



DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
ATLANTA REGIONAL OFFICE
101 MARIETTA TOWER
101 MARIETTA STREET, SUITE 2805
ATLANTA GA 30323-0008

E-24-X23

4330
243-Atl:ABB
g:\dtic1.wpd
01 August 1996

PHD
Mr. Duane Hutchison
Associate Director
GEORGIA TECH RESEARCH CORP
Office of Contract Administration
Atlanta, GA 30332-0420

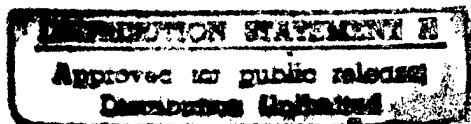
Mr. Hutchison:

Enclosed are one or more Technical Reports which were returned to this office by the Defense Technical Information Center (DTIC) in Cameron Station, Virginia. The reports were returned to this office because the report was not accompanied by a Standard Form(SF) 298, "Report Document Page". A copy of the Standard Form 298 is also enclosed, for your duplication and use. *Please ensure block 12a, "Distribution Availability Statement", of the SF 298 is marked "Unlimited" or it will be returned by DTIC.*

Please see that each enclosed report is resubmitted directly to DTIC, together with a properly completed SF 298. The new DTIC address is 8725 John J. Kingman Road, Suite 0944, Fort Belvoir, VA 22060-6218.

Please circulate the SF 298 to the researchers at your institution who work on Department of Defense contracts, grants, and cooperative agreements, so that it's use will become the standard for submitting any interim or final technical report to DTIC. We also ask that you continue to include the appropriate ONR office on the distribution for all future interim and final reports. We do not need the complete technical report, a properly completed SF 298 is sufficient.

Your assistance in this regard is greatly appreciated. If you have any questions, please contact the undersigned at (404) 730-9844.



Sincerely,

Antoinette Bigby
ANTOINETTE BIGBY
Grants Specialist

Enclosure(s)

19970227 075

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 06/10/96	3. REPORT TYPE AND DATES COVERED Final Technical Rept./ 6/1/93 - 5/31/96
----------------------------------	----------------------------	---

4. TITLE AND SUBTITLE
A Survey of Logistics Problems occurring in Container Transportation

5. FUNDING NUMBERS
N00014-93-1-0699

6. AUTHOR(S)
Greg Hunt, Donald Ratliff, Donna Llewellyn

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332

8. PERFORMING ORGANIZATION REPORT NUMBER

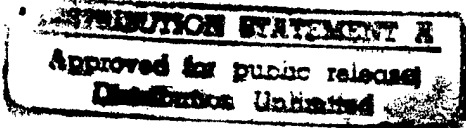
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE



13. ABSTRACT (Maximum 200 words)

This paper serves as an introduction to the types of operational problems that arise in the intermodal environment. We examine and describe problems from a wide range of topics including stowage of containers on both ship and rail, routing and scheduling of vehicles, routing of shipments on existing transportation networks, equipment management, and network design. It has become clear that the problems described here are inherently complex. In fact, for many of these problems, simply finding a feasible solution is not trivial. For selected problems, we have included initial research toward providing algorithmic solutions.

14. SUBJECT TERMS

15. NUMBER OF PAGES
16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE.

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

June 10, 1996

FINAL TECHNICAL REPORT

A Survey of Logistics Problems occurring in Container
Transportation

by

Greg Hunt
Donald Ratliff
Donna Llewellyn

Supported under ONR Contract N00014-93-1-0699

~~19961030 079~~

DTIC QUALITY INSPECTED 1
DTIC QUALITY INSPECTED 3

A Survey of Logistics Problems occurring in Container Transportation

Greg Hunt

Don Ratliff

Donna Llewellyn

June 4, 1996

1 Container Ship Stowage Problem (CSSP)

1.1 General Description

Schedule For CSSP, a container ship visits ports $1, 2, \dots, n$ in a circular route, i. e. after visiting port n , the ship returns to port 1. Furthermore, the ports are partitioned into two sets. At each port, the ship must unload any container which has that port as its destination and may load containers destined for any of the ports in the other partition. Currently, we have assumed that the ship starts and finishes empty, though this may not be realistic.

Storage Containers are stored in stacks on the ship. Usually, each slot on the ship can accommodate up to two TEUs. Each stack may only be accessed from the top. In addition, due to various restrictions, each stack on the ship may only contain a certain number of containers. This number differs from ship to ship and may differ from stack to stack within a particular ship due to its design. Also, the maximum number of containers in a stack may depend on attributes of containers in the stack, for example, weight limits on the lower containers may not be exceeded. Finally, there may be a large number of constraints due to types of containers and where they may be placed within the ship. This varies from ship to ship due to differences in design also.

Loading/Unloading Operations Containers are loaded and unloaded using a container crane which can access a stack only from the top. Also, a crane may carry up to two TEUs at once. From my observations, only one crane may work on a *bay* at a time and cranes must be at least one bay apart. In addition, when possible, the crane simultaneously loads and unloads containers. This might be a consideration when developing a stowage plan for a ship. Finally, due to the increasing width of the ships, some cranes may not be able to reach all containers in a bay.

Objectives Certainly, we want to minimize the time the ship is in port. One obvious way to reduce this is to reduce the time for loading/unloading operations. Two main problems must be addressed in developing a stowage plan, the placement of containers within a stack (overstowage) and the placement of stacks within a ship (efficiency of cranes).

1.2 Main Characteristics

1. Containers are stored in vertical stacks.
2. Each stack can store up to r containers. (r may vary upon location within the ship.)
3. A crane may access up to B (usually, this is the number of stacks in a bay) stacks without moving, i. e. incurring a cost.
4. The ship has c (port dependent) cranes loading/unloading containers simultaneously.
5. The ship visits n ports in a circular route.
6. The ports are partitioned into two sets A and B such that containers are shipped only between ports in different sets.

1.3 Operational Issues

1. How should the containers be loaded at each port in order to minimize total over-stowage?
2. In what order should the containers be loaded and unloaded to minimize the ship's total time in port?
3. How should the containers be loaded and unloaded to efficiently use the cranes?
4. How should the containers be stored in the yard?

1.4 Strategic Issues

1. How does the schedule of the ship affect the over-stowage problem?
2. At what percentage of capacity should the ship operate?

1.5 Circular Route versus Reverse Route for CSSP

1. A ship visits each port only once per cycle in a circular route, whereas it visits $n - 2$ ports twice per cycle in a reverse route. Thus the savings in over-stowage must be greater than the fixed costs of visiting several ports twice.
2. In most cases, the ship must travel farther in order to complete a reverse route cycle than the corresponding circular route.
3. The ship is empty twice per cycle in a reverse route and possibly never empty in a circular route. This certainly makes stowage planning at the end ports simpler.
4. A reverse route may allow the ship to provide better service (larger capacity) to some ports.

2 Iron Highway Stowage Problem (IHSP)

2.1 General Description

Schedule For IHSP, we have a train which, according to literature, will consist of ten horizontal stacks of fixed length. The train will visit stations $1, 2, \dots, n$. We assume that the train will then visit the stations in reverse order, i. e. $n, n-1, \dots, 1$. At each station, the train must unload any trailers destined for that station and may load containers originating at that station destined for any of the other stations.

Storage Each stack on the train is of a fixed length L and each trailer, t , has a fixed length l_t . Thus, the number of trailers a stack may hold depends on the length of the trailers in the stack.

Loading/Unloading Operations To access the trailers, two connected stacks, called an *element* must be split apart and special ramps lowered. A tractor may then either unload the first trailer in either stack or load a trailer onto the front of either stack if there is enough room. As little equipment is necessary to load/unload trailers, the main limiting factor is operating space needed alongside the tracks to accommodate any trailers to be loaded or unloaded. It is not unreasonable to assume that some stations will be larger than others and thus allow more elements to be simultaneously loaded and unloaded. However, we must be careful. If a station is large enough to accommodate two elements to be loaded/unloaded, but not large enough to accommodate three, then it would be unreasonable to assume that we could load/unload the first and last elements simultaneously. Also, if two elements are loaded/unloaded at the same station, but not simultaneously, then we assume that trailers may be unloaded from the first and loaded on the second, but not vice versa. Finally, containers may also be loaded onto the Iron Highway, but we currently do not address this additional fact.

Objectives As in CSSP, we wish to minimize the total time the train is at a terminal. Thus we want to minimize the loading/unloading time. Again, two main problems are the placement of the trailers within a stack and the placement of stacks within the train.

2.2 Main Characteristics

1. Trailers are stored in horizontal stacks (single access point per stack).
2. Each stack can store up to r trailers. (r may vary since trailers vary in length.) Typically, $10 \leq r \leq 15$.
3. Two stacks may be accessed simultaneously.
4. We will assume that the Iron Highway visits n stations in order, then reverses order.
5. Up to c (station dependent) elements may be loaded/unloaded simultaneously.

6. The order the trailers are loaded in and the order in which elements are accessed are not independent of each other.

2.3 Operational Issues

1. How should the trailers be loaded at each port in order to minimize overstorage?
2. In what order should the trailers be loaded?

2.4 Strategic Issues

1. How does the schedule of the train affect overstorage?
2. At what capacity should the train operate?
3. How should items be treated where it is possible to delay loading (because of the reverse route)?
4. Should the train stop at every terminal every time it passes it?
5. At what percentage of capacity should the train operate?
6. Should additional loading and unloading equipment be used at some terminals? If so, how should it be used?
7. Is a circular route possible? Is it practical?
8. Should the Iron Highway always transport a trailer to the terminal closest to its final destination? (Likewise, should it always transport a trailer from the terminal closest to its origin?)
9. Should routes for various trains overlap to provide larger service? If so, where should they overlap and how should items be transferred from one train to another?

3 Singlestack Railcar Access Problem (SRAP)

3.1 General Description

Schedule For SRAP, we have a train which consists of railcars which can carry trailers or containers. Each railcar has five slots. We will assume that all railcars are identical and each of the five slots on a railcar are identical, though we know that both of these assumptions are not true. In addition, the train visits n stations in order then reverse order, i. e. $1, 2, \dots, n, n-1, \dots, 1$. At each station, any trailers or containers with that station as the destination must be unloaded and trailers or containers may be loaded to be transported to any of the other stations. In addition, we may allow objects to be delayed until we return to this station if we return to this station before visiting

the object's destination. Additionally, we may have groups of objects with a common destination which are to be shipped on another train. In this case, if possible, we would like to transfer a full car of objects to be connected to the other train rather than unload five objects from one car on the current train only to load the same five on a railcar on the other train.

Storage In this scenario, the objects are not stacked. Requirements for unloading depend only on the loading/unloading equipment.

Loading/Unloading Operations Unfortunately, there is a wide variety of equipment used in the loading and unloading operations. Some of the types of equipment include:

1. Gantry cranes - These cranes may allow access to multiple rails at once. They have limited parallel movement to the tracks. They may be outfitted with either top loading mechanism or bottom loading mechanism. The bottom loading mechanism allows freedom in picking objects, but may inhibit which railcars objects may be loaded to.
2. Forklifts - Forklifts have restrictions on the weight of the container that it may pick. They have more freedom of movement around a yard than a gantry crane.

Objectives The objectives are unclear in this problem, since there is a wide variation in the equipment used. It seems reasonable that some grouping of objects with common origins and common destinations would be beneficial when possible. Also, it seems that it may be beneficial to reorder some objects at stations which are better equipped in order to save time at stations which are equipped with less efficient equipment.

3.2 Main Characteristics

1. Trailers or containers are stored on railcars.
2. A railcar can hold a total of five items.
3. We will erroneously assume that the five slots are identical.
4. We have some freedom in choosing when to load objects.
5. We particularly want to group objects to be transferred to another train.
6. This problem is extremely equipment dependent.

3.3 Strategic Issues

1. At what capacity should the train operate? I.e. should the train have empty railcars? If so, how many?

2. Given the existing rail system, what routes should be used? Assuming that routes overlap to allow shipment from terminals on one route to another, which terminals should overlap?
3. Assuming that we allow containers to be shipped from terminals in one route to terminals in a different route, how do we schedule the trains in order to provide this service?
4. Assuming that demand is not equal, how should empty railcars be dealt with?

4 Intermodal Core Network Design

Most of the container ship and intermodal rail services are driven by a core network. This core network provides regularly scheduled service between hubs. In fact, this schedule is published months in advance. In addition, several of the large LTL carriers are moving to a scheduled core network, i.e. truck loads between hubs are scheduled rather than load driven. For the problem, assume we are given a set of hubs, a geographical network connecting the hubs, a set of travel times over the geographical network, and the daily number of loads between hubs. We need to determine the route and schedule each load will follow, the route and schedule each vehicle will follow, and assign loads to each vehicle.

We first examine the case where we use vehicles with capacity one. For this case, the set of load routes will be a subset of the routes of all the vehicles.

4.1 Main Issues

Routing: Given the information above, how should each load be routed. It may be cheaper to route some loads along paths other than the shortest path in order to take advantage of backhaul miles of other loads. If we assume that each vehicle route consists of delivering a load from hub i to hub j , then returning to i , then the worst case set of vehicle routes is twice the set of load routes.

Scheduling: Given the route for each load, how do we schedule the loads? Since the schedule is constrained by desired service constraints and also crew (vehicle) constraints (i.e. limits on the number of hours a crew can operate the vehicle, union work rules, etc.), this is not trivial. Given that one load is a backhaul for a second load, the schedule must reflect this.

Vehicle Assignment: We need to tell each vehicle and crew which loads to deliver. In addition, we may have some vehicles and crews deadhead to deliver additional loads, to meet union restrictions, etc.

4.2 Additional Issues

1. Can we find a feasible solution?

2. How many vehicles do we need?
3. How many crews do we need?
4. How do we compare solutions?
5. Where should drivers be located?
6. Given a solution, how do we relocate existing resources to implement this solution?

4.3 Solution

4.3.1 Routing

For routing the loads, consider the following three restrictions: 1) we assume that all loads with the same origin and same destination must follow the same path, 2) we restrict our vehicle routes such that the vehicle must return along the same path, and 3) we assume that if load k_1 and load k_2 are routed in opposite directions then either k_2 is a backhaul for k_1 or vice versa. We can then model the load routing as the following special multi-commodity flow problem. We are given a graph $G = (U, V)$, a set of edge weights $tt_{ij} = tt_{ji}$, and a set of commodities $C = \{c_k = (\text{origin}_k, \text{dest}_k, \text{vol}_k)\}; k = 1, \dots, |C|$, we wish to find a route (path) for each commodity (load). Consider the following integer program:

$$\begin{array}{ll}
 \text{Min.} & R(x) = \sum_{i < j} tt_{ij} q_{ij} \\
 \text{ST} & \sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} 1 & \text{if } i = \text{origin}_k \\ -1 & \text{if } i = \text{dest}_k \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} i = 1, \dots, |U| \\ k = 1, \dots, |C| \end{matrix} \\
 & q_{ij} - \sum_k \text{vol}_k x_{ij}^k \geq 0 \quad (i, j) \in V \\
 & q_{ij} - \sum_k \text{vol}_k x_{ji}^k \geq 0 \quad (i, j) \in V \\
 & x_{ij}^k \in \{0, 1\}
 \end{array}$$

Then x is a solution to the routing problem if and only if x is a solution to the IP.

4.3.2 Load scheduling

Assume that the travel time on each edge is less than half of the total amount a single vehicle can operate at one time. Assume that a single vehicle will depart from node i , go to j , wait for a specified amount of time, then return to i . Thus, a vehicle assignment consists of a bipartite matching on each edge.

Assuming that we have a matching for the vehicle assignment problem and x is a solution to the routing algorithm, we can solve a linear program to find the departure times for each load of each commodity at each hub. Let $path^k = (hub_1^k, hub_2^k, \dots, hub_{|path^k|}^k)$ be the path all loads of commodity k will follow. We need to determine $dt_{hub_i^k}^{kl}, k = 1, \dots, |C|; l = 1, \dots, \text{vol}_k; i = 1, \dots, |path^k| - 1$, i. e. we simply need to set a departure time for each leg of each load. Now, we wish to minimize the vehicle slack time subject to the load constraints, i.e. each leg must have a departure time equal to the sum of the departure time

of the previous leg + the travel time of the previous leg + load slack time and load service constraints. Also, for each matching we will have an equation: the departure time of the second leg equals the departure time of the first leg + the travel time of the leg + vehicle slack time. We note that all variables will be non-negative as well. Let M be the set of matchings over all edges. Then our LP will have $|path^k| - 1$ equations for each path which implies $\sum x_{ij}^k - |C| + |M|$ total equations with approximately $2(\sum x_{ij}^k - |C|) + |M|$ variables.

However, we must be careful here. We can easily derive a matching that would yield an infeasible LP. However, if we solve the matching using feasible departure times, then the LP will be guaranteed to have a feasible solution. We also note that these two algorithms can be iterated.

4.3.3 Vehicle assignment

Assume that all loads have been routed and scheduled. (Assume that only loads have been scheduled - not empty loads for completing a round trip.) By scheduled, we have assigned departure times for each load at each hub. We wish to assign vehicles to routes. Assume that the travel time on each edge is less than half of the total amount a single vehicle can travel at one time. If we assume that a single vehicle will depart from node i , travel to j , wait for a specified amount of time, then return to i , then we can find the minimum weight vehicle assignment subject to vehicle and load constraints.

By our assumption of a vehicle assignment, we note that we can assign vehicle routes at each edge independently. Consider one edge (i, j) . Construct the auxiliary bipartite graph $G_1 = (A, B, E)$ in the following way. Let A = the loads from i to j and B = the loads from j to i . Add $|B|$ dummy nodes, dn to A and $|A|$ dummy nodes, dn to B . Now let $(job_a, job_b) \in E$ if either $dt_{job_a} + tt_{ij} \leq dt_{job_b}$ or $dt_{job_b} + tt_{ij} \leq dt_{job_a}$. We will give this edge a weight of driver fixed cost + vehicle slack time cost + travel time. Also, for each job_a we include $(job_a, dn) \in E$ with a weight of vehicle fixed cost + twice the edge travel time. Similarly, we include the edge $(job_b, dn) \in E$ for each job_b . Now, we can solve a minimum weight bipartite matching problem. Obviously, we can do this for each edge (i, j) to minimize subject to our assumed vehicle assignment description.

4.3.4 Initial solution

Algorithm

1. Generate the set of load-paths (one path per *load*).
2. Order the load-paths. (currently, according to total travel time of the path, in decreasing order)
3. Set $k = 1$.
4. Assume that the departure times of the $k - 1$ paths already scheduled is fixed.
5. Let $U_{ij} =$ the set of unmatched edges at edge (i, j) in the direction (i, j) .

6. Set the departure times of the k th path such that the departure times are feasible and the load segments can be matched to the largest weighted set of unmatched edges (by trial and error).
7. Increment k and repeat the last three steps until all load-paths have been scheduled and matched.

4.3.5 Additional Comments

1. The solution techniques for the vehicle assignment and load scheduling seem to be quite robust. We could easily limit the vehicle slack time or load slack time, restrict the departure time of certain loads, require two load parts to be matched, etc.
2. Both the vehicle assignment and the load scheduling parts are easily solvable—both optimize their respective objective functions.
3. At each iteration of LS, the previous solution to DA is feasible. Similarly, at each iteration of DA, the previous solution to LS is feasible. Thus TC can only improve at each iteration.
4. Our assumption of a vehicle schedule is unreasonable for short edges.
5. It is possible that we could cycle between degenerate matchings / LP solutions. Can we detect these cycles? If not, how do we know when to stop?
6. How does the solution to Routing impact the final solution?
7. We should consider adding in constraints in solving the routing problem to limit the total travel time of a commodity. Otherwise, we could easily increase the travel time of a shipment by an extremely large amount.
8. Once you have a final solution (matching) for the vehicle assignment, you can solve for the departure times in order to revise slack time.
9. What should the cycle time be?

5 Container Ship Scheduling Problems

Assume we are given a circular route of ports, $(1, 2, 3, \dots, n, 1)$, a set time in each port, and a set travel time for each leg. We are examining the class of problems where we are given a set of times when the ports are open and an amount of time that the ship must spend in port i . Can we find a feasible schedule?

5.1 Definitions and Notation

1. n = the number of ports
2. m = the number of weekends in the work schedule
3. p_i = time required in port i (in days)
4. tt_i = travel time from port i to the next port in the sequence (in days)
5. RTT = Round Trip Time = $\sum_{i=1}^n (p_i + tt_i)$
6. T_{ij} = trip time starting from the arrival at port i until departure from port j = $p_i + tt_i + p_{i+1} + tt_{i+1} + \dots + p_{j-1} + t_{j-1} + p_j$
7. $b(\cdot)$ = a function which returns the time a period begins
8. $e(\cdot)$ = a function which returns the time a period ends
9. Note: If we are referring to a time period in a ship schedule, then we will include a reference to a particular start time.

5.2 Problem Instances

To begin with, assume that all ports are open all during the week and closed on weekends. Also, assume that all ports are in the same time zone. The first scheduling problem we address is: "Given a 7 day Ship Schedule, can we schedule the weekend to occur during a travel time period?" This is extended to: "Given a $7 \times m$ day Ship Schedule, can we schedule the weekends to occur during travel time periods?" We then introduce slack by asking: "Given a < 7 day Ship Schedule, can we schedule the weekend to occur during a travel time period?" and "Given a $< 7 \times m$ day Ship Schedule, can we schedule the weekends to occur during travel time periods?" We generalize again by replacing the weekday / weekend time sequence by a more generic work / nonwork schedule. However, we still assume for the slack case that the Work Schedule is longer than the Ship Schedule. Finally, we address scheduling multiple ships. Note that we assume the Work Schedule is fixed in calendar time, however, the Ship Schedule is only in relative time.

5.2.1 1 ship / 1 week / no slack

Assume $RTT = 7$ (weekly service). Is there an initial start time when the weekend (Saturday and Sunday) does not coincide with a port time?

Solution: Is there an i such that $tt_i \geq 2$? If so, schedule the weekend to occur during that leg of the trip.

5.2.2 1 ship / 1 week / slack

Assume $RTT < 7$ (weekly service). Let $slack = 7 - RTT$. Is there a schedule when the weekend (Saturday and Sunday) does not coincide with a port time? (We are allowed to insert the slack anywhere.)

Solution: Is there an i such that $tt_i + slack \geq 2$? If so, schedule the weekend to occur during that leg of the trip.

5.2.3 1 ship / 2 weeks / no slack

Assume $RTT = 14$ (bi-weekly service). Is there an initial start time when the weekends (Saturday and Sunday) do not coincide with port times?

Solution: Is there an i_1 and i_2 such that $tt_{i_1}, tt_{i_2} \geq 2$ and $T_{i_1+1, i_2} \leq 5$ and $T_{i_2+1, i_1} \leq 5$? If so, schedule the weekends to occur during those legs of the trip.

Proof. Assume i_1 and i_2 exist as defined above. Assume also that $i_1 \neq i_2$, otherwise, the proof is trivial. Now, we are considering two 14 day cycles—one cycle represents the days of the week and the other cycle represents the alternating port and travel times. We wish to show that we can schedule the two weekends during the periods tt_{i_1} and tt_{i_2} . Define $ST = \max(0, tt_{i_1} + T_{i_1+1, i_2} - 7)$. Arrange the cycles such that the first Saturday begins at ST time into tt_{i_1} .

First, assume $ST = 0$. Thus tt_{i_1} starts at the same time as Saturday. Since $tt_{i_1} \geq 2$, then the first weekend is contained in tt_{i_1} . Also, $ST = 0 \Rightarrow tt_{i_1} + T_{i_1+1, i_2} - 7 \leq 0 \Rightarrow tt_{i_1} + T_{i_1+1, i_2} \leq 7 \Rightarrow$ the second weekend starts after $b_{ST}(tt_{i_2})$. Now, $T_{i_2+1, i_1} \leq 5 \Rightarrow tt_{i_1} + T_{i_1+1, i_2} + tt_{i_2} \geq 9 \Rightarrow$ that the second weekend ends before $e_{ST}(tt_{i_2})$ since the two time periods started at the same time. Thus, the second weekend is contained in tt_{i_2} .

Second, assume that $ST = tt_{i_1} + T_{i_1+1, i_2} - 7$. By design, the second weekend starts at $b_{ST}(tt_{i_2})$. Thus trivially, the second weekend is contained in tt_{i_2} since $tt_{i_2} \geq 2$. Now, since $ST \geq 0 \Rightarrow tt_{i_1} + T_{i_1+1, i_2} \geq 7 \Rightarrow$ the first weekend starts after $b_{ST}(tt_{i_1})$. Also, since $T_{i_1+1, i_2} \leq 5$, the first weekend ends before $e_{ST}(tt_{i_1})$. Thus either way, we can schedule the weekends to occur during travel legs of the trip.

Thus we have shown that these conditions are sufficient for scheduling the weekends into travel legs. For the 2 week case, it is obvious that these conditions are necessary. \square

Note that while this proof is not trivial, however, it is not terribly complex either. We tried a similar proof for the three week case. It quickly became clear that to provide a similar proof for the general case would be notationally difficult and almost algorithmically useless.

5.2.4 1 ship / m weeks / no slack

Assume $RTT = 7m$. Is there an initial start time when the weekends (Saturday and Sunday) do not coincide with port times?

Solution: Consider the following lemma:

Lemma 1 *If there is a feasible start time, then there is a feasible start time such that at least one weekend / travel leg pair has the same beginning time.*

Proof. Let an instance of the scheduling problem be given. Also, let ST be a feasible start time. If there is at least one weekend / travel leg pair that has the same beginning time, then we are done. Assume not. Let $Sat.j$ = the j^{th} Saturday and tt_i = the travel leg containing $Sat.j$ (see figure 5.1). Since ST is feasible and no pair has the same beginning time, then $b(Sat.j) > b_{ST}(tt_i) \Rightarrow b(Sat.j) - b_{ST}(tt_i) > 0, j = 1, \dots, m$. Let $\bar{ST} = ST + \min_j [b(Sat.j) - b_{ST}(tt_i)]$ (see figure 5.2). Then consider the schedule with initial start time of \bar{ST} (see figure 5.3). Now, since we simply shifted the travel / port schedule forward by $\min_j [b(Sat.j) - b_{ST}(tt_i)]$, then it is obvious from figure 5.3 that the weekends will still be contained in the same travel legs. Algebraically, this can be shown as:

$$\begin{aligned}
 b_{\bar{ST}}(tt_i) &= b_{ST}(tt_i) + \min_j [b(Sat.j) - b_{ST}(tt_i)] \\
 &\leq b_{ST}(tt_i) + b(Sat.j) - b_{ST}(tt_i) \\
 &= b(Sat.j) \\
 e(\bar{ST}.j) &\leq e_{ST}(tt_i) \\
 &< e_{ST}(tt_i) + \min_j [b(Sat.j) - b_{ST}(tt_i)] \\
 &= e_{ST}(tt_i)
 \end{aligned}$$

Since this is true for $j = 1, \dots, m$, then \bar{ST} is a feasible start time. In addition, at least one weekend / travel leg pair will have the same start time. Hence the statement is true. \square

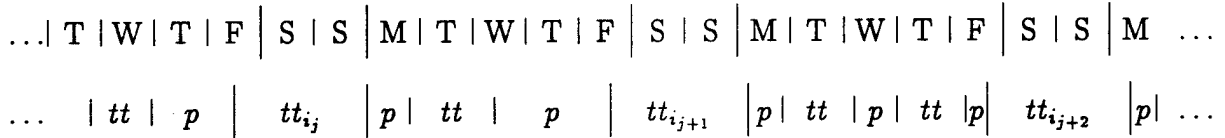


Figure 5.1: An example of a feasible schedule with start time ST

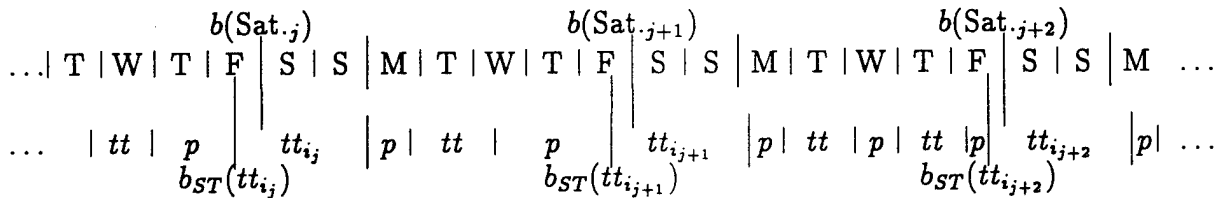


Figure 5.2: $b(Sat.) - b_{ST}(tt)$

This result leads to a simple algorithm for finding a feasible solution if one exists. We can check and see if starting a weekend at each travel leg results in a feasible schedule. For each start time, we must do no more than $2m$ comparisons to check for feasibility. Since we have n possible start times, then we have an $O(nm)$ algorithm.

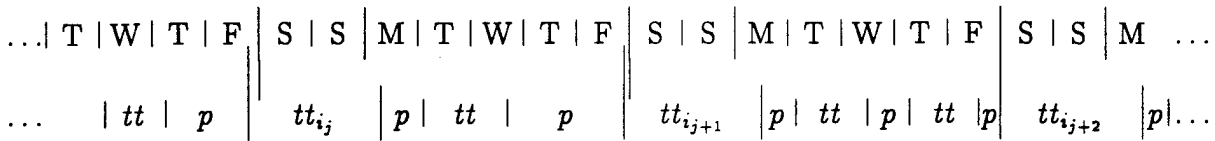


Figure 5.3: New schedule with start time $\bar{S}T$

5.2.5 1 ship / m weeks / slack

Assume $RTT < 7m$ (m -weekly service). Let $slack = 7m - RTT$. Is there a schedule when the weekends (Saturday and Sunday) do not coincide with a port time? (We are allowed to insert the slack anywhere and the slack can be broken up.)

Solution: Obviously we can easily generate a schedule if $slack + \max_i(tt_i) \geq 2m$. Thus we only consider cases in which $slack + \max_i(tt_i) < 2m$, i. e. we must use more than one travel leg.

Consider the following program. The main input to the program is the ship schedule. The work schedule is assumed to be m weeks long with Monday through Friday as a work period and Saturday and Sunday as a nonwork period.

Container Ship Scheduling Algorithm (CSSA)

1. Build a stack called `Work_Schedule` which defines the working schedule. Start the `Work_Schedule` with a weekend and it should be m weeks long. (Each element consists of a type, either *Work* or *Nonwork*, and length, l .)
2. Loop $j = 1$ to n ($j =$ travel time counter) {
 - Build a stack called `Ship_Schedule` which defines the ship's schedule starting with Travel time period j . (Each element consists of a type, either *Port* or *Travel*, and length, l .)
 - If `Find_Feasible(Ship_Schedule, Work_Schedule) = Feasible` then `return(Feasible and exit)` }
3. `return(Infeasible)`

Subroutine `Find_Feasible(Ship_Schedule, Work_Schedule)`

1. If `Ship_Schedule = ∅`, `return(Feasible)`
2. else if `Work_Schedule = ∅`, `return(Infeasible)`
3. else if `Top of Ship_Schedule = Travel Time` {
 - Define $temp =$ length of Top of `Ship_Schedule`
 - Remove $temp$ amount of time from the Top of `Work_Schedule`
 - Remove $temp$ amount of time from the Top of `Ship_Schedule`

- return(FindFeasible(Ship_Schedule, Work_Schedule)) }
4. else if Top of Work_Schedule = Work Period {
- Define $temp$ = length of Top of Work_Schedule
 - Remove $temp$ amount of time from the Top of Work_Schedule
 - Remove $temp$ amount of time from the Top of Ship_Schedule
 - return(FindFeasible(Ship_Schedule, Work_Schedule)) }
5. else (Top of Work_Schedule = Nonwork Period and Top of Ship_Schedule = Port Time) {
- Define $temp$ = length of Top of Work_Schedule
 - Remove $temp$ amount of time from Top of Work_Schedule
 - return(FindFeasible(Ship_Schedule, Work_Schedule)) }

Now, to show that the algorithm works, we must first show that if a feasible schedule exists, then a feasible schedule exists with a weekend and a travel time period starting at the same time. We note that this algorithm simply returns 'Feasible' or 'Infeasible', however, the algorithm can easily be modified to return the associated schedule. Thus, consider the following lemma:

Lemma 2 *If there is a feasible solution, then there exists a feasible solution such that at least one weekend and travel time period have the same beginning time.*

Proof. Let an instance of the problem with a feasible solution, ST , be given (see figure 5.4). Define a weekend containing time slot (WCTS) as any period or periods which contain a weekend. First, we observe that since we can insert slack anywhere, if we have a WCTS which is entirely slack, then we can easily modify the start time without affecting this WCTS's ability to contain a weekend. Secondly, we observe that given a WCTS containing slack and a travel time period, we can easily redistribute the slack before and after the travel period while retaining a feasible solution. Let $Sat.j$ = the j^{th} Saturday and $WCTS_{i,j}$ = the WCTS containing the j^{th} weekend (see figure 5.5). Define \mathcal{S} as the set of indices of WCTSs that are entirely composed of slack. If there is a j such that $b(Sat.j) = b_{ST}(WCTS_{i,j}), i_j \notin \mathcal{S}$, then we are done. Assume not. Since ST is feasible and no pair has the same beginning time, then $b(Sat.j) > b_{ST}(WCTS_{i,j}) \Rightarrow b(Sat.j) - b_{ST}(WCTS_{i,j}) > 0$. Let $\bar{ST} = ST + \min_{j|i_j \notin \mathcal{S}} [b(Sat.j) - b_{ST}(WCTS_{i,j})]$. Then consider the schedule with initial start time of \bar{ST} (see figure 5.6). Now, since we simply shifted the travel / port schedule forward by $\min_{j|i_j \notin \mathcal{S}} [b(Sat.j) - b_{ST}(WCTS_{i,j})]$, then it is obvious in figure 5.6 that the weekends will still be contained in the same WCTSs. This can be shown algebraically as:

$$\begin{aligned}
 b_{\bar{ST}}(WCTS_{i,j}) &= b_{ST}(WCTS_{i,j}) + \min_{j|i_j \notin \mathcal{S}} [b(Sat.j) - b_{ST}(WCTS_{i,j})] \\
 &\leq b_{ST}(WCTS_{i,j}) + b(Sat.j) - b_{ST}(WCTS_{i,j})
 \end{aligned}$$

$$\begin{aligned}
&= b(\text{Sat.}_j) \\
e(\text{Sat.}_j) &\leq e_{ST}(\text{WCTS}_{i_j}) \\
&< e_{ST}(\text{WCTS}_{i_j}) + \min_{j|i_j \notin S} [b(\text{Sat.}_j) - b_{ST}(\text{WCTS}_{i_j})] \\
&= e_{ST}(\text{WCTS}_{i_j})
\end{aligned}$$

Since this argument holds for each $j|i_j \notin S$ and there is a trivial argument for $j|i_j \in S$, then \bar{ST} is a feasible solution. In addition, at least one weekend and travel time period will have the same start time. Hence the lemma is true. \square

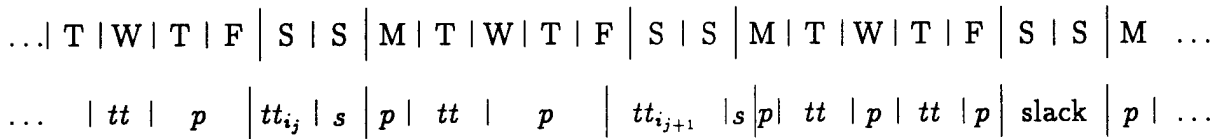


Figure 5.4: A feasible schedule, ST

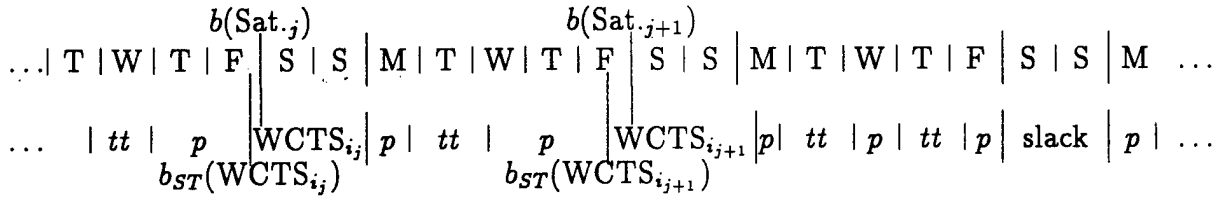


Figure 5.5: $b(\text{Sat.}) - b_{ST}(\text{WCTS})$



Figure 5.6: New solution, \bar{ST}

Secondly, we must show that if a feasible schedule exists with travel time period j and a weekend starting at the same time, then our algorithm can find a feasible schedule. First, we will give a brief discussion of how the algorithm works. The algorithm tests starting each travel time period at the same time as a weekend. Once the starting time is established, the algorithm ensures that only feasible times are matched between the two schedules. The only infeasible match is Port time matched with Nonwork time. Thus, in this case the algorithm inserts the necessary amount of slack to make a feasible match. The algorithm also takes

advantage of the recursive nature of the problem. If we have a current feasible match, then we have a feasible solution if the remainder of the two schedules can be feasibly matched. The algorithm terminates when one of the two schedules ends. If Ship_Schedule ends before or at the same time as Work_Schedule, then we have found a feasible solution. If Work_Schedule ends before Ship_Schedule, then we have inserted more slack than is available and thus we do not have a feasible solution.

Lemma 3 *CSSA finds a feasible schedule if one exists.*

Proof. Let an instance of the problem be given. Assume that a feasible solution ST exists. By the previous lemma, we may assume that a travel time period, say tt_1 , and a weekend begin at the same time in ST . Furthermore, we will assume that tt_1 is the first period in ST (if not, because of the circular nature of the schedules we can easily rewrite ST such that this will be true). Now, let AS be the infeasible solution generated by CSSA. Now consider the first point at which slack occurs in either ST or AS , but not both. (Note that ST and AS will be identical up to this point.) We have four cases:

1. $ST = \text{Travel Time}$, $AS = \text{Slack Time}$
2. $ST = \text{Port Time}$, $AS = \text{Slack Time}$
3. $ST = \text{Slack Time}$, $AS = \text{Port Time}$
4. $ST = \text{Slack Time}$, $AS = \text{Travel Time}$

Case 1: $ST = \text{Travel Time}$ and $AS = \text{Slack Time}$. Since this is the first point at which slack occurs in either ST or AS , but not both, then this implies that ST and AS must be identical up to this point. Thus, the period occurring after the Slack Time in AS must be Travel Time. However, the only way slack can be inserted by CSSA is before Port Time. Thus, by contradiction, this case is impossible!

Case 2: $ST = \text{Port Time}$ and $AS = \text{Slack Time}$. Again, since this is the first point at which slack occurs in either ST or AS , but not both, then this implies that ST and AS must be identical up to this point. However, we note that any slack inserted by CSSA corresponds to a Nonworking period in the Work Schedule. Thus, ST has Port Time matched to Nonworking Time. Since we have assumed that ST is a feasible schedule, then by contradiction, this case is impossible!

Case 3: $ST = \text{Slack Time}$ and $AS = \text{Port Time}$. Since this is the first point at which slack occurs in either ST or AS , but not both, then this implies that ST and AS must be identical up to this point. Thus, the period occurring after the Slack Time in ST must be Port Time. Since the schedule generated by our algorithm does not allow for an infeasible match, then the corresponding time slot in the Work Schedule must be a Work Period. Thus it is obvious that swapping the Slack Time period in ST with the Port Time period occurring after it will yield a new solution ST_1 which will be feasible and will match AS in one more period.

Case 4: $ST = \text{Slack Time}$ and $AS = \text{Travel Time}$. Since this is the first point at which slack occurs in either ST or AS , but not both, then this implies that ST and AS must be

identical up to this point. Thus, the period occurring after the Slack Time in ST must be Travel Time. It is obvious that swapping the Slack Time period in ST with the Travel Time period occurring after it will yield a new solution ST_1 which will be feasible and will match AS in one more period.

By recursively applying the above, we obtain that if a feasible solution exists, then AS is indeed a feasible solution. One final note is that for this procedure to work we need the additional restriction that ST has a finite number of periods. While it is possible to have solutions with an infinite number of periods, it is intuitive that if both the Work Schedule and the Ship Schedule have a finite number of periods then if a feasible solution exists, then there is a feasible solution with a finite number of periods. \square

5.2.6 m ships / m weeks / no slack

Assume $RTT = 7m$, is there an initial start time when the weekends (Saturday and Sunday) do not coincide with a port time?

Solution: Same as 1 ship / m weeks / no slack. Simply start the ships exactly seven days apart.

5.2.7 m ships / m weeks / slack

Assume $RTT < 7m$. Let $slack = 7m - RTT$. Is there an initial start time when the weekends (Saturday and Sunday) do not coincide with a port time? (We are allowed to insert the slack anywhere and the slack can be broken up.)

Solution: Same as 1 ship / m weeks / slack. Simply start the ships exactly seven days apart.

5.2.8 1 ship / General work schedule / no slack

Assume we have a Ship Schedule and a Work Schedule of equal length. The Ship Schedule has n alternating travel and port periods and the Work Schedule has m alternating work and nonwork periods. Is there a starting time where the port periods are matched to work periods? Consider the following lemma which is analogous to lemma 1:

Lemma 4 *If a feasible start time exists, then there exists a feasible start time where a nonwork period and a travel period start at the same time.*

Proof. Note that lemma 1 makes no assumption about the length of the individual weekends (nonwork periods). Thus, the proof holds for this more general case. \square

Now, we simply check to see if starting *each* nonwork period at the same time as each travel period results in a feasible schedule which is an $O(nm^2)$ algorithm.

5.2.9 1 ship / General work schedule / slack

Assume we have a Ship Schedule and a Work Schedule such that the Ship Schedule is shorter than the Work Schedule. The Ship Schedule has n alternating travel and port periods and the Work Schedule has m alternating work and nonwork periods. Is there a starting time where the port periods are matched to work periods? Consider the following lemma:

Lemma 5 *If there is a feasible solution, then there exists a feasible solution such that at least one nonwork period and travel time period have the same beginning time.*

Proof. Since the proof of lemma 2 does not use the length of the weekends nor the cyclic nature of the week, then the proof will work here also. \square

The previous lemma implies that we can simply run the algorithm for the 1 ship / m weeks / slack case m times modifying the Work Schedule to start with a different nonwork period each time.

5.2.10 1 ship / Multiple work schedules / no slack

Consider the following more general case of the problem: We are given a fixed time schedule for a ship including a route (ordered set of points) and a fixed amount of time to spend in each port and a fixed travel time for each leg. In addition, we are given a port work schedule for each port.

We will assume that each port work schedule is of length T and is cyclical. In addition, we will assume that the ship schedule is cyclical and of length T . Is there a starting time such that the ship is in port during that port's working hours (i. e. each p_i occurs during a w_{p_i})? Let $m_i =$ the number of nonwork periods in the work schedule for port $i, i = 1, \dots, n$.

Lemma 6 *If a feasible schedule exists, then there exists a feasible schedule such that the ship arrives at port at the beginning of a work period for at least one port.*

Proof. Let an instance of the problem be given. Assume we have a feasible schedule ST . Then each port time, p_i , is contained in a work period. Let j_{p_i} be the subscript of the work period containing it. If there is a p_i such that $b_{ST}(p_i) = b(w_{j_{p_i}})$, then we are done. Assume not. Now, let $t = \min[b_{ST}(p_i) - b(w_{j_{p_i}})]$. Define $ST_1 = ST - t$. Now we must show that ST_1 is a feasible start time. For any i ,

$$\begin{aligned}
 b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\
 &= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\
 &\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\
 &= b_{ST}(p_i) - t \\
 &= b_{ST_1}(p_i)
 \end{aligned}$$

Also, $e_{ST_1}(p_i) < e_{ST}(p_i) \leq e(w_{j_{p_i}})$. Thus, each p_i is contained in a work period which implies that ST_1 is feasible. \square

Thus, we can use an algorithm similar to the one described for the 1 **ship / General work schedule / no slack** case. However, instead of starting travel and nonwork periods at the same time, we are starting port and work periods at the same time. In addition, instead of comparing each port period to a master work schedule, we use the appropriate work schedule for each port.

5.2.11 1 ship / Multiple work schedules / slack

This is the same as the 1 **ship / Multiple work schedules / slack** problem except that the length of the ship schedule $< T$. Consider the following lemma:

Lemma 7 *If there is a feasible solution, then there exists a feasible solution such that at least one work time period and port time period have the same beginning time.*

Proof. Let an instance of the problem be given. Assume we have a feasible schedule ST . Then each port time, p_i , is contained in one or more work periods. However, if p_i is contained in more than one work period, then slack must be inserted during the intervening nonwork periods. Let j_{p_i} be the subscript of the work period containing the first portion of p_i . If there is a p_i such that $b_{ST}(p_i) = b(w_{j_{p_i}})$, then we are done. Assume not. Now, let $t = \min[b_{ST}(p_i) - b(w_{j_{p_i}})]$. Define $ST_1 = ST - t$. Now we must show that ST_1 is a feasible start time. Consider the following two cases:

Case 1: If p_i is contained in a single work period in ST , then the following argument holds:

$$\begin{aligned} b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\ &= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\ &\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\ &= b_{ST}(p_i) - t \\ &= b_{ST_1}(p_i) \end{aligned}$$

Also, $e_{ST_1}(p_i) < e_{ST}(p_i) \leq e(w_{j_{p_i}})$. Thus, p_i is contained in a work period.

Case 2: If p_i is contained in multiple work periods in ST , then there is at least enough work time in the period between $b_{ST}(p_i)$ and $b_{ST}(p_i)$ to contain p_i . Now, consider the time interval $(b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})], e_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})])$. Since

$$\begin{aligned} b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\ &= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\ &\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \end{aligned}$$

Then we have an interval the same length as the interval $(b_{ST}(p_i), b_{ST}(p_i))$ and since we have added $\min[b_{ST}(p_i) - b(w_{j_{p_i}})]$ amount of work time to the new interval, then the new interval has at least as much work time as the old interval. Thus, it is easy to see that by intelligently rearranging the slack in the new interval it is easy to see that we can schedule p_i to be contained by work time.

Thus, we have a new feasible schedule in which at least one port time period and one work period begin at the same time. Hence the lemma is true. \square

5.2.12 1 ship / Noncyclical work schedule / no slack

Assume we are given Ship Schedule and a Work Schedule. The Ship Schedule has n alternating travel and port periods and the Work Schedule has m alternating work and nonwork periods. In addition, the Ship Schedule is shorter than the Work Schedule and the schedules are not cyclical in nature. For the no slack case, we are not allowed to insert any slack time into the ship schedule. Is there a starting time where the port periods are matched to work periods?

First, we note that there are two immediate bounds on the starting time. Let L be the start time associated with starting the first port period at the same time as the first work period. Similarly, define U to be the start time associated with ending the last port period at the same time as the last work period. It is clear that any feasible start time ST will have the property, $L \leq ST \leq U$. Consider the following lemma:

Lemma 8 *If a feasible schedule exists, then there exists a feasible schedule such that the ship arrives at port at the beginning of a work period for at least one port.*

Proof. Let an instance of the problem be given. Assume we have a feasible schedule ST . Then each port time, p_i , is contained in a work period. Let j_{p_i} be the subscript of the work period containing it. If there is a p_i such that $b_{ST}(p_i) = b(w_{j_{p_i}})$, then we are done. Assume not. Now, let $t = \min[b_{ST}(p_i) - b(w_{j_{p_i}})]$. Define $ST_1 = ST - t$. Now we must show that ST_1 is a feasible start time. For any i ,

$$\begin{aligned}
 b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\
 &= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\
 &\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\
 &= b_{ST}(p_i) - t \\
 &= b_{ST_1}(p_i)
 \end{aligned}$$

Also, $e_{ST_1}(p_i) < e_{ST}(p_i) \leq e(w_{j_{p_i}})$. Thus, each p_i is contained in a work period which implies that ST_1 is feasible. However, we must also show that $L \leq ST_1 \leq U$.

$$\begin{aligned}
 L &= ST - ST + L \\
 &= ST - (ST - L)
 \end{aligned}$$

$$\begin{aligned}
&= ST - (b_{ST}(p_1) - b(nw_1)) \\
&\leq ST - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\
&= ST - t \\
&= ST_1 \\
&< ST \\
&\leq U
\end{aligned}$$

Thus, ST_1 is indeed a feasible start time. \square

This implies that we can use the algorithm for the 1 ship / General work schedule / no slack case with the slight modification that we calculate the corresponding start time when testing starting a port period and work period at the same time. In addition, we can improve the efficiency of the algorithm by making use of L and U .

5.2.13 1 ship / Noncyclical schedules / slack

Assume we are given Ship Schedule and a Work Schedule. The Ship Schedule has n alternating travel and port periods and the Work Schedule has m alternating work and nonwork periods. In addition, the Ship Schedule is shorter than the Work Schedule and the schedules are not cyclical in nature. For this case, we are allowed to insert slack time into the ship schedule. Is there a starting time where the port periods are matched to work periods?

We note that we can define upper and lower bounds on the starting time similar to the no slack case. In addition, the following lemma holds:

Lemma 9 *If there is a feasible solution, then there exists a feasible solution such that at least one work time period and port time period have the same beginning time.*

Proof. Let an instance of the problem be given. Assume we have a feasible schedule ST . Then each port time, p_i , is contained in one or more work periods. However, if p_i is contained in more than one work period, then slack must be inserted during the intervening nonwork periods. Let j_{p_i} be the subscript of the work period containing the first portion of p_i . If there is a p_i such that $b_{ST}(p_i) = b(w_{j_{p_i}})$, then we are done. Assume not. Now, let $t = \min[b_{ST}(p_i) - b(w_{j_{p_i}})]$. Define $ST_1 = ST - t$. Now we must show that ST_1 is a feasible start time. Consider the following two cases:

Case 1: If p_i is contained in a single work period in ST , then the following argument holds:

$$\begin{aligned}
b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\
&= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\
&\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\
&= b_{ST}(p_i) - t \\
&= b_{ST_1}(p_i)
\end{aligned}$$

Also, $e_{ST_1}(p_i) < e_{ST}(p_i) \leq e(w_{j_{p_i}})$. Thus, p_i is contained in a work period.

Case 2: If p_i is contained in multiple work periods in ST , then there is at least enough work time in the period between $b_{ST}(p_i)$ and $b_{ST}(p_i)$ to contain p_i . Now, consider the time interval $(b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})], e_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})])$. Since

$$\begin{aligned} b(w_{j_{p_i}}) &= b_{ST}(p_i) - b_{ST}(p_i) + b(w_{j_{p_i}}) \\ &= b_{ST}(p_i) - (b_{ST}(p_i) - b(w_{j_{p_i}})) \\ &\leq b_{ST}(p_i) - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \end{aligned}$$

Then we have an interval the same length as the interval $(b_{ST}(p_i), b_{ST}(p_i))$ and since we have added $\min[b_{ST}(p_i) - b(w_{j_{p_i}})]$ amount of work time to the new interval, then the new interval has at least as much work time as the old interval. Thus, it is easy to see that by intelligently rearranging the slack in the new interval it is easy to see that we can schedule p_i to be contained by work time.

In addition, consider the following argument:

$$\begin{aligned} L &= ST - ST + L \\ &= ST - (ST - L) \\ &= ST - (b_{ST}(p_1) - b(nw_1)) \\ &\leq ST - \min[b_{ST}(p_i) - b(w_{j_{p_i}})] \\ &= ST - t \\ &= ST_1 \\ &< ST \\ &\leq U \end{aligned}$$

Thus, we have a new feasible schedule in which at least one port time period and one work period begin at the same time. Hence the lemma is true. \square

However, once again we have to modify our algorithm for this particular problem. Since the schedule is not cyclical, then we do not have the freedom to choose where within the schedules to start. First, we note that since we are only concerned with matching port time to work time then we can remove any travel time that occurs at the beginning or ending of the Ship Schedule. Secondly, we note that to test starting p_i at the same time as w_j , we formulate two problems. We first test to see if we can find a feasible schedule given the Ship Schedule starting with p_i and the Work Schedule starting with w_j . This is easily done using Find_Feasible. Next, we use Find_Feasible again using the reverse of the ship schedule as Ship_Schedule ending with the period before p_i and the reverse of the work schedule ending with the period before w_j to test feasibility for the first portion of the schedule.

5.2.14 1 ship / Minimize overlap / no slack

Given one of the 1 ship / General work schedules / no slack problems, if a feasible solution does not exist, can we find a solution with minimum port / nonworking period overlap?

Let an instance be given. Let S be a start time. Note that there are four types of overlap:

1. $b_S(p_i) < b(nw_j) < e_S(p_i) < e(nw_j)$
2. $b(nw_j) < b_S(p_i) < e(nw_j) < e_S(p_i)$
3. $b_S(p_i) < b(nw_j) < e(nw_j) < e_S(p_i)$
4. $b(nw_j) < b_S(p_i) < e_S(p_i) < e(nw_j)$

and two types of potential overlap:

1. $b_S(p_i) < e_S(p_i) = b(nw_j) < e(nw_j)$
2. $b(nw_j) < e(nw_j) = b_S(p_i) < e_S(p_i)$

Let s be a small amount of time. Now consider the start time $S - s$. Assuming that s is small enough, each overlap of type 1 is reduced by s , each overlap of type 2 is increased by s , and each overlap of type 3 or 4 remains the same. Also, we have an additional overlap of s for each potential overlap of type 2. We will call S a *left local optimal* start time if $S - s$ has more overlap than S and $S + s$ has at least as much overlap as S . Thus if S is a left local optimal start time, then the number of type 1 overlaps plus the number of type 1 potential overlaps exceeds the number of type 2 overlaps and the number of type 2 overlaps plus the number of type 2 potential overlaps \geq the number of type 1 overlaps. Similarly, we define a *right local optimal* start time if $S + s$ has more overlap than S and $S - s$ has at least as much overlap as S .

Let O_L = the set of left local optimal start times and O_R = the set of right local optimal start times. Let O_G = the set of globally optimal start times. Define $z(S)$ to be the amount of overlap of start time S . Assume that O_G is not the entire set of start times. Note that S is continuous which implies that z is continuous. Then, there is at least one start time in O_G that is also in O_L . Thus if we can find all solutions in O_L , then we can find a globally optimal start time. Consider L , the set of solutions with at least one potential overlap. It is clear that $O_L \subseteq L$. Note that L is exactly the set of solutions that CSSA produces.

5.2.15 Multiple ships / Cyclical Schedules / No slack

Assume that we are given a route, travel and port times, and a work schedule. We wish to create a schedule with a *frequency policy*, F . (Note that we use *policy* instead of a number since there may be various descriptions such as 'daily' or 'weekly' but there also may be descriptions such as 'every weekday' or 'Mon., Wed., Fri.')

Is there a feasible schedule with policy F ? For the 'no slack' cases, we will assume that the Ship and Work Schedules are of length, T .

Fixed Time Frequency Policy A Fixed Time Frequency Policy provides a specific time link between the Work Schedule and the Ship Schedule. Some examples include:

1. Each ship leaves Port 1 at 9:00am Monday.
2. Ship 1 arrives at Port 1 at 10:00am Tuesday.
Ship 2 arrives at Port 3 at 5:00pm Thursday.
Ship 3 departs Port 6 at 2:30am Saturday.
Etc.

A Fixed Time Frequency Policy is easy to solve. We simply test the feasibility of the Work Schedule and Ship Schedule time link for each ship.

Fixed Time Windows Frequency Policy This case is similar to a Fixed Time Frequency Policy, except that a time window is given for each ship. Some examples include:

1. Ship 1 leaves Port 1 on Monday, Ship 2 leaves Port 1 on Wednesday, Ship 3 leaves Port 1 on Friday, etc.
2. Ship 1 leaves Port 2 between noon and midnight, Ship 2 leaves Port 4 between 6:00am and 10:00am, etc.

Solution: Test for a feasible schedule for each ship. Since the ships departure times are independent, then we are simply testing the feasibility of each ship, thus we can use a previous lemma regarding feasibility and a time window on the start time of the ship.

Lemma 10 *If a feasible schedule for a ship exists such that either a port / work period pair start at the same time or the event described happens at the earliest time possible in the time window.*

Relative Time Frequency Policy A Relative Time Frequency Policy provides a relative time link between the ships. Some examples include:

1. 5 ships equi-time apart ($T/5$)
2. Ship 2 is 3 days behind ship 1, ship 3 is 2 days behind ship 2, etc.

This can be translated into a *frequency schedule*, i. e. a schedule of length T which indicates the time spacing between the ships.

Now, assume that we have a series of feasible start times, ST_1, ST_2, \dots , which satisfy the frequency schedule. Consider the following lemma:

Lemma 11 *Given a series of feasible start times exists, then there exists a series of feasible start times such that at least one ship arrives in port at the beginning of a work period.*

Proof. Let an instance of the problem be given. Assume we have a series of feasible start times, ST_1, ST_2, \dots . Then each port time, $p_{s,i}$, is contained in a work period. Let $j_{s,i}$ be the subscript of the work period containing it. If there is a $p_{s,i}$ such that $b_{ST_s}(p_{s,i}) = b(w_{j_{s,i}})$, then we are done. Assume not. Now, let $t = \min[b_{ST_s}(p_{s,i}) - b(w_{j_{s,i}})]$. Define $\bar{ST}_s = ST_s - t$. Now we must show that \bar{ST}_s is a feasible start time. For any s and i ,

$$\begin{aligned}
 b(w_{j_{s,i}}) &= b_{ST_s}(p_{s,i}) - b_{ST_s}(p_{s,i}) + b(w_{j_{s,i}}) \\
 &= b_{ST_s}(p_{s,i}) - (b_{ST_s}(p_{s,i}) - b(w_{j_{s,i}})) \\
 &\leq b_{ST_s}(p_{s,i}) - \min[b_{ST_s}(p_{s,i}) - b(w_{j_{s,i}})] \\
 &= b_{ST_s}(p_{s,i}) - t \\
 &= b_{\bar{ST}_s}(p_{s,i})
 \end{aligned}$$

Also, $e_{\bar{ST}_s}(p_{s,i}) < e_{ST_s}(p_{s,i}) \leq e(w_{j_{s,i}})$. Thus, each $p_{s,i}$ is contained in a work period. In addition, since we subtracted the same amount of time from each schedule, then the new series of start times has the same time spacing as the old series of start times which implies that the new series of start times is feasible. \square

Thus we can use the 'no-slack' algorithm to find a single feasible schedule and then test it in each position of the relative schedule for feasibility.

Relative Time Windows Frequency Policy This case is not as straightforward as the previous cases. Both of the fixed time cases are straightforward in that we can reduce the problem to testing feasibility for each ship separately to determine whether the overall problem is feasible. Now, while the Relative Time Frequency Policy has interdependent ship start times, it has the benefit that given the start time of any ship, the start time of the remaining ships is then fixed. However, introducing time windows into the relative time case presents some problems. The first problem is problem definition. Consider the following two examples. While the problem statement is similar, they are structurally different.

1. Ship 2 starts between 28 and 35 hours after Ship 1.
 Ship 3 starts between 28 and 32 hours after Ship 2.
 Ship 4 starts between 28 and 32 hours after Ship 3.
2. Ship 2 starts between 28 and 35 hours after Ship 1.
 Ship 3 starts between 56 and 63 hours after Ship 1.
 Ship 4 starts between 84 and 90 hours after Ship 1.

The following lemma from the Relative Time Frequency Policy holds for this case as well.

Lemma 12 *Given a series of feasible start times exists, then there exists a series of feasible start times such that at least one ship arrives in port at the beginning of a work period.*

The proof for this case is identical to the proof for the relative case. Thus we can use the 'no-slack' algorithm to identify schedules which may be included in a feasible solution. However, for this case, even if we could identify a ship schedule which was guaranteed to be included in a feasible series of schedules, it is not clear how to identify the remaining schedules in the series.

Assume we are given a **Multiple ships / Cyclical Schedules / No slack** problem of the **Relative Time Windows Frequency Policy** variety. First, we examine the information we must be given.

1. A ship schedule = travel times, tt_i and port times, $p_i, i = 1, \dots, n$.
2. A work schedule = work period times, wp_i and nonwork period times, $nwp_i, i = 1, \dots, m$.
3. A frequency policy, F = a set of minimum times, $Fmin_i$, and window lengths, $Fwin_i, i = 1, \dots, f$ (I. e. f = number of ships).

First, we construct the set of feasible start times for the associated single ship problem, call it FST_1 . This we can do in $O(nm)$ time. We note that the set of feasible start times will consist of no more than nm intervals. Thus now we can easily determine whether a given start time is feasible or not and given an infeasible start time, we can determine the next occurring feasible start time.

Now, consider the following transformation:

1. Define $\bar{t}t_i = Fmin_i$ and $\bar{p}_i = 0, i = 1, \dots, f$.
2. Define $\bar{w}p_i =$ the length of the i th feasible interval and $n\bar{w}p_i =$ the length of the i th non-feasible interval in FST_1 .

We now have a **Single ship / Cyclical schedule / slack** problem where we are seeking a solution in which each port (starting of a ship) time occurs during a work (feasible starting time) period. However, we now have the added stipulation that we can insert no more than $Fwin_i$ slack at the i th port. This is a problem similar to, but more difficult than problems we have solved before. Thus, we have a dual purpose for solving this problem. It will extend existing work in two extremely different directions.

Lemma 13 *Given a Single Ship / Cyclical Schedule / No slack problem in which all data (both the ship schedule and work schedule) are integers, then any start time which has the ship arriving at the beginning of a work period will have an integral start time.*

Proof. Let a **Single Ship / Cyclical Schedule / No slack** problem with all integral data be given. Let ST be a start time in which the ship arrives at port i at the beginning of work period j . We wish to show that ST is integral. Now, $ST =$ the sum of all work and nonwork periods preceding work period j - the sum of all port and travel periods preceding port period i . Now since all data is integral, the sum of all work and nonwork periods preceding work period j is the sum of integers which implies the sum is integer. Similarly,

the sum of the port and travel periods is an integer. Thus, since ST is the difference of two integers, then ST is an integer. \square

Similarly, we can prove that any start time in which the ship leaves a port at the end of a work period will be integral.

These two results imply that if our **Multiple ships / Cyclical Schedules / No slack** problem has all integral data, then the feasible-non-feasible schedule generated will consist of integral data.

Conjecture 1 *If a Multiple ships / Cyclical Schedules / No slack problem has all integral data and a feasible solution exists, then a feasible integral solution exists.*

5.2.16 Multiple ships / Cyclical Schedules / Slack

Fixed Time Frequency Policy A Fixed Time Frequency Policy provides a specific time link between the Work Schedule and the Ship Schedule. Again, the Fixed Time Frequency Policy is easy to solve for the slack case. We simply test the feasibility of the Work Schedule and Ship Schedule time link for each ship.

Fixed Time Windows Frequency Policy This case is similar to a Fixed Time Frequency Policy, except that a time window is given for each ship. Since the ship departure times are independent, then we simply test the feasibility of each ship, thus we can use a previous lemma regarding feasibility and a time window on the start time of the ship.

Relative Time Policies For the slack version of the Relative Time Policies, we have problems similar to the 'no slack' case. We also have an additional definition problem. If we examine two final ship schedules which do not have slack occurring in the same places, then the time difference between fixed points in the schedules will not always be the same. For example, ship 2 may leave the first port three days after ship 1, however, ship 2 may leave the third port only two days after ship 1 due to the location of slack in the two schedules. Thus, we propose the following: relative time policies will ignore slack. In other words, relative time policies will be stated in terms of the ship schedules whereas fixed time policies are stated in terms of the work schedules. Note that this definition is reasonable and consistent with the 'no slack' case. In that case, the difference is transparent since the ship and work schedules are of the same length.

5.3 Extensions, Generalizations, and Questions

1. Assume $RTT \neq 7$. We could stack multiple round trips in hopes of obtaining a feasible repeating schedule with less slack. Note that we should be able to use the algorithms for the above cases to solve this problem.
2. Is there an easy way to determine alternative ways of placing ships? For example, given a three week schedule, is it possible to place 2 ships 10.5 days apart?

3. What is the limit of these algorithms? I. e. for what general cases do they NOT hold?
4. What insight can these algorithms provide for setting feasible ranges for the port and travel times?

6 Volvo Shipping Problem

Volvo is considering shipping all of its cars to the United States from the plant to the dealers entirely in containers. Thus it will contract with Shipping Company to transport the containers from Europe to the U.S. and with Trucking Company to transport the containers to the individual dealers. We wish to find the plan which minimizes total cost.

6.1 Main Issues

European Port(s): Assuming that Volvo is using its own drivers to move the containers from plant to port, then Volvo would have two considerations in choosing a European port: 1) distance from plant to port in order to minimize transport costs and 2) port storage costs in order to employ a small number of drivers which are busy continuously. In addition, assuming that variable costs (per container costs such as per container loading/unloading charges, tariffs, etc.) are passed directly to the shipper, Volvo would want these costs considered in choosing an European port. However, Shipping Company would consider the fixed port costs (docking fees, lease agreements, etc.) in choosing the European port to operate the service from. In addition, Shipping would consider the number of inbound and outbound containers that it would have access to in order to have a full ship. In fact, to ensure full loads, Shipping might have to operate out of multiple ports. This would increase the fixed costs Shipping would incur and the inventory carrying costs Volvo would incur and thus the total costs. In actuality, various government regulations may also affect the choice of European ports.

Service: Both Volvo and Trucking Company would want frequent service. Frequent service would reduce Volvo's inventory costs (storage, investment costs, etc.) and provide Trucking with a smaller, more manageable job (per shipment) on the American side. However, frequent service would require Shipping to operate more ships for the same route and would possibly require additional container bookings to ensure full ships.

American Port(s): We assume the per-container charge to Volvo from Shipping is based on the distance from the European port to the American port and that the change in distance from the European port to various American ports is minor. Then, assuming that it is cheaper to transport containers by ship than truck, Volvo would certainly want Shipping to provide service to many American ports. Also, again assuming variable costs are passed back to the shipper, Volvo would also want to use the ports with lower variable costs. However, Shipping would want to use as few ports as possible and choose these ports based on fixed port costs and the number of outbound and additional inbound containers necessary to insure full sailings. Finally, Trucking would want the

ports that Shipping serves to depend on per-container storage fees and would possibly want to use ports with which it has existing business.

6.2 Additional Issues

1. Whose containers should we use? How should we deal with the empty containers? It certainly seems that Volvo would use racks to ship the cars. Certainly these would need to be returned to the plant in some way.
2. What is the ideal capacity of the ships?
3. How many trucks do we need?
4. Given that we have a large number of containers arriving at a port, how do we schedule and route the trucks in order to balance delivery service requirements with steady work for the truck drivers?
5. Stowage of containers: A particular container would have to be shipped to a particular dealer. The order containers are handled in would affect cost. Given a loading order, can we determine the cost?