
  SV(@CNT1,1)
  AO(@PNADCHG,+,@PADCHG)
  AO(@PRCHG,+,@PNADCHG)
  JP(:A27)
:A1
  RV(@RANDOM,1,10000)                ; GENERATE A RANDOM NUMBER
  IF(@RANDOM,LE,@PADCHG,:A6)         ; ADJACENT SECTION CHANGE ?
  IF(@RANDOM,LE,@PNADCHG,:A13)       ; NON-ADJACENT SECTION CHANGE ?
  IF(@RANDOM,LE,@PRCHG,:A33)         ; ROW CHANGE ?
;
  SV(@RTOT,@@CLIST(7,@ROW))
  IF(@RTOT,EQ,0,:A1)                 ; CHECK IF RURRENT ROW IS EMPTY
:A2
  RV(@STK,1,6)                       ;
  SV(@STOT,@@CLIST(@STK,@ROW))
  IF(@STOT,EQ,0,:A2)                 ; CURRENT STACK IS EMPTY ?
  IF(@STOT,EQ,1,:A3)                 ; ONE CONTAINER IN CURRENT STACK ?
  RV(@RANDOM,1,10000)
  IF(@RANDOM,LE,@PRHD,:A4)           ; SCHEDULE REHANDLING ?
:A3
  SV(@LVL,@STOT)
  SV(@RHD,0)
  JP(:A5)
:A4
  RV(@LVL,1,3)
  IF(@LVL,GE,@STOT,:A4)              ; COMPUTE NUMBER OF CONTAINERS
  SV(@RHD,@STOT)                     ; TO BE REHANDLED
  AO(@RHD,-,@LVL)
:A5
  DV(@STOT)                          ; DECREASE CURRENT STACK AND ROW TOTAL
  SV(@@CLIST(@STK,@ROW),@STOT)       ; AND UPDATE @@CLIST
  DV(@RTOT)
  SV(@@CLIST(7,@ROW),@RTOT)
;
;                                    ;DECREASE SECTION TOTAL
  IF(@SEC,NE,1,:AA2)  DV(@S1TOT) SV(@SECTOT,@S1TOT)  JP(:AA7)
:AA2
  IF(@SEC,NE,2,:AA3)  DV(@S2TOT) SV(@SECTOT,@S2TOT)  JP(:AA7)
:AA3
  IF(@SEC,NE,3,:AA4)  DV(@S3TOT) SV(@SECTOT,@S3TOT)  JP(:AA7)
:AA4
  IF(@SEC,NE,4,:AA5)  DV(@S4TOT) SV(@SECTOT,@S4TOT)  JP(:AA7)
:AA5
                      DV(@S5TOT) SV(@SECTOT,@S5TOT)
:AA7
;
;                  ; STORE INFORMATION ABOUT SELECTED CONTAINER IN @@PLAN
  SV(@@PLAN(0,@CNT1),@RHD)
  SV(@@PLAN(1,@CNT1),@SEC)
```

```
  SV(@@PLAN(2,@CNT1),@RW)
  SV(@@PLAN(3,@CNT1),@STK)
  SV(@@PLAN(4,@CNT1),@LVL)
:AA8
;
  IV(@CNT1)
  IF(@CNT1,GT,@A5,:A50)          ; INCREMENT PLANNING INDEX AND CHECK IF
  JP(:A1)                        ; ALL CONTAINERS ARE INCLUDED IN.
;
;
;****************** ADJACENT SECTION SELECTION********************************
;
:A6
   IF(@SEC,NE,1,:A7)  SV(@A3,4)  SV(@A2,@S4TOT)  JP(:A12)
:A7
   IF(@SEC,NE,2,:A8)  SV(@A3,5)  SV(@A2,@S5TOT)  JP(:A12)
:A8
   IF(@SEC,NE,3,:A9)  JP(:A1)
:A9
   IF(@SEC,NE,4,:A10)  SV(@A3,1)  SV(@A2,@S1TOT)  JP(:A12)
:A10
   IF(@SEC,NE,5,:A11)  SV(@A3,2)  SV(@A2,@S2TOT)  JP(:A12)
:A11
                       SV(@A3,3)  SV(@A2,@S3TOT)
:A12
  JP(:A26)
;
;****************** NON-ADJACENT SECTION CHANGE*******************************
;
:A13
   RV(@A3,1,5)
   IF(@A3,EQ,@SEC,:A13)
   IF(@SEC,NE,1,:A14)
      IF(@A3,EQ,4,:A13)   JP(:A19)
:A14
   IF(@SEC,NE,2,:A15)
      IF(@A3,EQ,5,:A13)   JP(:A19)
:A15
   IF(@SEC,NE,3,:A16)
      IF(@A3,EQ,6,:A13)   JP(:A19)
:A16
   IF(@SEC,NE,4,:A17)
      IF(@A3,EQ,1,:A13)   JP(:A19)
:A17
   IF(@SEC,NE,5,:A18)
      IF(@A3,EQ,2,:A13)   JP(:A19)
:A18
      IF(@A3,EQ,3,:A13)
:A19
;
;                   ; CHECK IF SELECTED SECTION IS EMPTY
   IF(@A3,NE,1,:A20)  SV(@A2,@S1TOT)  JP(:A25)
:A20
   IF(@A3,NE,2,:A21)  SV(@A2,@S2TOT)  JP(:A25)
```

```
:A21
   IF(@A3,NE,3,:A22)  SV(@A2,@S3TOT)  JP(:A25)
:A22
   IF(@A3,NE,4,:A23)  SV(@A2,@S4TOT)  JP(:A25)
:A23
   IF(@A3,NE,5,:A24)  SV(@A2,@S5TOT)  JP(:A25)
:A24
:A25
;
;
:A26
 IF(@A2,EQ,0,:A1)   ; IF SELECTED SECTION IS EMPTY, REPEATE FROM BEGINNING
   SV(@SEC,@A3)
;
;******************* ROW SELECTION ********************************************
;
:A26                                 ; GENERATE A RANDOM NUMBEWR FOR ROW
      IF(@SEC,NE,1,:A28)             ; SELECTION. GET ROW TOTAL FROM @@CLIST
      IF(@S1TOT,EQ,0,:A1)            ; IF SELECTED ROW IS EMPTY, REPEATE ROW
      RV(@A7,1,34)                   ; SELECTION PROCEDURE.  OTHERWISE, ROW
      SV(@A6,@@CLIST(7,@A7))         ; IS SELECTED
      IF(@A6,EQ,0,:A27)
      SV(@ROW,@A7)
      SV(@RW,@ROW)
      JP(:A32)
:A28
   IF(@SEC,NE,2,:A29)
      IF(@S2TOT,EQ,0,:A1)
      RV(@A7,35,68)
      SV(@A6,@@CLIST(7,@A7))
      IF(@A6,EQ,0,:A27)
      SV(@ROW,@A7)
      SV(@RW,@ROW)
      AO(@RW,-,34)
      JP(:A32)
:A29
   IF(@SEC,NE,3,:A30)
      IF(@S3TOT,EQ,0,:A1)
      RV(@A7,69,102)
      SV(@A6,@@CLIST(7,@A7))
      IF(@A6,EQ,0,:A27)
      SV(@ROW,@A7)
      SV(@RW,@ROW)
      AO(@RW,-,68)
      JP(:A32)
:A30
   IF(@SEC,NE,4,:A31)
      IF(@S4TOT,EQ,0,:A1)
      RV(@A7,103,136)
      SV(@A6,@@CLIST(7,@A7))
      IF(@A6,EQ,0,:A27)
      SV(@ROW,@A7)
      SV(@RW,@ROW)
      AO(@RW,-,102)
```

```
      JP(:A32)
:A31
      IF(@S5TOT,EQ,0,:A1)
      RV(@A7,137,170)
      SV(@A6,@@CLIST(7,@A7))
      IF(@A6,EQ,0,:A27)
      SV(@ROW,@A7)
      SV(@RW,@ROW)
      AO(@RW,-,136)
:A32
      SV(@RTOT,@@CLIST(7,@ROW))
      JP(:A2)
;
;******************* SAME SECTION BUT ROW CHANGE ******************************
:A33
  SV(@A3,@SEC)
  JP(:A19)
;
;
:A50
;
  SV(@S1TOT,@Z1)
  SV(@S2TOT,@Z2)
  SV(@S3TOT,@Z3)
  SV(@S4TOT,@Z4)
  SV(@S5TOT,@Z5)
  SV(@START,1)
  SV(@CNT1,0)
  SV(@CNT2,1)
  SV(@CNT3,1)
  SV(@A9,0)
  SV(@B9,0)
  SV(@S1TOT,@Z1)
  SV(@S2TOT,@Z2)
  SV(@S3TOT,@Z3)
  SV(@S4TOT,@Z4)
  SV(@S5TOT,@Z5)
ER
```

```
;
;******************* ROUTING 2 ************************************************
;                                                   ; THIS ROUTING IS FOR
TRANSTAINER OPERATION
BR(2,*TTINIT,0)
:B1
  IV(@CNT1)          ; INCREMENT JOB INDEX
  CL(*TTUT)          ; PUT TRANSTAINER IN IDLE STATUS
  IF(@CNT1,GT,@A5,:B50)            ; NO MORE JOBS ?
;
:BB1
  DN
  SV(@B6,@@TTJOB(0,@CNT1))         ; GET INFORMATION FOR THE JOB
  SV(OBJ@1,@@TTJOB(1,@CNT1))
  SV(OBJ@2,@@TTJOB(2,@CNT1))
  SV(OBJ@3,@@TTJOB(3,@CNT1))
  SV(OBJ@4,@@TTJOB(4,@CNT1))
  SV(OBJ@6,@@ROWTOT(OBJ@1,OBJ@2))
  IF(OBJ@1,EQ,0,:BB1)
    PO(*TTUT)                      ; PUT TRANSTAINER IN BUSY STATUS
;
;
  IF(OBJ@1,EQ,@TTSEC,:B2)          ; NEDD A SECTION CHANGE ?
  LK(!TTSCHG)                      ; CHANGE SECTION
  JP(:B3)
;
:B2
  IF(OBJ@2,EQ,@TTROW,:B3)          ; NEED A ROW CHANGE ?
    IF(@TTROW,GT,OBJ@2,:BB2)       ; UP OR DOWN ?
    SV(@B2,OBJ@2)
    AO(@B2,-,@TTROW)
    LK(!TTMOVD)
    SV(@TTROW,OBJ@2)
    JP(:B3)
:BB2
    SV(@B2,@TTROW)
    AO(@B2,-,OBJ@2)
    LK(!TTMOVU)
    SV(@TTROW,OBJ@2)
;
:B3
    DN
    SV(@B5,@@PLAN(6,OBJ@4))
    CL(*TTUT)
    IF(@B5,EQ,0,:B3)               ; WAIT UNTIL TRUCK ARRIVES
    PO(*TTUT)                      ; PUT TRANSTAINER IN BUSY STATUS
    SV(@B4,OBJ@3)
    IF(@B6,NE,0,:BX1)              ; NEED A REHANDLE ?
    LK(!TTMOVR)
    ST(87)
    JP(:BX4)
;
:BX1
    SV(OBJ@ID,082)                 ; DISPLAY 'R' FOR REHANDLING
```

```
    LK(!TTMOVR)
    IF(@B6,NE,1,:BX2)
    ST(157)                             ; ONE CONTAINER TO BE REHANDLED
    JP(:BX3)
:BX2
    ST(314)                             ; TWO CONTAINERS TO BE REHANDLED
:BX3
    SV(OBJ@ID,084)
    AO(@RHDCNT,+,@B6)                   ; UPDATE TOTAL REHANDLE COUNT AND
    PV(#MSG3,@RHDCNT)                   ; DISPLAY AT #MSG3
;
:BX4
    DV(OBJ@6)                           ; DECREMENT ROW TOTAL AND DISPLAY IT
    SV(@@ROWTOT(OBJ@1,OBJ@2),OBJ@6)
    LK(!NUMBER)
    LK(!TTMOVL)
    SV(@@PLAN(7,OBJ@4),1)
    JP(:B1)                             ; REPEATE SAME PROCEDURE FROM BEGINNING
;
:B50
    PO(#FINAL)                          ; ALL CONTAINERS ARE LOADED. POST #FINAL
    MA(#EXIT,0)
ER
;
;
```

```
;****************** ROUTING 3 **************************************************
;                 ; THIS ROUTING IS FOR TRUCK OPERATION
;
BR(3,*YTRINIT,0)
:C0
  DN
  IF(@START,EQ,0,:C0)            ; WAIT UNTIL LOAD PLAN IS DEVELOPED
  MU(35,@TRMOVV)                 ; MOVE TO STARTING POSITION
;
:C1
  SV(@KK3,0)                     ; SET BUFFER JOB INDEX TO ZERO
:CCBF1
  IV(@KK3)                       ; INCREMENT SEARCHING INDEX
  IF(@KK3,GT,100,:CCBF2)         ; CHECKED ALL ENTRIES IN @@BFJOB ?
      SV(@KK4,@@BFJOB(0,@KK3))
    IF(@KK4,EQ,0,:CCBF2)         ; NO JOB AVAILABLE IN BUFFER ?
      SV(@KK1,@ORDER)
      IV(@KK1)
    IF(@KK1,LT,@KK4,:CCBF1)      ; CHECK TIMING OF BUFFER JOB
      SV(@KK4,@@BFJOB(1,@KK3))
    IF(@KK4,EQ,1,:CCBF1)
      SV(OBJ@6,@@BFJOB(0,@KK3))
      SV(@@BFJOB(1,@KK3),1)      ; TAKE THE JOB AND MOVE TO BUFFER AEA
      JP(:BF1)
;
:CCBF2
  JC(1,*RHD,:CF1)                ; REHANDLING OR SWEEPING GOING ON NOW ?
  SV(@DMCNT,@@RHD(@CC3,@CC2))    ; GET JOB INDEX
  IV(@CC3)
IF(@DMCNT,EQ,0,:CC0)             ; NO JOB LEFT FOR CURRENT @@RHD OPERATION?
  SV(OBJ@1,@@PLAN(1,@DMCNT))     ; GET JOB INFORMATIONS
  SV(OBJ@2,@@PLAN(2,@DMCNT))
  SV(OBJ@3,@@PLAN(3,@DMCNT))
  SV(OBJ@6,@DMCNT)  SV(OBJ@5,1)
  SV(@@PLAN(9,@DMCNT),1)
  IV(@DCNT)
  SV(@@TTJOB(0,@DCNT),0)         ; PUT THIS CONTAINER IN THE TRANSTAINER
  SV(@@TTJOB(1,@DCNT),OBJ@1)     ; OPERATION LIST
  SV(@@TTJOB(2,@DCNT),OBJ@2)
  SV(@@TTJOB(3,@DCNT),OBJ@3)
  SV(@@TTJOB(4,@DCNT),OBJ@6)
  DV(@CHANUM)                    ; DECREMENT NUMBER OF EMPTY CHASSIS
  JP(:CC3)                       ; MOVE TO CONTAINER LOCATION
;
:CC0
  CL(*RHD)                       ; TERMINATE CURRENT REHANDLE OR SWEEPING
  SV(@CC3,1)                     ; OPERATION AND REPEATE FROM BEGINNING
  JP(:C1)
;
;
:CF1
  SV(@CC1,@@PLAN(0,@CNT3))       ; GET JOB INFORMATION FROM @@PLAN
  SV(OBJ@1,@@PLAN(1,@CNT3))
  SV(OBJ@2,@@PLAN(2,@CNT3))
```

```
  SV(OBJ@3,@@PLAN(3,@CNT3))
  SV(OBJ@6,@CNT3)
  SV(OBJ@5,0)
  SV(@CC4,@@PLAN(9,@CNT3))
  SV(@CCC,@@PLAN(10,@CNT3))
;
  IF(@CNT3,GT,@A5,:C11)              ; ALL CONTAINERS ARE LOADED ?
    IV(@CNT3)
  IF(@CC4,EQ,0,:CC33)                ; IS IT STILL LOCATED IN SECTION ?
    SV(@@PLAN(11,OBJ@6),1)
  IF(@CCC,EQ,0,:C1)                  ; SIGNAL TO COMING CONTAINER FOR SHIPPING
    SV(@@PLAN(11,OBJ@6),0)
    SV(@KK1,@ORDER)                  ; CHECK TIMING OF TAKING IT TO CRANE
    IV(@KK1)
  IF(@KK1,GE,OBJ@6,:BF1)             ; IT IS RIGHT TIME TO TAKE IT TO CRANE ?
    IV(@KK2)
    SV(@@BFJOB(0,@KK2),OBJ@6)        ; IF NOT, PUT IT IN @@BFJOB LIST
    JP(:C1)
;
:CC33                                ; CONTAINER IS STILL LOCATED IN SECTION
   IF(OBJ@1,NE,1,:CC34)
      SV(@SECTOT,@S1TOT)  JP(:CC38)
:CC34
   IF(OBJ@1,NE,2,:CC35)
      SV(@SECTOT,@S2TOT)  JP(:CC38)
:CC35
   IF(OBJ@1,NE,3,:CC36)
      SV(@SECTOT,@S3TOT)  JP(:CC38)
:CC36
   IF(OBJ@1,NE,4,:CC37)
      SV(@SECTOT,@S4TOT)  JP(:CC38)
:CC37
      SV(@SECTOT,@S5TOT)
:CC38
     DV(@SECTOT)
     IV(@DCNT)  SV(@R4,@DCNT)
     SV(@@TTJOB(0,@DCNT),@CC1)       ; PUT THIS CONTAINER IN TRANSTAINER
     SV(@@TTJOB(1,@DCNT),OBJ@1)      ; OPERATION LIST
     SV(@@TTJOB(2,@DCNT),OBJ@2)
     SV(@@TTJOB(3,@DCNT),OBJ@3)
     SV(@@TTJOB(4,@DCNT),OBJ@6)
;
  IF(@SECTOT,EQ,0,:CC39)             ; IS IT THE LAST CONTAINER IN THE SECTION?
  IF(@SECTOT,GT,@SWEEP,:CC39)        ; SWEEPING CRITERIA IS MET ?
  IF(@CHANUM,LT,@SECTOT,:CC39)       ; ENOUGH CHASSIS FOR SWEEPING ?
;
     SV(@B10,OBJ@6)
:CAA
    IV(@B10)
    IF(@B10,GT,@A5,:CC3)             ; CONTAINER SEARCH FOR SWEEPING OPERATION
       SV(@B11,@@PLAN(13,@B10))
    IF(@B11,EQ,1,:CAA)
       SV(@B11,@@PLAN(1,@B10))
    IF(@B11,EQ,OBJ@1,:CC39)
```

```
       SV(@D7,0)
:CAB
       IV(@B10)
    IF(@B10,GT,@A5,:CAF)
       SV(@D1,@@PLAN(1,@B10))
    IF(@D1,NE,OBJ@1,:CAB)
       SV(@D1,@@PLAN(13,@B10)
    IF(@D1,EQ,1,:CAB)
       IV(@D7)
       SV(@@RHD(@D7,@NRHD),@B10)       ; FILL @@RHD LIST
       SV(@@PLAN(13,@B10),1)
       JP(:CAB)
;
:CAF
  SV(@@TTJOB(0,@R4),0)
  PO(#RHD)  IV(@NRHD)   IV(@CC2)       ; POST #RHD FOR THIS SWEEPING OPERATION
  JP(:CC3)                             ; MOVE TO CONTAINER LOCATION
;
:CC39
    JB(1,#RHD,:CC3)
  IF(@CC1,EQ,0,:CC3)                   ; NEED TO SCHEDULE A REHANDLING OPERATION
?
     IF(@CHANUM,EQ,0,:CC3)             ; NO CHASSIS AVAILABLE ?
        SV(@D1,OBJ@6)
                      SV(@D7,0)
  IF(@CC1,EQ,2,:RCA)                   ; TWO CONTAINERS ARE TO BE REHANDLED ?
:RBA
    IV(@D1)
    SV(@D2,@@PLAN(1,@D1))              ; CONTAINER SEARCH FOR THID REHANDLE
  IF(@D2,NE,OBJ@1,:RBA)
   SV(@D3,@@PLAN(2,@D1))
  IF(@D3,NE,OBJ@2,:RBA)
   SV(@D4,@@PLAN(3,@D1))
  IF(@D4,NE,OBJ@3,:RBA)
;
      SV(@D5,@@PLAN(13,@D1))
      IF(@D5,EQ,1,:CC3)                ; IS IT ALREADY TAKEN TO BUFFER AREA ?
      IV(@D7)
      SV(@@RHD(@D7,@NRHD),@D1)         ; FILL @@RHD LIST
      SV(@@PLAN(13,@D1),1)
;
      DV(@SECTOT)  SV(@D1,OBJ@6)       ; CHECK IF SWEEPING OPERATION NEED TO
      IF(@SECTOT,GT,@SWEEP,:RBB)       ; FOLLOW IMMEDIATELY
      SV(@D3,@CHANUM)
      DV(@D3)
      IF(@D3,LT,@SECTOT,:RBB)          ; NOT ENOUGH EMPTY CHASSIS ?
:RBC
      IV(@D1)                          ; CONTAINER SEARCH FOR SWEEPING OPERATION
      IF(@D1,GT,@A5,:RBB)              ; ALL LIST SEARCHED ?
        SV(@D4,@@PLAN(1,@D1))
      IF(@D4,NE,OBJ@1,:RBC)
        SV(@D4,@@PLAN(13,@D1))
      IF(@D4,EQ,1,:RBC)
         IV(@D7)
```

```
        SV(@@PLAN(13,@D1),1)
        SV(@@RHD(@D7,@NRHD),@D1)     ; FILL @@RHD LIST FOR THIS SWEEPING
        JP(:RBC)
:RBB
      SV(@@TTJOB(0,@R4),0)
      PO(#RHD) IV(@NRHD)  IV(@CC2)    ; POST #RHD AND MOVE TO TRANSTAINER
      JP(:CC3)
;
:RCA
    IV(@D1)                           ; CONTAINER SEARCH FOR REHANDLING TWO
  IF(@D1,GT,@A5,:RCC)                 ; CONTAINERS
    SV(@D2,@@PLAN(1,@D1))
  IF(@D2,NE,OBJ@1,:RCA)
    SV(@D3,@@PLAN(2,@D1))
  IF(@D3,NE,OBJ@2,:RCA)
    SV(@D4,@@PLAN(3,@D1))
  IF(@D4,NE,OBJ@3,:RCA)
;
    SV(@D5,@@PLAN(13,@D1))
  IF(@D5,EQ,1,:RCA)
    IV(@D7)
    SV(@@RHD(@D7,@NRHD),@D1)          ; FILL @@RHD LIST
    SV(@@PLAN(13,@D1),1)
    JP(:RCA)
:RCC
  IF(@D7,EQ,0,:CC3)
  IF(@CHANUM,LT,@D7,:RCE)             ; ENOUGH CHASSIS FOR THIS REHANDLING ?
     AO(@SECTOT,-,@D7)
     SV(@D1,OBJ@6)
  IF(@SECTOT,GT,@SWEEP,:RHB)          ; NEED TO SCHEDULE SWEEPING RIGHT AFTER ?
     SV(@D3,@CHANUM)
     AO(@D3,-,@D7)
  IF(@D3,LT,@SECTOT,:RHB)             ; ENOUGH CHASSIS FOR SWEEPING ?
:RHC
  IV(@D1)                             ; CONTAINER SEARCH FOR SWEEPING
  IF(@D1,GT,@A5,:RHB)                 ; ALL LIST SEARCHED ?
     SV(@D4,@@PLAN(1,@D1))
  IF(@D4,NE,OBJ@1,:RHC)
     SV(@D4,@@PLAN(13,@D1))
  IF(@D4,EQ,1,:RHC)
     IV(@D7)
     SV(@@PLAN(13,@D1),1)
     SV(@@RHD(@D7,@NRHD),@D1)         ; FILL @@RHD LIST
     JP(:RHC)
:RHB
     SV(@@TTJOB(0,@R4),0)
     PO(#RHD)                         ; POST #RHD
     IV(@NRHD) IV(@CC2)               ; MOVE TO CONTAINER LOCATION
     JP(:CC3)
:RCE
     SV(@@RHD(1,@NRHD),0)
     SV(@@RHD(2,@NRHD),0)
     JP(:CC3)
;
```

```
:CC3                              ; TRUCK MOVES TO CONTAINER LOCATION
    IF(OBJ@1,NE,1,:C3)
;**************SECTION 1*********
  DV(@S1TOT)
  MR(8,@TRMOV)
  MD(1,@TRMOVV)
  SV(OBJ@4,OBJ@2)
  LK(!TRMOVD)
  SV(@@PLAN(6,OBJ@6),1)
:C2
  DN
  SV(@C1,@@PLAN(7,OBJ@6))
 IF(@C1,EQ,0,:C2)                 ; WAIT IF TRANSTAINER IS BUSY
  SV(OBJ@ID,067)
  SV(OBJ@4,36)
  AO(OBJ@4,-,OBJ@2)
  LK(!TRMOVD)
 IF(OBJ@5,EQ,0,:CCC2)             ; CHECK IF THIS CONTAINER NEED GO TO
BUFFER
  ML(8,@TRMOV)
  LK(!GOBF)
  JP(:C1)
:CCC2
  ML(2,@TRMOV)
  SV(@ORDER,OBJ@6)                ; UPDATE CHECKING COUNT FOR BUFFER JOB
  ML(10,@TRMOV)
  MU(17,@TRMOVV)
:WT1
  DN
 JB(1,*CRANE,:WT1)                ; WAIT IF CRANE IS BUSY
  PO(*CRANE)
  MU(1,@TRMOVV)
  ST(175)                       ; CRANE WORK
  CL(*CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(*MSG2,@CNT4)                 ; DISPLAY TOTAL NUMBER LOADED
  MU(19,@TRMOVV)
  MR(4,@TRMOV)
  JP(:C1)
;
;***************SECTION 2**********
:C3
 IF(OBJ@1,NE,2,:C5)
  DV(@S2TOT)
  MR(22,@TRMOV)
  MD(1,@TRMOVV)
  SV(OBJ@4,OBJ@2)
  LK(!TRMOVD)
  SV(@@PLAN(6,OBJ@6),1)
:C4
  DN
  SV(@C2,@@PLAN(7,OBJ@6))
 IF(@C2,EQ,0,:C4)
```

```
  SV(OBJ@ID,067)
  SV(OBJ@4,36)
  AO(OBJ@4,-,OBJ@2)
  LK(!TRMOVD)
 IF(OBJ@5,EQ,0,:CCC3)
  ML(22,@TRMOV)
  LK(!GOBF)
  JP(:C1)
:CCC3
  ML(16,@TRMOV)
  SV(@ORDER,OBJ@6)
  ML(10,@TRMOV)
  MU(17,@TRMOVV)
:WT2
  DN
  JB(1,#CRANE,:WT2)
  PO(#CRANE)
  MU(1,@TRMOVV)
  ST(175)
  CL(#CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(#MSG2,@CNT4)
  MU(19,@TRMOVV)
  MR(4,@TRMOV)
  JP(:C1)

;################SECTION 3##############
:C5
 IF(OBJ@1,NE,3,:C7)
  DV(@S3TOT)
  MR(36,@TRMOV)
  MD(1,@TRMOVV)
  SV(OBJ@4,OBJ@2)
  LK(!TRMOVD)
  SV(@@PLAN(6,OBJ@6),1)
:C6
  DN
  SV(@C3,@@PLAN(7,OBJ@6))
 IF(@C3,EQ,0,:C6)
  SV(OBJ@ID,067)
  SV(OBJ@4,36)
  AO(OBJ@4,-,OBJ@2)
  LK(!TRMOVD)
 IF(OBJ@5,EQ,0,:CCC4)
  ML(36,@TRMOV)
  LK(!GOBF)
  JP(:C1)
:CCC4
  ML(30,@TRMOV)  SV(@ORDER,OBJ@6)
  ML(10,@TRMOV) MU(17,@TRMOVV)
:WT3
  DN
 JB(1,#CRANE,:WT3)
```

```
  PO(*CRANE)
  MU(1,@TRMOVV)
  ST(175)
  CL(*CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(*MSG2,@CNT4)
  MU(19,@TRMOVV)
  MR(4,@TRMOV)
  JP(:C1)
;
;***************SECTION 4******************
:C7
 IF(OBJ@1,NE,4,:C9)
  DV(@S4TOT)
  MR(50,@TRMOV)
  MD(37,@TRMOVV)
  ML(40,@TRMOV)
  TP(6,XY(34,41),XY(34,42),XY(35,39),XY(35,40),XY(35,41),XY(35,42))
  ML(2,@TRMOV)
  MD(2,@TRMOVV)
  SV(OBJ@4,OBJ@2)
  LK(!TRMOVD)
  SV(@@PLAN(6,OBJ@6),1)
:C8
  DN
  SV(@C4,@@PLAN(7,OBJ@6))
 IF(@C4,EQ,0,:C8)
  SV(OBJ@ID,067)
  SV(OBJ@4,36)
  AO(OBJ@4,-,OBJ@2)
  LK(!TRMOVD)
 IF(OBJ@5,EQ,0,:CCC5)
  ML(8,@TRMOV)
  MU(38,@TRMOVV)
  LK(!GOBF)
  JP(:C1)
:CCC5
  ML(10,@TRMOV)
  MU(35,@TRMOVV)
  SV(@ORDER,OBJ@6)
  MU(20,@TRMOVV)
:WT4
  DN
 JB(1,*CRANE,:WT4)
  PO(*CRANE)
  MU(1,@TRMOVV)
  ST(175)
  CL(*CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(*MSG2,@CNT4)
  MU(19,@TRMOVV)
  MR(2,@TRMOV)
```

```
  JP(:C1)
;
;*************SECTION 5****************
:C9
  DV(@S5TOT)
  MR(50,@TRMOV)
  MD(37,@TRMOVV)
  ML(26,@TRMOV)
  TP(7,XY(48,40),XY(48,41),XY(48,42),XY(48,43),XY(49,40),XY(49,41),XY(49,42))
  ML(2,@TRMOV)
  MD(2,@TRMOVV)
  SV(OBJ@4,OBJ@2)
  SV(OBJ@4,OBJ@2)
  LK(!TRMOVD)
  SV(@@PLAN(6,OBJ@6),1)
:C10
  DN
  SV(@C5,@@PLAN(7,OBJ@6))
 IF(@C5,EQ,0,:C10)
  SV(OBJ@ID,067)
  SV(OBJ@4,36)
  AO(OBJ@4,-,OBJ@2)
  LK(!TRMOVD)
 IF(OBJ@5,EQ,0,:CCC6)
  ML(22,@TRMOV)
  MU(38,@TRMOVV)
  LK(!GOBF)
  JP(:C1)
:CCC6
  ML(24,@TRMOV)
  MU(35,@TRMOVV)
  SV(@ORDER,OBJ@6)
  MU(20,@TRMOVV)
:WT5
  DN
 JB(1,*CRANE,:WT5)
  PO(*CRANE)
  MU(1,@TRMOVV)
  ST(175)
  CL(*CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(*MSG2,@CNT4)
  MU(19,@TRMOVV)
  MR(2,@TRMOV)
  JP(:C1)
;
;
;*******************TRUCK MOVES TO BUFFER AREA FROM CRANE *******************
:BF1
  IV(@CHANUM)
  MR(6,@TRMOV)
:BF2
  IV(@N3)                         ; FIND AN EMPTY SPACE FOR CHASSIS
```

```
  SV(@S5,@@BUFFER(1,@N3))
  IF(@S5,NE,0,:BF2)
  SV(OBJ@4,@N3)
  SV(@@BUFFER(1,@N3),1)
  SV(@N3,0)
  MD(1,@TRMOVV)
  LK(!TRMOVD)
  ML(4,@TRMOV)
  ST(10)                        ; DISCONNECT THE EMPTY CHASSIS
  MR(2,@TRMOV)
  LK(!PUTCHA)
:BF3
  IV(@N4)
  SV(@S7,@@BUFFER(1,@N4))
 IF(@S7,NE,OBJ@6,:BF3)          ; SEARCH THE CONTAINER TO PICK OUT
  SV(OBJ@5,@N4)
  SV(@N4,0)
  IF(OBJ@5,GT,OBJ@4,:BF33)      ; MOVE UP OR DOWN ?
    SV(OBJ@3,OBJ@5)
    AO(OBJ@4,-,OBJ@5)
    SV(OBJ@5,OBJ@4)
    LK(!TRMOVU)
    ML(2,@TRMOV)
    ST(10)
    SV(OBJ@ID,067)
    MR(4,@TRMOV)
    SV(@@BUFFER(1,OBJ@3),0)
    LK(!CLEAR)
    SV(OBJ@4,36)
    AO(OBJ@4,-,OBJ@3)
    DV(OBJ@4)
    LK(!TRMOVD)
    MD(1,@TRMOVV)
    ML(8,@TRMOV)
    MU(17,@TRMOVV)
    JP(:WT6)
:BF33
    MR(2,@TRMOV)
    SV(OBJ@3,OBJ@5)
    AO(OBJ@5,-,OBJ@4)
    SV(OBJ@4,OBJ@5)
    LK(!TRMOVD)
    ML(4,@TRMOV)
    ST(10)
    SV(OBJ@ID,067)
    MR(4,@TRMOV)
    SV(@@BUFFER(1,OBJ@3),0)
    LK(!CLEAR)
    SV(OBJ@4,36)
    AO(OBJ@4,-,OBJ@3)
    DV(OBJ@4)
    LK(!TRMOVD)
    MD(1,@TRMOVV)
    ML(8,@TRMOV)
```

```
    SV(@ORDER,OBJ@6)
    MU(17,@TRMOVV)
;
:WT6
  DN
 JB(1,#CRANE,:WT6)             ; WAIT IF CRANE IS BUSY
  PO(#CRANE)
  MU(1,@TRMOVV)
  ST(175)                      ; CRANE WORK
  CL(#CRANE)
  SV(OBJ@ID,254)
  IV(@CNT4)
  PV(#MSG2,@CNT4)              ; DISPLAY TOTAL NUMBER LOADED
  MU(19,@TRMOVV)
  MR(2,@TRMOV)
  JB(1,#FINAL,:CT12)           ; CHECK IF LAST CONTAINERS ARE IN BUFFER
  JP(:C1)                      ; BEFORE TERMINATE OPERATION
:C11
  DN
  JC(1,#FINAL,:C11)
    SV(@ORDER,500)
:CT12
    IF(@CNT4,GE,@A5,:CT13)
     SV(@KK3,0)
:CT14
     IV(@KK3)                  ; THIS PORTION IS FOR THE CASE WHEN LAST
     IF(@KK3,GT,100,:CT13)     ; SEVERAL CONTAINERS ARE TO BE TRANSFERED
     SV(@KK4,@@BFJOB(0,@KK3))  ; FROM BUFFER AREA.
     IF(@KK4,EQ,0,:CT14)
       SV(@KK1,@@BFJOB(1,@KK3))
       IF(@KK1,EQ,1,:CT14)
       SV(OBJ@6,@@BFJOB(0,@KK3))
       SV(@@BFJOB(1,@KK3),1)
     JP(:BF1)
:CT13
  PO(#DATA)
  MA(#EXIT,0)
ER
;
```

```
;******************* ROUTING 4 **************************************************
BR(4,*NUMINIT,0)
  IF(@SWITCH,EQ,0,:WW5)           ; CHECK IF USER WANTED A OUTPUT FILE
:WW
  DN
  JC(1,*DATA,:WW)                 ; GENERATE OUTPUT FILE AFTER LOADING
  SV(@CNT1,1)                     ; OPERATION IS COMPLETED
  PV(F,@SWEEP)
  PV(F,@CHANUM)
:A52
  SV(@A8,@@PLAN(0,@CNT1))   PV(F,@A8,)
  SV(@A1,@@PLAN(1,@CNT1))   PV(F,@A1,)
  SV(@A2,@@PLAN(2,@CNT1))   PV(F,@A2,)
  SV(@A3,@@PLAN(3,@CNT1))   PV(F,@A3,)
; SV(@A4,@@PLAN(4,@CNT1))   PV(F,@A4,)
; SV(@A6,@@PLAN(5,@CNT1))   PV(F,@A6,)
; SV(@C1,@@PLAN(6,@CNT1))   PV(F,@C1,)
; SV(@C2,@@PLAN(7,@CNT1))   PV(F,@C2,)
; SV(@C3,@@PLAN(8,@CNT1))   PV(F,@C3,)
; SV(@C4,@@PLAN(9,@CNT1))   PV(F,@C4,)
; SV(@C6,@@PLAN(10,@CNT1))  PV(F,@C6,)
; SV(@C7,@@PLAN(11,@CNT1))  PV(F,@C7,)
  SV(@A9,@@PLAN(12,@CNT1))  PV(F,@A9)
; SV(@B9,@@PLAN(13,@CNT1))  PV(F,@B9,)
; SV(@B1,@@PLAN(14,@CNT1))  PV(F,@B1,)
; SV(@B2,@@PLAN(15,@CNT1))  PV(F,@B2,)
; SV(@B3,@@PLAN(16,@CNT1))  PV(F,@B3)
  IV(@CNT1)
 IF(@CNT1,LE,@A5,:A52)
:WW5
  MA(*EXIT,0)
ER
;
```

```
;
;******************* LINK DEFINITIONS ******************************************
;
BL(!TRMOVD)                          ;THIS LINK IS FOR TRUCK'S DOWNWARD MOVEMEN
  IF(OBJ@4,NE,1,:C20)
     MD(1,@TRMOVV)  JP(:C60)
:C20
  IF(OBJ@4,NE,2,:C21)
     MD(2,@TRMOVV)  JP(:C60)
:C21
  IF(OBJ@4,NE,3,:C22)
     MD(3,@TRMOVV)  JP(:C60)
:C22
  IF(OBJ@4,NE,4,:C23)
     MD(4,@TRMOVV)  JP(:C60)
:C23
  IF(OBJ@4,NE,5,:C24)
     MD(5,@TRMOVV)  JP(:C60)
:C24
  IF(OBJ@4,NE,6,:C25)
     MD(6,@TRMOVV)  JP(:C60)
:C25
  IF(OBJ@4,NE,7,:C26)
     MD(7,@TRMOVV)  JP(:C60)
:C26
  IF(OBJ@4,NE,8,:C27)
     MD(8,@TRMOVV)  JP(:C60)
:C27
  IF(OBJ@4,NE,9,:C28)
     MD(9,@TRMOVV)  JP(:C60)
:C28
  IF(OBJ@4,NE,10,:C29)
     MD(10,@TRMOVV)  JP(:C60)
:C29
  IF(OBJ@4,NE,11,:C30)
     MD(11,@TRMOVV)  JP(:C60)
:C30
  IF(OBJ@4,NE,12,:C31)
     MD(12,@TRMOVV)  JP(:C60)
:C31
  IF(OBJ@4,NE,13,:C32)
     MD(13,@TRMOVV)  JP(:C60)
:C32
  IF(OBJ@4,NE,14,:C33)
     MD(14,@TRMOVV)  JP(:C60)
:C33
  IF(OBJ@4,NE,15,:C34)
     MD(15,@TRMOVV)  JP(:C60)
:C34
  IF(OBJ@4,NE,16,:C35)
     MD(16,@TRMOVV)  JP(:C60)
:C35
  IF(OBJ@4,NE,17,:C36)
     MD(17,@TRMOVV)  JP(:C60)
```

```
:C36
  IF(OBJ@4,NE,18,:C37)
     MD(18,@TRMOVV)  JP(:C60)
:C37
  IF(OBJ@4,NE,19,:C38)
     MD(19,@TRMOVV)  JP(:C60)
:C38
  IF(OBJ@4,NE,20,:C39)
     MD(20,@TRMOVV)  JP(:C60)
:C39
  IF(OBJ@4,NE,21,:C40)
     MD(21,@TRMOVV)  JP(:C60)
:C40
  IF(OBJ@4,NE,22,:C41)
     MD(22,@TRMOVV)  JP(:C60)
:C41
  IF(OBJ@4,NE,23,:C42)
     MD(23,@TRMOVV)  JP(:C60)
:C42
  IF(OBJ@4,NE,24,:C43)
     MD(24,@TRMOVV)  JP(:C60)
:C43
  IF(OBJ@4,NE,25,:C44)
     MD(25,@TRMOVV)  JP(:C60)
:C44
  IF(OBJ@4,NE,26,:C45)
     MD(26,@TRMOVV)  JP(:C60)
:C45
  IF(OBJ@4,NE,27,:C46)
     MD(27,@TRMOVV)  JP(:C60)
:C46
  IF(OBJ@4,NE,28,:C47)
     MD(28,@TRMOVV)  JP(:C60)
:C47
  IF(OBJ@4,NE,29,:C48)
     MD(29,@TRMOVV)  JP(:C60)
:C48
  IF(OBJ@4,NE,30,:C49)
     MD(30,@TRMOVV)  JP(:C60)
:C49
  IF(OBJ@4,NE,31,:C50)
     MD(31,@TRMOVV)  JP(:C60)
:C50
  IF(OBJ@4,NE,32,:C51)
     MD(32,@TRMOVV)  JP(:C60)
:C51
  IF(OBJ@4,NE,33,:C52)
     MD(33,@TRMOVV)  JP(:C60)
:C52
  IF(OBJ@4,NE,34,:C53)
     MD(34,@TRMOVV)  JP(:C60)
:C53
  IF(OBJ@4,NE,35,:C54)
     MD(35,@TRMOVV)  JP(:C60)
```

```
:C54
  IF(OBJ@4,NE,36,:C55)
     MD(36,@TRMOVV)  JP(:C60)
:C55
  IF(OBJ@4,NE,37,:C56)
     MD(37,@TRMOVV)  JP(:C60)
:C56
  IF(OBJ@4,NE,38,:C57)
     MD(38,@TRMOVV)  JP(:C60)
:C57
     MD(39,@TRMOVV)
:C60
EL
;
;
;
BL(!TRMOVU)         ; THIS LINK IS FOR TRUCK'S UPWARD MOVEMENT
  IF(OBJ@5,NE,1,:U20)
     MU(1,@TRMOVV)  JP(:U60)
:U20
  IF(OBJ@5,NE,2,:U21)
     MU(2,@TRMOVV)  JP(:U60)
:U21
  IF(OBJ@5,NE,3,:U22)
     MU(3,@TRMOVV)  JP(:U60)
:U22
  IF(OBJ@5,NE,4,:U23)
     MU(4,@TRMOVV)  JP(:U60)
:U23
  IF(OBJ@5,NE,5,:U24)
     MU(5,@TRMOVV)  JP(:U60)
:U24
  IF(OBJ@5,NE,6,:U25)
     MU(6,@TRMOVV)  JP(:U60)
:U25
  IF(OBJ@5,NE,7,:U26)
     MU(7,@TRMOVV)  JP(:U60)
:U26
  IF(OBJ@5,NE,8,:U27)
     MU(8,@TRMOVV)  JP(:U60)
:U27
  IF(OBJ@5,NE,9,:U28)
     MU(9,@TRMOVV)  JP(:U60)
:U28
  IF(OBJ@5,NE,10,:U29)
     MU(10,@TRMOVV)  JP(:U60)
:U29
  IF(OBJ@5,NE,11,:U30)
     MU(11,@TRMOVV)  JP(:U60)
:U30
  IF(OBJ@5,NE,12,:U31)
     MU(12,@TRMOVV)  JP(:U60)
:U31
  IF(OBJ@5,NE,13,:U32)
```

```
    MU(13,@TRMOVV)  JP(:U60)
:U32
  IF(OBJ@5,NE,14,:U33)
    MU(14,@TRMOVV)  JP(:U60)
:U33
  IF(OBJ@5,NE,15,:U34)
    MU(15,@TRMOVV)  JP(:U60)
:U34
  IF(OBJ@5,NE,16,:U35)
    MU(16,@TRMOVV)  JP(:U60)
:U35
  IF(OBJ@5,NE,17,:U36)
    MU(17,@TRMOVV)  JP(:U60)
:U36
  IF(OBJ@5,NE,18,:U37)
    MU(18,@TRMOVV)  JP(:U60)
:U37
  IF(OBJ@5,NE,19,:U38)
    MU(19,@TRMOVV)  JP(:U60)
:U38
  IF(OBJ@5,NE,20,:U39)
    MU(20,@TRMOVV)  JP(:U60)
:U39
  IF(OBJ@5,NE,21,:U40)
    MU(21,@TRMOVV)  JP(:U60)
:U40
  IF(OBJ@5,NE,22,:U41)
    MU(22,@TRMOVV)  JP(:U60)
:U41
  IF(OBJ@5,NE,23,:U42)
    MU(23,@TRMOVV)  JP(:U60)
:U42
  IF(OBJ@5,NE,24,:U43)
    MU(24,@TRMOVV)  JP(:U60)
:U43
  IF(OBJ@5,NE,25,:U44)
    MU(25,@TRMOVV)  JP(:U60)
:U44
  IF(OBJ@5,NE,26,:U45)
    MU(26,@TRMOVV)  JP(:U60)
:U45
  IF(OBJ@5,NE,27,:U46)
    MU(27,@TRMOVV)  JP(:U60)
:U46
  IF(OBJ@5,NE,28,:U47)
    MU(28,@TRMOVV)  JP(:U60)
:U47
  IF(OBJ@5,NE,29,:U48)
    MU(29,@TRMOVV)  JP(:U60)
:U48
  IF(OBJ@5,NE,30,:U49)
    MU(30,@TRMOVV)  JP(:U60)
:U49
  IF(OBJ@5,NE,31,:U50)
```

```
     MU(31,@TRMOVV)  JP(:U60)
:U50
  IF(OBJ@5,NE,32,:U51)
     MU(32,@TRMOVV)  JP(:U60)
:U51
  IF(OBJ@5,NE,33,:U52)
     MU(33,@TRMOVV)  JP(:U60)
:U52
  IF(OBJ@5,NE,34,:U53)
     MU(34,@TRMOVV)  JP(:U60)
:U53
  IF(OBJ@5,NE,35,:U54)
     MU(35,@TRMOVV)  JP(:U60)
:U54
  IF(OBJ@5,NE,36,:U55)
     MU(36,@TRMOVV)  JP(:U60)
:U55
  IF(OBJ@5,NE,37,:U56)
     MU(37,@TRMOVV)  JP(:U60)
:U56
  IF(OBJ@5,NE,38,:U57)
     MU(38,@TRMOVV)  JP(:U60)
:U57
     MU(39,@TRMOVV)
:U60
EL
;
;
;
BL(!PUTCON)          ; THIS LINK IS FOT DISPLAY OF CONTAINER IN BUFFER SPACE
  IF(OBJ@3,NE,1,:P2)      PM(#BF1,C)  JP(:P35)
:P2
  IF(OBJ@3,NE,2,:P3)      PM(#BF2,C)  JP(:P35)
:P3
  IF(OBJ@3,NE,3,:P4)      PM(#BF3,C)  JP(:P35)
:P4
  IF(OBJ@3,NE,4,:P5)      PM(#BF4,C)  JP(:P35)
:P5
  IF(OBJ@3,NE,5,:P6)      PM(#BF5,C)  JP(:P35)
:P6
  IF(OBJ@3,NE,6,:P7)      PM(#BF6,C)  JP(:P35)
:P7
  IF(OBJ@3,NE,7,:P8)      PM(#BF7,C)  JP(:P35)
:P8
  IF(OBJ@3,NE,8,:P9)      PM(#BF8,C)  JP(:P35)
:P9
  IF(OBJ@3,NE,9,:P10)     PM(#BF9,C)  JP(:P35)
:P10
  IF(OBJ@3,NE,10,:P11)    PM(#BF10,C)  JP(:P35)
:P11
  IF(OBJ@3,NE,11,:P12)    PM(#BF11,C)  JP(:P35)
:P12
  IF(OBJ@3,NE,12,:P13)    PM(#BF12,C)  JP(:P35)
:P13
```

```
  IF(OBJ@3,NE,13,:P14)    PM(#BF13,C)  JP(:P35)
:P14
  IF(OBJ@3,NE,14,:P15)    PM(#BF14,C)  JP(:P35)
:P15
  IF(OBJ@3,NE,15,:P16)    PM(#BF15,C)  JP(:P35)
:P16
  IF(OBJ@3,NE,16,:P17)    PM(#BF16,C)  JP(:P35)
:P17
  IF(OBJ@3,NE,17,:P18)    PM(#BF17,C)  JP(:P35)
:P18
  IF(OBJ@3,NE,18,:P19)    PM(#BF18,C)  JP(:P35)
:P19
  IF(OBJ@3,NE,19,:P20)    PM(#BF19,C)  JP(:P35)
:P20
  IF(OBJ@3,NE,20,:P21)    PM(#BF20,C)  JP(:P35)
:P21
  IF(OBJ@3,NE,21,:P22)    PM(#BF21,C)  JP(:P35)
:P22
  IF(OBJ@3,NE,22,:P23)    PM(#BF22,C)  JP(:P35)
:P23
  IF(OBJ@3,NE,23,:P24)    PM(#BF23,C)  JP(:P35)
:P24
  IF(OBJ@3,NE,24,:P25)    PM(#BF24,C)  JP(:P35)
:P25
  IF(OBJ@3,NE,25,:P26)    PM(#BF25,C)  JP(:P35)
:P26
  IF(OBJ@3,NE,26,:P27)    PM(#BF26,C)  JP(:P35)
:P27
  IF(OBJ@3,NE,27,:P28)    PM(#BF27,C)  JP(:P35)
:P28
  IF(OBJ@3,NE,28,:P29)    PM(#BF28,C)  JP(:P35)
:P29
  IF(OBJ@3,NE,29,:P30)    PM(#BF29,C)  JP(:P35)
:P30
  IF(OBJ@3,NE,30,:P31)    PM(#BF30,C)  JP(:P35)
:P31
  IF(OBJ@3,NE,31,:P32)    PM(#BF31,C)  JP(:P35)
:P32
  IF(OBJ@3,NE,32,:P33)    PM(#BF32,C)  JP(:P35)
:P33
  IF(OBJ@3,NE,33,:P34)    PM(#BF33,C)  JP(:P35)
:P34
                     PM(#BF34,C)
:P35
EL
;
;
;
BL(!PUTCHA)          ; THIS LINK IS FOR DISPLAY OF CHASSIS IN FUFFER SPACE
  IF(OBJ@4,NE,1,:Q2)      PM(#BF1,)  JP(:Q35)
:Q2
  IF(OBJ@4,NE,2,:Q3)      PM(#BF2,)  JP(:Q35)
:Q3
  IF(OBJ@4,NE,3,:Q4)      PM(#BF3,)  JP(:Q35)
```

```
:Q4
 IF(OBJ@4,NE,4,:Q5)      PM(#BF4,)  JP(:Q35)
:Q5
 IF(OBJ@4,NE,5,:Q6)      PM(#BF5,)  JP(:Q35)
:Q6
 IF(OBJ@4,NE,6,:Q7)      PM(#BF6,)  JP(:Q35)
:Q7
 IF(OBJ@4,NE,7,:Q8)      PM(#BF7,)  JP(:Q35)
:Q8
 IF(OBJ@4,NE,8,:Q9)      PM(#BF8,)  JP(:Q35)
:Q9
 IF(OBJ@4,NE,9,:Q10)     PM(#BF9,)  JP(:Q35)
:Q10
 IF(OBJ@4,NE,10,:Q11)    PM(#BF10,)  JP(:Q35)
:Q11
 IF(OBJ@4,NE,11,:Q12)    PM(#BF11,)  JP(:Q35)
:Q12
 IF(OBJ@4,NE,12,:Q13)    PM(#BF12,)  JP(:Q35)
:Q13
 IF(OBJ@4,NE,13,:Q14)    PM(#BF13,)  JP(:Q35)
:Q14
 IF(OBJ@4,NE,14,:Q15)    PM(#BF14,)  JP(:Q35)
:Q15
 IF(OBJ@4,NE,15,:Q16)    PM(#BF15,)  JP(:Q35)
:Q16
 IF(OBJ@4,NE,16,:Q17)    PM(#BF16,)  JP(:Q35)
:Q17
 IF(OBJ@4,NE,17,:Q18)    PM(#BF17,)  JP(:Q35)
:Q18
 IF(OBJ@4,NE,18,:Q19)    PM(#BF18,)  JP(:Q35)
:Q19
 IF(OBJ@4,NE,19,:Q20)    PM(#BF19,)  JP(:Q35)
:Q20
 IF(OBJ@4,NE,20,:Q21)    PM(#BF20,)  JP(:Q35)
:Q21
 IF(OBJ@4,NE,21,:Q22)    PM(#BF21,)  JP(:Q35)
:Q22
 IF(OBJ@4,NE,22,:Q23)    PM(#BF22,)  JP(:Q35)
:Q23
 IF(OBJ@4,NE,23,:Q24)    PM(#BF23,)  JP(:Q35)
:Q24
 IF(OBJ@4,NE,24,:Q25)    PM(#BF24,)  JP(:Q35)
:Q25
 IF(OBJ@4,NE,25,:Q26)    PM(#BF25,)  JP(:Q35)
:Q26
 IF(OBJ@4,NE,26,:Q27)    PM(#BF26,)  JP(:Q35)
:Q27
 IF(OBJ@4,NE,27,:Q28)    PM(#BF27,)  JP(:Q35)
:Q28
 IF(OBJ@4,NE,28,:Q29)    PM(#BF28,)  JP(:Q35)
:Q29
 IF(OBJ@4,NE,29,:Q30)    PM(#BF29,)  JP(:Q35)
:Q30
 IF(OBJ@4,NE,30,:Q31)    PM(#BF30,)  JP(:Q35)
```

```
:Q31
  IF(OBJ@4,NE,31,:Q32)     PM(#BF31,)  JP(:Q35)
:Q32
  IF(OBJ@4,NE,32,:Q33)     PM(#BF32,)  JP(:Q35)
:Q33
  IF(OBJ@4,NE,33,:Q34)     PM(#BF33,)  JP(:Q35)
:Q34
                     PM(#BF34,)
:Q35
EL
;
;
;
BL(!CLEAR)           ; THIS IS FOR ERASING CONTAINER OR CHASSIS FROM BUFFER
  IF(OBJ@3,NE,1,:T2)      PM(#BF1, )  JP(:T35)
:T2
  IF(OBJ@3,NE,2,:T3)      PM(#BF2, )  JP(:T35)
:T3
  IF(OBJ@3,NE,3,:T4)      PM(#BF3, )  JP(:T35)
:T4
  IF(OBJ@3,NE,4,:T5)      PM(#BF4, )  JP(:T35)
:T5
  IF(OBJ@3,NE,5,:T6)      PM(#BF5, )  JP(:T35)
:T6
  IF(OBJ@3,NE,6,:T7)      PM(#BF6, )  JP(:T35)
:T7
  IF(OBJ@3,NE,7,:T8)      PM(#BF7, )  JP(:T35)
:T8
  IF(OBJ@3,NE,8,:T9)      PM(#BF8, )  JP(:T35)
:T9
  IF(OBJ@3,NE,9,:T10)     PM(#BF9, )  JP(:T35)
:T10
  IF(OBJ@3,NE,10,:T11)    PM(#BF10, )  JP(:T35)
:T11
  IF(OBJ@3,NE,11,:T12)    PM(#BF11, )  JP(:T35)
:T12
  IF(OBJ@3,NE,12,:T13)    PM(#BF12, )  JP(:T35)
:T13
  IF(OBJ@3,NE,13,:T14)    PM(#BF13, )  JP(:T35)
:T14
  IF(OBJ@3,NE,14,:T15)    PM(#BF14, )  JP(:T35)
:T15
  IF(OBJ@3,NE,15,:T16)    PM(#BF15, )  JP(:T35)
:T16
  IF(OBJ@3,NE,16,:T17)    PM(#BF16, )  JP(:T35)
:T17
  IF(OBJ@3,NE,17,:T18)    PM(#BF17, )  JP(:T35)
:T18
  IF(OBJ@3,NE,18,:T19)    PM(#BF18, )  JP(:T35)
:T19
  IF(OBJ@3,NE,19,:T20)    PM(#BF19, )  JP(:T35)
:T20
  IF(OBJ@3,NE,20,:T21)    PM(#BF20, )  JP(:T35)
:T21
```

```
  IF(OBJ@3,NE,21,:T22)    PM(#BF21, )  JP(:T35)
:T22
  IF(OBJ@3,NE,22,:T23)    PM(#BF22, )  JP(:T35)
:T23
  IF(OBJ@3,NE,23,:T24)    PM(#BF23, )  JP(:T35)
:T24
  IF(OBJ@3,NE,24,:T25)    PM(#BF24, )  JP(:T35)
:T25
  IF(OBJ@3,NE,25,:T26)    PM(#BF25, )  JP(:T35)
:T26
  IF(OBJ@3,NE,26,:T27)    PM(#BF26, )  JP(:T35)
:T27
  IF(OBJ@3,NE,27,:T28)    PM(#BF27, )  JP(:T35)
:T28
  IF(OBJ@3,NE,28,:T29)    PM(#BF28, )  JP(:T35)
:T29
  IF(OBJ@3,NE,29,:T30)    PM(#BF29, )  JP(:T35)
:T30
  IF(OBJ@3,NE,30,:T31)    PM(#BF30, )  JP(:T35)
:T31
  IF(OBJ@3,NE,31,:T32)    PM(#BF31, )  JP(:T35)
:T32
  IF(OBJ@3,NE,32,:T33)    PM(#BF32, )  JP(:T35)
:T33
  IF(OBJ@3,NE,33,:T34)    PM(#BF33, )  JP(:T35)
:T34
                    PM(#BF34, )
:T35
EL
;
;
;
BL(!NUMBER)          ;THIS LINK IS USED TO DISPLAY TOTAL NUMBER OF CONTAINERS IN
   IF(OBJ@1,NE,1,:ROW2)               ; EACH ROWS
     IF(OBJ@2,NE,1,:G102)   PV(#N11,OBJ@6)   JP(:DONE)
:G102
     IF(OBJ@2,NE,2,:G103)   PV(#N12,OBJ@6)   JP(:DONE)
:G103
     IF(OBJ@2,NE,3,:G104)   PV(#N13,OBJ@6)   JP(:DONE)
:G104
     IF(OBJ@2,NE,4,:G105)   PV(#N14,OBJ@6)   JP(:DONE)
:G105
     IF(OBJ@2,NE,5,:G106)   PV(#N15,OBJ@6)   JP(:DONE)
:G106
     IF(OBJ@2,NE,6,:G107)   PV(#N16,OBJ@6)   JP(:DONE)
:G107
     IF(OBJ@2,NE,7,:G108)   PV(#N17,OBJ@6)   JP(:DONE)
:G108
     IF(OBJ@2,NE,8,:G109)   PV(#N18,OBJ@6)   JP(:DONE)
:G109
     IF(OBJ@2,NE,9,:G110)   PV(#N19,OBJ@6)   JP(:DONE)
:G110
     IF(OBJ@2,NE,10,:G111)  PV(#N110,OBJ@6)  JP(:DONE)
:G111
```

```
     IF(OBJ@2,NE,11,:G112)  PV(#N111,OBJ@6)  JP(:DONE)
:G112
     IF(OBJ@2,NE,12,:G113)  PV(#N112,OBJ@6)  JP(:DONE)
:G113
     IF(OBJ@2,NE,13,:G114)  PV(#N113,OBJ@6)  JP(:DONE)
:G114
     IF(OBJ@2,NE,14,:G115)  PV(#N114,OBJ@6)  JP(:DONE)
:G115
     IF(OBJ@2,NE,15,:G116)  PV(#N115,OBJ@6)  JP(:DONE)
:G116
     IF(OBJ@2,NE,16,:G117)  PV(#N116,OBJ@6)  JP(:DONE)
:G117
     IF(OBJ@2,NE,17,:G118)  PV(#N117,OBJ@6)  JP(:DONE)
:G118
     IF(OBJ@2,NE,18,:G119)  PV(#N118,OBJ@6)  JP(:DONE)
:G119
     IF(OBJ@2,NE,19,:G120)  PV(#N119,OBJ@6)  JP(:DONE)
:G120
     IF(OBJ@2,NE,20,:G121)  PV(#N120,OBJ@6)  JP(:DONE)
:G121
     IF(OBJ@2,NE,21,:G122)  PV(#N121,OBJ@6)  JP(:DONE)
:G122
     IF(OBJ@2,NE,22,:G123)  PV(#N122,OBJ@6)  JP(:DONE)
:G123
     IF(OBJ@2,NE,23,:G124)  PV(#N123,OBJ@6)  JP(:DONE)
:G124
     IF(OBJ@2,NE,24,:G125)  PV(#N124,OBJ@6)  JP(:DONE)
:G125
     IF(OBJ@2,NE,25,:G126)  PV(#N125,OBJ@6)  JP(:DONE)
:G126
     IF(OBJ@2,NE,26,:G127)  PV(#N126,OBJ@6)  JP(:DONE)
:G127
     IF(OBJ@2,NE,27,:G128)  PV(#N127,OBJ@6)  JP(:DONE)
:G128
     IF(OBJ@2,NE,28,:G129)  PV(#N128,OBJ@6)  JP(:DONE)
:G129
     IF(OBJ@2,NE,29,:G130)  PV(#N129,OBJ@6)  JP(:DONE)
:G130
     IF(OBJ@2,NE,30,:G131)  PV(#N130,OBJ@6)  JP(:DONE)
:G131
     IF(OBJ@2,NE,31,:G132)  PV(#N131,OBJ@6)  JP(:DONE)
:G132
     IF(OBJ@2,NE,32,:G133)  PV(#N132,OBJ@6)  JP(:DONE)
:G133
     IF(OBJ@2,NE,33,:G134)  PV(#N133,OBJ@6)  JP(:DONE)
:G134
                            PV(#N134,OBJ@6)  JP(:DONE)
:ROW2
   IF(OBJ@1,NE,2,:R4)
     IF(OBJ@2,NE,1,:G202)   PV(#N21,OBJ@6)   JP(:DONE)
:G202
     IF(OBJ@2,NE,2,:G203)   PV(#N22,OBJ@6)   JP(:DONE)
:G203
     IF(OBJ@2,NE,3,:G204)   PV(#N23,OBJ@6)   JP(:DONE)
```

```
:G204
     IF(OBJ@2,NE,4,:G205)   PV(#N24,OBJ@6)   JP(:DONE)
:G205
     IF(OBJ@2,NE,5,:G206)   PV(#N25,OBJ@6)   JP(:DONE)
:G206
     IF(OBJ@2,NE,6,:G207)   PV(#N26,OBJ@6)   JP(:DONE)
:G207
     IF(OBJ@2,NE,7,:G208)   PV(#N27,OBJ@6)   JP(:DONE)
:G208
     IF(OBJ@2,NE,8,:G209)   PV(#N28,OBJ@6)   JP(:DONE)
:G209
     IF(OBJ@2,NE,9,:G210)   PV(#N29,OBJ@6)   JP(:DONE)
:G210
     IF(OBJ@2,NE,10,:G211)  PV(#N210,OBJ@6)  JP(:DONE)
:G211
     IF(OBJ@2,NE,11,:G212)  PV(#N211,OBJ@6)  JP(:DONE)
:G212
     IF(OBJ@2,NE,12,:G213)  PV(#N212,OBJ@6)  JP(:DONE)
:G213
     IF(OBJ@2,NE,13,:G214)  PV(#N213,OBJ@6)  JP(:DONE)
:G214
     IF(OBJ@2,NE,14,:G215)  PV(#N214,OBJ@6)  JP(:DONE)
:G215
     IF(OBJ@2,NE,15,:G216)  PV(#N215,OBJ@6)  JP(:DONE)
:G216
     IF(OBJ@2,NE,16,:G217)  PV(#N216,OBJ@6)  JP(:DONE)
:G217
     IF(OBJ@2,NE,17,:G218)  PV(#N217,OBJ@6)  JP(:DONE)
:G218
     IF(OBJ@2,NE,18,:G219)  PV(#N218,OBJ@6)  JP(:DONE)
:G219
     IF(OBJ@2,NE,19,:G220)  PV(#N219,OBJ@6)  JP(:DONE)
:G220
     IF(OBJ@2,NE,20,:G221)  PV(#N220,OBJ@6)  JP(:DONE)
:G221
     IF(OBJ@2,NE,21,:G222)  PV(#N221,OBJ@6)  JP(:DONE)
:G222
     IF(OBJ@2,NE,22,:G223)  PV(#N222,OBJ@6)  JP(:DONE)
:G223
     IF(OBJ@2,NE,23,:G224)  PV(#N223,OBJ@6)  JP(:DONE)
:G224
     IF(OBJ@2,NE,24,:G225)  PV(#N224,OBJ@6)  JP(:DONE)
:G225
     IF(OBJ@2,NE,25,:G226)  PV(#N225,OBJ@6)  JP(:DONE)
:G226
     IF(OBJ@2,NE,26,:G227)  PV(#N226,OBJ@6)  JP(:DONE)
:G227
     IF(OBJ@2,NE,27,:G228)  PV(#N227,OBJ@6)  JP(:DONE)
:G228
     IF(OBJ@2,NE,28,:G229)  PV(#N228,OBJ@6)  JP(:DONE)
:G229
     IF(OBJ@2,NE,29,:G230)  PV(#N229,OBJ@6)  JP(:DONE)
:G230
     IF(OBJ@2,NE,30,:G231)  PV(#N230,OBJ@6)  JP(:DONE)
```

```
:6231
     IF(OBJ@2,NE,31,:6232)  PV(#N231,OBJ@6)  JP(:DONE)
:6232
     IF(OBJ@2,NE,32,:6233)  PV(#N232,OBJ@6)  JP(:DONE)
:6233
     IF(OBJ@2,NE,33,:6234)  PV(#N233,OBJ@6)  JP(:DONE)
:6234
                            PV(#N234,OBJ@6)  JP(:DONE)
:R4
   IF(OBJ@1,NE,4,:ROW5)
     IF(OBJ@2,NE,1,:6402)   PV(#N41,OBJ@6)   JP(:DONE)
:6402
     IF(OBJ@2,NE,2,:6403)   PV(#N42,OBJ@6)   JP(:DONE)
:6403
     IF(OBJ@2,NE,3,:6404)   PV(#N43,OBJ@6)   JP(:DONE)
:6404
     IF(OBJ@2,NE,4,:6405)   PV(#N44,OBJ@6)   JP(:DONE)
:6405
     IF(OBJ@2,NE,5,:6406)   PV(#N45,OBJ@6)   JP(:DONE)
:6406
     IF(OBJ@2,NE,6,:6407)   PV(#N46,OBJ@6)   JP(:DONE)
:6407
     IF(OBJ@2,NE,7,:6408)   PV(#N47,OBJ@6)   JP(:DONE)
:6408
     IF(OBJ@2,NE,8,:6409)   PV(#N48,OBJ@6)   JP(:DONE)
:6409
     IF(OBJ@2,NE,9,:6410)   PV(#N49,OBJ@6)   JP(:DONE)
:6410
     IF(OBJ@2,NE,10,:6411)  PV(#N410,OBJ@6)  JP(:DONE)
:6411
     IF(OBJ@2,NE,11,:6412)  PV(#N411,OBJ@6)  JP(:DONE)
:6412
     IF(OBJ@2,NE,12,:6413)  PV(#N412,OBJ@6)  JP(:DONE)
:6413
     IF(OBJ@2,NE,13,:6414)  PV(#N413,OBJ@6)  JP(:DONE)
:6414
     IF(OBJ@2,NE,14,:6415)  PV(#N414,OBJ@6)  JP(:DONE)
:6415
     IF(OBJ@2,NE,15,:6416)  PV(#N415,OBJ@6)  JP(:DONE)
:6416
     IF(OBJ@2,NE,16,:6417)  PV(#N416,OBJ@6)  JP(:DONE)
:6417
     IF(OBJ@2,NE,17,:6418)  PV(#N417,OBJ@6)  JP(:DONE)
:6418
     IF(OBJ@2,NE,18,:6419)  PV(#N418,OBJ@6)  JP(:DONE)
:6419
     IF(OBJ@2,NE,19,:6420)  PV(#N419,OBJ@6)  JP(:DONE)
:6420
     IF(OBJ@2,NE,20,:6421)  PV(#N420,OBJ@6)  JP(:DONE)
:6421
     IF(OBJ@2,NE,21,:6422)  PV(#N421,OBJ@6)  JP(:DONE)
:6422
     IF(OBJ@2,NE,22,:6423)  PV(#N422,OBJ@6)  JP(:DONE)
:6423
```

```
     IF(OBJ@2,NE,23,:G424)  PV(#N423,OBJ@6)  JP(:DONE)
:G424
     IF(OBJ@2,NE,24,:G425)  PV(#N424,OBJ@6)  JP(:DONE)
:G425
     IF(OBJ@2,NE,25,:G426)  PV(#N425,OBJ@6)  JP(:DONE)
:G426
     IF(OBJ@2,NE,26,:G427)  PV(#N426,OBJ@6)  JP(:DONE)
:G427
     IF(OBJ@2,NE,27,:G428)  PV(#N427,OBJ@6)  JP(:DONE)
:G428
     IF(OBJ@2,NE,28,:G429)  PV(#N428,OBJ@6)  JP(:DONE)
:G429
     IF(OBJ@2,NE,29,:G430)  PV(#N429,OBJ@6)  JP(:DONE)
:G430
     IF(OBJ@2,NE,30,:G431)  PV(#N430,OBJ@6)  JP(:DONE)
:G431
     IF(OBJ@2,NE,31,:G432)  PV(#N431,OBJ@6)  JP(:DONE)
:G432
     IF(OBJ@2,NE,32,:G433)  PV(#N432,OBJ@6)  JP(:DONE)
:G433
     IF(OBJ@2,NE,33,:G434)  PV(#N433,OBJ@6)  JP(:DONE)
:G434
                            PV(#N434,OBJ@6)  JP(:DONE)
:ROW5
   IF(OBJ@1,NE,5,:R3)
     IF(OBJ@2,NE,1,:G502)   PV(#N51,OBJ@6)   JP(:DONE)
:G502
     IF(OBJ@2,NE,2,:G503)   PV(#N52,OBJ@6)   JP(:DONE)
:G503
     IF(OBJ@2,NE,3,:G504)   PV(#N53,OBJ@6)   JP(:DONE)
:G504
     IF(OBJ@2,NE,4,:G505)   PV(#N54,OBJ@6)   JP(:DONE)
:G505
     IF(OBJ@2,NE,5,:G506)   PV(#N55,OBJ@6)   JP(:DONE)
:G506
     IF(OBJ@2,NE,6,:G507)   PV(#N56,OBJ@6)   JP(:DONE)
:G507
     IF(OBJ@2,NE,7,:G508)   PV(#N57,OBJ@6)   JP(:DONE)
:G508
     IF(OBJ@2,NE,8,:G509)   PV(#N58,OBJ@6)   JP(:DONE)
:G509
     IF(OBJ@2,NE,9,:G510)   PV(#N59,OBJ@6)   JP(:DONE)
:G510
     IF(OBJ@2,NE,10,:G511)  PV(#N510,OBJ@6)  JP(:DONE)
:G511
     IF(OBJ@2,NE,11,:G512)  PV(#N511,OBJ@6)  JP(:DONE)
:G512
     IF(OBJ@2,NE,12,:G513)  PV(#N512,OBJ@6)  JP(:DONE)
:G513
     IF(OBJ@2,NE,13,:G514)  PV(#N513,OBJ@6)  JP(:DONE)
:G514
     IF(OBJ@2,NE,14,:G515)  PV(#N514,OBJ@6)  JP(:DONE)
:G515
     IF(OBJ@2,NE,15,:G516)  PV(#N515,OBJ@6)  JP(:DONE)
```

```
:G516
     IF(OBJ@2,NE,16,:G517)  PV(#N516,OBJ@6)  JP(:DONE)
:G517
     IF(OBJ@2,NE,17,:G518)  PV(#N517,OBJ@6)  JP(:DONE)
:G518
     IF(OBJ@2,NE,18,:G519)  PV(#N518,OBJ@6)  JP(:DONE)
:G519
     IF(OBJ@2,NE,19,:G520)  PV(#N519,OBJ@6)  JP(:DONE)
:G520
     IF(OBJ@2,NE,20,:G521)  PV(#N520,OBJ@6)  JP(:DONE)
:G521
     IF(OBJ@2,NE,21,:G522)  PV(#N521,OBJ@6)  JP(:DONE)
:G522
     IF(OBJ@2,NE,22,:G523)  PV(#N522,OBJ@6)  JP(:DONE)
:G523
     IF(OBJ@2,NE,23,:G524)  PV(#N523,OBJ@6)  JP(:DONE)
:G524
     IF(OBJ@2,NE,24,:G525)  PV(#N524,OBJ@6)  JP(:DONE)
:G525
     IF(OBJ@2,NE,25,:G526)  PV(#N525,OBJ@6)  JP(:DONE)
:G526
     IF(OBJ@2,NE,26,:G527)  PV(#N526,OBJ@6)  JP(:DONE)
:G527
     IF(OBJ@2,NE,27,:G528)  PV(#N527,OBJ@6)  JP(:DONE)
:G528
     IF(OBJ@2,NE,28,:G529)  PV(#N528,OBJ@6)  JP(:DONE)
:G529
     IF(OBJ@2,NE,29,:G530)  PV(#N529,OBJ@6)  JP(:DONE)
:G530
     IF(OBJ@2,NE,30,:G531)  PV(#N530,OBJ@6)  JP(:DONE)
:G531
     IF(OBJ@2,NE,31,:G532)  PV(#N531,OBJ@6)  JP(:DONE)
:G532
     IF(OBJ@2,NE,32,:G533)  PV(#N532,OBJ@6)  JP(:DONE)
:G533
     IF(OBJ@2,NE,33,:G534)  PV(#N533,OBJ@6)  JP(:DONE)
:G534
                            PV(#N534,OBJ@6)  JP(:DONE)
:R3
     IF(OBJ@2,NE,31,:G332)  PV(#N331,OBJ@6)  JP(:DONE)
:G332
     IF(OBJ@2,NE,32,:G333)  PV(#N332,OBJ@6)  JP(:DONE)
:G333
     IF(OBJ@2,NE,33,:G334)  PV(#N333,OBJ@6)  JP(:DONE)
:G334
                            PV(#N334,OBJ@6)
:DONE
EL
;
;
;
BL(!TTSCHG)         ; THIS LINK IS FOR TRANSTAINER SECTION CHANGING MOVEMENT
  IF(@TTSEC,NE,1,:B8)
    IF(OBJ@1,NE,2,:B5)
```

```
    SV(@B3,1)  SV(@B4,14)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B5
    IF(OBJ@1,NE,3,:B6)
    SV(@B3,1)  SV(@B4,28)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B6
    IF(OBJ@1,NE,4,:B7)
    SV(@B1,1)               IV(@TADCH6)   PV(#MS65,@TADCH6)
    LK(!TTV)  JP(:B24)
:B7
                            IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TT15)  JP(:B24)
:B8
  IF(@TTSEC,NE,2,:B12)
    IF(OBJ@1,NE,1,:B9)
    SV(@B3,0)  SV(@B4,14)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B9
    IF(OBJ@1,NE,3,:B10)
    SV(@B3,1)  SV(@B4,14)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B10
    IF(OBJ@1,NE,4,:B11)
                            IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TT2435)  JP(:B24)
:B11
    SV(@B1,1)               IV(@TADCH6)   PV(#MS65,@TADCH6)
    LK(!TTV)  JP(:B24)
:B12
  IF(@TTSEC,NE,3,:B16)
    IF(OBJ@1,NE,1,:B13)
    SV(@B3,0)  SV(@B4,28)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B13
    IF(OBJ@1,NE,2,:B14)
    SV(@B3,0)  SV(@B4,14)   IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TTH)  JP(:B24)
:B14
    IF(OBJ@1,NE,4,:B15)
                            IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TT34)  JP(:B24)
:B15
                            IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TT2435)  JP(:B24)
:B16
  IF(@TTSEC,NE,4,:B20)
    IF(OBJ@1,NE,1,:B17)
    SV(@B1,0)               IV(@TADCH6)   PV(#MS65,@TADCH6)
    LK(!TTV)  JP(:B24)
:B17
    IF(OBJ@1,NE,2,:B18)
                            IV(@TNADCH6)  PV(#MS66,@TNADCH6)
    LK(!TT4253)  JP(:B24)
```

```
:B18
    IF(OBJ@1,NE,3,:B19)
                            IV(@TNADCHG)  PV(*MS66,@TNADCHG)
    LK(!TT43)  JP(:B24)
:B19
    SV(@B3,1)  SV(@B4,14)   IV(@TNADCHG)  PV(*MS66,@TNADCHG)
    LK(!TTH)  JP(:B24)
:B20
    IF(OBJ@1,NE,1,:B21)
                            IV(@TNADCHG)  PV(*MS66,@TNADCHG)
    LK(!TT51)  JP(:B24)
:B21
    IF(OBJ@1,NE,2,:B22)
    SV(@B1,0)               IV(@TADCHG)   PV(*MS65,@TADCHG)
    LK(!TTV)  JP(:B24)
:B22
    IF(OBJ@1,NE,3,:B23)
                            IV(@TNADCHG)  PV(*MS66,@TNADCHG)
    LK(!TT4253)  JP(:B24)
:B23
    SV(@B3,0)  SV(@B4,14)   IV(@TNADCHG)  PV(*MS66,@TNADCHG)
    LK(!TTH)
:B24
  SV(@TTSEC,OBJ@1)
EL
;
;
;
BL(!TTH)            ;THIS IS FOR TRANSTAINER HORIZONTAL MOVEMENT, LEFT & RIGHT
  SV(@B1,@TTROW)
  AO(@B1,+,OBJ@2)
  IF(@B1,GT,34,:B27)
     SV(@B2,@TTROW)
     LK(!TTMOVU)
     IF(@B3,EQ,1,:B25)
     LK(!TTMOVL)
     JP(:B26)
:B25
     LK(!TTMOVR)
:B26
     SV(@B2,OBJ@2)
     LK(!TTMOVD)
     JP(:B30)
:B27
     SV(@B2,35)
     AO(@B2,-,@TTROW)
     LK(!TTMOVD)
     IF(@B3,EQ,1,:B28)
     LK(!TTMOVL)
     JP(:B29)
:B28
     LK(!TTMOVR)
:B29
     SV(@B2,35)
```

```
     AD(@B2,-,OBJ@2)
     LK(!TTMOVU)
:B30
     SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TTV)            ; TRANSTAINER'S VERTICAL MOVEMENT ON SCREEN, UP & DOWN
  IF(@B1,EQ,1,:B31)
     SV(@B2,@TTROW)
     AD(@B2,+,38)
     AD(@B2,-,OBJ@2)
     LK(!TTMOVU)
     JP(:B32)
:B31
     SV(@B2,OBJ@2)
     AD(@B2,+,38)
     AD(@B2,-,@TTROW)
     LK(!TTMOVD)
:B32
     SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT15)           ; TRANSTAINER'S MOVEMENT FROM SECTION 1 TO SECTION 5
  SV(@B2,35)
  AD(@B2,-,@TTROW)
  LK(!TTMOVD)
  MR(14,@TTMRL)
  SV(@B2,3)
  AD(@B2,+,OBJ@2)
  LK(!TTMOVD)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT2435)         ; TRANSTAINER MOVEMENT FROM SECTION 2 TO 4, OR 3 TO 5
  SV(@B2,35)
  AD(@B2,-,@TTROW)
  LK(!TTMOVD)
  ML(14,@TTMRL)
  SV(@B2,3)
  AD(@B2,+,OBJ@2)
  LK(!TTMOVD)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT34)           ; TRANSTAINER MOVEMENT FROM SECTION 3 TO 4
  SV(@B2,35)
```

```
  AD(@B2,-,@TTROW)
  LK(!TTMOVD)
  ML(28,@TTMRL)
  SV(@B2,3)
  AD(@B2,+,OBJ@2)
  LK(!TTMOVD)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT43)           ; TRANSTAINER MOVEMENT FROM SECTION 4 TO 3
  SV(@B2,@TTROW)
  LK(!TTMOVU)
  MR(28,@TTMRL)
  SV(@B2,38)
  AD(@B2,-,OBJ@2)
  LK(!TTMOVU)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT51)           ; TRANSTAINER MOVEMENT FROM SECTION 5 TO 1
  SV(@B2,@TTROW)
  LK(!TTMOVU)
  ML(14,@TTMRL)
  SV(@B2,38)
  AD(@B2,-,OBJ@2)
  LK(!TTMOVU)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TT4253)         ; FROM SECTION 4 TO 2, OR 5 TO 3
  SV(@B2,@TTROW)
  LK(!TTMOVU)
  MR(14,@TTMRL)
  SV(@B2,38)
  AD(@B2,-,OBJ@2)
  LK(!TTMOVU)
  SV(@TTROW,OBJ@2)
EL
;
;
;
BL(!TTMOVU)         ; TRANSTAINER'S UPWARD MOVEMENT ON SCREEN
:B40
  MU(1,@TTMOV)
  IV(@K1)
  IV(@TROWCHG)
  PV(#MS64,@TROWCHG)
  IF(@K1,LT,@B2,:B40)
```

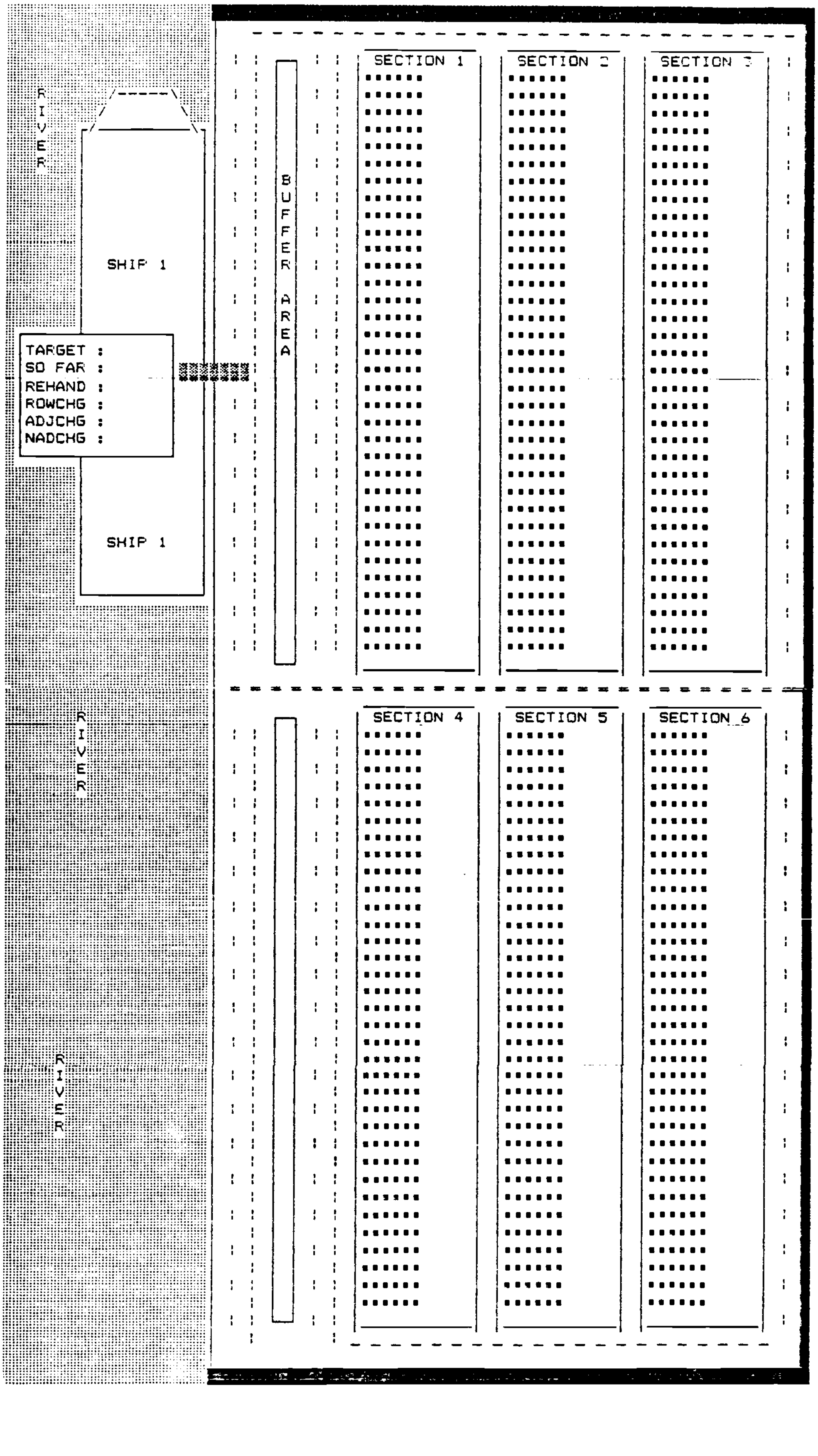
```
  SV(@K1,0)
EL
;
;
;
BL(!TTMOVD)          ; TRANSTAINER'S DOWNWARD MOVEMENT ON SCREEN
:B41
  MD(1,@TTMOV)
  IV(@K2)
  IV(@TROWCHG)
  PV(*MSG4,@TROWCHG)
  IF(@K2,LT,@B2,:B41)
  SV(@K2,0)
EL
;
;
;
BL(!TTMOVL)          ; TRANSTAINER'S LEFT MOVEMENT
:B42
  ML(1,@TTMRL)
  IV(@K3)
  IF(@K3,LT,@B4,:B42)
  SV(@K3,0)
EL
;
;
;
BL(!TTMOVR)          ; TRANSTAINER'S RIGHT MOVEMENT
:B43
  MR(1,@TTMRL)
  IV(@K4)
  IF(@K4,LT,@B4,:B43)
  SV(@K4,0)
EL
;
;
;
BL(!GOBF)            ; THIS LINK IS FOR TRUCKS'S MOVING TO BUFFER AREA FROM
:CCC1                ; STORAGE SECTIONS
    IV(@N1)
    SV(@S1,@@BUFFER(1,@N1))
    IF(@S1,NE,0,:CCC1)
    SV(@@BUFFER(1,@N1),OBJ@6)
    SV(@@PLAN(10,OBJ@6),1)
    SV(OBJ@5,36)
    AO(OBJ@5,-,@N1)
    SV(OBJ@3,@N1)
    SV(@N1,0)
    LK(!TRMOVU)
    MR(2,@TRMOV)
    SV(@RRR,@@PLAN(11,OBJ@6))
    IF(@RRR,EQ,0,:CCCC8)
    IV(@CHANUM)
    MR(2,@TRMOV)
```

```
    SV(OBJ@4,OBJ@5)
    SV(@@BUFFER(1,OBJ@3),0)
    LK(!TRMOVD)
    SV(@ORDER,OBJ@6)
    ML(6,@TRMOV)
    MU(17,@TRMOVV)
:WTO
    DN
    JB(1,#CRANE,:WTO)
    PO(#CRANE)
    MU(1,@TRMOVV)
    ST(175)
    CL(#CRANE)
    SV(OBJ@ID,254)
    IV(@CNT4)
    PV(#MSG2,@CNT4)
    MU(19,@TRMOVV)
    MR(2,@TRMOV)
    JP(:C1)
:CCCC8
    ST(10)
    SV(OBJ@ID,254)
    MR(2,@TRMOV)
    LK(!PUTCON)
    SV(@S2,OBJ@5)
    SV(OBJ@5,36)
    AO(OBJ@5,-,@S2)
:CCCC9
    IV(@N7)
    SV(@S2,@@BUFFER(1,@N7))
    IF(@S2,NE,1,:CCCC9)
    SV(OBJ@6,@N7)
    SV(@@BUFFER(1,@N7),0)
    SV(@N7,0)
    IF(OBJ@6,GT,OBJ@5,:CCCC10)
      AO(OBJ@5,-,OBJ@6)
      SV(OBJ@3,OBJ@6)
      LK(!TRMOVU)
      ML(2,@TRMOVV)
      ST(10)
      ML(2,@TRMOV)
      LK(!CLEAR)
      SV(OBJ@5,OBJ@6)
      LK(!TRMOVU)
      MU(1,@TRMOVV)
      JP(:ENDLK)
:CCCC10
      MR(2,@TRMOV)
      SV(OBJ@3,OBJ@6)
      AO(OBJ@6,-,OBJ@5)
      SV(OBJ@4,OBJ@6)
      LK(!TRMOVD)
      ML(4,@TRMOV)
      ST(10)
```

```
    ML(2,@TRMOV)
    LK(!CLEAR)
    SV(OBJ@5,OBJ@3)
    LK(!TRMOVU)
    MU(1,@TRMOVV)
:ENDLK
EL
;
```

APPENDIX E

Overlay of Program YARD


RIVER
SHIP 1
TARGET :
SO FAR :
REHAND :
ROWCHG :
ADJCHG :
NADCHG :
SHIP 1
BUFFER AREA
SECTION 1
SECTION 2
SECTION 3
SECTION 4
SECTION 5
SECTION 6
RIVER
RIVER

APPENDIX F

User Input

User input specifies the rows of the sections that will contain the containers for the voyage and the length of the container for each specified rows. Values for the lenght can assume one of the following four values:

0 - the row doesn't have containers for the voyage
20 - the row has 20 ft. containers for the voyage
40 - the row has 40 ft. containers for the voyage
41 - the row is used as an extra space for 40 ft. containers

User Input for Plan A

SECTION ROW NO	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	20	0	0	40	20
5	40	0	0	41	40
6	41	0	0	20	41
7	0	0	0	20	0
8	20	0	0	0	0
9	40	0	0	0	0
10	41	0	0	0	40
11	0	0	0	20	41
12	20	20	0	40	20
13	20	40	0	41	20
14	0	41	0	0	0
15	40	20	0	0	0
16	41	20	0	0	0
17	0	0	0	20	0
18	20	20	0	0	0
19	0	20	0	0	0
20	0	0	0	20	0
21	0	0	0	40	0
22	20	40	0	41	0
23	0	41	0	0	0
24	0	20	0	20	0
25	20	40	0	0	0
26	20	41	0	0	0
27	40	0	0	0	0
28	41	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	20	0	0
33	0	0	40	0	0
34	0	0	41	0	0

User Input for Plan B

SECTION ROW NO.	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	20	0	0	40	20
5	40	0	0	41	40
6	41	0	0	20	41
7	0	0	0	20	0
8	20	0	0	0	0
9	40	0	0	0	0
10	41	0	0	0	40
11	0	0	0	20	41
12	20	20	0	40	20
13	20	40	0	41	20
14	0	41	0	0	0
15	40	20	0	0	0
16	41	20	0	0	0
17	0	0	0	20	0
18	20	20	0	0	0
19	0	20	0	0	0
20	0	0	0	20	0
21	0	0	0	40	0
22	20	40	0	41	0
23	0	41	0	0	0
24	0	20	0	20	0
25	20	40	0	0	0
26	20	41	0	0	0
27	40	0	0	0	0
28	41	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	20	0	0
33	0	0	40	0	0
34	0	0	41	0	0

APPEDIX G

Load Plans

Ith row - Ith container in the initial load plan
1st column - number of containers to be rehandled for the Ith container
2nd column - the section number for the Ith container
3rd column - the row number for the Ith container
4th column - the stack number for the Ith container
5th column - the level for the Ith container

Load Plan A					Load Plan B				
0	1	8	1	1	0	1	12	2	1
0	1	26	6	1	0	1	22	4	3
0	1	12	3	3	0	4	20	3	2
0	1	12	1	2	0	4	20	3	1
0	1	9	5	1	0	2	13	3	1
0	1	9	6	1	0	2	22	4	1
0	1	9	4	1	0	2	22	1	2
0	1	9	1	2	0	2	22	6	2
0	1	15	1	2	0	2	25	5	3
0	1	26	3	3	0	2	18	4	2
0	3	32	5	2	0	2	18	6	3
0	3	33	3	3	0	2	18	4	1
0	3	33	1	3	0	2	18	3	2
0	3	33	2	3	0	2	12	2	1
0	3	33	1	2	0	2	15	5	1
0	3	32	4	3	0	2	15	2	1
0	3	32	5	1	0	2	25	1	2
0	3	33	6	3	0	2	25	1	1
0	3	33	6	2	0	2	25	6	1
0	3	33	1	1	0	2	16	2	3
0	3	33	2	2	0	3	33	6	2
0	3	33	2	1	0	3	33	4	2
0	4	4	1	3	0	3	33	5	1
0	3	33	3	2	0	3	32	2	3
0	3	33	6	1	0	3	32	6	3
0	3	33	3	1	0	3	33	4	1
0	5	13	3	3	0	3	33	3	2
0	5	12	6	2	0	3	33	6	1
0	5	12	3	1	0	3	33	3	1
0	5	12	2	3	0	3	33	1	1
0	5	12	5	1	0	3	33	2	3
0	5	12	6	1	0	3	33	2	2
0	5	12	1	2	0	3	33	2	1
0	5	12	2	2	0	3	32	6	2
0	5	12	1	1	0	3	32	2	2
0	5	10	4	3	0	3	32	1	2
0	5	10	1	2	0	3	32	2	1
0	5	10	1	1	0	3	32	1	1
0	5	10	5	1	0	3	32	5	3
0	5	10	2	2	0	3	32	5	2
0	5	10	2	1	0	3	32	5	1
0	5	4	5	2	0	3	32	6	1
0	5	4	6	3	0	2	25	4	2
0	5	10	6	1	0	5	5	5	2

0	5	13	2	2
1	5	5	3	1
0	5	4	2	1
0	5	4	5	1
0	2	13	6	3
1	2	13	2	2
0	2	13	2	2
0	2	13	5	2
0	2	13	6	2
0	2	13	2	1
0	2	19	2	3
0	2	19	4	1
0	2	19	2	2
0	2	16	5	1
0	2	15	3	2
0	2	22	5	1
0	2	22	1	3
0	2	22	6	2
0	2	25	2	1
0	2	25	5	1
0	2	12	5	3
0	2	12	5	2
0	2	12	1	1
0	2	13	4	3
0	2	25	4	1
0	2	15	5	1
0	2	18	4	1
0	2	12	2	2
0	2	12	3	1
0	2	12	4	2
0	2	12	4	1
0	2	13	6	1
0	1	18	2	3
1	1	18	2	1
0	1	18	2	1
0	1	18	5	2
0	1	18	6	2
0	1	27	2	1
0	1	27	5	3
0	1	27	6	1
0	1	27	1	3
0	1	27	5	2
0	1	27	5	1
0	2	24	3	2
2	2	16	4	1
0	2	16	4	2
0	2	16	3	3
0	2	15	3	1
0	1	25	4	2
0	1	12	6	1
1	1	4	4	1
0	1	4	1	1
0	1	4	4	1

0	5	5	5	1
0	5	5	3	2
0	5	12	5	2
0	5	10	4	3
0	5	5	4	2
0	5	5	3	1
0	5	12	2	3
0	5	12	3	1
0	5	12	2	2
0	5	12	2	1
0	5	4	6	2
0	5	4	6	1
0	5	12	5	1
0	5	12	1	2
0	5	12	1	1
0	5	5	4	1
0	5	4	3	2
0	5	13	4	1
0	5	4	2	3
0	5	4	2	2
0	5	13	3	2
0	5	10	1	3
0	5	4	2	1
1	5	10	3	1
0	5	10	1	2
0	5	10	2	3
0	5	10	4	2
0	2	15	3	2
0	2	15	3	1
0	5	10	3	1
0	5	10	2	2
0	4	20	5	2
0	4	20	6	1
0	4	20	2	1
0	4	20	5	1
0	4	17	2	2
1	4	17	4	2
0	4	17	2	1
0	4	21	3	1
0	4	17	3	1
0	4	12	2	3
0	4	12	4	3
0	4	12	4	2
0	4	12	4	1
1	4	7	6	1
0	1	15	3	1
2	1	15	4	1
0	1	15	6	1
0	1	9	6	2
0	1	13	4	3
0	1	9	1	1
0	1	8	5	2
0	1	8	6	3

0	1	22	1	3
0	1	22	2	3
0	4	17	1	2
0	4	12	4	3
0	4	12	2	2
0	4	24	6	2
0	4	24	5	2
0	4	11	2	1
0	4	11	1	1
0	4	12	1	2
0	4	20	4	1
0	4	17	5	3
0	4	17	5	2
0	4	7	2	1
0	4	11	4	1
0	4	11	5	2
0	4	11	5	1
0	4	11	3	1
0	4	24	2	1
0	4	21	2	1
0	4	20	2	2
0	4	20	6	2
0	4	24	6	1
0	1	18	6	1
2	1	5	1	1
0	1	5	4	1
0	1	5	2	2
0	1	5	1	2
0	1	5	3	3
0	1	9	1	1
0	1	9	3	1
0	5	13	2	1
0	5	13	6	3
0	5	13	6	2
0	5	4	3	2
0	5	4	6	2
0	5	4	3	1
0	5	4	6	1
0	5	4	4	3
0	5	4	4	2
0	5	4	4	1
0	5	13	3	2
0	5	10	4	2
0	5	13	6	1
0	5	12	2	1
0	2	13	4	2
0	2	13	4	1
0	2	13	5	1
0	2	13	1	2
0	2	13	1	1
0	2	24	5	2
0	2	24	4	1
0	2	16	1	2

0	1	8	3	3
0	1	8	3	2
0	1	4	4	1
0	1	4	3	1
0	1	25	2	1
0	1	22	5	1
0	1	8	1	1
0	1	4	1	1
0	1	13	3	1
0	1	9	6	1
0	1	9	4	2
0	1	9	3	1
0	1	9	4	1
0	1	5	4	1
0	1	5	5	3
0	1	5	3	2
0	1	22	6	3
0	1	22	6	2
0	1	12	6	1
0	1	12	3	3
0	1	12	3	2
0	1	12	3	1
0	1	13	6	2
0	1	13	1	2
0	1	27	3	3
1	1	8	4	2
0	1	25	5	2
0	1	13	5	3
0	5	13	5	1
0	2	16	2	2
0	2	16	5	2
0	2	15	6	1
0	2	13	4	1
0	2	18	2	1
0	2	25	4	1
0	5	4	3	1
0	5	13	2	2
0	5	13	6	1
0	5	10	6	2
0	5	10	1	1
0	5	10	4	1
0	5	10	6	1
0	5	10	2	1
0	5	13	3	1
0	5	13	2	1
0	2	13	1	3
0	2	13	1	2
0	2	18	3	1
0	2	18	6	2
0	2	18	6	1
0	2	24	6	3
0	2	25	2	3
0	4	7	2	2

0	2	12	5	1
0	2	18	2	1
0	2	18	5	3
0	2	16	3	2
0	2	16	4	1
0	2	18	6	2
0	2	16	1	1
0	2	16	3	1
0	2	22	3	3
0	2	22	1	2
0	2	24	5	1
0	2	24	2	2
0	2	24	3	1
0	2	24	1	2
0	2	19	1	1
0	2	19	2	1
0	2	19	5	1
0	2	19	3	2
0	2	19	3	1
0	4	6	3	2
0	4	6	3	1
0	4	17	2	1
0	4	12	1	1
0	4	12	6	3
0	4	17	4	1
0	4	17	5	1
0	4	17	3	1
0	1	5	1	1
0	1	5	2	1
0	1	5	3	2
0	4	12	4	2
0	4	24	3	2
0	4	20	2	1
0	4	7	4	2
0	4	20	6	1
0	5	5	6	3
0	5	5	6	2
0	5	13	4	3
0	5	13	3	1
0	5	13	4	2
0	2	12	2	1
0	2	18	6	1
0	2	15	1	1
0	2	15	6	2
0	2	24	1	1
0	2	24	2	1
0	5	5	3	1
0	5	5	4	1
0	5	10	4	1
0	5	5	6	1

0	4	7	5	1
0	4	12	3	2
0	4	12	6	3
0	4	12	6	2
0	4	12	6	1
0	4	12	2	2
0	4	12	2	1
0	4	12	3	1
0	4	12	5	1
0	1	13	4	2
0	1	26	1	2
0	1	26	3	2
0	1	8	2	3
0	1	8	6	2
0	1	8	4	2
0	1	8	3	1
0	1	27	2	2
0	1	27	2	1
0	1	18	1	3
0	1	18	3	2
0	1	15	5	1
0	1	15	1	1
0	1	15	4	2
0	1	25	3	1
0	1	9	2	1
0	1	5	2	2
0	1	18	6	1
0	1	18	2	1
0	1	18	5	1
0	1	18	1	2
0	1	25	4	3
0	1	5	2	1
0	1	5	5	2
0	1	8	5	1
0	1	8	6	1
0	1	5	1	2
0	1	5	3	1
0	1	5	1	1
0	1	5	5	1
0	1	15	4	1
0	1	26	5	1
0	1	18	3	1
0	4	20	4	1
0	4	17	4	2
0	4	17	5	3
2	4	6	3	1
0	4	6	5	2
0	4	24	3	3
0	4	24	2	1
0	4	24	5	2

0	5	5	1	3
0	5	13	4	1
1	5	5	1	1
0	5	5	1	1
0	2	15	4	3
0	2	15	6	1
0	2	15	4	2
0	2	15	4	1
0	2	15	2	3
0	2	15	2	2
0	2	15	2	1
0	2	18	5	2
0	2	18	5	1
0	2	22	1	1
0	2	22	6	1
0	2	22	3	2
0	2	22	2	2
0	2	22	4	2
0	2	22	4	1
0	2	22	3	1
0	2	22	2	1
0	3	32	2	3
0	3	32	2	2
0	3	32	4	2
0	3	32	2	1
0	3	32	4	1
0	4	24	3	1
0	4	24	1	1
0	4	24	5	1
0	4	4	4	3
0	4	4	2	1
0	4	12	6	2
0	4	21	5	2
0	4	21	3	1
0	1	5	3	1
0	1	4	5	1
0	1	18	5	1
0	1	18	1	2
0	1	12	3	2
0	1	12	1	1
0	1	8	6	3
0	1	8	5	3
0	1	8	2	1
0	1	13	3	2
0	1	27	1	2
0	1	27	1	1
0	1	15	2	1
0	1	18	1	1
0	1	15	4	2
0	1	13	4	1

0	4	11	2	3
0	4	21	1	1
0	4	6	3	2
0	4	6	5	1
0	4	6	4	3
0	1	8	4	1
0	1	26	6	2
0	1	26	2	3
0	1	26	3	1
0	1	26	2	2
0	1	26	1	1
0	1	22	1	3
0	1	22	4	2
0	1	22	6	1
0	1	27	3	2
0	1	27	3	1
0	1	26	2	1
0	1	26	6	1
0	1	22	4	1
0	4	6	4	2
0	4	6	4	1
0	4	6	2	3
0	4	6	2	2
0	4	6	3	1
0	4	7	4	1
0	4	7	3	3
0	4	7	6	1
0	4	21	5	3
0	4	21	5	2
0	4	7	3	2
0	4	11	4	3
0	4	11	3	2
0	4	11	1	1
0	4	11	2	2
0	4	11	4	2
0	4	11	4	1
0	4	11	3	1
0	4	11	2	1
0	4	6	2	1
0	4	4	1	1
1	4	4	5	1
0	1	25	4	2
0	1	18	1	1
0	1	8	2	2
0	4	21	5	1
0	4	21	4	1
0	4	21	2	2
0	4	21	2	1
0	4	4	3	2
0	4	4	6	3

0	1	13	2	2
0	1	13	3	1
0	1	13	2	1
0	1	13	5	3
0	1	13	1	1
0	1	15	5	2
0	1	15	3	2
0	1	15	6	2
0	1	15	3	1
0	1	26	1	1
0	1	26	2	1
0	1	15	4	1
0	1	15	1	1
0	1	18	4	1
0	1	25	3	3
0	1	22	4	2
0	1	12	4	3
0	1	25	3	2
0	1	25	4	1
0	1	25	1	3
0	1	26	3	2
0	1	13	5	2
0	1	13	5	1
0	1	12	4	2
0	1	26	4	3
0	1	26	4	2
0	1	26	4	1
0	1	26	3	1
0	1	12	3	1
0	1	12	4	1
0	1	12	2	2
0	1	12	2	1
0	1	25	1	2
1	1	22	2	1
0	4	4	4	2
0	4	4	3	1
0	4	17	6	3
0	4	6	5	1
0	4	6	6	1
1	4	7	6	1
0	4	7	1	3
0	4	6	4	1
0	4	7	4	1
0	4	7	6	1
0	4	4	6	1
0	4	4	5	1
0	4	4	1	2
0	4	4	4	1
0	4	21	5	1
0	4	21	6	3
0	4	17	4	1
0	4	17	6	2
0	4	17	6	1
0	4	17	5	2
0	4	17	5	1
0	2	16	4	2
0	2	16	6	3
0	2	19	1	3
0	2	16	2	1
0	2	16	5	1
0	2	24	4	3
0	2	24	3	3
0	2	24	4	2
0	2	12	5	2
1	2	25	2	1
0	1	13	5	2
0	1	13	1	1
0	1	8	2	1
0	1	22	1	2
0	1	22	1	1
0	1	13	4	1
0	1	13	6	1
0	1	13	5	1
0	1	25	4	1
0	1	25	5	1
0	1	25	1	1
0	2	16	4	1
0	2	16	6	2
0	2	24	1	3
0	2	24	5	3
1	2	25	5	1
1	2	25	3	2
0	2	24	3	2
0	2	16	6	1
0	2	12	1	3
0	2	24	6	2
0	2	24	6	1
0	2	19	4	2
0	2	24	3	1
0	2	24	5	2
0	2	24	1	2
0	2	24	4	1
0	2	25	2	1
0	2	25	3	2
0	2	25	3	1
0	2	25	5	1
0	2	24	1	1
0	2	24	5	1
0	2	13	1	1
0	2	19	4	1

0	4	21	6	2
0	5	5	2	2
0	5	5	2	1
0	1	15	6	1
0	1	15	5	1
0	4	17	6	2
0	4	17	1	1
0	4	17	6	1
0	4	7	1	2
0	4	7	1	1
0	4	4	1	1
0	4	12	4	1
0	4	12	6	1
0	4	21	6	1
0	4	12	5	3
0	4	12	2	1
0	4	12	5	2
0	4	12	5	1
0	1	22	4	1
0	1	25	1	1
0	1	8	6	2
0	1	8	3	3
0	1	8	6	1
0	1	22	2	1
0	1	8	5	2
0	1	8	3	2
0	1	8	3	1
0	1	8	5	1
0	1	22	1	2
0	1	22	1	1
0	1	25	6	2
0	1	25	3	1
0	1	25	6	1

0	2	19	3	2
0	2	12	5	1
0	2	22	1	1
0	2	22	2	1
0	2	22	3	1
0	2	19	1	2
0	2	19	1	1
0	2	19	3	1
0	2	12	1	2
0	2	12	4	3
0	2	12	1	1
0	2	13	6	2
0	2	13	2	3
0	2	19	2	1
0	2	22	6	1
0	2	22	5	3
0	2	13	6	1
1	2	13	2	1
0	2	22	5	2
0	2	22	5	1
0	2	13	2	1
0	2	12	4	2
0	2	12	4	1
0	4	4	5	1
0	4	4	2	2
0	4	4	4	2
0	4	4	2	1
0	4	7	3	1
0	4	7	2	1
0	4	24	3	2
0	4	4	6	2
0	4	4	4	1
0	4	4	6	1
0	4	4	3	1
0	4	24	4	2
0	4	24	5	1
0	4	24	6	3
0	4	24	4	1
0	4	24	6	2
0	4	24	6	1
0	4	24	3	1

APPENDIX H

Sample Echo Report

Ith row	-	Ith container in the initial load plan A
1st column	-	truck arrival at the Ith container location
2nd column	-	transtainer loads the Ith container onto a truck
3rd column	-	a truck has been already assigned to the Ith container
4th column	-	the Ith container is placed in buffer area
5th column	-	the Ith container is called for loading without discharging in buffer space
6th column	-	the Ith container is included in @@RHD list

Segment of Echo Report for Plan A

1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	1	1	0	1
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	1	1	0	1
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0

1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0

APPENDIX I

Sample Output

Output file contains information on tanstainer and crane utilization during the loading operation. Following is the utilization statistics for plan A (T=6, C=10, S=4).

HOUR	1	2	3	4	5	6	7	8	9	10
CRANE	60.88	73.02	100	91.13	58.44	73.75	91.44	80.63	71.36	93.11
TRANSTAIN	76.91	91.36	96.91	87.08	65.8	91.69	95.55	91.97	79.19	85.77

HOUR	11	12	13	14	15	16	17	18	19	20
CRANE	94.11	64.66	78.16	68.83	65.66	59	61.19	65	70.3	90.83
TRANSTAIN	94.38	83.55	77.11	69.22	86.61	53.66	53.25	67.3	84.75	96.69

APPENDIX J

Summary of Results

Appendix J.1 Total Loading Times for Plan A

T=2 C\S	0	4	8	12	BEST
0	35.0875	35.0875	35.0875	35.0875	35.0875
5	34.99472	35.0375	35.06611	35.06611	34.99472
10	34.99472	35.0375	34.94638	34.94638	34.94638
15	34.99472	35.0375	34.94638	35.11277	34.94638
BEST	34.99472	35.0375	34.94638	34.94638	34.94638

T=4 C\S	0	4	8	12	BSET
0	22.8575	22.8575	22.8575	22.8575	22.8575
5	21.92111	22.40888	22.82833	22.82833	21.92111
10	21.92111	22.40888	22.42694	22.42694	21.92111
15	21.92111	22.40888	22.42694	22.33055	21.92111
BEST	21.92111	22.40888	22.42694	22.33055	21.92111

T=6 C\S	0	4	8	12	BEST
0	22.01527	22.01527	22.01527	22.01527	22.01527
5	21.20638	21.53222	21.75777	21.75777	21.20638
10	21.20638	21.53222	21.41194	21.41194	21.20638
15	21.20638	21.53222	21.41194	21.48972	21.20638
BEST	21.20638	21.53222	21.41194	21.41194	21.20638

Appendix J.2 Total Loading Time for Plan B

T=2 C\S	0	4	8	12	BEST
0	36.71527	36.71527	36.71527	36.71527	36.71527
5	36.77222	36.77222	36.77222	36.77222	36.77222
10	36.77222	36.77222	36.77222	36.77222	36.77222
15	36.77222	36.77222	36.77222	37.11916	36.77222
BEST	36.71527	36.71527	36.71527	36.71527	36.71527

T=4 C\S	0	4	8	12	BEST
0	23.62555	23.62555	23.62555	23.62555	23.62555
5	23.13416	23.13416	23.13416	23.13416	23.13416
10	23.13416	23.13416	23.13416	23.13416	23.13416
15	23.13416	23.13416	23.13416	23.85777	23.13416
BEST	23.13416	23.13416	23.13416	23.13416	23.13416

T=6 C\S	0	4	8	12	BEST
0	23.24972	23.24972	23.24972	23.24972	23.24972
5	22.07138	22.07138	22.07138	22.07138	22.07138
10	22.07138	22.07138	22.07138	22.07138	22.07138
15	22.07138	22.07138	22.07138	22.22	22.07138
BEST	22.07138	22.07138	22.07138	22.07138	22.07138

Appendix J.3 Maximum Reduction in Total Loading time

X(i,j) implies the maximum reduction in the loading time when the buffer area is utilized and the truck level is increased from i to j. For example, 7.22% (5,0) for plan A indicates that the use of the buffer area with 5 chassis and the sweeping criterion of zero along with the increase of truck level from 4 to 6 resulted in 7.22% reduction in the loading time for plan A, compared to the PoP' result with 4 trucks.

for Plan A

T\T	2		4		6	
2	0.4 %	(10,8)	37.5 %	(5,0)	39.57 %	(5,0)
4	0		4.09 %	(5,0)	7.22 %	(5,0)
6	0		0		3.67 %	(5,0)

for Plan B

T\T	2		4		6	
2	-0.155 %	(5,0)	36.99 %	(5,0)	39.885 %	(5,0)
4	0		2.079 %	(5,0)	6.578 %	(5,0)
6	0		0		5.068 %	(5,0)

Appendix J.4 Material Handling Results for Plan A

(C - Chassis, S - Sweeping criterion)

ROW CHANGE

C\S	0	4	8	12
0	1658	1658	1658	1658
5	1638	1620	1626	1626
10	1638	1620	1606	1606
15	1638	1620	1606	1568

NON-ADJACENT SECTION CHANGE

C\S	0	4	8	12
0	14	14	14	14
5	14	12	13	13
10	14	12	10	10
15	14	12	10	10

ADJACENT SECTION CHANGE

C\S	0	4	8	12
0	13	13	13	13
5	13	14	13	13
10	13	14	14	14
15	13	14	14	12

NO. OF CONTAINERS REHANDLED

C\S	0	4	8	12
0	11	0	0	0
5	0	0	9	0
10	0	0	0	0
15	0	0	0	0

Appendix J.5 Material Handling Resultd for Plan B

(C - Chassis, S - Sweeping criterion)

C\S	0	4	8	12
0	1757	1757	1757	1757
5	1757	1757	1757	1757
10	1757	1757	1757	1757
15	1757	1757	1757	1601

C\S	0	4	8	12
0	10	10	10	10
5	10	10	10	10
10	10	10	10	10
15	10	10	10	8

C\S	0	4	8	12
0	14	14	14	14
5	14	14	14	14
10	14	14	14	14
15	14	14	14	10

C\S	0	4	8	12
0	13	0	0	0
5	0	0	0	0
10	0	0	0	0
15	0	0	0	1

Appendix J.6 Utilization of Crane and Transtainer for Plan A

	CRANE					TRANSTAINER				
T=2	C\S	0	4	8	12	C\S	0	4	8	12
	0	46.1215	46.1215	46.1215	46.1215	0	51.1965	51.1965	51.1965	51.1965
	5	46.2569	46.2	46.1628	46.1628	5	51.2462	50.955	50.874	50.874
	10	46.2569	46.2	46.3209	46.3209	10	51.2462	50.955	51.094	51.094
	15	46.2569	46.2	46.3209	46.1014	15	51.2462	50.955	51.094	50.81633
T=4	C\S	0	4	8	12	C\S	0	4	8	12
	0	72.0255	72.0255	72.0255	72.0255	0	78.26	78.26	78.26	78.26
	5	74.483	72.358	71.0585	71.0585	5	80.1855	77.7035	76.9595	76.9595
	10	74.483	72.358	72.268	72.268	10	80.1855	77.7035	76.935	76.935
	15	74.483	72.358	72.268	72.7505	15	80.1855	77.7035	76.935	76.986
T=6	C\S	0	4	8	12	C\S	0	4	8	12
	0	74.972	74.972	74.972	74.972	0	82.0975	82.0975	82.0975	82.0975
	5	77.187	75.575	74.865	74.865	5	83.5175	81.4375	81.4655	81.4655
	10	77.187	75.575	76.1755	76.1755	10	83.5175	81.4375	81.291	81.291
	15	77.187	75.575	76.1755	75.7865	15	83.5175	81.4375	81.291	80.6105

Appendix J.7 Utilization of Crane and Transtainer for Plan B

T=2	C\S	0	4	8	12	C\S	0	4	8	12
	0	46.3675	46.3675	46.3675	46.3675	0	53.813	53.813	53.813	53.813
	5	45.723	45.723	45.723	45.723	5	52.3465	52.3465	52.3465	52.3465
	10	45.723	45.723	45.723	45.723	10	52.3465	52.3465	52.3465	52.3465
	15	45.723	45.723	45.723	44.96	15	52.3465	52.3465	52.3465	51.0775
T=4	C\S	0	4	8	12	C\S	0	4	8	12
	0	69.291	69.291	69.291	69.291	0	76.9995	76.9995	76.9995	76.9995
	5	70.38	70.38	70.38	70.38	5	79.412	79.412	79.412	79.412
	10	70.38	70.38	70.38	70.38	10	79.412	79.412	79.412	79.412
	15	70.38	70.38	70.38	68.055	15	79.412	79.412	79.412	73.644
T=6	C\S	0	4	8	12	C\S	0	4	8	12
	0	72.2705	72.2705	72.2705	72.2705	0	79.443	79.443	79.443	79.443
	5	74.3965	74.3965	74.3965	74.3965	5	80.523	80.523	80.523	80.523
	10	74.3965	74.3965	74.3965	74.3965	10	80.523	80.523	80.523	80.523
	15	74.3965	74.3965	74.3965	73.658	15	80.523	80.523	80.523	78.557