Specialization: Transport Engineering and Logistics

Report number: 2013.TEL.7775

Title: Building an integer programming model to plan Inter Terminal Transportation

Author: F.E. Nieuwkoop

Title (in Dutch) Integer programming model om Inter Terminal Transport te modeleren.

Assignment: Computer assignment

Confidential: no

Initiator (university): Dr. R.R. Negenborn

Supervisor: Dr. R.R. Negenborn and Ir. M.B. Duinkerken

Date: May 7, 2013

This report consists of 40 pages and 6 appendices. It may only be reproduced literally and as a whole. For commercial purposes only with written authorization of Delft University of Technology. Requests for consult are only taken into consideration under the condition that the applicant denies all legal rights on liabilities concerning the contents of the advice.
Building an integer programming model to plan Inter Terminal Transportation

The large container ports around the world use inter terminal transportation (ITT), to transport containers between different container terminals, distribution parks and empty container depots in the port area. The investment costs are high and sufficient space has to be allocated for the ITT and therefore proper planning and design is vital for an efficient and congestion free transportation system.

In the case of the Maasvlakte 2, a port expansion project in the Port of Rotterdam, an ITT system still has to be developed. One way in determining and optimizing the properties of such a system is by building a model, which represents the situation and can simulate its operations. In Tierney et al. (2012) an integer programming model was built to analyze the ITT at i.a. the Maasvlakte 2. This model uses time space graphs with congestion to model vehicle flows through a network of transfer stations and intersections focusing on minimizing the overall delay of the cargo.

The goal of this assignment is to get a better understanding of this model by building a new general applicable model in Matlab using the same mathematical principles as in Tierney et al. In short your assignment is to:

1. Determine the boundary conditions and constraints of the model.
2. Create arcs and nodes, implement vehicles properties and build an supply-demand generator;
3. Create the minimal cost flow algorithm using the defined input;
4. Implement time into your model and increase the complexity by:
   - build a time-place diagram;
   - let the vehicles move with constant speeds and infinite arc capacity;
   - implement congestion in the model.

Based on your assignment, it is expected that you conclude with a recommendation for future research opportunities and potential for more ideas and/or applications. The report must be written in English and must comply with the guidelines of the section. Details can be found on the website.

For more information, contact Dr. Rudy Negenborn (8B-1-05; r.r negerborn@tudelft.nl) or Ir. M.B. Duinkerken (34B-1-260; m.b. duinkerken@tudelft.nl).

The professor,

Prof.dr.ir. G. Lodewijks
Summary

Since the introduction of containerized trade in 1956, containerization has spread across the globe and still shows a strong yearly volume growth. This growth supported port expansions all around the world and in the case of the port of Rotterdam the growth in container volume contributed in the port expansion project Maasvlakte 2. The Maasvlakte 2 is constructed next to the already existing Maasvlakte 1 area and increases the ports total container capacity up to almost 35 million TEU annually. To be able to transport all this cargo efficiently through the combined Maasvlakte area an Inter Terminal Transport (ITT) system is considered.

In Diekman and Koeman [2010] a few scenarios are presented containing future ITT demands. These scenarios were used by Tierney et al. [2012] as an input for building an integer optimization model, which optimizes the flow of containers and vehicles through the Maasvlakte area by minimizing the occurred delays of containers. With this model the researchers try to provide a tool for the Port of Rotterdam Authority to be able to determine the specifics of an ITT system which can guide them in their decision making process. In order to determine the performance of the integer model this report will find an answer to the research question: to what extent will the model of Tierney et al. [2012] give reliable and realistic results and what improvements could be made to increase its reliability?

To be able to give an answer to this question a model based on the same working principles as in Tierney et al. [2012] has been build in Matlab using the CPLEX for Matlab integration to solve the optimization. First the basic principles of the model are presented, followed by model structure including assumptions, constraints and objective. Finally a verification is made where the model is verified for small scale models after which a conclusion can be drawn about the reliability and realism of the model.

This research will contribute to the understanding of the feasibility and realism of integer programming models for ITT systems. Also a critical view on the research presented in Tierney et al. [2012] is given, which may result in an update of their paper. It will also provide some recommendations for further research to make an integer programming model more realistic and reliable.

The mathematical principles of the integer model are based on the minimal cost flow theory using a graph or network consisting of nodes and arcs, where flows can travel through this network from a source node to a sink node. A time expansion is added, which copies the base graph and makes layers of the base graph on top of each other. The layers are connected by adding arcs between the same nodes through time which are called stationary arcs. The arcs connecting different nodes to each other in the base graph will now be connected from a node to the other node one or several layers above. A representation of the time expanded model containing two nodes is shown in Figure 2.2. To add multiple demands of containers to model, the multi-commodity flow theory is introduced. This theory adds a new set of variables and constraints for each individual demand to the network ensuring that demands will flow correctly through the network.

The ITT system is build from a base network, where each terminal represents a node which is connected to an intersection, terminal or LT node by arcs. Intersections are also modelled by nodes and combine the flows from all directions entering the intersection and directs them through a funnel before directed out of the intersection towards the destination. This funnel is in place to represent congestion arising at the intersection when the supply of vehicles exceeds a certain value. This funnel is modelled by creating two nodes for every intersection. These nodes
are connected by so called fan arcs as presented in Kohler et al. [2002], where each fan arc is assigned with capacity and a transverse time.

Next to road vehicles the model also provides the opportunity to use other transportation options such as barges or trains. Because these options require additional loading and unloading time LT nodes are introduced to be able to implement a correct representation of long term vehicle usage. The LT node network is a separate set of variables and can therefore be added or removed from the total network when LT vehicles are present or not.

As already has been mentioned both road vehicles as well as long term vehicles can be implemented in the model. There are three possible road vehicles: Automated Guided Vehicles (AGVs), Automated Lifting Vehicles (ALVs) and Multi Trailer Systems (MTSs). At this stage, one type of LT vehicle can be incorporated: barges. These vehicles have several assigned properties such as speed, handling rate and load quantity. These properties are used to realistically constrain the flows through the network.

The vehicles will transport the container flows through the network. These flows are generated by a demand generator determining the number of containers to be transported, the priority of these containers and assigns an origin and destination terminal. The generator also determines the start time and the delivery or due time for the containers. This data is combined in a demand vector and can contain multiple demands, thanks to the introduction of multi-commodity flow theory. The amount of demands generated is influenced by the user who can give a total number of containers flowing through the network as an input. This total number is divided in individual demands with a random amount of containers between 1 and 50 containers.

The constraints that are introduced will constrain the model in such a way that the flow is only able to follow the arcs through the network. The flow is bounded by the demand vector, which stores the information on number of containers, origin/destination and start/delivery period. Every container flowing through the network requires a transport vehicle to do so. These transport vehicles are constraint in number and a balancing constraint ensures that it is not possible that more vehicles are able to leave a node than there are vehicles present in that node. The vehicle type also constraints the loading and unloading rate of containers at a terminal. To model congestion some additional constraints are required identifying which fan arc is allowed to have a flow over the arc. Finally all stationary arcs can be used by all vehicle types and are unconstrained in the amount of flow over the arc.

These constraints are combined and are used by the objective function to calculate the minimum amount of delay for all container demands to reach their destination. This objective function assigns a penalty to arcs entering the destination node after the given delivery time of the demand. The height of this penalty is determined by the amount of delay and the priority of the cargo.
The model has been verified for small size problems, with 2 nodes and a maximum of three time periods. This allowed the writer to manually verify the flows through the network and the optimization results. The model is verified for all vehicle types and for multi-commodity demands. The verification showed that the behaviour of the model was as expected on forehand and therefore the constraints are implemented correctly. It is also shown that the size of the problem quickly grows in size when additional vehicles or commodities are added. This makes it important to optimize the layout of the network and to reduce the amount of base variables as much as possible to limit the calculation times and required memory.

The main research question of this research is to what extent will the model presented by Tierney et al. give realistic and reliable results for the application of integer optimization models in the development process of inter terminal transportation systems? The following conclusions answer this question:

First of all it is important to realize that it is an optimization model optimizing transport routes by minimizing the total delay of containers. This means that the results are only useful when optimization of transport routes in the real ITT implementation is also incorporated in the system. Otherwise the results should be interpreted as the minimum requirements for an ITT system. However the implementation of an objective function can result in less overcapacity in the system and reduces the overall implementation and operational costs. Integer programming is easily adaptable to other port situations and a proper constructed base model can quickly provide guidelines for ITT development and decision making processes.

Unfortunately the mathematical principles also impose some drawbacks and the implementation of the model can be improved in some ways. The following itemization will give an overview of drawbacks and points for improvement.

- The discrete nature of the model imposes a drawback on the realism of operations that are performed, because operations having a duration less than one or a part of a time period will either be neglected or rounded to the nearest time period. This also makes it impossible to implement time distributions to container or vehicle operations.
- It can be questioned whether congestion will create any problems at all, because the total area covered by the ITT system is almost 30 km with a maximum amount of 125 vehicles. This means a low vehicle density which makes the chance that more than 40 vehicles at the same time will cross each other at an intersection will be small.
- ITT by barge might be unrealistic, because there are long wait times before being handled at a terminal in the Port of Rotterdam.
- The demand generator has to be improved and validated to generate more realistic demands.
- Vehicle properties used in the model are outdated and should be updated to modern standards. Also the application in ITT might change the vehicle properties, because AGVs, ALVs and MTSs are only applied on terminals, where they have a different functionality.

It is recommended for future research to validate the integer programming model to real situations or other valid models to determine its own reliability for the use in ITT development. It is also recommended to investigate how ITT vehicle planning optimization should be implemented in ports to be able to have a meaningful model.
# Contents

1 Introduction ............................ 7
   1.1 Problem statement .................................. 8
   1.2 Main research question .......................... 9
   1.3 Approach ........................................... 9
   1.4 Contribution of the research .................. 9

2 Basic mathematical principles .......... 11
   2.1 Base structure ................................... 11
   2.2 Time expansion .................................. 12
   2.3 Total layout in matrix representation ........ 13
   2.4 Flow and multi-commodity flows ............. 13
   2.5 Discussion ...................................... 14

3 Model structure .......................... 15
   3.1 Key assumptions ................................ 15
   3.2 Node and arc types .............................. 15
   3.3 Demand generator ............................... 17
   3.4 Vehicles, vehicle types and containers ....... 19
   3.5 Parameters ...................................... 19
   3.6 Objective function .............................. 19
   3.7 Constraints ..................................... 20
   3.8 Total model ..................................... 21
   3.9 Discussion ...................................... 22
   3.10 Concluding remarks ............................ 24

4 Verification of the model ............... 25
   4.1 Input parameters ................................. 25
   4.2 Model output .................................. 27
   4.3 Concluding remarks ............................ 29

5 Conclusion and future research ........ 30

A Results of scenario 1 ................... 33

B Results of scenario 2 .................... 34

C Results of scenario 3 .................... 35

D Results of scenario 4 .................... 36

E Results of scenario 5 .................... 38

F Results of scenario 6 .................... 39
Chapter 1

Introduction

Since the introduction of containerized trade in 1956, containerization has spread across the globe. The main functions of containers are protection against damages and theft, but foremost, the standardization of containers made this form of transport very cost effective and economies of scale became possible. The standard size of a container was set in 1961 by the International Organization for Standardization (ISO) and is 20ft long, 8ft wide and 8ft 6in high. This container is commonly referred to as a twenty foot equivalent unit or TEU. The widespread use of containers in worldwide trade resulted in hundreds of ports with specialized terminals to load and unload containers from ships.

The largest port in Europe, Rotterdam (The Netherlands), has multiple terminals handling containers. With the expansion of the port by the Maasvlakte 2 project, the port will add 5 new container terminals increasing the annual container throughput up to 34.6 million TEU in 2040 [Visser et al., 2012]. In Figure 1.1 an artist impression is shown how the Maasvlakte 1&2 will look like in 2030 when all infrastructure and terminals will be build.

![Figure 1.1: Maasvlakte 1&2 (Maasvlakte2.com)](image)

The container flows through Rotterdam are heading for destinations across Europe by truck, train and barge. In order to effectively handle these large amounts of containers infrastructure inside the port, as well as throughout the entire nation and European Union must have sufficient capacity to handle this flow. In the case of the Maasvlakte 2 expansion the capacity of the hinterland transport has been enlarged by i.e. widening parts of the highway A15, the construction of...
the dedicated freight railway line ‘the Betuweroute’ and the construction of an inland container transferium just outside Rotterdam, where containers will be fed by barge [Port of Rotterdam Authority, 2010].

The previous examples served the goal of providing fast connections to the hinterland of the port. However the transport of containers inside the port, between terminals and other service providers, should have a sufficient capacity as well. In Diekman and Koeman [2010] the Inter Terminal Transport (ITT) on Maasvlakte 1 & 2 has been investigated whether the current infrastructure would have sufficient capacity or that a new closed transport route should be constructed. This closed transport route is closed for public traffic and therefore vehicles on this road do not have to apply to national laws for vehicles on public roads. This makes it possible to choose for other transport options than trucks such as Automated Guided Vehicles (AGVs), Automated Lift Vehicles (ALVs) or Multi Trailer Systems (MTS).

This research provided three scenarios about the amount of containers transported between terminals based on different assumptions. The results shown in Table 1.1 showed that a minimum amount of 2.1 million TEU per year will be transported internally across the Maasvlakte in 2035. Although these numbers are large, it is expected that until 2020 current infrastructure is capable of handling these ITT flows. However after 2020 it is advised to build a closed transport route with 1 lane in both directions to efficiently handle all container flows across the port.

<table>
<thead>
<tr>
<th>Findings</th>
<th>Total ITT in million TEU/year</th>
<th>Total ITT rush hour (3x avg.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>scenario 1</td>
<td>4.9</td>
<td>997</td>
</tr>
<tr>
<td>scenario 2</td>
<td>3.4</td>
<td>685</td>
</tr>
<tr>
<td>scenario 3</td>
<td>2.1</td>
<td>439</td>
</tr>
</tbody>
</table>

Table 1.1: Amount of containers in TEU/year for different scenarios

1.1 Problem statement

The route itself has been determined, but the Port of Rotterdam Authority (PoRA) still needs to make a decision about which type of vehicles will be used for transportation. In Diekman and Koeman [2010] 4 different vehicle concepts for the closed transport route and two different concepts for the public road have been investigated in relation to operational costs:

1. Terminal tractor with chassis (closed)
2. Multi Trailer System (closed)
3. AGV (closed)
4. ALV (closed)
5. Truck (public)
6. 3-TEU truck (public)

This resulted in the average cost per TEU as shown in Figure 1.2.

However a proper conceptual choice not only depends on operational costs and in this study factors like i.e. investment cost of transport equipment, reliability and maintenance costs have not been included. Also a tool that analyses the improvements in port efficiency versus the costs of the installed infrastructure and equipment would be welcome to support the decision making process for the PoRA. This tool has now been developed by Tierney et al. [2012], which uses a mathematical model based on integer programming to model the inter terminal container and vehicles flows and minimizes the delay in the arrival of containers at their destination. The model includes traffic congestion, multiple vehicle types, loading and unloading times and port layout configuration. Although this model looks very promising no validation of the model has been done yet and therefore one is interested in the reliability and realism of the proposed model.
1.2 Main research question

To investigate the properties of the model presented by Tierney et al. [2012] this report will give an answer to what extent the integer programming model will give reliable and realistic results and what improvements could be made to increase its reliability?

1.3 Approach

To answer the main research question an integer programming model is build and verified, which uses the same mathematical principles as the model presented in Tierney et al. [2012]. The model has to be able to simulate different vehicle types and combinations of vehicle types carrying containers between different locations on the Maasvlakte 1 and 2 in the Port of Rotterdam. The model will optimize the way vehicles and containers move between terminals in order to receive the lowest amount of delay, while experiencing congestion, loading/unloading times and driving times. It will be constructed in Matlab and solved by the CPLEX for Matlab optimizer.

The new model will provide a better understanding to the researcher about the working principles of integer modelling, the model structure and to what extent the model presented in Tierney et al. [2012] is able to provide realistic and reliable results. This enables the researcher to discuss the realism and reliability of this model and come up with some recommendations for future research.

The report containing the results of the research has been structured as followed. First the mathematical principles on which this model has been build are presented. Secondly an overview on the model has been given, containing the variables and constraints after which the realism of the model structure is discussed. Finally the model will be verified by running a small size model and comparing the results with the writers expectations.

1.4 Contribution of the research

This research will contribute to the understanding of the feasibility and realism of integer programming models for ITT systems. Also a critical view on the research presented in Tierney
et al. [2012] is given, which may result in an update of their paper. It will also provide some recommendations for further research to make an integer programming model more realistic and reliable.
Chapter 2

Basic mathematical principles

This chapter presents the basic mathematical principles on which the model described in Tierney et al. [2012] is build. The basic structure is commonly used in Operational Research based on network flow theory. To improve the comparability between this report and Tierney et al. [2012] the used parameters in Tierney et al. [2012] are also used in this report and their symbols are kept the same in most instances.

2.1 Base structure

The model is based on the graph theory which uses nodes and arcs. The nodes represent vertices and the arcs, representing the edges, are used to connect the nodes into a network. The arcs also have a capacity and can be directed or undirected. This representation can than be used to model a flow through the network. Flows have an incoming and an outgoing node who are called the sink and source respectively. All nodes in the network have the restriction that the flow entering a node equals the flow exiting the node, making it impossible to ‘stick’ in a certain node. These network flow models are widely used to model various types of relations and processes in subjects ranging from logistical, chemical, biological, social, information and communication systems. In the case of this research network flows are used to model the ITT through the Maasvlakte 1&2, where transport vehicles and containers will flow through a network of roads connecting each terminal. To get a better visual understanding of how a network is mathematically modelled a representation is given in Figure 2.1.

![Basic network containing 2 nodes connected by arcs](image)

Figure 2.1: Basic network containing 2 nodes connected by arcs

To be able to solve problems using networks Ford and Fulkerson [1956] were the first to construct
an algorithm which was able to find the maximum flow through a network. This algorithm makes use of a given graph \( G(V, A) \) where \( V \) is a set containing all nodes and \( A \) is a set containing all arcs \((i, j)\) where \( i, j \in V \). Each arc \((i, j)\) \(\in A\) has a positive capacity \(c(i, j)\). Two nodes are distinguished of \( V \), one as the source \( s \) and one as the sink \( t \). The algorithm now determines the maximum flow \( f(s, t) \) from node \( s \) to node \( t \) in graph \( G \) as \( \sum_{(i,j) \in A} f(i, j) \) where value \( f(s, t) \) is the max flow from node \( s \) to node \( t \) and under the constraints of:

\[
\begin{align*}
f(i, j) &\leq c(i, j) \quad \forall (i, j) \in A \quad \text{(capacity constraint)} \\
f(i, j) & = -f(i, j) \quad \forall (i, j) \in A \quad \text{(anti symmetry constraint)} \\
\sum_{j \in V} f(i, j) &= 0 \quad \forall i \in V - \{s, t\} \quad \text{(flow conservation constraint)}
\end{align*}
\]

In the scope of this research we are not interested in finding the maximum flow on itself, but in finding the minimum delay of containers in the network. To be able to provide such a solution the previous described network has to be expanded to a minimum cost flow problem as described in Edmonds and Karp [1972]. A cost function is added assigning a non negative cost \( p(i, j) \) to every arc \((i, j) \in A\), which makes the cost of flow \( f(s, t) \): \( \sum_{(i,j) \in A} p(i, j)f(i, j) \) and take the value of \( f(s, t) \) as the total cost of the maximum possible flow.

In most instances however one is interested in a finite amount of flow from source to sink instead of the maximum possible flow. In this case some additional constraints are required. A flow is introduced of \( \theta \) from node \( s \) to node \( t \). To ensure that this flow is created and drained from the system a constraint is added:

\[
\sum_{j \in V} f(s, j) = \theta \quad \text{and} \quad \sum_{j \in V} f(j, t) = -\theta
\]

### 2.2 Time expansion

A time space network is used to model nodes in time in order to be able to incorporate factors like driving times, waiting times and loading times experienced by vehicles in the inter terminal transport. Time is introduced in the model by copying the base network on top of each other. Let \( \tau \) be the number of time periods, than graph \( G^T = (V^T, A^T) \). Let \( n \) be the number of terminal nodes and \( m \) be the number of intersections in the base graph. Because the nodes will be copied through time this will mean that the length of vector \( V^T = \tau(n + 2m) \). Two nodes are introduced for every intersection, the reason for this will be explained in Section 3.2. This is represented in Figure 2.2, where the two terminals represented by node 1 and 2 from Figure 2.1 are copied. In this case time is represented on the vertical axis and the location on the horizontal axis. Arcs are connecting the nodes through time, where the arcs between node 1 and 2 could represent driving and loading time while the arcs between the same node in time could represent waiting or idling time. This means that arcs not only have capacity and cost properties, but also time properties [Yan and Shih, 2007]. The arcs between the same node trough time are called stationary arcs and are applied to every node.

In this way time is modelled discreet and therefore the time step size and the actions taking place in these steps should be proportional in order to have a reasonable amount of slack time within a period. The reliability of a model increases when the amount of slack time per time period is lower, because this would certain tasks take more or less time than in reality and over larger time horizons these errors could add up to large deviations from reality. On the other hand having a small time step over a large time horizon would increase calculation times significantly. Therefore a good balance between time step size and time horizon has to be found in order for the model to be meaningful while at the same time preventing over excessive calculation times [Haghani and Oh, 1996].
2.3 Total layout in matrix representation

Before any mathematical algorithm is able to perform calculations on network flows it needs an input which represents the arcs and nodes in matrix notation. The base network can be represented in matrix notation by making a vector $\alpha_v$ representing each arc $(i, j) \in A$. In this case the created vectors are combined in matrix $A_{eq}$ as columns representing the basic network, where all arcs are represented by a vector column and the nodes are represented as an element in the column. Therefore let the length of $\alpha_v$ be the number of nodes in the network, where

$$\alpha_v(i) = \begin{cases} 1 & \text{if } \alpha(i) \text{ is the origin node}, \\ -1 & \text{if } \alpha(i) \text{ is the destination node}, \\ 0 & \text{otherwise.} \end{cases}$$

In the case of the base network as represented in Figure 2.1 the two arcs will create vector $\alpha_v(1) = [1 \ -1]$ and $\alpha_v(2) = [-1 \ 1]$ into matrix $A_{eq}$.

$$A_{eq} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$ 

When the time expansion is applied, extra columns will be introduced because extra arcs are added, but also additional rows will be added because more nodes are present. The matrix representing the time extended network in Figure 2.2 would be:

$$A_{eq}^\tau = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 \end{bmatrix}.$$ 

2.4 Flow and multi-commodity flows

To model a flow through the network demand $\theta$, introduced in constraint 2.4, is incorporated in the set of flow conservation constraints presented in constraint 2.3. This creates the following set of constraints incorporated the network of matrix $A_{eq}^\tau$, where a flow of value $a_\theta$ is introduced from node 1 in timestep 1 to node 2 in timestep 2:

$$f_{1,2} + f_{1,1} = a_\theta$$
$$f_{2,1} + f_{2,2} = 0$$
$$-f_{2,1} - f_{1,1} = 0$$
$$-f_{1,2} + f_{2,2} = -a_\theta$$
for simplicity demand vector $b$ is introduced, which incorporates the values after the equality sign of constraints 2.3. In the case of previously introduced flow this will shape vector $b$ in

$$b = \begin{bmatrix} a_0 \\ 0 \\ 0 \\ -a_0 \end{bmatrix}.$$ 

This vector combined with matrix $A_{eq}$ in: $A_{eq}^T \cdot f = b$, forms a basic system of linear equations introducing the flow conservation constraints for the network.

**Multi-commodity flows**

In the scope of this research it is required to model more than one flow in a network, to represent flows of containers between different terminals. It is possible to create multiple demands by introducing the multi-commodity flow principle, where a commodity represents a single demand for the network. This was presented for the first time by Tomlin [1966].

A multi-commodity flow can be created by introducing $\Theta$ number of commodities. The generated flow vectors $b_{(\theta)}$, where $\theta$ is a single demand generation, will be combined in one large demand $b$ vector, where $b = [b_1, b_2, ..., b_\Theta]^T$. To create matrix $A_{eq}$, let $I$ be the unit matrix of size $\Theta$, then matrix $A_{eq}$ becomes $IA_{eq}$ or

$$A_{eq} = \begin{bmatrix} A_{eq_{\tau_1}} & 0 & 0 & 0 \\ 0 & A_{eq_{\tau_2}} & 0 & 0 \\ & & \ddots & \vdots \\ 0 & 0 & \cdots & A_{eq_{\tau_\Theta}} \end{bmatrix}.$$ 

### 2.5 Discussion

The network flow theory is used for modelling ITT, because it can simply represent the ports road network and integrate its terminals along it. This makes it easily adaptable to various port around the world. The mathematical principles behind the model provide the possibility to optimize the flows through the network, which could result in better cost expectations for infrastructure and equipment. It also creates the possibility to include any specific wishes into the model and analysing the effects on costs or other performance indicators.

A downside of the model is however that the discrete nature of integer programming results in a lower realism of the model, because operations with time requirements less than one time step will be neglected. It is also not possible to include time distributions on vehicle specific operations such as loading and unloading, or waiting for an intersection. When small time steps are chosen, the size of the problem will increase rapidly resulting in longer calculation times and a possible shortage of memory. Therefore it is inevitable to neglect some of the short term operations and to round of time requirements of operations to full time steps.

To investigate to what extent network flow models can represent reality in ITT operations it is recommended for future research to investigate how to validate these models with real port situations.
Chapter 3

Model structure

In this chapter the model structure based on the mathematical principles explained in Section 2.1 are laid out. The total model, as already mentioned in the Introduction, has to be able to simulate different vehicle types and combinations of vehicle types carrying containers between different locations on the Maasvlakте 1 and 2 in the Port of Rotterdam. The model will optimize the way vehicles and containers move between terminals in order to receive the lowest amount of delay, while experiencing congestion, loading/unloading times and driving times. The overall goal of the model is to identify whether the planned infrastructure is sufficient to handle the given flow and how many vehicles of which type are required to handle the inter terminal container flow. The model provides the option to include AGVs, ALVs, Multi-trailer systems and barges to be used for the ITT, but can be easily extended to include trains and trucks.

This chapter is structured as followed: first different types of nodes and arcs are discussed providing the overall layout of the Maasvlakте area. Secondly the demand generator is presented followed by some key assumptions of the model. Finally the variables, the objective and constraints are given.

3.1 Key assumptions

The model presented in this report will use the same assumptions used in Tierney et al. [2012] and are summarized below:

- Arcs in the model have only one vehicle type travelling on them. This results in that vehicle interaction is only possible during container handling at terminals;
- ITT is penalized for late arrival only. Arriving early is considered acceptable;
- All container types are considered to require a single unit of capacity on vehicles, i.e. no distinction between 20 and 40 feet containers is made;
- Short vehicle activities such as connecting a tractor to a trailer are neglected;
- No distinction in handling is made between different types of containers, such as dangerous goods, refrigerated or out-of-gauge containers;

3.2 Node and arc types

To assign constraints in Section 3.7 a few sets are introduced containing certain arcs. Let $In(i) = \{j | (j, i) \in A_T^T\}$ be a set of nodes with arcs entering node $i \in V_T$ and $Out(i) = \{j | (j, i) \in A_T^T\}$ be a set of nodes with arcs leaving node $i \in V_T$. The same holds for set $V_{(ih)}^T = \{j | (j, i) \in A_T^T \land h_{j,i} = h\}$ and $V_{(ih)} = \{j | (i, j) \in A_T \land h_{i,j} = h\}$, where the first set is a set of nodes with arcs entering
node \( i \in V^T \) of vehicle type \( h \) and the latter set contains nodes with arcs leaving node \( i \in V^T \) of vehicle type \( h \).

**Stationary arcs**

In order to model the possibility for vehicles to stay at the same place in time stationary arcs are introduced as already has been mentioned in Section 2.2. Let these stationary arcs be collected in set \( A^S = \bigcup_{1 \leq t \leq \tau} \bigcup_{1 \leq i \leq n+m} [(i, i_{t+1})] \). Stationary arcs are not constraint in the number of vehicles or containers flowing over them. They also allow containers flowing over them without the presence of vehicles and all vehicle types are allowed on the stationary arcs. Let \( \delta_{ij\theta} \) indicate whether arc \((i, j) \in A^T\) is a stationary arc for demand \( \theta \), where

\[
\delta_{ij\theta} = \begin{cases} 
0 & \text{if } \delta_{ij\theta} \text{ is a stationary arc,} \\
1 & \text{otherwise.} 
\end{cases}
\]

**Intersections and congestion**

When more than 2 terminals exist, which are connected to each other by road, an intersection is included to connect the roads. In the model the intersection is not only used as a crossroad, but also as the location where congestion will occur. In order to be able to do this an intersection between bi-directional roads will be modelled by two nodes, one input node and one output node. The terminals connected to the intersection will get a directed arc towards the intersection entering in the input node and a directed arc away from the intersection connected to the output node. A directed arc between the input and output node is used to transport all the containers and vehicles coming from all directions through the intersection. A representation of an intersection is given in Figure 3.1.

![Figure 3.1: Representation of an intersection](image)

To come up with realistic results from this model, congestion is modelled in order to measure its effect on the performance of vehicles in combination with various road and intersection profiles. Congestion is modelled in an intersection by creating so called fan arcs from the input node to the output node as explained in Kohler et al. [2002]. For every intersection three fan arcs have been used to model congestion, where each congestion arc \((i', j') \in A\) is assigned with \( \psi_{i'j'} \) tuples containing a time and capacity property \((\phi, \psi)\). \( \phi_{ij} \) is the maximum amount of vehicles that can transverse an arc in \( \psi_{ij} \) time steps. One arc has a travel time \( \psi \) of 0 time steps, but a low capacity \( \phi \) of only 40 vehicles. The second arc has a travel time \( \psi \) of 1 time step and a maximum vehicle capacity \( \phi \) of 80 vehicles and the final arc has a capacity \( \phi \) of 120 vehicles and a travel time \( \psi \) of 2 time steps. Each created fan arc over all time periods is collected in set \( A^F_{i'j'} = \bigcup_{1 \leq t \leq \tau} \bigcup_{1 \leq g_{i'j'}} [(i'_{tg_{i'j'}}, j'_{tg_{i'j'}})] \).

In order to make sure that only one of the three arcs is used, a new set of variables \( z \) is introduced. In this set, \( z_{ij} \in [0, 1] \) indicates whether fan arc \((i, j) \in A^F\) is allowed to have flow on it. The three different fan arcs with their assigned properties is represented in Figure 3.2.

**Long Term (LT) loading/unloading**
In order to incorporate barges and trains more realistically a LT node is introduced for each terminal, enabling the ability to create a parallel network with the correct properties for the transportation of containers. This is necessary, because unrealistic handling and transportation times would be used for LT vehicles. The arcs \((i^{LT}, j^{LT})\) who connect the terminal node with the LT node are called LT arcs and are represented in set \(A^{LT} = \bigcup_{1 \leq t \leq \tau} \bigcup_{1 \leq i \leq n} (i^{LT}, j^{LT})\) All arcs will have a vehicle property assigned, which makes flows over the LT arcs only possible for barges or trains and not for road vehicles. The fact that it runs parallel to the road network enables the possibility to remove the network for non barge or train instances, which reduces calculation time significantly. As can be seen in Figure 3.3 vehicles can move from node 1 to node 2 by either road or LT connection. The loading and unloading of LT vehicles is done independently from road vehicles by separate equipment. The equipment available at a terminal for LT vehicle handling operations is shared by all docked LT vehicles and for that reason an undirected arc is modelled for the loading and unloading operations of LT vehicles.

**Total layout**

The total port layout is build from the previous discussed arcs and nodes. All arcs will be assigned with properties for allowing certain vehicle types, handling times, driving times, capacity and cost. For the Maasvlakte area, Tierney et al. [2012] used the layout as presented in Figure 3.4. The notes on the arcs give an indication of the representation of the arc.

### 3.3 Demand generator

To force a flow through the model, a demand is generated which forces a flow of containers from one terminal to another terminal. Let \(o_\theta \in V, d_\theta \in V, a_\theta \in \mathbb{Z}^+, r_\theta \in \{1, ..., \tau\}, u_\theta \in \{1, ..., \tau\}\) be the origin node, destination node, number of containers, release period and delivery period. All this data is determined by the demand generator and is combined in a vector, which can be fed as input in the algorithm.

The origin and destination node are determined uniformly at random from the available terminal
nodes. In order to identify the terminal nodes from the intersection nodes and LT nodes, vector \( t\texttt{ster} = \{\text{terminal}_1, ..., \text{terminal}_n\} \) is asked as input before the model runs, where the terminal numbers correspond to the node represented by the terminal in matrix \( A_{eq} \). The number of containers, which have to flow between the two terminals, is also determined uniformly at random with a value between 1 and 50 containers.

The release period is determined uniformly at random in the range \([0, t^{max} - 1.5 \text{time}(a,b)]\), where \( t^{max} \) is the last time period of the model and \( \text{time}(a,b) \) is the shortest time required for the slowest vehicle to travel from the origin node to the destination node. \( \text{time}(a,b) \) is determined by an all pairs shortest path algorithm. The delivery time is determined uniformly at random in the range \([r_\theta + \text{time}(a,b), t^{max}]\).

As already mentioned in Section 2.4 combining all this information into a vector we introduce parameter \( b \) to be the demand vector with length \( n \) and

\[
b(i) = \begin{cases} 
  a_\theta & \text{if } b(i) \text{ is the origin node in release period } r_\theta, \\
  -a_\theta & \text{if } b(i) \text{ is the destination node in delivery period } u_\theta, \\
  0 & \text{otherwise.}
\end{cases}
\]

So i.e. if a demand is generated from node 1 to node 2 with \( r_\theta = 1, u_\theta = 2 \) and \( a_\theta = 25 \) in the time extended network of Figure 2.2,

\[
b = \begin{bmatrix} 25 \\ 0 \\ 0 \\ -25 \end{bmatrix}.
\]

**Multi-commodity flow**

To generate a realistic volume of containers transported by ITT on the Maasvlakte area, one generated demand is not sufficient. In order to do this the demand generation process is repeated \( \Theta \) times to create flows of realistic volume. The generated flow vectors \( b(\theta) \), where \( \theta \) is a single demand generation, are combined in one large demand \( b \) vector as already mentioned in Section 2.4.

For the determination of the number of commodities \( \theta \), parameter \( C \) is introduced representing the total amount of containers flowing through the model. Let \( a_{\text{tot}} \) be the sum of the amount of containers generated from the individual demands. Individual demands are generated until reaching this amount \( C \) minus the maximum possible amount of containers generated in a single demand. To get the total demand up to \( C \) a new demand is added with an amount of containers equal to \( C \) minus \( a_{\text{tot}} \).
3.4 Vehicles, vehicle types and containers

The model can use different types of vehicles $h \in H$, where $H$ is a set consisting of all vehicle types. Each vehicle type has its own cargo carrying capacity $h_{ij}$ per vehicle on arc $(i, j) \in A^\tau$, and is set to 1 container for AGVs and ALVs, up to 5 containers for MTSs and up to 50 containers for barges. The amount of vehicles of vehicle type $h$ that is present on a node is $s_{ih}$, with $i \in V^T$ and $h \in H$. Each arc $(i, j) \in A^T$ is also given a property of which vehicle type it carries by parameter $\eta_{ij}$. It is assumed that all terminal nodes start with a certain amount of vehicles $s_{ih}$ in the first time period. For all intersection and LT nodes $s_{ih}$ in the first time period is set to zero.

To implement vehicles in the model, so that the algorithm can make a distinction between different vehicle types and containers, is to create a new set of variables for each vehicle type $h \in H$, where $x_{ijh} \in [0, ..., s_{ih}]$ represents the amount of vehicles of vehicle type $h$ on arc $(i, j) \in A^\tau$. Also a set of containers corresponding to vehicle type $h$ is required, represented by variable $y_{ij\theta h}$, where $y_{ij\theta h} \in [0, ..., a]$ is the amount of containers on arc $(i, j) \in A^T$ for demand $\theta$ and of vehicle type $h$.

At terminals containers are loaded and unloaded on vehicles by Automated Stacking Cranes for AGVs and by Reach Stackers or Straddle Carriers for MTSs. Because of the automated lifting system of an ALV, ALVs do not require any equipment for getting loaded or unloaded and are therefore not restricted by a handling rate. For every vehicle type the handling rate $m_{ih}$ at terminal node $i \in V^T$ is based on the available handling equipment and the handling process. The handling rate of AGVs is set to 30 moves per hour per crane and the handling rate for MTSs is set to 35 moves per hour per crane. The rate for MTS is higher, because efficiency gain is reached with handling multiple containers in a set of trailers. Next to road vehicles barges whom are being unloaded by special barge cranes having a rate of 30 moves per hour per crane. Loading and unloading moves only apply for non-stationary arcs entering and leaving time space terminal node $i$ of vehicle type $h$ and therefore set $V^T_{ih} = \{\text{arc}(i, j) \in A^T \setminus A^S \land \eta_{ij} = h\}$ is a set of terminal nodes connected by non-stationary arcs of vehicle type $h$.

All vehicles types have been assigned with a driving speed $v_h$ in order to calculate the transverse period $t_{ijh} \in t$ for each arc depending on the different properties for each vehicle type, where $t$ is a set containing the transverse periods for all arcs. The speeds set for an AGV, ALV, MTS and barge are 5.0, 4.0, 6.6 and 2.2 m/s respectively.

3.5 Parameters

For a clear overview on all used parameters in the objective function and constraints, the parameters are summarized in this section and are shown in Table ??.

3.6 Objective function

The objective of the model is to minimize the lateness of container delivery at the requested terminal. To model this in an objective function a penalty is assigned to each stationary arc entering the destination node after the assigned delivery time of the demand, collected in set $In^S(ih) = \{j|(j, i) \in A^S\}$. The height of the penalty is depending on both the priority of the cargo and the total amount of time periods the cargo is late.

A priority $\rho_{\theta}$ is given to each individual commodity. Three priority levels will be distinguished: low, medium and high. The priority can be determined by a triangular distribution, where around 60% of the demands will have a low priority, around 30% will have a medium priority and around the remaining 10% will have a high priority. The penalty for low priority containers will only be affected by the amount of time periods the cargo is late, where for medium priority cargo the penalty is multiplied by 2 and for high priority by 3.

Combining these parameters the objective will be:
\( V^T \) | Set of time space nodes. \\
\( A^T \) | Set of time space arcs. \\
\( Aeq^T \) | Matrix representing the time space network of arcs and nodes. \\
b | Demand vector. \\
\( A_{Fe}^T \) | Set of fan arcs. \\
\( A_{St}^T \) | Set of stationary arcs. \\
\( A^{LT} \) | Set of LT arcs. \\
z_{ij} | Indicate whether fan arc \((i, j)\) is allowed to have flow on. \\
\( \delta_{ij} \) | Indicate whether arc \((i, j)\) is a stationary arc. \\
\( \phi_{ij} \) | Capacity of fan arc \((i, j)\). \\
\( \Theta \) | Number of demands. \\
o_\theta | Origin node \(\in V^T\) of demand \(\theta\). \\
d_\theta | Destination node \(\in V^T\) of demand \(\theta\). \\
a_\theta | Amount of containers in demand \(\theta\). \\
r_\theta | Release time period of demand \(\theta\). \\
u_\theta | Delivery time period of demand \(\theta\). \\
t_{ster} | Set of terminal nodes \(i \in V\). \\
H | Set containing all vehicle types. \\
h | Vehicle type. \\
\( \eta_{ij} \) | Vehicle type allowed on arc\((i, j)\). \\
\( h_{ij} \) | Carrying capacity per vehicle on arc\((i, j)\). \\
s_{ih} | Amount of vehicles of type \(h\) present at node \(i \in V^T\). \\
v_h | Driving speed of vehicle type \(h\). \\
t | Set containing the transverse times of all arcs. \\
t_{ijh} | Transverse time of arc \((i, j)\) for vehicle type \(h\). \\
n_{ih} | Handling capacity at terminal \(i \in V^T\) for vehicle type \(h\). \\
x_{ijh} | The amount of vehicles on arc\((i, j)\) of vehicle type \(h\). \\
y_{ij\theta} | The amount of containers on arc\((i, j)\) for demand \(\theta\). \\
\rho_\theta | Priority of demand \(\theta\). \\
In(ih) | A set of nodes with arcs entering node \(i \in V^T\). \\
Out(ih) | A set of nodes with arcs leaving node \(i \in V^T\). \\
In^S(ih) | A set of nodes with stationary arcs entering node \(i \in V^T\). \\
V_{ih}^T | A set of nodes connected by non-stationary arcs of vehicle type \(h\). \\
V_{(ih)}^T | A set of nodes with arcs entering node \(i \in V^T\) of vehicle type \(h\). \\
V_{(ih)}^T | A set of nodes with arcs leaving node \(i \in V^T\) of vehicle type \(h\). \\

\[
\min \quad \sum_{1 \leq \theta \leq \Theta} \left( \sum_{u \leq \tau} \sum_{i \in In^S(ih)} \rho_\theta (t - u_\theta) y_{ij} \right) \\
\tag{3.1}
\]

### 3.7 Constraints

This section presents the constraints incorporated in the model. Based on these constraints and the objective function presented in Section 3.6 the model can make a calculation of the minimal amount of delay in the network given the number of vehicles of vehicle type \(h\) and a number of demands. The following constraints are used:
\[ Aeq^\tau y_{ij\theta} = b \quad \forall (i,j) \in A^T, \quad 1 \leq \theta \leq \Theta \quad (3.2) \]

\[ \sum_{j \in Out(i,h)} x_{ijh} = s_{ih} \quad \forall \quad i \in V, h \in H \quad (3.3) \]

\[ \sum_{j \in Out(i,h)} x_{ijh} = 0 \quad \forall \quad i \in V \text{ tster}, h \in H \quad (3.4) \]

\[ \sum_{j \in Out(i,h)} x_{ijh} - \sum_{k \in In(i,h)} x_{kijh} \leq s_{ih} \quad \forall \quad i \in V^T, h \in H \quad (3.5) \]

\[ \sum_{j \in Out(i,h)} x_{ijh} \leq h_{ih} \quad \forall \quad i \in V^T, h \in H \quad (3.6) \]

\[ \sum_{1 \leq \theta \leq \Theta} \left( \sum_{j \in V_{ih}^T} y_{ij\theta} + \sum_{j \in V_{ih}^T} y_{ij\theta} \right) \leq m_{ih} \quad \forall \quad i \in V_{ih}^T, h \in H \quad (3.7) \]

\[ \sum_{1 \leq \theta \leq \Theta} \sum_{A_{ijh}^{LT}} y_{ijh\theta} \leq m_{ih} \quad \forall \quad (i,j) \in A^{LT}, 1 \leq t \leq \tau, 1 \leq i \leq n, h \in H^{LT} \quad (3.8) \]

\[ x_{ij} \leq \phi_{ij} z_{ij} \quad \forall \quad (i,j) \in A^F \quad (3.9) \]

\[ \sum_{(j,k) \in A_{ijf}^{LT}} z_{ij} \leq 1 \quad \forall \quad (i',j') \in A, 1 \leq t \leq \tau \quad (3.10) \]

- Constraint 3.12 restricts the model in such a way that demand \( \theta \) will flow through the network via the arcs and nodes implemented in matrix \( Aeq^\tau \). Vector \( b \) constrains the origin and demand node in having an outflow respectively inflow of the number of containers as indicated in the vector.

- Constraint 3.15 constrains the sum of vehicles leaving a node, minus the sum of vehicles entering a node to be less or equal than the number of vehicles currently available at a node. For the first time period, because no vehicles are entering the node, constraints 3.13 and 3.14 can be seen as initializing the number of available vehicles at the terminal nodes. The number of available vehicles is set to 0 for intersection nodes.

- Constraint 3.16 connects the containers to the vehicles, preventing containers from flowing over an arc connecting the different nodes without a vehicle. The containers are however unconstrained while flowing over the stationary arcs indicated by parameter \( \delta_{ij\theta} \).

- Constraints 3.17 and 3.18 constrain the amount of handling moves performed per time period and per vehicle type over all demands at the terminal nodes. Constraint 3.17 does this for the road vehicles and constraint 3.18 restricts the amount of loading and unloading moves for LT vehicles.

- Constraints 3.19 and 3.20 bound the flow over fan arcs in such a way that only one fan arc can be used per time period. It also ensures that all vehicles encounter the same amount of congestion, while flowing over arcs, also considering the congestion of previous time periods at the same intersection. As an example when flow x1 encounters a congestion of 1 time period in period \( t \) at intersection node \( i \), flow x2 entering intersection \( i \) at \( t+1 \) will also encounter a congestion of 1 time period.

### 3.8 Total model

When the objective and the constraints are combined the total model is obtained.

\[ \min \sum_{1 \leq \theta \leq \Theta} \sum_{u \leq t \leq \tau} \sum_{i \in In^\theta(ih)} \rho_{\theta}(t-u_{\theta}) y_{i} \quad (3.11) \]
subject to

\[ \text{Aeq} \sum_{j \in \text{Out}(i,h)} x_{ijh} = b \quad \forall (i,j) \in A^T, \quad 1 \leq \theta \leq \Theta \quad (3.12) \]
\[ \sum_{j \in \text{Out}(i,h)} x_{ijh} = s_{ih} \quad \forall i \in V, h \in H \quad (3.13) \]
\[ \sum_{j \in \text{Out}(i,h)} x_{ijh} = 0 \quad \forall i \in V \text{ tster}, h \in H \quad (3.14) \]
\[ \sum_{j \in \text{Out}(i,h)} - \sum_{k \in \text{In}(i,h)} \leq s_{ih} \quad \forall i \in V^T, V, h \in H \quad (3.15) \]
\[ \sum_{\theta \leq \theta} \left( \sum_{j \in V_{ih}} y_{ij}\theta + \sum_{j \in V_{ih}} y_{ij}\theta \right) \leq m_{ih} \quad \forall i \in V_{ih}, h \in H \quad (3.16) \]
\[ \sum_{\theta \leq \theta} \sum_{A^T_{ih}} y_{ij}\theta \leq m_{ih} \quad \forall (i,j) \in A^T, 1 \leq t \leq \tau, 1 \leq i \leq n, h \in H^T \quad (3.17) \]
\[ x_{ij} \leq \phi_{ij} \leq \tau \quad \forall (i,j) \in A^F \quad (3.19) \]
\[ \sum_{(j,k) \in A^F, \tau} z_{ij} \leq 1 \quad \forall (i', j') \in A, 1 \leq t \leq \tau \quad (3.20) \]

The objective function and constraints are programmed by Matlab and will be solved by the CPLEX for Matlab integration, which uses the CPLEX algorithm to solve the integer problem but can use Matlab double matrices as an input for the algorithm. The large amount of constraints and variables tend to create memory issues. Fortunately the relative low amount of non-zero elements make it possible to use sparse matrices to prevent any memory issues. For large problem sizes the model therefore uses sparse matrices. To decrease the calculation time of the model variables representing flows before the start time of the demand are set to zero. Also the variables representing the flow of containers over arcs entering the origin node are set to zero. The latter adjustment also eases the termination of the algorithm for infeasible instances according to Tierney et al. [2012].

### 3.9 Discussion

This section discusses the way the model is build as presented in Tierney et al. [2012].

**Intersections and congestion**

Modelling an intersection in this way has some drawbacks. Intersections can only take congestion into account by checking the absolute number of vehicles entering the intersection. The direction of vehicles is not taken into account, which results in cases i.e. when 41 vehicles coming from the same node entering an intersection will incur 1 time step of congestion, while they normally would not have to slow down under the condition that no other vehicles from other directions are entering the intersection. Another case i.e. is that two approaching vehicle flows incur congestion at an intersection, while they are normally able to pass each other on a bi-directional road. It is also not possible to incorporate details such as priority given to certain streets or longer wait times for left turns, but the main idea of Tierney et al. [2012] by modelling intersections in this way was that it is able to slow down the overall flow when many vehicles enter an intersection.

A second point of criticism is that the incurred congestion, in time steps, and the capacity of each fan arc taken by Tierney et al. [2012] are not supported by arguments and therefore it is advised to provide some literature references to increase the reliability of the incorporation of congestion in this model. On the other hand the question can be raised to what extent congestion will occur.
and if it significantly increases the occurring delays? The maximum number of vehicles modelled by Tierney et al. [2012] in the whole Maasvlake area is only 125 vehicles, handling up to 1500 container movements in 6 hours. In the writer's opinion, congestion will probably only increase the travelling times slightly and mostly only for low priority cargo or cargo which still has some slack time before being too late. This occurs because the optimization algorithm will minimize the amount of cargo delivered too late and therefore gives priority on intersections to vehicles carrying or heading for cargo which is in danger of late delivery.

**LT loading/unloading**

The introduction of barges or trains in the way it is proposed by Tierney et al. [2012] is only realistic, when the LT vehicles will be the only vehicles handled by the handling equipment. However in most cases the handling equipment is also used for import/export barges and trains who are transporting containers into or from the hinterland and therefore queues are formed made up by these vehicles as well. Currently waiting times are already very long due to the fact that quays are also shared with ocean going vessels who have priority over barges for contractual reasons. With increasing container volumes in the future, this can only add up to the problem. On the other hand the current process of handling barges provides an opportunity for ITT. In port it is normal for barges to call multiple ports (sometimes up to 10) before it heads into the hinterland [Konings et al., 2010]. When calling multiple of these ports the barges could be used for ITT as well.

**The generation of demands**

A main assumption of the demand generator is that it does not make a distinction between different types of cargo such as various sizes of containers, dangerous goods or refrigerated containers which might require other handling procedures. However most containers will have a standard size and therefore these containers are neglected in the model. Also the call sizes between 1 and 50 containers have to be investigated whether this is a realistic value, because this is not supported by references in Tierney et al. [2012]. Also the distribution for the determination of the origin and destination nodes should be investigated whether or not it is a realistic distribution or that a distribution based on the expected weighted annual ITT volume per terminal should be applied. Especially the long distance of ECT Euromax and empty depot Maasvlakte 1 from the other terminals could have an reasonable influence on the amount of required vehicles to guarantee an acceptable delay. If the generated flow by these terminals in reality is not proportional to the flows generated by the other terminals the amount of simulated vehicle kilometres will be to high, which would indicate that more vehicles are necessary than in reality. A final comment on the demand generation is that for the determination of the time window of a commodity only the shortest path is considered and that the amount of available vehicles is neglected, while when the amount of generated containers is larger than the amount of available vehicles the additional vehicles will have to come from other terminals taken some extra time to arrive. This could create some instances in which one knows on forehand that the given time window is just not sufficient for all vehicles to travel to the assigned origin terminal and deliver their containers in time.

In the current model the handling rate over an arc is fixed to a certain amount per time step, but when a number of containers less than this amount is handled it still takes one time period to load/unload. This is unfortunately a downside of the modelling method and can only be improved by using smaller time periods to reduce the slack time if the total capacity of that arc is not fully used.

**Properties of vehicles, vehicle types and vehicle handling**

All vehicle properties presented in Section 3.4 are based on the properties used in Ottjes et al. [1996]. However in the writers opinion the vehicle speeds for road vehicles that are assumed by Ottjes et al. [1996] and adopted by Tierney et al. [2012] are too low. These speeds are reasonable for intra-terminal transportation, where relatively short distances are driven and a lot of vehicles interact in a confined area. This automatically limits the average speed, because of both safety issues and accelerations. However for the use in ITT where long distances are driven and vehicles only interact at intersection and handling stations of terminals the limitation for the average speed compared with the case of intra terminal transportation could be a lot lower. Also AGV and ALV technology have matured in the years since 1996 and therefore
technological implications may have been solved to increase the maximum speeds of AGVs, and ALVs. To make a reliable assumption of vehicle travelling speeds the current technological state of the vehicles and what possibilities there are when applying these technology in ITT should be investigated.

**Objective function**

The objective function only takes the delay of containers into account, but a cost optimization might be more wanted for the port authority to use in their decision making process. It could also eliminate unnecessary behaviour such as for example vehicles flowing freely through the model without being penalised for using fuel and increasing maintenance costs.

Minimizing the delay in container delivery by optimizing transport routes will give a result giving the amount of equipment required to achieve an acceptable level of delay. This amount of equipment should be considered as a minimum amount, because when in the real implementation optimization of transport routes is not applied it can be expected that more equipment is required than the simulation had shown. Optimizing also means that knowledge about future demands is required to be able to get a realistic outcome for the optimal transport routes for vehicles. When information about future demands is not available for an ITT system the optimization model as presented in this research can not on itself be used for the determination of ITT equipment specifics.

### 3.10 Concluding remarks

Following from the discussion it can be concluded that there are still some questions to be answered and also some improvements can be made. The model presented by Tierney et al. [2012] is still in an early development stage meaning that a lot of iterations and improvements will be applied before a final version is available to be used by port authorities.

It however remains important to realize that using an optimization for the determination of equipment and infrastructure requirement is only realistic when the transport routes of the implemented ITT system are also optimized and the required information for this optimization is available. Otherwise simulation models without optimization are advised to be used. On the other hand when implementing the optimization of transport routes in ITT a mixed integer model as presented in this research would be a good choice to use.

Besides the negative sides and problems which are discussed in the previous section, the mixed integer model is useful in finding whether a network with a given number of vehicles and handling equipment is able to handle all the demands in certain scenarios and will give an indication about equipment and vehicle requirements to achieve the pre defined performance indicators.
Chapter 4

Verification of the model

In this chapter the Matlab integer inter terminal transport model will be verified, by analysing
the results of the Matlab model with the expectations of the writer and this analysis is also
discussed. The question to what extent the model will simulate reality is not answered in this
verification.

The results are generated using the network and input parameters as given in Section 4.1, followed
by the results of this network. Finally a conclusion can be drawn whether both models are
generating the same results.

4.1 Input parameters

To be able to verify the working principles of the Matlab model, a simple network containing
2 terminals is used. An intersection is not required in inter terminal situations of only two
terminals, therefore intersections are excluded in this verification. The model is run for all road
vehicle types and in the case of ALVs, the model is run either with and without barges as well
as for 1 and 2 commodities. ALVs are chosen to model in combination with barges and multiple
commodities, because the loading constraints for ALVs are unnecessary and therefore the runtime
of the model can be limited by using less time periods. This will contribute in minimizing the
amount of time needed for verification. The network with LT nodes is given in Figure 4.1.

![Network Layout with Barges](image)

Figure 4.1: Graphical network layout with barges

The model will be run for various ITT vehicle configurations which create 6 different scenarios
as follows:

1. AGVs,
2. MTSs,
3. ALVs,
4. ALVs and barges,
5. ALVs with 2 commodities,
6. ALVs and barges with 2 commodities.

The first three scenarios have the same network matrix $A_{eq}$ and $A_{eq}^\tau$ and for scenario four these matrices will be extended with variables and arcs for barges. The last two scenarios have two demands to be served, resulting in an extra set of variables to do so. The transverse time $t$ for all arcs is set to 1. Matrix $A_{eq}^\tau$ for scenario 1 to 3 is presented in Figure 4.2a having a time expansion over 2 time periods, because it takes one time step to travel from terminal 1 to terminal 2 or vice versa and if the amount of vehicles is less than the number of containers it can be checked if the vehicle constraints work properly because it now first has to drive some equipment to the origin terminal before it can transport containers. For the first three scenarios the number of containers to be transported will be 50 with 25 ALVs per terminal available for transport. In case of the MTSs and AGVs the handling rate of containers is also included and to be able to complete the demand in time the rate is set to 25 containers per time step. For MTSs only 6 vehicles per terminal will be available, because every MTS can carry up to 5 containers. The demand vector $b$ is shown in Figure 4.2b.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>time</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22</td>
<td>-50</td>
</tr>
</tbody>
</table>

(a) $A_{eq}^\tau$ for a 2 terminal network over 2 time period. (b) Demand vector $b$.

Figure 4.2: Matrix $A_{eq}^\tau$ and vector $b$ to construct the first set of constraints $A_{eq}^\tau \cdot x = b$.

The time space network including barges for scenario 4 is presented in Figure 4.3 having 3 time periods, which is the minimum time it takes for one barge to sail to another terminal. The handling rate for barges is set to 48 containers per time step, the number of barges is set to 1 and the number of ALVs per terminal is set to 2. The loading rate now requires the usage of the 2 vehicles to be able to complete the total demand in time.

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 4.3: $A_{eq}^\tau$ including barges.

The matrices for scenario 5 and 6 are extended for two commodities by copying matrix $A_{eq}^\tau$ one time by multiplying the matrix with the identity matrix of size 2, giving the result for scenario 5 as shown in Figure 4.4. The matrix for scenario 6 is created in the the same way, but because of the size it is shown in Appendix F.
4.2 Model output

This section presents the outputs of the various scenarios and discusses the validity of the results. The figures representing the results not only show the values for every variable, but also matrix $A$ is shown, which show which arc the variable represents. This will provide a clear view on the validity of the result. A note should be made that in most cases more than one possible optimum can be found, but the algorithm finds only one. Therefore only the found optimum is discussed whether it is a possible optimum according to the expectations of the researcher. There are a few points which are checked:

- Is the demand satisfied with the correct origin and destination?
- Is every transported container properly connected to a vehicle?
- When containers are transported from an origin, are there enough vehicles available to be able to start the transportation?
- Is the objective value corresponding to the expectations?

**Scenario 1: AGVs**

The model outcome of the first scenario is shown in Appendix A. In Figure 4.5a the flows through the network are shown according to the results. The time steps are given by $t_0$, $t_1$, and $t_2$, the vehicle flows are represented by symbol $V$ and container flows are represented by symbol $C$. The number before the symbol represents the amount of flow over the arc.

The flow over the arcs is according to the expectations by giving a solution which requires two time steps to fully complete the demand. It can be seen that 25 vehicles are moving from terminal 2 to terminal 1 before the last 25 containers are transported which also explains the flow of 25 containers over the stationary arc of terminal 1 meaning that the containers are waiting before being picked up. The vehicle flow over the arc connecting terminal 2 with terminal 1 from $t_1$ to $t_2$ is valid because vehicle movement is not penalized and once those vehicles have completed their demand before the run time of the model is expired they can move freely through the network.

The objective value is zero as expected, because the demand is completed in time.

**Scenario 2: MTSs**

The results of scenario 2 are shown in Figure 4.5b. The results are as expected, with a note to the values for vehicles where all 12 available MTS vehicles are flowing, where only 10 MTSs are required to transport all the containers of the demand. The reason for this is the same as in scenario 1, because the vehicle movements are not penalized they can move freely through the model as long as the demand is satisfied. The model output is shown in Appendix B.

The objective value is zero which is according to the expectations.

**Scenario 3: ALVs**

As can be seen in the results shown in Figure 4.6a they correspond to the results that can be expected. In this solution the vehicles wait until enough vehicles are present to transport all containers at once. This is possible, because the absence of a handling rate for ALVs makes it
possible that the flow over the arcs between different terminals is unconstrained. The real model output is shown in Appendix C, where also the matrix network representation is shown.

The objective value is zero as expected.

**Scenario 4: ALVs with barges**

The demand vector $b$ for this scenario has its delivery time set to period 2, which should give a late arrival of one time period and therefore the objective value should be higher than zero, because a barge requires a minimum of 3 time steps to reach the other terminal. Also the number of road vehicles is not sufficient to transport the containers over the road.

Because of the added nodes for barges a graphical representation, such as presented for the previous scenarios, is not giving a clear representation. Therefore only the model output is presented in Appendix D. It shows a negative value of -46 for variable 40, which represents the stationary arc from node two connecting period two to three. This negative value represents the number of containers which are delivered late and is also the value of the objective function. The value of variable 40 is only -46, because the two vehicles of terminal 1 and 2 can pick up and deliver 4 containers before the delivery time is expired. The other vehicle movements can also be expected, which makes this a feasible solution.

**Scenario 5: ALVs with 2 commodities**

The demand vector $b$ is set such that the first commodity will transport 20 containers from terminal 2 to terminal 1 with delivery period 2 and the second commodity will transport 30 containers from terminal 2 to terminal 1 as well, but with delivery period 1. The results of this simulation shown in Figure 4.6b are according to the expectations. The objective value is 5, because the 30 containers from commodity 2 can not be delivered in 1 time period, due to the number of starting vehicles at each terminal is set to only 25 ALVs. This will result in a late delivery of 5 containers, shown by the value of -5C2 in Figure 4.6b, because the vehicles of terminal 1 have a travel time of one time period to reach terminal 2. The negative value basically means that a flow travels against the direction of the arc and can be explained as travelling back in time to reach its destination in time. This time travel is not allowed in real transport situations and means that a delay has occurred. The model output is shown in Appendix E.

**Scenario 6: ALVs and barges with 2 commodities**

For the final scenario demand vector $b$ is given a flow of 9 containers for the first commodity flowing from terminal 2 to terminal 1 with delivery period 3. The second commodity has a flow demand of 42 containers also flowing from terminal 1 to terminal 2 with delivery period 3. The number of ALVs and barges is set to 10 and 1 per terminal respectively.

![Diagram](image_url)
Because of the matrix size, the results are shown in Appendix F. All flows are corresponding to values which can be expected and therefore it is a feasible solution. The objective value is zero which is the only possible solution, because both delivery times are in the last time period.

### 4.3 Concluding remarks

From the results shown in Section 4.2 it can be concluded that the Matlab model with CPLEX integration gives feasible solutions for the six presented scenarios. These results show that the current model is able to simulate small size networks and has the potential to be extended to large scale port models with more functionalities. The model is already able to run situations with different vehicle types and multiple commodities and although not verified the model is also able to run situations with more than two nodes and the additional required arcs.

It can be seen that even for small networks with only 2 nodes and a maximum of three time periods, the amount of variables and constraints are increasing rapidly when adding additional vehicle types or commodities. Therefore it is advised for future large scale models to optimize the layout of arcs and nodes to reduce the size of the model. Also additional pre solving is performed as explained in Tierney et al. [2012] by setting variables to zero representing arcs operating before the start time of a commodity and for variables representing container flows trying to re-enter the origin node of a commodity to ease the optimization in finding a solution.

To reduce the memory requirement the model presented in this research also makes use of storing the constraint matrix in a sparse matrix, which only stores the non-zero entries of the matrix and their value.

Unfortunately it is not possible to validate the current model with the model presented by Tierney et al. [2012], because of the lack of sufficient input data such as arc transverse times and the amount of available handling equipment at a terminal. This data will hopefully be available for future research.
Chapter 5

Conclusion and future research

In this report an investigation is presented about the working principles, the benefits and the drawbacks of the integer programming model presented in Tierney et al. [2012]. Also the assumptions made in the paper of Tierney et al. are discussed. By investigating the working principles and properties of the model an answer can be given to the main research question: To what extent will the model presented by Tierney et al. give realistic and reliable results for the application of mixed integer optimization models in the development process of inter terminal transportation systems? Also some improvements are proposed in order to make the model more realistic and reliable.

The mathematical principles of a mixed integer optimization problem makes that the model gives results for an optimal situation. This means that when an ITT system is integrated in a port environment where transport routes of vehicles are not optimized, the amount of vehicles required to achieve the desired performance can differ because non optimal situation will occur. In the researchers opinion it is important to keep this in mind when using integer optimization models in the decision making process and development of ITT systems.

The real drawback of integer programming is that the discrete nature of the model will use time periods of 5 minutes in the case of this model. This makes that operations taking up a part of a time period will either be neglected or rounded off to the nearest time period, reducing the overall realism of the model. Choosing smaller time periods is possible, but will rapidly increase the model size resulting in longer calculation times and possible memory problems. Also implementing time distributions for vehicle or container specific operations, such as loading and unloading is not possible, because arc transverse times are known prior to running the simulation which provides the possibility for the optimization algorithm to choose when to use the loading or unloading arc in its own favour resulting in a change of the loading/unloading distribution.

Despite these drawbacks the model is easily adaptable for various port situations by changing the number of nodes and arcs giving the possibility to get fast results to use in ITT development processes. Also the possibility to optimize the problem can result in better networks reducing the overall implementation and operation costs for ITT systems.

Besides the benefits and drawbacks of simulating ITT with an integer model, the assumptions made in Tierney et al. [2012] for the port of Rotterdam are also discussed. The conclusions that can be drawn from these assumptions in relation with the main research question are as follows:

- Congestion is modelled in a way that the connected roads at an intersection basically act as a funnel, where all incoming traffic is slowed down. This way of modelling is not taking the direction of the vehicles into account but only the volume of traffic, resulting in congestion for vehicles who normally would not suffer from congestion. Also the congestion that occurs is based on fixed time periods with a maximum of 2 time periods or 10 minutes. In the writers opinion it can be questioned whether congestion will occur at all, because a maximum of 125 vehicles are present in the ITT system which covers a total length of almost 30 km. If also the loading and unloading is considered where vehicles remain in
one place for a certain amount of time the question can be raised how often more than
40 vehicles will coincide at one intersection. If this however happens the optimization
algorithm will make sure that only the low priority cargo will suffer from congestion. For
future research it is suggested to investigate the differences in results for instances with and
without congestion. If it turns out that the differences are negligible the fan arc principle
can be excluded from the model which reduces the amount of variables and constraints.

- Including barges into the model is only realistic when priority will be given for ITT barge
handling at the different terminals, because the congestion for barges in the port of Rot-
tterdam nowadays can add up to one day making it impossible to implement a reliable ITT
barge service in the model otherwise.

- The demand generator in the model creates demands with a random determined origin
and destination. However in the report of Diekman and Koeman [2010] it becomes clear
that each terminal creates a certain percentage of the total ITT demand per time period.
Combined with the geographical layout of the Maasvlakte area this creates areas in the
network which are more densely populated with vehicles than other areas. A proper im-
plementation of the demand generator combined with the geographical network layout is
required to be able to investigate the to what extent congestion will occur in the ITT
system at the Maasvlakte.

- The vehicle properties used in Tierney et al. [2012] are all based on outdated information
from the report of Ottjes et al. [1996]. To get a reliable result of the model these properties
need to be update to current industry standards and technological possibilities for these
vehicle types.

With the above mentioned arguments an answer is given to what extent mixed integer pro-
gramming on itself is able to provide reliable and realistic solutions for the development of ITT
systems. The current model presented by Tierney et al. [2012] still needs some improvements to
provide more realistic results and by changing the objective function into a function optimizing
costs instead of delays more information can be included in the model to determine an optimal
ITT system.

**Future research**

In future research the model should be validated with a real situation or another already validated
model in order to be able to prove the reliability of integer programming models for ITT systems.
It would also be advised to investigate how optimization of transport routes for ITT vehicles can
be implemented in a real ITT system. Once the model is validated to be reliable and realistic and
optimization is implemented in the ITT vehicle planning as well, mixed integer programming
models can provide a useful tool for port planners and authorities to use and optimize port
development and decision making processes. It is also recommended to investigate how ITT
vehicle planning optimization should be implemented in ports to be able to create a meaningful
minimum cost flow model.

A summary for the updates which have to improve the reliability and realism of the model are
as follows:

- The way congestion is modelled by the integer programming model have to be reconsidered
  and properly founded;

- Investigate whether it is realistic to use barges for ITT;

- Provide the model with a realistic demand generator;

- Investigate and update the vehicle properties used in the model.
Bibliography


S. Voss, K. Tierney, and R. Stahlbock. Inter-terminal transportation (itt) - challenges and instruments, 12 2012. Presentation slides from meeting with the Port of Rotterdam authority 04-12-2012.

Appendix A

Results of scenario 1

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure A.1: Result of scenario 1.
Appendix B

Results of scenario 2

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Figure B.1: Result of scenario 2.
Appendix C

Results of scenario 3

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D

Results of scenario 4
Figure D.1: Result of scenario 4.
Appendix E

Results of scenario 5

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure E.1: Result of scenario 5.
Appendix F

Results of scenario 6
Figure F.1: Result of scenario 6.