

Optimising the sourcing strategy of a grain importing company, given uncertain backhaul availability

T.C. Seppen



Optimising the sourcing strategy of a grain importing company, given uncertain backhaul availability

by

T.C. Seppen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday July 5, 2019 at 10:30 AM.

Student number: 4209680
Project duration: September 17, 2018 – July 5, 2019
Thesis committee: Dr. ir. J. T. van Essen, TU Delft, supervisor
Dr. ir. K. I. Aardal, TU Delft
Dr. ir. M. B. van Gijzen, TU Delft
■. ■■■■■ ■■■■■ ■■■■■ ■■■■■ ■■■■■

This thesis is confidential and cannot be made public until July 5, 2021.

Preface

[illegible]

Furthermore, I want to thank my supervisor at the TU Delft, Theresia van Essen. Without your guidance, I would not have been able to complete this thesis. Discussing my work with you the last nine months was immensely insightful, just as your feedback on both my writing and research.

Finally, I want to thank my family, friends and girlfriend for all the support, comfort and fun during this last year. All of you enabled me to relax and have fun, especially during the more stressful periods of the last year.

T.C. Seppen
Delft, July 2019

Contents

1	Introduction	1
1.1	Research motivation	1
1.2	Problem description	2
1.3	Structure	3
2	Literature	5
2.1	Vehicle routing problem	5
2.2	Transportation problem.	6
2.3	Stochastic programming	7
2.4	Conclusion	8
3	Problem Description	9
3.1	Challenges	9
3.2	Sets, parameters and variables	10
3.3	Penalty functions	12
3.4	Objective function	15
3.5	Constraints	15
4	Exact Solution Method	19
4.1	Linearisation of supplier results.	19
4.2	Linearisation of penalty functions	20
4.3	Reducing the number of variables and constraints	21
4.3.1	Variables	22
4.3.2	Constraints	22
5	Local Search Method	25
5.1	Simulated annealing	25
5.2	Initial solution	26
5.2.1	Mixed integer linear program initial solution	26
5.2.2	Mixed integer linear program contracts	27
5.2.3	Objective value.	29
5.3	Neighbours	29
5.3.1	Adding or deleting a truck	30
5.3.2	Changing the product of the truck	30
5.3.3	Moving the truck forward or backward in time.	31
5.3.4	Changing supplier location of the truck	31
5.3.5	Changing customer location of the truck.	32
5.3.6	Neighbour selection	33
5.3.7	Processing algorithm.	34
5.4	Changing the step size of the changes.	35
6	Problem Data	37
6.1	Data description	37
6.2	Locations and routes	38
6.3	Transportation costs and capacity	40
6.4	Contracts and lateness	42
6.5	Products and prices.	43
6.6	Defining the test instance.	44
6.7	Results of reducing the number of variables and constraints	45

7	Truck Availability	47
7.1	State space models	48
7.1.1	Exponential smoothing	48
7.1.2	State space methods	49
7.2	Results of the state space models	50
8	Results	53
8.1	Exact solution method	53
8.1.1	Determining the penalty function values	53
8.1.2	Result analysis	55
8.1.3	Simulating the time horizon	55
8.1.4	Run time	56
8.1.5	Limited run time	56
8.2	Simulated annealing	57
8.2.1	Initial solution	58
8.2.2	Start temperature	59
8.2.3	Cooling schedule.	61
8.2.4	Step size	63
8.2.5	Final selection of parameters	64
8.3	Comparison between methods	64
9	Conclusions	65
9.1	Conclusions.	65
9.2	Future Research.	66
9.3	Recommendations	67
	Appendices	69
A	Appendix A: Algorithms	71
	Bibliography	73

Glossary

backhaul	In trucking, a backhaul is the truckload going back from destination point B to the departure point A.
contract	A contract is an agreement to pick up a stipulated amount of goods in a specified time period. A contract is said to be live if the current time period is in the specified time period. A contract is said to be depleted if the total amount of goods specified in the contract is picked up. A contract is said to be unfulfilled if less than the stipulated amount of goods is picked up in the specified period.
FOT	Free On Truck, terms of delivery of a contract, meaning that the customer is responsible for the transport of the goods from the storage of the supplier.
franco	Terms of delivery of a contract, meaning that the transport of the goods to the customer is included in the price and that the supplier is responsible for the transport.
postal zone	An area for which first two digits of a postal code are the same within a country.

1.2. Problem description

A schematic representation of the problem is given in Figure 1.1. [REDACTED] has a number of customers with a demand for different products. This customer demand is represented by a farm and a product with a corresponding contract next to it. Furthermore, each customer pays a price for the product. The demand and price fluctuate over a time horizon, which is represented by the clock in the figure. Furthermore, the question mark above the truck in the middle represents the backhaul availability. The suppliers of [REDACTED] are represented by the factories on the left in the figure. These suppliers have different products (with different prices) available, represented by the products next to the factories in the figure. Based on these aspects, contracts at the supplier (i.e., the variables) need to be purchased such that the profit is maximised.

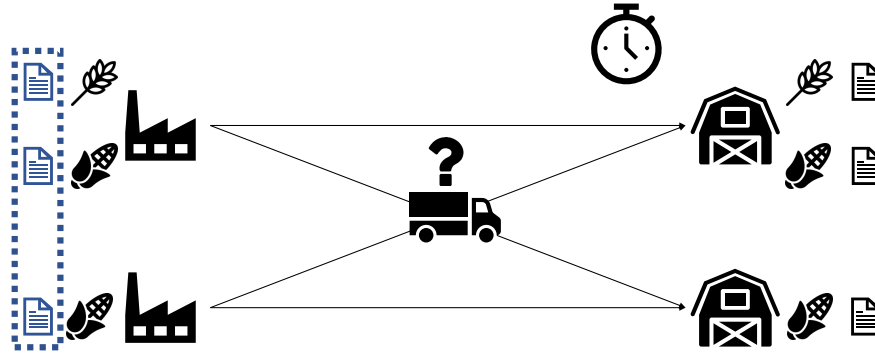


Figure 1.1: Problem representation in schematic form

At the center of the problem is the backhaul availability. Currently, there is a regular shortage in backhaul availability in the sourcing strategy of [REDACTED] in relation to their transportation by truck. This is due to the reliance on backhauls from third party logistic providers for the transportation of goods. The upside of this is that [REDACTED]'s transportation prices are low. Backhauls are often cheaper than primary cargo, as the trucks need to go back to their base after their primary cargo shipment and going back without a truckload yields no revenue for the third party logistic provider. The downside of this is that the third party logistic providers are not inclined to book a backhaul a month in advance for such low prices, as this lowers their flexibility for their more profitable primary cargo. Therefore, only using backhauls has as side effect that the truck planning cannot be made months in advance, but only a week or a couple of days before the backhaul is available. Usually, supplier contracts are purchased months in advance. Therefore, [REDACTED] has no certainty that the goods they have bought months in advance can be picked up and delivered, yielding possible unfulfilled contracts and under delivered customer demands.

Clearly, unfulfilled contracts and under delivered customer demands are undesirable. In time periods that customer demands cannot be fulfilled on time, [REDACTED] needs to provide a solution, as [REDACTED] always wants to deliver their products to the customers. Usually, [REDACTED] opts to buy franco contracts, which means that the supplier transports the goods to wherever [REDACTED] wants, in this case to the unsatisfied customer. These franco contracts are more expensive than the FOT contracts plus transportation costs. If goods are picked up late at suppliers, [REDACTED] does not necessarily make extra costs, but the professional relation with the supplier does suffer. Both these monetary and nonmonetary expenses need to be incorporated in the sourcing strategy of [REDACTED].

Finally, [REDACTED] desires a short run time (of at most one hour). Waiting multiple hours before making a decision regarding which contracts to purchase is not an option. The product prices are prone to change every day, so decisions need to be made quickly.

Based on the research motivation and problem description, we derive the research question:

How can we optimise [REDACTED]'s sourcing strategy pertaining to its backhaul availability in relation to profit maximisation?

Based on the problem description and research motivation, the answer to the question is to construct a model which has the following characteristics:

- The model maximises expected profits
- The model minimises unfulfilled contracts and under delivered customer demands
- The model incorporates the backhaul availability
- The model has a computing time of less than one hour

1.3. Structure

With this research question in mind, we define the sub-questions to structure the report and answer the main research question. Each sub-question corresponds to one or two chapters in the report. Thus, following these sub-questions, we find the outline of the report.

What relevant models and frameworks are available in literature?

Chapter 2 contains a literature study that gives an overview of relevant literature. Two relevant mathematical problems (and their relation to this thesis topic) are discussed: the transportation problem and the vehicle routing problem. Furthermore, stochastic programming (a framework for modelling optimisation problems that involve uncertainty) is discussed.

How should the sourcing strategy be optimised?

Chapter 3 describes the mathematical model used to optimise ██████'s sourcing strategy for truck import in relation to expected revenue and backhaul availability.

What solution methods are used?

Chapter 4 describes the exact solution method, obtained by linearising the nonlinear functions and variables of the model described in Chapter 3.

Chapter 5 describes the metaheuristic based on simulated annealing.

How can we construct an input instance from the data of ██████?

Chapter 6 describes the relevant data of ██████ regarding the model defined in Chapter 3. From this data, the input for the calculation of the model is obtained.

How can we describe the backhaul availability distribution?

Chapter 7 describes how a forecast for the backhaul availability is obtained from data. This forecast is obtained using state space models, which are based on exponential smoothing methods.

What are the results of these solution methods?

Chapter 8 describes the results the solution methods discussed in Chapters 4 and 5, using the test instance defined in Chapter 6 and 7.

What are the conclusions and recommendations?

Chapter 9 describes the conclusions regarding the solution methods and their performance in terms of quality and run time. Finally, suggestions for future research and recommendations to ██████ are given.

2

Literature

In this chapter, we discuss the relevant problems and frameworks found in literature. For two mathematical problems, the vehicle routing problem and the transportation problem, we describe the problem, its characteristics and the forms that are similar to our problem. We define our problem in terms of these two mathematical problems. It is found that both problems have similar characteristics to our problem. However, both problems do not account for the probabilistic availability of backhauls. In order to deal with that characteristic, we examine the framework of stochastic programming.

The vehicle routing problem is discussed in Section 2.1. In Section 2.2, the transportation problem is discussed. Stochastic programming is discussed in Section 2.3. Finally, the conclusion of this literature review is given.

2.1. Vehicle routing problem

One of the possibilities of optimising the sourcing strategy of [REDACTED] is to first define the ideal routes for the backhauls. From the ideal routes (with as objective function to maximise profit), we can calculate the contracts that are needed for the number of backhaul that are scheduled to pick up products at the supplier, yielding the sourcing strategy for [REDACTED]. From here on, we use the word truck instead of backhaul to improve readability.

A mathematical problem that deals with finding routes is the vehicle routing problem. Clarke and Wright [3] describe the vehicle routing problem as a linear optimisation problem that is commonly encountered in the domain of logistics and transport. The problem is to figure out how to serve (geographically dispersed) customers around a (central) depot, while using a fleet of trucks, which need to return to the central depot after their deliveries minimising the total length of the routes of the trucks. The routes the trucks need to take to minimise the total length of the routes of the trucks is one of the results of this problem.

In order to define our problem as a vehicle routing problem, some characteristics need to be added to the standard vehicle routing problem. Thus, we examine literature describing different vehicle routing problems. Multiple review papers have been written on the topic of vehicle routing, for example, by Braekers et al. [2] and Oyola et al. [15]. These review papers give an overview of the latest innovations in the descriptions (and solutions methods) of the vehicle routing problems. Most papers examine a vehicle routing problem with one or two additional characteristics. However, if we define our problem as a vehicle routing problem, many characteristics (more than one or two) need to be added. Thus, in order to define our problem as a vehicle routing problem, we take relevant characteristics of the different vehicle routing problems described in the literature. These different vehicle routing problems and their relevant characteristics are described below.

Defining our problem as a vehicle routing problem

We can model the suppliers of [REDACTED] as uncapacitated depots (uncapacitated means that there is no limit to how much product can be picked up at the depots). It does not matter from which depot the customer is supplied. The number of trucks that originate from that depot yield the amount of goods purchased at that

supplier. Each supplier is modelled as a separate depot. Thus, our problem has multiple depots, whereas a standard vehicle routing problem has one depot. Such a multiple depot vehicle routing problem was introduced in Tillman [19] and is called the multiple depot vehicle routing problem.

In a standard vehicle routing problem, the trucks need to return to their depot after delivering all the products to the customers. However, as [REDACTED] only uses the backhauls of third party logistic providers, our problem does not have the requirement that trucks need to return to their depot. Such a variant of the vehicle routing problem was introduced in Sariklis and Powell [17] and is called the open vehicle routing problem.

Furthermore, in a standard vehicle routing problem, the customers generally do not require a full truckload delivery. [REDACTED] only delivers full truckloads to its customers. These truckloads carry one type of product. There do exist vehicle routing problems with full truckload deliveries, such as the problem described in Arunapuram et al. [1]. This variant of the vehicle routing problem is called the vehicle routing problem with full truckloads.

Moreover, a standard vehicle routing problem considers one time period. However, as [REDACTED] is in need of a sourcing strategy over multiple time periods, we want to solve the vehicle routing problem for multiple time periods. Each time period yields constraints which need to be taken into account. Such a problem was introduced in Kim and Kim [12] and is called the multi-period vehicle routing problem.

The trucks in a standard vehicle routing problem are often homogeneous, i.e., all the trucks in the problem have the same capacity and transportation prices. The trucks used for the transport of goods for [REDACTED] have different capacities and transportation prices. A vehicle routing problem which allows for different capacities (and costs) of trucks was introduced in Golden et al. [7] and is called the mixed fleet vehicle routing problem.

The fleet size in a standard vehicle routing problem is given. As [REDACTED] does not know the availability of backhauls of third party logistic providers in advance, our problem has an uncertain fleet size. Such a problem has not been found in the literature. There do exist stochastic vehicle routing problems. Gendreau et al. [5] identify three sets of parameters that are commonly considered to be stochastic: customer demands, customers (i.e., the customers may be absent or not) and travel times. Overviews for stochastic vehicle routing, for example Gendreau et al. [5], Gendreau et al. [6] and Oyola et al. [15], do not mention a probabilistic or stochastic number of trucks that may traverse a route. There do exist vehicle routing problems which incorporate forecasts in their problem. For example, Potvin et al. [16] describe a vehicle routing problem based on both long and short demand forecasts. However, a vehicle routing problem having an uncertain or forecasted fleet size has not been found in the literature.

If we define our problem as a vehicle routing problem, a lot of characteristics need to be added to the standard vehicle routing problem definition. Furthermore, with the final definition of the problem, we still need to account for the fact that the probabilistic availability of trucks is not part of the problem definition. For this final definition, we combine the different aspects described from the previous paragraphs into one vehicle routing problem. Defined as a vehicle routing problem, our problem is an open, multi-period, multiple depot, mixed fleet vehicle routing problem with full truckloads. Solving the problem, we obtain the scheduled trucks. From this, we can calculate how much truckloads are picked up from the suppliers (depots). This way, we can calculate the contracts to buy, yielding the sourcing strategy.

However, defining our problem as a vehicle routing problem may be overly complicated for what our problem is. We can define the problem as a set of suppliers and a set of customers. The suppliers have a purchase price and the customers have a demand and a selling price. There are some more characteristics to our problem, but defined like this, the problem definition is more similar to the standard transportation problem.

2.2. Transportation problem

In a transportation problem, one has m suppliers with heterogeneous capacities and n customers with heterogeneous demands. Each supplier i has a transportation cost M_{ij} to customer j . The traditional objective of a transportation problem is to minimise the overall transportation costs, but the objective can be changed

to maximise profits, if one has the purchase price C_i and selling price R_j . In order to define our problem as a transportation problem, we take relevant characteristics of the different transportation problems defined in literature. Below, we define our problem as a transportation problem.

Defining our problem as a transportation problem

In a standard transportation problem, two dimensions of constraints are considered: customer constraints and supplier constraints. The solid transportation problem is an extension of the transportation problem and was introduced in Haley [8]. The solid transportation problem deals with three dimensions of constraints instead of two (customer and supplier). This extra dimension of constraints is usually due to different modes of conveyance. Our problem does not have an extra dimension for the heterogeneous capacities of the trucks. However, in our problem, we do need extra dimensions of constraints, for example, due to different products and time periods.

Essentially, this is the only relevant characteristic of the transportation problem described in the literature for our problem. Adding these dimensions of constraints creates a transportation problem similar to our problem. However, similar to the vehicle routing problem, there is no transportation problem in the literature that incorporates an uncertain fleet size as a characteristic. Thus, we turn to a relevant framework (stochastic programming) in order to deal with this characteristic.

2.3. Stochastic programming

A framework that handles modelling with stochastic parameters is stochastic programming. There exist two main approaches in stochastic programming: chance constrained programming and stochastic programming with recourse.

In chance constrained programming, constraints need to be satisfied with some probability. This method does not offer much potential for our problem. We do not, for example, want to satisfy all the customer demand with a certain probability. We want to satisfy as much customer demand as possible without a chance of overdelivering, as long it is expected to be profitable. This profitability depends for a part on penalties suffered due to uncertainty, i.e., penalties given due to scheduled trucks that are not expected to be realised.

Stochastic programming with recourse does offer a method to include the expected penalties incurred by the realisation of stochastic parameters in its objective function. In stochastic programming with recourse, a two-stage model is proposed. In the first stage, we solve the problem before the realisation of the stochastic variables and in the second stage (after the realisation of the variables) the recourse action is defined. This recourse is chosen to compensate the violation of the constraint (if there is one), i.e., an estimation of the losses (penalties) incurred by the violation. Violating a constraint usually leads to a penalty, which is subtracted from the objective value. The objective function of the model includes (an expected value of) the objective value of both stages. This framework offers potential for modelling the penalties incurred by not fulfilling contracts and demands.

A general stochastic model with recourse is defined as follows, as in Kall and Wallace [11]:

$$\min c^T x + \mathbb{Q}(x) \quad (2.1)$$

$$\text{s.t. } Ax = b, x \geq 0, \quad (2.2)$$

where

$$\mathbb{Q}(x) = \sum_j p^j \mathbb{Q}(x, \xi^j) \quad (2.3)$$

and

$$\mathbb{Q}(x, \xi) = \min(q(\xi)^T y \mid W(\xi)y = h(\xi) - T(\xi)x, y \geq 0) \quad (2.4)$$

where p^j is the probability that $\tilde{\xi} = \xi^j$, the j th realisation of $\tilde{\xi}$. Furthermore, $h(\xi) = h_0 + H\xi = h_0 + \sum_i h_i \xi_i$, $T(\xi) = T_0 + \sum_i T_i \xi_i$ and $q(\xi) = q_0 + \sum_i q_i \xi_i$. The function $\mathbb{Q}(x, \xi)$ is called the recourse function, and $\mathbb{Q}(x)$ the expected recourse function.

In the first stage, the cost $c^T x$ plus the expected cost of the optimal second stage decision is optimised. After the realisations of the stochastic information have been made available, in the second stage, a recourse is applied to the first stage solution. The solution of this second stage problem is the cost of the recourse action, where the term $W y$ compensates for a possible inconsistency of the system $T x - h$ (i.e., a realisation of a stochastic parameter that causes a constraint to be violated). The objective function $q^T y$ represents the cost of this recourse action.

Defining the penalties with stochastic programming with recourse

If we relate our problem to this framework, the first stage is the scheduling of the trucks and purchasing the corresponding contracts. Thus, in the first stage, we maximise expected profits, under the constraints that we deliver all the customer demand and that we pick up all products purchased at the suppliers. In the second stage, the recourse is defined, i.e., the penalties are calculated for under delivered customer demand and unfulfilled contracts.

In the model we propose, we include the expected penalties that are expected from this second stage. The expected penalties are subtracted from our expected profits. By including the expected penalties in our objective function, we are able to incorporate the availability of backhauls in the objective function of the model.

2.4. Conclusion

In this chapter, we have discussed two mathematical problems which are similar to our problem: the vehicle routing problem and the transportation problem. Considering the vehicle routing problem, the number of additional characteristics to the standard vehicle routing problem needed was high. Most characteristics were already defined before in literature. However, incorporating an uncertain fleet size in a vehicle routing problem has not been addressed before. Similarly, literature concerning the transportation problem also did not address an uncertain fleet size.

Therefore, to deal with an uncertain fleet size and the potential penalties it leads to, we introduced the mathematical framework of stochastic programming with recourse. This framework describes a method to handle an optimisation problem with stochastic input. In this manner, an expected violated constraint yields an expected recourse (penalty). In a similar manner, we can model penalties if scheduled trucks are expected not to be realised. Using this framework, and inspired by the transportation problem, we can model our problem.

3

Problem Description

In the previous chapter, we have discussed the relevant literature and have decided to base our problem on the transportation problem and the framework of stochastic programming. In this chapter, we model our problem.

As some constraints and definitions may be easier to grasp when described in an example, we start the chapter with an example. In Section 3.2, we describe the sets, parameters and variables associated with our model. Then, we describe the penalty functions for unfulfilled contracts and under delivered customer demand. In Section 3.4, we describe the objective function of our model. Finally, we describe the constraints of our model.

3.1. Challenges

In this section, we examine an example, as the definitions of expected and maximum supplier and customer result can be difficult to understand. The goal of this example is to show the necessity of probabilistic parameters (i.e., the truck availability) in the expected results of the model, while in maximum constraints we do not use these probabilistic parameters. Furthermore, we show the necessity of both types of results.

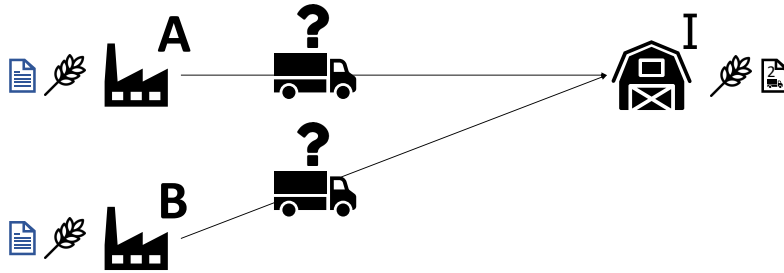


Figure 3.1: A situation with 2 suppliers and 1 customer

Consider 2 suppliers and 1 customer as in Figure 3.1, for which there is no difference between the suppliers in product and transportation costs. Furthermore, let the probability that 1, 2, or 3 trucks are available from both suppliers individually be 0.25 and let the customer demand be 2 truckloads. Because of this, it is not possible to schedule 3 trucks (in total, from all suppliers) to the customer, even though this would result in 1.50 expected delivered truckloads ($0.25 \cdot 3 + 0.25 \cdot 2 + 0.25 \cdot 1 + 0.25 \cdot 0 = 1.50$), i.e., less than the customer demand. Over delivering to a customer is not acceptable, as the customer might unexpectedly have to handle more product than requested. Thus, in our model, we do not want to schedule more trucks than the customer ordered. Therefore, we define the maximum customer result: the customer demand minus the maximum amount of product (i.e., not taking probabilistic parameters into account) delivered to the customer by scheduled trucks. The maximum customer result must be a nonnegative value, i.e., we cannot deliver more than the customer

demanded.

Now that the maximum customer results constraints the number of scheduled trucks, we need to quantify the amount we expect to under deliver at customers. To model this, we define the expected results which include probabilities. If we schedule 2 truckloads, we expect $0.50 \cdot 2 + 0.25 \cdot 1 = 1.25$ delivered truckloads, which leads to an expected customer result of $2 - 1.25 = 0.75$ truckloads, which leads to an expected customer penalty.

Now, let the probability that 1, 2, or 3 trucks are available from supplier A be 0.30 (and so 0.10 for 0 available trucks) and 0.10 for supplier B (and so 0.70 for 0 available trucks). Now, we pick supplier A, as supplier A has a higher truck availability, and the only difference between the suppliers is in truck availability. Thus, we schedule 2 trucks from supplier A to customer I. Now, the expected amount of product we pick up and deliver to customer I is 1.50 truckloads ($0.60 \cdot 2 + 0.30 \cdot 1 + 0.10 \cdot 0 = 1.50$). However, we need to purchase a contract for two truckloads. Otherwise, if we were to purchase a smaller contract and two trucks actually are available, then we are not able to pick up the second truckload, because we do not have a contract for two truckloads. To deal with this in the model, we define the maximum supplier result which exclude the probabilities. This maximum supplier result is the amount of product on the contract minus the amount of product picked up by scheduled trucks and must be a nonnegative value, i.e., we cannot schedule to pick up more product than we have purchased.

Now that the maximum supplier results constraints the minimum number of tons that needs to be purchased, we need to quantify the amount we expect to unfulfill. To do this, we define the expected supplier result, i.e., the amount of product we expect to leave (i.e., not pick up) on a contract (in the current example: $2 - 1.50 = 0.50$), such that we can minimise this quantity.

Thus, the optimal solution is that we schedule two truckloads from supplier A to customer I and purchase a contract for two truckloads at supplier A. Note that we schedule these two truckloads based on truck availability, as the product and transport costs are the same for all suppliers. Therefore, the decision is based on the minimisation of the expected under delivery in customer demand (expected customer penalty), as well as the expected deficiency at the supplier (expected supplier penalty).

So, the maximum customer prohibits us from over delivering and the maximum supplier result prohibits us from picking up more than we purchased. We can use the truck availability to calculate the expected supplier results and the expected customer results. Then, a penalty can be given over these expected results to incorporate the truck availability in the objective function.

3.2. Sets, parameters and variables

In this section, we define the sets, parameters and (decision) variables in our model. The sets and parameters are the input of our model. The decision variables are part of the output of our model.

As our problems calls for transportation from supplier locations to customer locations, we define the set I , which contains all the locations. This set contains both the supplier locations and customer locations. There are two important subsets of this set: the set of supplier locations, $I_N \subseteq I$ and the set of customer locations, $I_M \subseteq I$. Some supplier locations are also customer locations, i.e., $I_N \cap I_M \neq \emptyset$. The set G represents the set of products. Furthermore, the set T represents the planning horizon.

Our goal is to optimise the sourcing strategy for each time period $t \in T$. This means that we want to know for each supplier location $i \in I_N$, how much product $g \in G$ (in tons) to purchase for a given price C_{igt} . The decision variable $y_{igt_1 t_2}$ is called a contract and specifies the number of tons of product $g \in G$, which need to be picked up at supplier location $i \in I_N$ in the period between $t_1 \in T$ and $t_2 \in T$. The start time of the contract is $t_1 \in T$ and $t_2 \in T$ is the end time of the contract, with $t_1 \leq t_2$. A contract is live when the current time period $t \in T$ is in $[t_1, t_2]$. A contract is called depleted when the total amount of product bought for period $[t_1, t_2]$ is picked up.

We define \bar{t} to be the latest time period for which there is realised data available, i.e., the truck transport has been realised until that time period. The planning horizon, T , starts after \bar{t} . The time period \bar{t} is the latest

time period before the planning horizon T starts. So, $\bar{t} \notin T$.

Any live contract starting before realised time period \bar{t} is modelled as starting at time period \bar{t} . [REDACTED] keeps data on the remaining amount on the contract, i.e., how much product has been picked up from which contract. So, as we have realised data of time period \bar{t} and the time periods before \bar{t} , we can update the live and realised contracts (in the terms of the model starting at \bar{t}) with the amount of product remaining on the contract. So, we can subtract the amount of product already picked up in realised time periods from the original contract amount. Thus, for our model there does not exist a time period before time \bar{t} .

If at time period $t \in T$, a supplier location $i \in I_N$ has a live, nondepleted contract for product $g \in G$, then it is possible to pick up product $g \in G$ from that supplier location i at that time period t . The transportation cost to deliver one ton of product $g \in G$ from supplier location $i \in I_N$ to customer location $j \in I_M$ is defined as M_{ij} . The amount of product a truck can carry, i.e., the capacity of a truck, Q_{ij} depends on the country of the supplier location $i \in I_N$ and the country of customer location $j \in I_M$. This is due to trade legislation. Further information is given in Section 6.3. The transportation cost is not dependent on the product $g \in G$ or time period $t \in T$. When product $g \in G$ of a contract is delivered to customer location $j \in I_M$ at time period $t \in T$, a given revenue of R_{jgt} per ton delivered is generated. Each customer location $j \in I_M$ has a demand $D_{jgt} \geq 0$ for each product $g \in G$ for all time periods $t \in T$.

[REDACTED] provides the delivery of product using third party logistic providers. This means that trucks only need to be planned for a one-way trip. When trucks have delivered their full truckload of a single product $g \in G$ (no mixing allowed) from supplier location $i \in I_N$ to customer location $j \in I_M$, they do not need to return to the supplier location $i \in I_N$. It is possible that trucks need to deliver from supplier location $i \in I_N$ to customer location $j \in I_M$, with $i = j$, i.e., that a truck delivers the product to the same location as the location the truck picked up the product.

Supplier locations $i \in I_N$ have no capacity constraints, i.e., each supplier has an unlimited supply of product $g \in G$. How much product $g \in G$ is purchased depends multiple factors, such as the cost price C_{igt} , transportation costs M_{ij} , and the availability of trucks from supplier location $i \in I_N$ towards customer location $j \in I_M$. Each route has an uncertain availability of trucks. We define P_{ijt}^z as the cumulative probability that $z \in [1, 2, \dots, Z]$ trucks can traverse the route from $i \in I_N$ to $j \in I_M$ in time period $t \in T$, i.e., if we were to schedule z trucks on the route (i, j) at time period $t \in T$, we expect that $P_{ijt}^z z$ trucks are realised. We define Z as follows:

$$Z = \arg \max_{\substack{t \in T \\ i \in I_N \\ j \in I_M}} \left(\frac{\sum_{g \in G} D_{jgt}}{Q_{ij}} \right). \quad (3.1)$$

Z is an upper bound on the maximum number of trucks that may traverse any route (i, j) . The number Z is the maximum number of trucks needed to supply the biggest demand (over all $g \in G$) from a single supplier. The sets and parameters are summarised below.

Sets:

- I Set of locations, with $|I| = \beta$.
- I_N Set of supplier locations, $I_N \subseteq I$ with $|I_N| = n \leq \beta$.
- I_M Set of customer locations, $I_M \subseteq I$ with $|I_M| = m \leq \beta$.
- T Time horizon, $T = [0, 1, 2, \dots, \bar{T}]$, with $|T| = \bar{T} + 1$.
- G Set of products.

Parameters:

- C_{igt} Purchase costs of one ton of product $g \in G$ bought at supplier location $i \in I_N$ in time period $t \in T$.
- D_{jgt} Demand of product $g \in G$ (in tons) of location $j \in I_M$ in time period $t \in T$.
- M_{ij} Transportation costs of one ton of product from supplier location $i \in I_N$ to customer location $j \in I_M$ in time period $t \in T$.
- Q_{ij} Capacity of the truck based on route (i, j) .
- Z Maximum number of vehicles that may traverse any route (i, j) .
- P_{ijt}^z Probability that z trucks traverse route (i, j) in time period $t \in T$, with $z = 1, 2, \dots, Z$.
- R_{jgt} Revenue of one ton of product $g \in G$ delivered at customer location $j \in I_M$ in time period $t \in T$.

The binary decision variable x_{ijt}^z is one when z trucks are scheduled from supplier location $i \in I_N$ to customer location $j \in I_M$ in time period $t \in T$ and zero otherwise. The nonnegative integer number σ_{ijgt}^z is the number of trucks that transport product $g \in G$ on route (i, j) in time period $t \in T$ from the z number of trucks that are scheduled from supplier location $i \in I_N$ to customer location $j \in I_M$. The real, nonnegative variable $y_{igt_1 t_2}$ is the contract decision variable, which represents the amount of product $g \in G$ (in tons) that should be bought from supplier location $i \in I_N$ for the time period $[t_1, t_2]$, with $t_1 \leq t_2$. The decision variables are summarised below.

Decision variables:

- x_{ijt}^z Binary variable that is one when z trucks are scheduled from supplier location $i \in I_N$ to customer location $j \in I_M$ in time period $t \in T$ and zero otherwise.
- σ_{ijgt}^z The number of trucks that transport product $g \in G$ on route (i, j) from the z number of trucks that are scheduled from supplier location $i \in I_N$ to customer location $j \in I_M$ in time period $t \in T$.
- $y_{igt_1 t_2}$ Number of tons of product $g \in G$ that is scheduled to be picked up at supplier location $i \in I_N$ in time period $[t_1, t_2]$, with $t_1 \leq t_2$ and $t_1, t_2 \in T$.

It is uncertain whether the decision variables x_{ijt}^z , σ_{ijgt}^z and $y_{igt_1 t_2}$ are carried out as scheduled. The time period \bar{t} is the most recent time for which there exist realised decision parameters. We define the realised contract by $\bar{y}_{igt_1 t_2}$. We allow an overlap of scheduled contracts $y_{igt_1 t_2}$ and realised contracts, $\bar{y}_{igt_1 t_2}$. Thus, if there are already (multiple) realised contract(s) $\bar{y}_{igt_1 t_2}$, then we may schedule a contract $y_{igt_1 t_2}$ in that same time period. The realised decision variables \bar{x}_{ijt}^z and $\bar{\sigma}_{ijgt}^z$ are dealt with in another way. This is explained later. The realised decision variables are summarised below.

Realised decision parameters:

- $\bar{y}_{igt_1 t_2}$ Number of tons of product $g \in G$ that need to be picked at supplier location i in the time period between t_1 and t_2 .

3.3. Penalty functions

In order to include the truck (backhaul) availability in the model, we need to quantify the penalties of under delivered demand and unfulfilled contracts (i.e., not picking up all the product $g \in G$ at supplier location $i \in I_N$ in time period t). To do this, we include penalty functions for unfulfilled demand and unfulfilled contracts. The penalty functions and their input are described in this section.

The penalty functions express the unfulfilled demand and unfulfilled contracts as a monetary cost and are included in the objective function in order to account for the truck availability in the optimisation model. The functions are linear, i.e., a penalty is incurred per ton of product that is under delivered. However, if we deliver a little more than the customer demanded or pick up a little more than the contract stipulated (these scenarios will be explained later), no penalty will be given.

Functions:

- $\theta_{jgt}(x)$ Penalty function for under delivering x number of tons of product $g \in G$ to customer location $j \in I_M$ in time period $t \in T$.
- $\phi_{igt}(x)$ Penalty function for picking up x number of tons of product $g \in G$ too few from supplier location $i \in I_N$ in time period $t \in T$.

The penalty function for under delivered demand of product $g \in G$ for a customer location $j \in I_M$ in time period $t \in T$, θ_{jgt} , has as input the expected customer result, l_{jgt} , i.e., the difference between the demand of the customer location $j \in I_M$ of product $g \in G$ up to time period $t \in T$ and the expected delivered amount to the customer location $j \in I_M$ of product $g \in G$ up to time period $t \in T$. We define the set $T^* = [0, t]$.

$$l_{jgt} = D_{jg\bar{t}} + \sum_{t^* \in T^*} \sum_{i \in I_N} \sum_{z=1}^Z (D_{jgt} - Q_{ij} \sigma_{ijgt^*}^z P_{ijt^*}^z), \quad \forall j \in I_M, \forall t \in T, \forall g \in G. \quad (3.2)$$

The parameter $D_{jg\bar{t}}$ has a different definition than D_{jgt} , as the former represents the remaining demand, i.e., the difference in demand and realised delivery (obtained from the realised decision variables \bar{x}_{ijt}^z and $\bar{\sigma}_{ijt}^z$) in time period \bar{t} , whereas the latter represents the original demand at time period $t \in T$.

For example, in Figure 3.2, we can see that the original demand is the same for all time periods. If the realised customer result, l_{jgt} , is positive, such as in $t = \bar{t}$ (note that in time period \bar{t} , the result is realised, not expected), then the demand of supplier location $j \in I_M$ of product $g \in G$ is under delivered. This can be noted by comparing the demand of product $g \in G$ the customer location $j \in I_M$ demanded in time period \bar{t} , $D_{jg\bar{t}}$ (the yellow bar at $t = \bar{t}$), and the realised delivery in that time period \bar{t} (dark green bar at $t = \bar{t}$). In that case, \square pays a penalty of $\theta_{jg\bar{t}}(l_{jg\bar{t}})$, and the remaining amount of product $g \in G$ (the red bar at $t = \bar{t}$) to be delivered to customer location $j \in I_M$ will be added to the original demand of the next time period $t = 0$, D_{jg0} .

However, if we schedule to deliver all demand, but expect that we have an under delivery for time $t = 0$, we cannot add that expected under delivery to the demand of the next time period, $t = 1$. Therefore, the penalty of $t = 0$ is also suffered at $t = 1$, as we cannot schedule to over deliver.

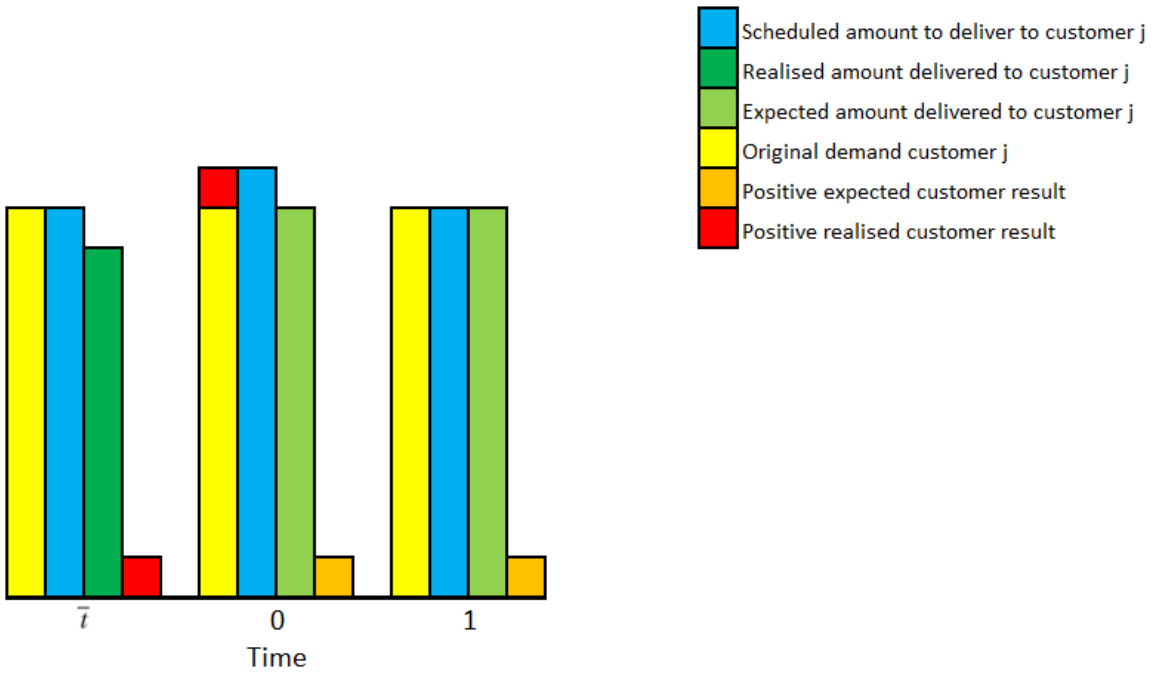


Figure 3.2: An example for the expected customer result and updated demand

Furthermore, the expected customer result l_{jgt} of product $g \in G$ of customer location $j \in I_M$ at the time period $t \in T$ may be positive or negative. If $l_{jgt} \leq 0$, then $\theta_{jgt}(l_{jgt}) = 0$. Therefore, this penalty function is not linear.

The other penalty function, ϕ_{igt} , is used to determine the penalty for unfulfilled contracts. This function has as input the expected supplier result v_{igt} . The expected supplier result v_{igt} of supplier location $i \in I_N$ for product $g \in G$ at time period $t \in T$ may be positive or negative. The expected supplier result v_{igt} represents the number of tons that we expect to pick up above or below the stipulated amount of product $g \in G$ to be picked up for the contract ending in time period $t \in T$. If the expected supplier result v_{igt} is negative, we expect to pick up more product $g \in G$ than the contract(s) ending at time period $t \in T$ stipulate. In this case, no penalty is expected. Thus, for $v_{igt} \leq 0$, we have that $\phi_{igt}(v_{igt}) = 0$. This means that this function is not linear. A linearisation of this can be found in Chapter 4. A negative expected supplier result is possible as there may be a live contract from which we can pick up the product $g \in G$. However, if a contract ends at period $t \in T$, and we expect to pick up less than the contract stipulated, the expected supplier result v_{igt} is

positive and an expected penalty of $\phi_{igt}(v_{igt})$ will be paid. When this expected penalty is paid, we do not need to pick up the expected product $g \in G$ remaining from the expired contract. This yields the following expected supplier result v_{igt} , for $t \geq 1$:

$$v_{igt} = \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{\substack{t_1 \leq t \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ijt}^z + \min(v_{ig(t-1)}, 0), \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T. \quad (3.3)$$

The first two terms are the scheduled and realised contracts ending at time period t . Note that $t_1 = \bar{t}$ is added to the summation for realised contracts, as $\bar{t} \notin T$ and realised contracts are able to start at \bar{t} , which is the starting point of our model. As \bar{t} is the most recent realised time period, it is not able to schedule a contract $y_{igt_1 t_2}$ with $t_1 = \bar{t}$.

From these contracts ending at time period t , we subtract the product $g \in G$ that is expected to be picked up, the third term in (3.3): $\sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ijt}^z$. Finally, the last term represents the expected supplier result of the previous period, $\min(v_{ig(t-1)}, 0)$.

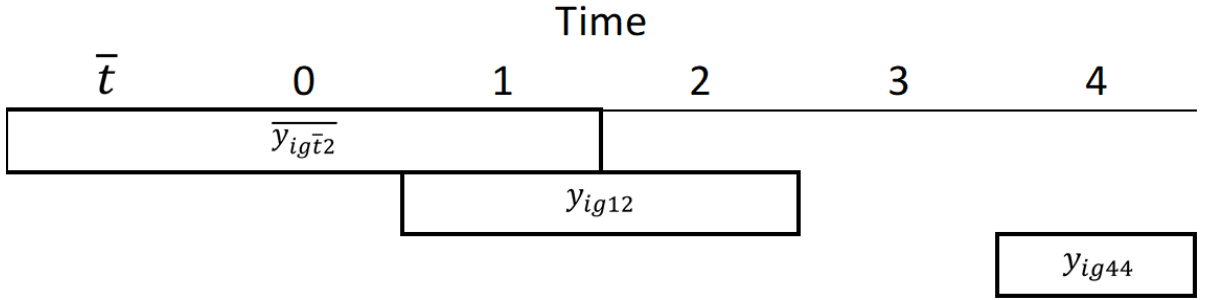


Figure 3.3: An example of contract variables and the expected supplier result

Taking the minimum of $v_{ig(t-1)}$ and 0 in (3.3) is needed such that we do not need to pick up the expected product remaining from an expired contract. For example, see Figure 3.3. If the expected supplier result v_{ig2} of a supplier location $i \in I_N$ of product $g \in G$ in time period 2 is positive, we expect to pay a penalty for the expected remaining product $g \in G$ from the expiring contract. If at a later time period, in this case $t = 4$, we again have a contract at that supplier location $i \in I_N$, we do not need to pick up this expected remaining product $g \in G$ from time period $t = 2$. Furthermore, we do not expect to pay a penalty for every time period between 2 and 4. Therefore, the expected supplier result after an expected positive result needs to be 0.

Furthermore, if the expected supplier result of supplier location $i \in I_N$ of product $g \in G$ from the previous time period $t - 1 \in T$, $v_{ig(t-1)}$, is negative, we need to subtract it from the expected supplier result of the next period, v_{igt} . Again, we refer to Figure 3.3. When calculating the expected supplier result for y_{ig12} , we cannot simply sum all the product $g \in G$ that is expected to be picked up between $t = 1$ and $t = 2$, because of contract $\bar{y}_{ig\bar{t}1}$. At time period $t = 1$, we expect to pick up product $g \in G$ from both contracts ($\bar{y}_{ig\bar{t}1}$ and y_{ig12}). Therefore, we need the expected supplier result of the previous time period $t - 1 \in T$, such that we can keep track of all the ending contracts and how much product $g \in G$ is expected to be picked up every time period. Otherwise, we would not take the contract ending on or after $t_1 \in T$ and before $t \in T$ into account.

However, equation (3.3) is not linear. The assumption that a product $g \in G$ does not need to be picked up after a live (not depleted) contract has ended, results in the fact that the expected supplier result of the previous time needs to be reset to zero. This causes the nonlinearity. A linearisation of this equation can be found in Chapter 4.

For the first time period $t = 0$, we need an alternative equation:

$$v_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1=\bar{t}}} \bar{y}_{igt_1 0} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijg0}^z P_{ij0}^z + v_{ig\bar{t}}, \quad \forall i \in I_N, \forall g \in G. \quad (3.4)$$

We need equation (3.4) due to the fact that $-1 \notin T$ and so $t - 1$ would yield a problem for $t = 0$ for equation (3.3). The parameter v_{igt} is the realised supplier result. It is possible to calculate this result from the available data of [REDACTED]. Therefore, we do need to take a minimum in this equation. As v_{igt} is calculated beforehand, we set $v_{igt} = 0$, if v_{igt} is positive. It is not possible for a scheduled contract to appear in the realised supplier result. This is due to the fact that we cannot (plan to) purchase contracts for a passed time period. So, v_{igt} is a parameter.

Now that we have described the input for the penalty functions, we can describe the objective function, which includes these penalty functions.

3.4. Objective function

In this section, we define the objective function of our model. The objective function of our model consists of three parts: the expected revenue, the expected penalties and the expected costs.

The total expected revenue ω_1 is determined by summing over the expected amount of product $g \in G$ delivered to customer location $j \in I_M$ at time period $t \in T$, $\sum_{i \in I_N} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ijt}^z$, multiplied by the selling price per ton, R_{jgt} . Note that we expect to receive revenue for product that we expect to deliver, which is not the same as the demanded amount of product. The total expected revenue equals:

$$\omega_1 = \sum_{t \in T} \sum_{g \in G} \sum_{i \in I_N} \sum_{j \in I_M} \sum_{z=1}^Z R_{jgt} Q_{ij} \sigma_{ijgt}^z P_{ijt}^z. \quad (3.5)$$

The total expected penalties ω_2 relate to the two different penalty functions:

$$\omega_2 = \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \theta_{jgt} (l_{jgt}) + \sum_{t \in T} \sum_{g \in G} \sum_{i \in I_N} \phi_{igt} (v_{igt}). \quad (3.6)$$

The expected costs of a product $g \in G$ consist of the transportation cost per ton from $i \in I_N$ to $j \in I_M$ at time period $t \in T$, M_{ij} , plus the purchasing cost of product $g \in G$ at supplier location $i \in I_N$ at time period $t \in T$, C_{igt} , multiplied by the expected amount purchased, which is equivalent to the expected number of scheduled trucks from $i \in I_N$ to $j \in I_M$ at time period t , $\sum_{z=1}^Z \sigma_{ijgt}^z P_{ijt}^z$, times the capacity of those trucks, Q_{ij} . Note that we do not pay for the amount of product $g \in G$ the contract specified, but for the amount we pick up.

$$\omega_3 = \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^Z (M_{ij} + C_{igt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z. \quad (3.7)$$

Thus, we have the following objective function in order to optimise [REDACTED]'s sourcing strategy:

$$\omega = \max_{\sigma_{ijgt}^z, y_{it_1 t_2}} \omega_1 - \omega_2 - \omega_3, \quad (3.8)$$

$$\begin{aligned} &= \max_{\sigma_{ijgt}^z, y_{it_1 t_2}} \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^Z (R_{jgt} - M_{ij} - C_{igt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z - \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \theta_{jgt} (l_{jgt}) \\ &\quad - \sum_{t \in T} \sum_{g \in G} \sum_{i \in I_N} \phi_{igt} (v_{igt}). \end{aligned} \quad (3.9)$$

3.5. Constraints

In this section, we define and describe the constraints associated with the model. The constraints of the model yield the solution space of our model.

In section 3.3, we already defined three constraints for the model: constraints (3.2), (3.3) and (3.4). These

constraints define the expected customer result l_{jgt} and the expected supplier result v_{igt} :

$$l_{jgt} = D_{jg\bar{t}} + \sum_{t^* \in T^*} \sum_{i \in I_N} \sum_{z=1}^Z \left(D_{jgt^*} - Q_{ij} \sigma_{ijgt^*}^z P_{ijt^*}^z \right), \quad \forall j \in I_M, \forall g \in G, \forall t \in T, \quad (3.2)$$

$$v_{igt} = \sum_{t_1 \leq t} y_{igt_1} + \sum_{\substack{t_1 \leq t \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ijt}^z + \min(v_{ig(t-1)}, 0), \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (3.3)$$

$$v_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ij0}^z + v_{ig\bar{t}}, \quad \forall i \in I_N, \forall g \in G. \quad (3.4)$$

Next to the expected supplier result v_{igt} , we define the maximum supplier result h_{igt} in equations (3.10) and (3.11). The difference between the maximum supplier result h_{igt} and v_{igt} is that the maximum supplier result does not consider the probability P_{ijt}^z of z trucks being available on route (i, j) at time period $t \in T$. It ensures that the model will generate contracts for the product $g \in G$ it schedules to pick up, and not for the amount of product $g \in G$ it expects to pick up. For an example, see Section 3.1. Note that the parameter $v_{ig\bar{t}}$ is the realised supplier result.

$$h_{igt} = \sum_{t_1 \leq t} y_{igt_1} + \sum_{\substack{t_1 \leq t \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z + \min(h_{ig(t-1)}, 0), \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (3.10)$$

$$h_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijg0}^z + v_{ig\bar{t}}, \quad \forall i \in I_N, \forall g \in G. \quad (3.11)$$

Furthermore, constraint (3.12) ensures no more trucks can be planned than there is product $g \in G$ remaining on a contract. The maximum supplier result h_{igt} is the maximum number of tons that we pick up above or below the stipulated amount of product $g \in G$ to be picked up for the contract ending in time period $t \in T$. If the maximum supplier result is nonpositive, but there are live contracts $(y_{igt_1 t_2} \text{ or } \bar{y}_{igt_1 t_2})$ with $t_2 > t$, then we can still pick up from those contracts at supplier location $i \in I_N$ in time period $t \in T$, but not more than the amount remaining of product $g \in G$ on those live contracts. If there are no live (realised or scheduled) contracts, then the maximum supplier result, h_{igt} , cannot be equal to or lower than minus one truckload. We allow to pick up just short of one truckload more from the contract due to the fact that demands are specified in tons and not in number of trucks. This extra truckload is modelled as $-\arg\min_{j \in I_M} Q_{ij} + \epsilon$. Due to this constraint, the amount of product $g \in G$ picked up from the contracts from supplier location $i \in I_N$ ending at time period $t \in T$ is not exceeded by one full truckload or more. Note that if we were to use the expected supplier result v_{igt} in constraint (3.12), then it could be possible to schedule multiple pickups from one supplier, even though we only have one truckload remaining on the contract.

$$h_{igt} + \sum_{\substack{t_2 > t \\ t_1 \leq t}} y_{igt_1 t_2} + \sum_{\substack{t_2 > t \\ t_1 \leq t \\ \cup t_1 = \bar{t}}} \bar{y}_{igt_1 t_2} \geq -\arg\min_{j \in I_M} Q_{ij} + \epsilon, \quad \forall i \in I_N, \forall g \in G, \forall t \in T, \quad (3.12)$$

Constraint (3.13) is the demand constraint. Due to the fact that demands are specified in tons and not in number of trucks, we allow for a small over delivery, similar to constraint (3.12). No penalty is expected if up to one truckload more is delivered than demanded. [REDACTED] wants to deliver as much as possible, without a possible over delivery. Therefore, for each time time period $t \in T$, it is allowed to deliver up to the total demand up until time period $t \in T$. For each time period $t \in T$, we cannot deliver more than the total demanded product up until that time period (remember the set $T^* = [0, t]$), plus just less than one truckload:

$$\sum_{z=1}^Z \sum_{i \in I_N} \sum_{t^* \in T^*} Q_{ij} \sigma_{ijgt^*}^z \leq \sum_{t^* \in T^*} D_{jgt^*} + \arg\min_{j \in I_M} Q_{ij} - \epsilon, \quad \forall j \in I_M, \forall g \in G, \forall t \in T. \quad (3.13)$$

Constraint (3.14) ensures that for one route (i, j) and time period $t \in T$ there is one integer number of vehicles scheduled, i.e., such that not both 1 and 4 trucks can be scheduled for route (i, j) in time period $t \in T$. Constraint (3.15) forces $\sum_{g \in G} \sigma_{ijgt}^z$ to be equal to $x_{ijt}^z z$, such that all trucks are distributed over the products $g \in G$.

Constraint (3.16) forces σ_{ijgt}^z to be a nonnegative integer, such that we cannot schedule a truckload which contains multiple products. Constraint (3.17) forces x_{ijgt}^z to be binary. Constraint (3.18) forces the scheduled contracts to be a nonnegative real number. Finally, constraints (3.19) and (3.20) force the variables l_{jgt} , v_{igt} and h_{igt} to be real numbers.

$$\sum_{z=1}^Z x_{ijgt}^z \leq 1, \quad \forall i \in I_N, \forall j \in I_M, \forall t \in T, \quad (3.14)$$

$$\sum_{g \in G} \sigma_{ijgt}^z = x_{ijgt}^z, \quad \forall i \in I_N, \forall j \in I_M, \forall t \in T, \forall z \in [1, 2, \dots, Z], \quad (3.15)$$

$$\sigma_{ijgt}^z \in \mathbb{N}^+, \quad \forall i \in I_N, \forall j \in I_M, \forall g \in G, \forall t \in T, \forall z \in [1, 2, \dots, Z], \quad (3.16)$$

$$x_{ijgt}^z \in \{0, 1\}, \quad \forall i \in I_N, \forall j \in I_M, \forall t \in T, \forall z \in [1, 2, \dots, Z], \quad (3.17)$$

$$y_{igt_1 t_2} \in \mathbb{R}^+, \quad \forall i \in I_N, \forall g \in G, \forall t_1 \in T, \forall t_2 \geq t_1 \in T, \quad (3.18)$$

$$l_{jgt} \in \mathbb{R}, \quad \forall j \in I_M, \forall g \in G, \forall t \in T, \quad (3.19)$$

$$v_{igt}, h_{igt} \in \mathbb{R}, \quad \forall i \in I_N, \forall g \in G, \forall t \in T. \quad (3.20)$$

4

Exact Solution Method

The model described in Chapter 3 contains two nonlinear balance equations (the supplier results), as well as two nonlinear penalty functions in the objective function. In this chapter, we linearise the nonlinear equations and nonlinear functions in the objective function in order to obtain an exact solution method.

The linearisation of the supplier results and penalty functions is given in Sections 4.1 and 4.2, respectively. However, building the model with all the introduced variables from the sets defined in Section 3.2 may yield an unnecessarily big problem, as it is possible that some variables may be excluded without affecting the problem. Thus, all combinations of the sets of our model (defined in Section 3.2) are analysed. From this analysis, sets are constructed which exclude all unnecessary variables. This results in a method which reduces the number of variables and constraints of the model. This method is described in Section 4.3.

At the end of the chapter, the linearised model is given.

4.1. Linearisation of supplier results

Our model, defined Chapter 3, introduced nonlinearities. In this section, we linearise the nonlinearities associated with the expected supplier result and the maximum supplier result.

The expected supplier result (defined in constraint (3.3)) takes the minimum of $v_{ig(t-1)}$ and 0, while the maximum supplier result (defined in constraint (3.10)) takes the minimum of $h_{ig(t-1)}$ and 0. Therefore, to linearise these constraints, we need to linearise $\min(v_{ig(t-1)}, 0)$ and $\min(h_{ig(t-1)}, 0)$. Due to the similarity in the two constraints, we go into detail for the linearisation for expected supplier result and only give the constraints resulting from the linearisation of the maximum supplier result.

We introduce the variable b_{igt} , which takes the value of $\min(v_{ig(t-1)}, 0)$. We also define the big M constant Γ_{1igt} . Furthermore, we introduce the binary variable a_{igt} . The binary variable a_{igt} determines whether b_{igt} should be 0 or $v_{ig(t-1)}$ such that $b_{igt} = \min(v_{ig(t-1)}, 0)$. If $b_{igt} = \min(v_{ig(t-1)}, 0) = 0$ at time period $t \in T$, then $a_{igt} = 1$. If $b_{igt} = \min(v_{ig(t-1)}, 0) = v_{ig(t-1)}$, then $a_{igt} = 0$.

The value of Γ_{1igt} needs to be at least $v_{ig(t-1)}$. Otherwise, the value b_{igt} may take the value of Γ_{1igt} instead of $v_{ig(t-1)}$ when constrained by big M constraints. To define Γ_{1igt} as a parameter, we define Γ_{1igt} using only input parameters:

$$\Gamma_{1igt} = \sum_{j \in I_M} \sum_{t^* \in T^*} \left(Q_{ij} Z \left(1 - P_{ijt}^Z \right) \right) \quad \forall i \in I_N, \forall g \in G, \forall t \in T. \quad (4.1)$$

The term $\left(\sum_{j \in I_M} \sum_{t^* \in T^*} Q_{ij} Z \left(1 - P_{ijt}^Z \right) \right)$ represents the maximum number of tons (Z trucks times capacity Q_{ij} , where Z is the maximum number of trucks that can be scheduled on route (i, j)) of product $g \in G$ that can be ordered from any supplier $i \in I_M$ during the whole time horizon T , which we multiply by the probability that it cannot be picked up $\left(1 - P_{ijt}^Z \right)$. This is the maximum amount of product transported from any

supplier location $i \in I_N$, any product $g \in G$, and any time period $t \in T$, which we multiply by the maximum probability of not picking up product, $(1 - P_{ijt}^Z)$. Thus, this number is at least v_{igt} .

Below, the linear constraints that ensure that $b_{igt} = \min(v_{ig(t-1)}, 0)$ are given. Adding these constraints and altering the balance equation (3.3) results in the linearisation of balance equation (3.3).

$$b_{igt} \leq v_{ig(t-1)}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.2)$$

$$b_{igt} \geq -\Gamma_{1igt}(1 - a_{igt}), \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.3)$$

$$b_{igt} \geq v_{ig(t-1)} - \Gamma_{1igt}a_{igt}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.4)$$

$$a_{igt} \in \{0, 1\}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.5)$$

$$b_{igt} \in \mathbb{R}^-, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T. \quad (4.6)$$

As $v_{igt} \leq \Gamma_{1igt}$ (by definition of Γ_{1igt}), b_{igt} takes the value of $\min(v_{ig(t-1)}, 0)$ due to the constraints above. Constraints (4.2) and (4.6) ensure that b_{igt} is bounded from above by 0 and $v_{ig(t-1)}$. In constraints (4.3) and (4.4), the binary variable a_{igt} is used. We identify four cases in order to show that these constraints ensure that $b_{igt} = \min(v_{ig(t-1)}, 0)$.

If $a_{igt} = 0$ and $\min(v_{ig(t-1)}, 0) = v_{ig(t-1)}$, then (4.3) ensures that $b_{igt} \geq -\Gamma_{1igt}$, as $1 - a_{igt} = 1$. This constraint is made obsolete by (4.4), which ensures that $b_{igt} \geq v_{ig(t-1)}$, as $a_{igt} = 0$. Thus, we find the correct result: $b_{igt} = v_{ig(t-1)}$. The other correct case is $a_{igt} = 1$ and $\min(v_{ig(t-1)}, 0) = 0$. As $v_{igt} \leq \Gamma_{1igt}$, it follows that $v_{ig(t-1)} - \Gamma_{1igt} \leq 0$. Therefore, equations (4.3) and (4.4) ensure that $b_{igt} \geq 0$. Together with (4.6), this results in $b_{igt} = 0$.

To show the constraints do not allow for different combinations of a_{igt} and b_{igt} , assume that $a_{igt} = 0$ and $\min(v_{ig(t-1)}, 0) = 0$. In this case, b_{igt} is equal to $v_{ig(t-1)}$ by constraints (4.2) and (4.4). However, since $\min(v_{ig(t-1)}, 0) = 0$, $v \geq 0$, which violates constraint (4.6) when $v > 0$. For the final case, $a_{igt} = 1$ and $\min(v_{ig(t-1)}, 0) = v_{ig(t-1)} \leq 0$. In this case, constraint (4.3) ensures that $b_{igt} \geq 0$, while from (4.2) we find $b_{igt} \leq v_{ig(t-1)} \leq 0$. Thus, this case is also infeasible, if $v_{ig(t-1)} < 0$.

Now, we can change balance equation (3.3) to the following linear equation:

$$v_{igt} = b_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z P_{ijt}^z, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T. \quad (4.7)$$

Following the same steps, but replacing b_{igt} with f_{igt} and a_{igt} with e_{igt} , we can linearise h_{igt} as follows:

$$h_{igt} = f_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \sigma_{ijgt}^z, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T. \quad (4.8)$$

with

$$f_{igt} \leq h_{ig(t-1)}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.9)$$

$$f_{igt} \geq -\Gamma_{1igt}(1 - e_{igt}), \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.10)$$

$$f_{igt} \geq h_{ig(t-1)} - \Gamma_{1igt}e_{igt}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.11)$$

$$e_{igt} \in \{0, 1\}, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T, \quad (4.12)$$

$$f_{igt} \in \mathbb{R}^-, \quad \forall i \in I_N, \forall g \in G, \forall t \geq 1 \in T. \quad (4.13)$$

The equations for the expected and maximum supplier result for $t = 0$, i.e., equations (3.4) and (3.11), do not need to be changed.

4.2. Linearisation of penalty functions

Our model contains other nonlinearities than the expected supplier result and the maximum supplier result. The penalty functions defined in Section 3.3 are also nonlinear. In this section, we linearise these penalty functions.

We can linearise the penalty function for the unfulfilled contracts θ_{igt} in a similar way as we linearised the expected supplier result. We use a_{igt} again, the binary variable that has value 1 if the result of v_{igt} is positive, and 0 otherwise. However, for this linearisation, we also need a_{igt} for $t = 0$. We also use big M constant Γ_{1igt} again, with $\Gamma_{1igt} \geq |v_{igt}|$. Furthermore, we define the parameter Φ_{igt} , the penalty per ton of product $g \in G$ picked up less than the contract stipulated for supplier location $i \in I_M$ at time period $t \in T$. The variable ϕ_{igt} represents the total penalty for product $g \in G$ inflicted at supplier location $i \in I_N$ at time period $t \in T$. We need to define constraints such that ϕ_{igt} gets the value of Φ_{igt} times the value of the expected supplier result v_{igt} , if v_{igt} is bigger than 0 (δ_{igt} is 1) and 0 otherwise. In order to set ϕ_{igt} equal to Φ_{igt} times the value of the expected supplier result v_{igt} , if v_{igt} is bigger than 0 (a_{igt} is 1) and 0 otherwise, we need the following constraints:

$$v_{ig0} \leq \Gamma_{1igt} a_{ig0}, \quad \forall i \in I_N, \forall g \in G, \quad (4.14)$$

$$v_{ig0} \geq -\Gamma_{1igt}(1 - a_{ig0}), \quad \forall i \in I_N, \forall g \in G, \quad (4.15)$$

$$\Phi_{igt} v_{igt} - \Gamma_{1igt} \Phi_{igt}(1 - a_{igt}) \leq \phi_{igt}, \quad \forall i \in I_N, \forall g \in G, \forall t \in T, \quad (4.16)$$

$$\Phi_{igt} v_{igt} + \Gamma_{1igt} \Phi_{igt}(1 - a_{igt}) \geq \phi_{igt}, \quad \forall i \in I_N, \forall g \in G, \forall t \in T, \quad (4.17)$$

$$\Gamma_{1igt} \Phi_{igt} a_{igt} \geq \phi_{igt}, \quad \forall i \in I_N, \forall g \in G, \forall t \in T, \quad (4.18)$$

$$\phi_{igt} \in \mathbb{R}^+, \quad \forall i \in I_N, \forall g \in G, \forall t \in T, \quad (4.19)$$

$$a_{igt} \in \{0, 1\}, \quad \forall i \in I_N, \forall g \in G, \forall t \in T. \quad (4.20)$$

Note that constraints (4.14) and (4.15) have the same function as constraints (4.2) and (4.3), but now for $t = 0$. Furthermore, constraint (4.20) is an update of constraint (4.5), which now holds for all $t \in T$. If $\delta_{igt} = 0$, constraints (4.2) and (4.14) ensure that $v_{igt} \leq 0$. Constraints (4.14) and (4.15) ensure that if $v_{ig0} \geq 0$ then δ_{ig0} must be equal to 1. Therefore, if $v_{igt} \geq 0$ and so $\delta_{igt} = 1$, we want that ϕ_{igt} to become positive. In this case, constraints (4.16) and (4.17) ensure that, as $\delta_{igt} = 1$, $\phi_{igt} \geq \Phi_{igt} v_{igt}$ and $\phi_{igt} \leq \Phi_{igt} v_{igt}$ and so $\phi_{igt} = \Phi_{igt} v_{igt}$. In words, this means that the penalty of supplier location $i \in I_N$ at time period $t \in T$ is the penalty per ton of product $g \in G$ picked up less than the contract stipulated times the expected supplier result.

In the case that $v_{igt} \leq 0$ and thus $a_{igt} = 0$, equation (4.18) ensures that $\phi_{igt} \geq 0$. Furthermore, constraint (4.19) (always) ensures that $\phi_{igt} \leq 0$. Therefore, if $a_{igt} = 0$, then $\phi_{igt} = 0$.

Similarly, we linearise the function θ_{igt} . We define the big M constant Γ_{2jgt} :

$$\Gamma_{2jgt} = D_{jg\bar{t}} + \sum_{t^* \in T^*} D_{jgt^*} \quad \forall j \in I_M, \forall g \in G, \forall t \in T. \quad (4.21)$$

To do this, we introduce Θ_{jgt} , the penalty of under delivering one ton of product $g \in G$ in customer location $j \in I_M$ at time period $t \in T$. Also, we introduce the variable θ_{jgt} , which represents the total penalty inflicted at supplier location $j \in I_M$ at time period $t \in T$ for product $g \in G$, and η_{jgt} , a binary variable that has value 1 if the expected customer result, $l_{jgt} \left(= D_{jg\bar{t}} + \left(\sum_{t \in [0, t]} \sum_{i \in I_N} \sum_{z=1}^Z D_{jgt} - Q_{ij} \sigma_{ijgt}^z P_{ijgt}^z \right) \right)$, is positive, and 0 otherwise. Now the linearisation works in the same way as the linearisation for ϕ_{igt} , replacing v_{igt} with l_{jgt} , a_{igt} with η_{jgt} , Θ_{igt} with Φ_{jgt} and Γ_{1igt} with Γ_{2jgt} . The constraints associated with this linearisation are given at the end of the chapter.

Having linearised the penalty functions (θ_{jgt} and ϕ_{igt}) in our model, the total expected penalties are a linear function now:

$$\omega_2 = \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \theta_{jgt} + \sum_{t \in T} \sum_{g \in G} \sum_{i \in I_N} \phi_{igt}. \quad (4.22)$$

4.3. Reducing the number of variables and constraints

The number of variables that would arise from the different locations, products and number of trucks is high. It is expected that a lot of these variables are unnecessary. Therefore, we analyse the introduced variables to determine which variables are necessary. Furthermore, we analyse which constraints are necessary. In Subsection 4.3.1, we construct sets that only contain necessary variables. We use these sets to construct the necessary constraints in Subsection 4.3.2.

4.3.1. Variables

In this subsection, we determine which variables are necessary, and what conditions make the variables necessary to be included in the model. From these conditions, we construct sets that only contain necessary variables.

Consider the routes (i, j) of our model. For some routes (i, j) , there is no probability that a scheduled truck is realised on this route. Therefore, we do not generate the variables that have this route (i, j) connection. These routes have no probability $P_{ijt}^z > 0$ for a $t \in T$ and $z > 0$. Thus, we only consider routes for which $\sum_{z=1}^Z P_{ijt}^z > 0$. All these routes (i, j) are collected in the set $\Lambda = \{(i, j) : i \in I_N, j \in I_M, \sum_{z=1}^Z P_{ijt}^z > 0\}$. Variables that depend on supplier location $i \in I_N$ and customer location $j \in I_M$ are only generated if $(i, j) \in \Lambda$.

Similarly, some routes (i, j) have no probability $p_{ijt}^z > 0$ for a certain $z > 0$. Adding more trucks on the route (i, j) than this z only yields more penalties, as this truck has no probability of generating revenue, and does lead to supplier penalties (as a contract needs to be purchased for the amount of product $g \in G$ that is scheduled to be picked up). Therefore, we do not generate the x_{ijt}^z variables that have a combination of (i, j, t, z) such that route (i, j) for time period t yields $p_{ijt}^z = 0$. To do this, we define number Z^{ijt} . Z^{ijt} is the number such that for all $z \leq Z$ for which $p_{ijt}^z > 0$ we have $z \leq Z^{ijt}$, and if $p_{ijt}^z = 0$ then $z > Z^{ijt}$. Variables that depend on $z \leq Z$ are only generated if $z \leq Z^{ijt}$.

Furthermore, it is possible that a supplier location $i \in I_N$ does not have all products $g \in G$ available at any time period $t \in T$. If a supplier location $i \in I_N$ does not have product $g \in G$ available at any time period $t \in T$, then variables that contain this combination (e.g., the expected supplier result v_{igt}) do not need to be included in our model. To do this, we define the set $\Xi = \{(i, g) : i \in I_N, g \in G, \text{supplier location } i \in I_N \text{ has product } g \in G \text{ available } \exists t \in T\}$. Variables that depend on supplier location $i \in I_N$ and product $g \in G$ are only generated if $(i, g) \in \Xi$.

Moreover, it is possible that a customer location $j \in I_M$ does not have a demand for product $g \in G$ for any time period $t \in T$. If this is the case, the variables related to this (j, g, t) combination are not needed in our model. To exclude these variables, we define the set $\Psi = \{(j, g) : j \in I_M, g \in G, D_{igt} > 0 \exists t \in T\}$. Variables that depend on both customer $j \in I_M$ and product $g \in G$ are only generated if $(j, g) \in \Psi$.

Lastly, we need a set which contains all the possible combinations of (i, j, g, t) of the variable σ_{ijgt}^z . This set contains all possible combinations of the three previously mentioned sets. We define a set Π that contains all combinations of routes (i, j) , available products at supplier locations (i, g) and customer demand (j, g) . Thus, $\Pi = \{(i, j, g) : (i, j) \in \Lambda, (i, g) \in \Xi, (j, g) \in \Psi\}$.

4.3.2. Constraints

Now that we have constructed sets that only contain necessary variables for our model, we need to adjust our constraints. Similar to the variables, some constraints are unnecessary to be generated as the variables in these constraints are not necessary.

For example, take constraint (3.14):

$$\sum_{z=1}^Z x_{ijt}^z \leq 1, \quad \forall i \in I_N, \forall j \in I_M, \forall t \in T. \quad (3.14)$$

Constraint (3.14) does not need to be generated for $\forall i \in I_N, \forall j \in I_M$ and $\forall t \in T$. This is due to the fact that not all variables x_{ijt}^z are generated. Specifically, some routes (i, j) have been deleted, and for some routes (i, j) and time period t , (high) values of $z < Z$ have been deleted. Therefore, the constraint should only hold for routes (i, j) that exist. These routes (i, j) are collected in the set $\Lambda = \{(i, j) : i \in I_N, j \in I_M, \sum_{z=1}^Z P_{ijt}^z > 0\}$. For the routes $(i, j) \in \Lambda$, the original number Z has been replaced by the number Z^{ijt} . The number Z^{ijt} is defined (in the previous subsection) such that all $z \leq Z$ for which $p_{ijt}^z > 0$ we have $z \leq Z^{ijt}$, and if $p_{ijt}^z = 0$

then $z > Z^{ijt}$. Thus, we redefine constraint 3.14 as follows:

$$\sum_{z=1}^{Z^{ijt}} x_{ijt}^z \leq 1, \quad \forall (i, j) \in \Lambda, \forall t \in T. \quad (4.23)$$

Similarly, we defined sets which include only combinations of (i, g) and (j, g) if they are necessary, namely Ξ and Ψ . The set Π contains all combinations of routes (i, j) , available products at supplier locations (i, g) and customer demand (j, g) . Now, we can properly redefine the constraints.

For example, constraint (3.15) becomes:

$$\sum_{g \in G} \sigma_{ijgt}^z = x_{ijt}^z z, \quad \forall (i, j) \in \Lambda, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijt}]. \quad (4.24)$$

Note that we still write $\sum_{g \in G}$ in constraint 4.24, even though this is not strictly correct. All the sets regarding summation are unaltered. This is done to preserve the readability of the summations.

Now, our complete, linearised model is as follows:

$$\max_{\sigma_{ijgt}^z, y_{it_1 t_2}} \sum_{i \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^Z (R_{jgt} - M_{ij} - C_{igt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z - \sum_{i \in T} \sum_{g \in G} \sum_{j \in I_M} \theta_{jgt} - \sum_{i \in T} \sum_{g \in G} \sum_{i \in I_N} \phi_{igt} \quad (4.25)$$

such that

$$l_{jgt} = D_{jg\bar{t}} + \sum_{t \in [0, t]} \sum_{i \in I_N} \sum_{z=1}^{Z^{ijt}} (D_{jgt} - Q_{ij} \sigma_{ijgt}^z P_{ijt}^z), \quad \forall (j, g) \in \Psi \forall t \in T, \quad (3.2)$$

$$v_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1=\bar{t}}} \bar{y}_{igt_1 0} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \sigma_{ijg0}^z P_{ij0}^z + v_{ig\bar{t}}, \quad \forall (i, g) \in \Xi, \quad (3.4)$$

$$v_{igt} = b_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \sigma_{ijgt}^z P_{ijt}^z, \quad \forall (i, g) \in \Xi \forall t \geq 1 \in T, \quad (4.7)$$

$$h_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1=\bar{t}}} \bar{y}_{igt_1 0} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \sigma_{ijg0}^z P_{ij0}^z + v_{ig\bar{t}}, \quad \forall (i, g) \in \Xi, \quad (3.11)$$

$$h_{igt} = f_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \sigma_{ijgt}^z P_{ijt}^z, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.8)$$

$$h_{igt} + \sum_{\substack{t_2 > t \\ t_1 \leq t}} y_{igt_1 t_2} + \sum_{\substack{t_2 > t \\ t_1 \leq t}} \bar{y}_{igt_1 t_2} \geq -\argmin_{j \in I_M} Q_{ij} + \epsilon, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (3.12)$$

$$\sum_{z=1}^{Z^{ijt}} \sum_{i \in I_N} \sum_{t^* \in T^*} Q_{ij} \sigma_{ijgt^*}^z \leq \sum_{t^* \in T^*} D_{jgt^*} + \argmin_{i \in I_N} Q_{ij} - \epsilon, \quad \forall (j, g) \in \Psi \forall t \in T. \quad (3.13)$$

$$\sum_{z=1}^{Z^{ijt}} x_{ijt}^z \leq 1, \quad \forall (i, j) \in \Lambda, \forall t \in T, \quad (3.14)$$

$$\sum_{g \in G} \sigma_{ijgt}^z = x_{ijt}^z z, \quad \forall (i, j) \in \Lambda, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijt}], \quad (3.15)$$

$$b_{igt} \leq v_{ig(t-1)}, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.2)$$

$$b_{igt} \geq -\Gamma_{1igt} (1 - a_{igt}), \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.3)$$

$$b_{igt} \geq v_{ig(t-1)} - \Gamma_{1igt} a_{igt}, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.4)$$

$$f_{igt} \leq h_{ig(t-1)}, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.9)$$

$$f_{igt} \geq -\Gamma_{1igt} (1 - e_{igt}), \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.10)$$

$$f_{igt} \geq h_{ig(t-1)} - \Gamma_{1igt} e_{igt}, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.11)$$

$$v_{ig0} \leq \Gamma_{1igt} a_{ig0}, \quad \forall (i, g) \in \Xi, \quad (4.14)$$

$$v_{ig0} \geq -\Gamma_{1igt} (1 - a_{ig0}), \quad \forall (i, g) \in \Xi, \quad (4.15)$$

$$\Phi_{igt} v_{igt} - \Gamma_{1igt} \Phi_{igt} (1 - a_{igt}) \leq \phi_{igt}, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (4.16)$$

$$\Phi_{igt} v_{igt} + \Gamma_{1igt} \Phi_{igt} (1 - a_{igt}) \geq \phi_{igt}, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (4.17)$$

$$\Gamma_{1igt} \Phi_{igt} a_{igt} \geq \phi_{igt}, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (4.18)$$

$$l_{jgt} \geq -\Gamma_{2jgt} (1 - \eta_{jgt}), \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.26)$$

$$l_{jgt} \leq \Gamma_{2jgt} \eta_{jgt}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.27)$$

$$\Theta_{jgt} l_{jgt} - \Gamma_{2jgt} \Theta_{jgt} (1 - \eta_{jgt}) \leq \theta_{jgt}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.28)$$

$$\Theta_{jgt} l_{jgt} + \Gamma_{2jgt} \Theta_{jgt} (1 - \eta_{jgt}) \geq \theta_{jgt}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.29)$$

$$\Gamma_{2jgt} \Theta_{jgt} \eta_{jgt} \geq \theta_{jgt}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.30)$$

$$\sigma_{ijgt}^z \in \mathbb{N}^+, \quad \forall (i, j, g) \in \Pi, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijgt}], \quad (3.16)$$

$$x_{ijgt}^z \in \{0, 1\}, \quad \forall (i, j) \in \Lambda, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijgt}], \quad (3.17)$$

$$y_{igt_1 t_2} \in \mathbb{R}^+, \quad \forall (i, g) \in \Xi, \forall t_1 \in T, \forall t_2 \geq t_1 \in T, \quad (3.18)$$

$$l_{jgt} \in \mathbb{R}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (3.19)$$

$$v_{igt}, h_{igt} \in \mathbb{R}, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (3.20)$$

$$\phi_{igt} \in \mathbb{R}^+, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (4.19)$$

$$\eta_{jgt} \in \{0, 1\}, \quad \forall (j, g) \in \Psi, \forall t \in T, \quad (4.31)$$

$$a_{igt}, e_{igt} \in \{0, 1\}, \quad \forall (i, g) \in \Xi, \forall t \in T, \quad (4.32)$$

$$b_{igt}, f_{igt} \in \mathbb{R}^-, \quad \forall (i, g) \in \Xi, \forall t \geq 1 \in T, \quad (4.33)$$

$$\theta_{jgt} \in \mathbb{R}^+, \quad \forall (j, g) \in \Psi, \forall t \in T. \quad (4.34)$$

5

Local Search Method

The time needed to obtain a good solution in terms of contracts to be purchased is essential for [REDACTED]. Therefore, they are interested in the trade off between optimality and run time. [REDACTED] requires a run time of less than an hour, preferably as short as possible. Thus, we turn to a metaheuristic. Specifically, a simulated annealing method is proposed.

The proposed simulated annealing method is described in Section 5.1. The initial solution for this method is described in Section 5.2. The different neighbourhoods used by this method are detailed in Section 5.3. In Section 5.4, an alteration to improve the method is described.

5.1. Simulated annealing

In this section, we describe the proposed metaheuristic for our problem. A metaheuristic is a general solution method (i.e., not problem specific) that provides strategies to guide the search process to efficiently explore the solution space to find a near optimal solution. The metaheuristic used in this thesis is simulated annealing. Simulated annealing is a stochastic algorithm which searches for a global optimum and avoids getting stuck in local, nonglobal optimum [4]. It is based on the heating and cooling process of materials.

In order to start the metaheuristic, we need an initial solution s . In Section 5.2, the construction of the initial solution is described. For each iteration of the metaheuristic, a random neighbour s' is generated from the current solution s . A neighbour is a (feasible) solution obtained by performing an operation on the current solution. In Section 5.3, the construction of a neighbour is described.

If the neighbour has a better objective function value, the neighbour solution is accepted and becomes the new current solution. If the neighbour has a worse objective function value, the neighbour will be accepted with a certain probability, which depends on the current temperature and the difference between the objective function value ω' of the generated neighbour s' and the objective function value ω of the current solution s . This probability is the Boltzmann probability:

$$P(\omega', \omega, T_k) = e^{-\frac{|\omega' - \omega|}{T_k}} \quad (5.1)$$

where T_k is the temperature during iteration k . The cooling schedule $g(T)$ yields the way in which temperature T_k is decreased. This cooling schedule depends on k and M_{\max} . This function is called after M_{\max} iterations of the algorithm. A decrease in temperature T_k leads to a decrease in the probability of acceptance. Furthermore, the probability of acceptance depends on the quality of the neighbour solution is, i.e., the worse the neighbour solution, the lower the probability of acceptance. Due to the possibility of accepting worse solutions, the algorithm can escape a local optimum. The algorithm stops after one hour has passed. This is done in order to accommodate the wishes [REDACTED] detailed in Section 1.2.

To determine the start temperature, we use the acceptance ratio a_0 of solutions, which is defined as follows

by Laarhoven and Aarts [13]:

$$T_0 = \frac{\Delta^+}{\ln(a_0^{-1})} \quad (5.2)$$

where Δ^+ is the average increase of increased objective function values based on preliminary experiments. a_0 is the acceptance ratio of solutions, i.e., the number of solutions accepted divided by the total number of solutions. According to Laarhoven and Aarts [13], a_0 should be in the interval $[0.4, 0.5]$.

Furthermore, we use the geometric cooling schedule, which is defined as follows:

$$g(T) = \alpha T \quad (5.3)$$

where $\alpha \in (0, 1)$. This update of the temperature is done after a predetermined M_{\max} iterations. Previous research has shown that α should be between 0.5 and 0.99 [13].

The implemented simulated annealing algorithm, based on the algorithm in Talbi [18], is given in Algorithm 1, which is found in Appendix A.

5.2. Initial solution

To start the simulated annealing method, we need an initial solution. In this section, we construct two different initial solutions. Both initial solutions are constructed using the solution obtained by solving a mixed integer linear program. This mixed integer linear program is described in Subsection 5.2.1. The construction of the initial solution from this mixed integer linear program is described in Subsections 5.2.2 and 5.2.3.

5.2.1. Mixed integer linear program initial solution

In this subsection, we construct two initial solutions by using the solution obtained by solving a mixed integer linear program with an objective function based on the total expected profit excluding the total expected penalties. The total expected profit excluding the total expected penalties is defined as

$$\max_{\sigma_{ijgt}^z} \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^{Z^{ijt}} \left((R_{jgt} - M_{ij} - C_{igt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z \right). \quad (5.4)$$

In this paragraph, a small recap concerning the parameters in equation (5.4) is given. The revenue per delivered ton of product $g \in G$ to customer location $j \in I_M$ at time period $t \in T$ is R_{jgt} . The transportation costs per ton of product from supplier location $i \in I_N$ to customer location $j \in I_M$ is M_{ij} . The costs of a product $g \in G$ at supplier location $i \in I_N$ at time period $t \in T$ is C_{igt} . The capacity of the truck from supplier location $i \in I_N$ to customer location $j \in I_M$ is Q_{ij} and P_{ijt}^z is the cumulative probability that $z \in [1, 2, \dots, Z^{ijt}]$ trucks can traverse the route from $i \in I_N$ to $j \in I_M$ at time period $t \in T$. The truck scheduling variables are σ_{ijgt} .

If we only consider the expected profit (equation (5.4)) as the objective function, the number of variables in the model is decreased significantly, as we only need to consider the truck scheduling variables (σ_{ijgt}^z and x_{ijgt}^z) for the expected profit. Thus, we do not consider the supplier and customer results and penalties, which are included in the program defined in Chapter 3. Furthermore, this model does not need all the constraints and extra variables concerning the linearisation of the suppliers result and the penalties. It is expected that this simplified model has a much shorter run time.

However, only maximising the expected profit does necessarily deliver as much demand as possible, as some products may only be possible to transport with a loss. This does not necessarily mean that the original model described in Chapter 3 would not schedule trucks on that route, as avoiding customer penalties may yield that the truck is profitable (compared to not scheduling the truck). We say a truck is profitable if adding the truck yields a higher value for the original objective value defined in Chapter 3 than not adding the truck. For our initial solution, we do not want to schedule trucks that are not profitable. Therefore, we maximise the number of possibly profitable trucks. We define two objective functions based on (5.4). One objective function yields a higher number of potentially profitable scheduled trucks. The initial solution with that objective function limits the options of the neighbour search for the first change from the initial solution. It is expected that the simulated annealing method starting from this initial solution is more promising than for the other

initial solution. The other objective function is a better estimation of the objective function of the original model described in Chapter 3.

For the first objective function, we add the customer penalty Θ_{jgt} to the expected profit. This increases the profitability of the scheduled truck compared to the original model, as a small over delivery is rewarded extra, whereas that is not the case in the original model. Using this objective value leads to at least as many scheduled trucks as the original program from Chapter 3, as the supplier penalties are neglected.

$$\hat{\omega}_1 = \max_{\sigma_{ijgt}^z} \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^{Z^{ijt}} \left((R_{jgt} + \Theta_{jgt}) - M_{ij} - C_{igt} \right) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z \quad (5.5)$$

A scheduled truck on a route (i, j) at time period $t \in T$ may yield an expected supplier penalty of $\Phi_{igt}(1 - P_{ijt}^z)$, if there are z trucks planned on that route. For our second objective function, we subtract that from our expected profit. In practice, this value may be a little lower, as a contract for one truck on a route may be ϵ big, as such a contract gives the possibility to schedule a truck with product truck and such a contract generally does not yield a penalty. It is expected that adding an estimation of the expected supplier penalty yields a closer approximation of the original objective value function than the objective function defined in equation (5.5).

$$\hat{\omega}_2 = \max_{\sigma_{ijgt}^z} \sum_{t \in T} \sum_{g \in G} \sum_{j \in I_M} \sum_{i \in I_N} \sum_{z=1}^{Z^{ijt}} \left(((R_{jgt} + \Theta_{jgt}) - M_{ij} - C_{igt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z - (1 - P_{ijt}) Q_{ij} \sigma_{ijgt}^z P_{ijt}^z \Phi_{igt} \right) \quad (5.6)$$

The constraints that are associated with this mixed integer linear program are similar to the constraints concerning the demand and customer results of the mixed integer linear program defined in Chapter 3. In fact, all constraints in this section were originally constructed in Chapter 3. We give a small recap concerning these constraints. Furthermore, we need all the sets defined in Chapter 4. In that chapter, sets such as Π were constructed. Π is the set which contains all the necessary (i, j) , (i, g) and (j, g) combinations, i.e., the route (i, j) has a truck availability, supplier location $i \in I_N$ has the product $g \in G$ available and the customer $j \in I_M$ has a demand for product $g \in G$ in at least one of the time periods $t \in T$. Constructing such sets ensures that not too many variables are generated for the mixed integer linear program. Constraint (3.13) ensures that customer demand is not over delivered for any time period. Again, we need the set $T^* = [0, t]$:

$$\sum_{z=1}^{Z^{ijt}} \sum_{i \in I_N} \sum_{t^* \in T^*} Q_{ij} \sigma_{ijgt^*}^z \leq \sum_{t^* \in T^*} D_{jgt^*} + \arg \min_{i \in I_N} (Q_{ij}) - \epsilon, \quad \forall (j, g) \in \Psi, \forall t \in T. \quad (3.13)$$

Furthermore, constraint (3.14) ensures that at most one number of trucks, z , is scheduled on route (i, j) at time period $t \in T$. Constraint (3.15) ensures all trucks are distributed over the products $g \in G$. Constraint (3.16) ensures that σ_{ijgt}^z is a nonnegative integer number, while constraint (3.17) ensures that x_{ijt}^z is a binary number.

$$\sum_{z=1}^{Z^{ijt}} x_{ijt}^z \leq 1, \quad \forall (i, j) \in \Lambda, \quad (3.14)$$

$$\sum_{g \in G} \sigma_{ijgt}^z = x_{ijt}^z z, \quad \forall (i, j) \in \Lambda, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijt}], \quad (3.15)$$

$$\sigma_{ijgt}^z \in \mathbb{N}^+, \quad \forall (i, j, g) \in \Pi, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijt}], \quad (3.16)$$

$$x_{ijt}^z \in \{0, 1\}, \quad \forall (i, j) \in \Lambda, \forall t \in T, \forall z \in [1, 2, \dots, Z^{ijt}]. \quad (3.17)$$

This program yields the truck scheduling variables σ_{ijgt}^z . From this, we need to convert the results from this mixed integer linear program to a feasible solution for our original model. Thus, we need to add contracts and calculate the corresponding objective value for the objective function.

5.2.2. Mixed integer linear program contracts

To calculate the contracts which yield the least amount of penalties for the scheduled trucks obtained by solving the mixed integer linear program of the previous subsection, we construct another mixed integer

linear program. In this program, we minimise the penalties of the contracts for one supplier location and product. Thus, with θ_{igt} as the penalty at time period $t \in T$ for the given supplier location $i \in I_N$ and product $g \in G$, the objective function is:

$$\max - \sum_{t \in T} (\theta_{igt})$$

This model computes the optimal contracts to purchase based on the scheduled trucks, $\hat{\sigma}_{ijgt}^z$, of the initial solution. All constraints in this section were originally constructed in Chapters 3 and 4. We give a small recap concerning these constraints. The contracts may lead to a penalty of the supplier. The penalty of the supplier, θ_{igt} , is the expected supplier result, v_{igt} , multiplied by the related penalty Θ_{igt} . The maximum supplier result, h_{igt} , ensures that not more trucks are scheduled from a supplier location than the contract allows. The expected supplier result was originally defined (in Chapter 3) as $v_{igt} = \min(v_{igt}, 0) + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^Z Q_{ij} \hat{\sigma}_{ijgt}^z P_{ijt}^z$, but these are linearised in the equations below (h_{igt} had a similar nonlinearity).

$$v_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1=\bar{t}}} \bar{y}_{igt_1 0} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \hat{\sigma}_{ijg0}^z P_{ij0}^z + v_{ig\bar{t}}, \quad (3.4)$$

$$h_{ig0} = y_{ig00} + \sum_{\substack{t_1=0 \\ \cup t_1=\bar{t}}} \bar{y}_{igt_1 0} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \hat{\sigma}_{ijg0}^z P_{ij0}^z + v_{ig\bar{t}}, \quad (3.11)$$

$$v_{igt} = b_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \hat{\sigma}_{ijgt}^z P_{ijt}^z, \quad \forall t \geq 1 \in T, \quad (4.7)$$

$$h_{igt} = f_{igt} + \sum_{t_1 \leq t} y_{igt_1 t} + \sum_{t_1 \leq t} \bar{y}_{igt_1 t} - \sum_{j \in I_M} \sum_{z=1}^{Z^{ijt}} Q_{ij} \hat{\sigma}_{ijgt}^z P_{ijt}^z, \quad \forall t \geq 1 \in T, \quad (4.8)$$

$$b_{igt} \leq v_{ig(t-1)}, \quad \forall t \geq 1 \in T, \quad (4.2)$$

$$b_{igt} \geq -\Gamma_{1igt} (1 - a_{igt}), \quad \forall t \geq 1 \in T, \quad (4.3)$$

$$b_{igt} \geq v_{ig(t-1)} - \Gamma_{1igt} a_{igt}, \quad \forall t \geq 1 \in T, \quad (4.4)$$

$$f_{igt} \leq h_{ig(t-1)}, \quad \forall t \geq 1 \in T, \quad (4.9)$$

$$f_{igt} \geq -\Gamma_{1igt} (1 - e_{igt}), \quad \forall t \geq 1 \in T, \quad (4.10)$$

$$f_{igt} \geq h_{ig(t-1)} - \Gamma_{1igt} e_{igt}, \quad \forall t \geq 1 \in T, \quad (4.11)$$

The expected supplier result yields a penalty, but only if the expected supplier result is positive. The supplier penalty for time period $t \in T$, θ_{igt} , is equal to $\min(v_{igt} \Theta_{igt}, 0)$, where Θ_{igt} is the penalty per ton of product of the contract not picked up. This penalty is linearised in the equations below. More details about the linearisation are given in Section 4.2.

$$v_{igt} \leq \Gamma_{1igt} \delta_{igt}, \quad \forall t \in T, \quad (4.14)$$

$$v_{igt} \geq -\Gamma_{1igt} (1 - \delta_{igt}), \quad \forall t \geq 1 \in T, \quad (4.15)$$

$$\Phi_{igt} v_{igt} - \Gamma_{1igt} \Phi_{igt} (1 - \delta_{igt}) \leq \phi_{igt}, \quad \forall t \in T, \quad (4.16)$$

$$\Phi_{igt} v_{igt} + \Gamma_{1igt} \Phi_{igt} (1 - \delta_{igt}) \geq \phi_{igt}, \quad \forall t \in T, \quad (4.17)$$

$$\Gamma_{1igt} \Phi_{igt} \delta_{igt} \geq \phi_{igt}, \quad \forall t \in T, \quad (4.18)$$

Finally, we have the variable declarations. As in the model described in Chapter 4, the constraints (3.18) and (4.19) ensure that the contracts and penalties are real, nonnegative numbers. Furthermore, constraint (4.32) ensures that a_{igt} , e_{igt} and δ_{igt} are binary numbers. Constraint (3.20) ensures that the supplier results (both the expected and maximum) are real numbers. Finally, constraint (4.33) ensures that the variables v_{igt} and

h_{igt} are real, nonpositive numbers.

$$y_{igt_1 t_2} \in \mathbb{R}^+, \quad \forall t_1 \in T, \forall t_2 \geq t_1 \in T, \quad (3.18)$$

$$v_{igt}, h_{igt} \in \mathbb{R}, \quad \forall t \in T, \quad (3.20)$$

$$\theta_{igt} \in \mathbb{R}^+, \quad \forall t \in T. \quad (4.19)$$

$$a_{igt}, e_{igt}, \delta_{igt} \in \{0, 1\}, \quad \forall t \in T, \quad (4.32)$$

$$b_{igt}, f_{igt} \in \mathbb{R}^-, \quad \forall t \in T, \quad (4.33)$$

This program yields the contract scheduling variables $y_{igt_1 t_2}$ and the expected supplier penalty and result for each time period $t \in T$: ϕ_{igt} and v_{igt} . By running this problem for all supplier product combinations which have a truck scheduled (i.e., $\forall (i, g) : \exists \sigma_{ijgt}^z > 0$), we obtain the corresponding (and optimal) contracts for our initial solution.

5.2.3. Objective value

Now that we have a feasible initial solution (i.e., scheduled trucks and corresponding contracts), we need to calculate the corresponding objective value. This is done by Algorithm 2, which is described in the following paragraphs. The algorithm itself is found in Appendix A.

In Step 1, we initialise the values of the expected supplier result and penalty, the expected customer result and penalty, and the number of scheduled trucks. Initially, there are no trucks scheduled. However, there may be some realised contracts which lead to an expected supplier penalty. Any realised contract ending at time period $t \in T$ yields a penalty of $\bar{y}_{igt_1 t} \Phi_{igt}$, as there are no trucks scheduled yet. In Step 2, we calculate the first objective value, the sum of the expected supplier and customer penalties. Then, we run the initial solution formulated in Section 5.2.1 (with either the objective function ω_1 or ω_2) and update the scheduled trucks resulting from the initial solution.

Then, for each route (i, j) on which trucks are scheduled, i.e., $\forall (i, j)$ s.t. $\exists \sigma_{ijgt}^z > 0$, we select the individual trucks with their product $g \in G$, total number of trucks on the route, $z \leq Z$, and time period $t \in T$. In Step 7, we can update the objective value, ω , with the expected profit of the planned trucks $\left(\sigma_{ijgt}^z Q_{ij} P_{ijt}^z (R_{jgt} - C_{igt} - M_{ijt}) \right)$. The previous value of the expected profit of the truck was zero, so no value needs to be subtracted from the objective value ω .

Furthermore, we need to recalculate the expected customer results and penalties. When trucks are scheduled in the first time period, the total demand for the following time periods is also decreased. This is due to the fact that under delivered demand in time period $t \in T$ needs to be delivered in the next time period, $t + 1 \in T$. Therefore, if there are some truckloads delivered in time period $t \in T$, this not only decreases the expected customer penalty of time period $t \in T$, but also of all the time periods after $t \in T$. So, for all $t' \in \{t' \in T : t' \geq t\}$, we decrease the expected customer result for time period $t' \in T$, $l_{jgt'}$ with the expected number of tons of the scheduled trucks in time period $t \in T$, $Q_{ij} \sigma_{ijgt}^z P_{ijt}^z$ (Step 9). Then, we update the objective value, by adding the old expected customer penalty, $\theta_{jgt'}$, and subtracting the new expected customer penalty, $\max(\sum_{j \in I_M} \sum_{g \in G} \sum_{t' \in \{t' \in T : t' \geq t\}} \sum (\Theta_{jgt'} l_{jgt'}), 0)$. Finally, we update the expected customer penalty. The steps described in this paragraph are called the CUSTOMER PROCEDURE.

Lastly, we need to recalculate the expected supplier results and penalties. To do this, we first run the program described in Section 5.2.2. From this, we obtain the new contracts to schedule, $y_{igt_1 t_2}$, which we update in Step 13. Then, for all $t \in T$, we update the new expected supplier penalty ϕ_{igt} (Step 15), calculate the new objective value (using the old expected supplier result, v_{igt} , Step 16), and then update the expected supplier penalty, v_{igt} . The steps described in this paragraph are called the SUPPLIER PROCEDURE.

5.3. Neighbours

Now that we have constructed an initial solution, we need to construct a method for finding a neighbour of this solution. For a neighbour of the initial (or more broadly: current) solution, we only look for changes in the number of scheduled trucks. A neighbour of a current solution will differ one change from the current solution. It is assumed that the contracts can be computed quickly from the scheduled trucks.

There are many possible changes to be made in the number of scheduled trucks. In the following subsections, we detail the possible changes one scheduled truck can make. For all these changes, we assume that the customer may always receive one truckload more of any product (so adding a truck does not result in an infeasible solution).

5.3.1. Adding or deleting a truck

It is possible that a scheduled truck is added (if $z < Z$) or removed (if $z > 0$) on a route (i, j) at a time period $t \in T$, as in Figure 5.1. It may be beneficial to delete a scheduled truck, if a truck has, for instance, a low probability, p_{ijt}^z , or a high purchase price of the product $g \in G$ it is carrying, C_{igt} . This scheduled truck could lead to more expected supplier penalties than that the truck is expected to earn. Adding a scheduled truck is beneficial, if the expected earnings is higher than the expected penalties it causes. Essentially, every other change (in the next subsections) is based on this concept of adding or deleting trucks.

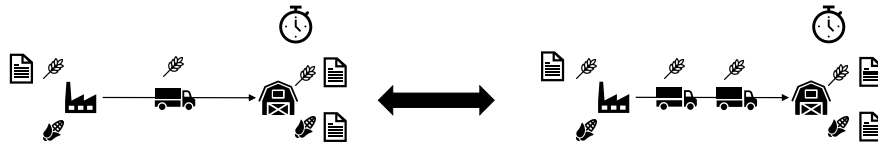


Figure 5.1: Adding or deleting a scheduled truck

To add a scheduled truck on the route, the number of scheduled trucks $c \leq z$ planned for product $g \in G$ on route (i, j) at time period $t \in T$ is increased to $c + 1$ and the total number of trucks on this route $z < Z$ is also increased to $z + 1 \leq Z$:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^{(z+1)} = c + 1 \quad (5.7)$$

For all the other products $g^* \neq g \in G$ that had $c^* \leq z$ scheduled trucks planned on route (i, j) at time period $t \in T$, c^* stays the same for the new total number of scheduled trucks planned for the route, $z + 1$:

$$\sigma_{ijg^*t}^z = c^* \Rightarrow \sigma_{ijg^*t}^{(z+1)} = c^* \quad (5.8)$$

Furthermore, for all products $g \in G$, the previous value σ_{ijgt}^z needs to be set to zero, in order to keep the solution feasible:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^z = 0 \quad (5.9)$$

Removing a truck works similarly.

5.3.2. Changing the product of the truck

It is also possible that a scheduled truck changes the product it is carrying, as in Figure 5.2. Some routes may be more profitable for some products than others, i.e., if $R_{jgt} - C_{igt} < R_{jg't} - C_{ig't}$ for some products $g \in G$ and $g' \neq g \in G$. It may also be that the customer penalty for product $g' \neq g \in G$, $\Theta_{jg't}$, is relatively high compared to the customer penalty for the other product $g \in G$ (or vice versa considering supplier penalties). For this action to be possible, the supplier location $i \in I_N$ needs to carry both products and the customer location $j \in I_M$ needs to have a demand for both products.

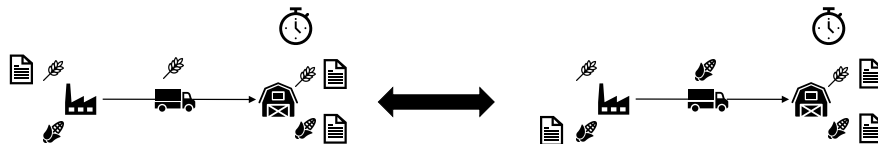


Figure 5.2: Changing the product of a scheduled truck

In order to change the product a scheduled truck is carrying from $g \in G$ to $g' \neq g \in G$, the number of scheduled trucks $c \leq z$ planned for product $g \in G$ on route (i, j) at time period $t \in T$ is decreased to $c - 1$ and the number of scheduled trucks $c' \leq z'$ for product $g' \neq g \in G$ on route (i, j) at time period $t \in T$ is increased to $c' + 1$:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^z = c - 1 \quad \wedge \quad \sigma_{ijg't}^{z'} = c' \Rightarrow \sigma_{ijg't}^{z'} = c' + 1 \quad (5.10)$$

For all products $g^* \neq (g \vee g') \in G$ that had $c^* \leq z$ scheduled trucks planned on route (i, j) for time t , c^* stays the same:

$$\sigma_{ijg^*t}^z = c^* \Rightarrow \sigma_{ijg^*t}^z = c^*$$

No values need to be set to zero.

5.3.3. Moving the truck forward or backward in time

Furthermore, it is possible for a scheduled truck to be scheduled a time period later (unless it is scheduled on the last time period in the planning horizon) or earlier (unless it is scheduled on the first time period in the planning horizon), as in Figure 5.3. It may be beneficial to change the time period $t \in T$ of a scheduled truck due to a low probability (p_{ijgt}^z), a low selling price (R_{ijgt}), a high purchase (C_{ijgt}) or high penalties (Θ_{ijgt} or Φ_{ijgt}) compared to the time period before, $t - 1 \in T$, or after it, $t + 1 \in T$.

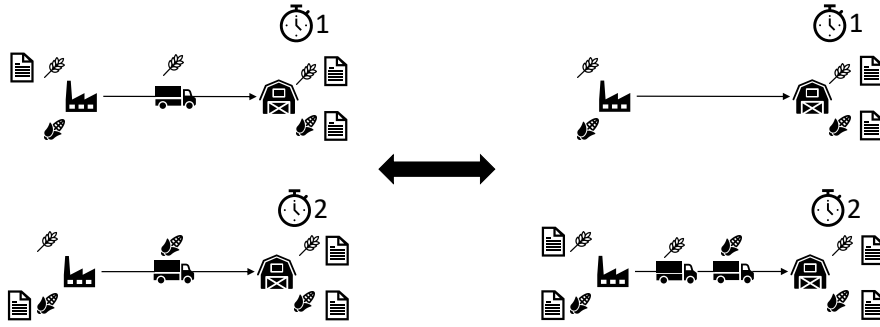


Figure 5.3: Moving a scheduling truck forward or backward in time

To move a scheduled truck to a later time period, the number of scheduled trucks $c \leq z$ planned for product $g \in G$ on route (i, j) for time $t \in T$ is decreased to $c - 1$ (and the total number of scheduled trucks is decreased from $z < Z$ to $z - 1 \geq 0$) and the number of scheduled trucks $c' \leq z'$ for product $g \in G$ on route (i, j) for time $t + 1 \in T$ is increased to $c' + 1$ (and the total number of scheduled trucks is increased from $z' < Z$ to $z' + 1 \leq Z$):

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^{(z-1)} = c - 1 \quad \wedge \quad \sigma_{ijg(t+1)}^{z'} = c' \Rightarrow \sigma_{ijg(t+1)}^{(z'+1)} = c' + 1 \quad (5.11)$$

For all other products $g^* \neq g \in G$ that had $c^* \leq z - 1$ and $c^{**} \leq z'$ scheduled trucks planned on route (i, j) for time $t \in T$ and time $t + 1 \in T$ respectively, c^* and c^{**} stay the same for the new number of scheduled trucks planned for the route (i, j) , $z - 1 \geq 0$ and $z' + 1 \leq Z$:

$$\sigma_{ijg^*t}^z = c^* \Rightarrow \sigma_{ijg^*t}^{(z-1)} = c^* \quad \wedge \quad \sigma_{ijg^*(t+1)}^{z'} = c^{**} \Rightarrow \sigma_{ijg^*(t+1)}^{(z'+1)} = c^{**} \quad (5.12)$$

Furthermore, for all products $g \in G$, the previous value of σ_{ijgt}^z , c , needs to be set to zero, for both time t and $t + 1$, in order to keep the solution feasible:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^z = 0 \quad \wedge \quad \sigma_{ijg(t+1)}^{z'} = c' \Rightarrow \sigma_{ijg(t+1)}^{z'} = 0 \quad (5.13)$$

Moving a scheduled truck to an earlier time period works similarly.

5.3.4. Changing supplier location of the truck

It is also possible that a scheduled truck changes its supplier location, as in Figure 5.4. This may be beneficial for the objective function value if the new supplier location, $i' \neq i \in I_N$, has a relatively low purchase price for

the product $g \in G$, $C_{i'gt}$, or supplier penalty, $\Phi_{i'gt}$. It may also be that the probability of the next scheduled truck for $i' \neq i \in I_N$, $p_{i'jt}^{(z'+1)}$, is relatively high compared to the probability of the highest numbered scheduled truck for supplier location $i \in I_N$, p_{ijt}^z . This move is only possible if the new supplier location $i' \neq i \in I_N$ carries the product $g \in G$.

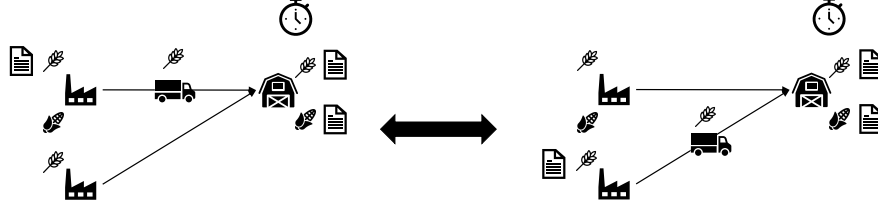


Figure 5.4: Changing the supplier of a scheduled truck

In order to move a scheduled truck from a supplier location $i \in I_N$ to $i' \in I_N$, the number of scheduled trucks $c \leq z$ planned for product $g \in G$ on route (i, j) at time period $t \in T$ is decreased to $c - 1$ (and the total number of scheduled trucks is decreased from $z < Z$ to $z - 1 \geq 0$), while the number of scheduled trucks $c' \leq z'$ for product g on route (i', j) at time period $t \in T$ is increased to $c' + 1$ (and the total number of scheduled trucks is increased from z' to $z' + 1$):

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^{(z-1)} = c - 1 \quad \wedge \quad \sigma_{i'jgt}^{z'} = c' \Rightarrow \sigma_{i'jgt}^{(z'+1)} = c' + 1 \quad (5.14)$$

For all other products $g^* \neq g \in G$ that had $c^* \leq z - 1$ and $c^{**} \leq z'$ scheduled trucks planned on route (i, j) and (i', j) respectively, at time period $t \in T$, c^* and c^{**} stay the same for the new number of scheduled trucks planned for the route (i, j) , $z - 1$, and route (i', j) , $z' + 1$:

$$\sigma_{ijg^*t}^z = c^* \Rightarrow \sigma_{ijg^*t}^{z-1} = c^* \quad \wedge \quad \sigma_{i'jg^*t}^{z'} = c^{**} \Rightarrow \sigma_{i'jg^*t}^{z'+1} = c^{**} \quad (5.15)$$

Furthermore, for all products $g \in G$ and both routes (i, j) and (i', j) , the previous value of σ_{ijgt}^z and $\sigma_{i'jgt}^{z'}$ needs to be set to zero at time period $t \in T$, in order to keep the solution feasible:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^z = 0 \quad \wedge \quad \sigma_{i'jgt}^{z'} = c' \Rightarrow \sigma_{i'jgt}^{z'} = 0 \quad (5.16)$$

5.3.5. Changing customer location of the truck

Similar to changing the supplier location of a truck, we can change the customer location of a truck, as in Figure 5.5. This may be beneficial for the expected profit if the new customer location, $j' \neq j \in I_M$, has a relatively high revenue price for the product $g \in G$, $R_{j'gt}$, or customer penalty, $\Theta_{j'gt}$. It may also be that the probability of the next scheduled truck for $j' \neq j \in I_M$, $p_{ij't}^{(z'+1)}$, is relatively high compared to the probability of the highest numbered scheduled truck for customer location $j \in I_M$, $p_{ij't}^z$. This move is only possible if the new customer location $j' \neq j \in I_M$ has a demand the product $g \in G$.

In order to move a scheduled truck from a customer location $j \in I_M$ to $j' \neq j \in I_M$, the number of scheduled trucks $c \leq z$ planned for product $g \in G$ on route (i, j) at time period $t \in T$ is decreased to $c - 1$ (and the total number of scheduled trucks is decreased from $z < Z$ to $z - 1 \geq 0$), while the number of scheduled trucks $c' \leq z'$ for product g on route (i, j') at time period t is increased to $c' + 1$ (and the total number of scheduled trucks is increased from z' to $z' + 1$):

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^{(z-1)} = c - 1 \quad \wedge \quad \sigma_{ij'gt}^{z'} = c' \Rightarrow \sigma_{ij'gt}^{(z'+1)} = c' + 1 \quad (5.17)$$

For all other products $g^* \neq g \in G$ that had $c^* \leq z - 1$ and $c^{**} \leq z'$ scheduled trucks planned on route (i, j) and (i, j') respectively, at time period $t \in T$, c^* and c^{**} stay the same for the new number of scheduled trucks planned for the route (i, j) , $z - 1$, and route (i, j') , $z' + 1$:

$$\sigma_{ijg^*t}^z = c^* \Rightarrow \sigma_{ijg^*t}^{z-1} = c^* \quad \wedge \quad \sigma_{ij'g^*t}^{z'} = c^{**} \Rightarrow \sigma_{ij'g^*t}^{z'+1} = c^{**} \quad (5.18)$$

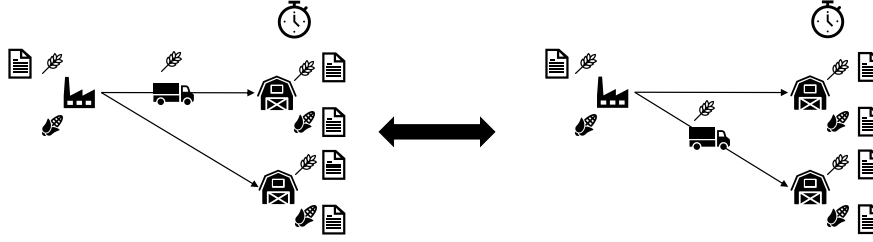


Figure 5.5: Changing the customer of a scheduled truck

Furthermore, for all products $g \in G$ and both routes (i, j) and (i, j') , the previous value of σ_{ijgt}^z and $\sigma_{ij'gt}^{z'}$ needs to be set to zero at time period $t \in T$, in order to keep the solution feasible:

$$\sigma_{ijgt}^z = c \Rightarrow \sigma_{ijgt}^z = 0 \quad \wedge \quad \sigma_{ij'gt}^{z'} = c' \Rightarrow \sigma_{ij'gt}^{z'} = 0 \quad (5.19)$$

5.3.6. Neighbour selection

Now that we have stated the different changes that can be made to a variable in the current solution, we need an algorithm to select a change for a variable in the current solution. In other words, we construct an algorithm to select a variable from the current solution and make a feasible change to the variable. From this change, a neighbour (i.e., a feasible solution with the corresponding objective function value) can be constructed. This algorithm (Algorithm 3) is called the neighbour selection algorithm and is given in Appendix A.

In Step 1, we initialise four lists: *PossibleChanges*, *PossibleProducts*, *PossibleSuppliers* and *PossibleCustomers*. How these lists are filled is described in later paragraphs. In *PossibleChanges*, every neighbourhood which leads to at least one feasible neighbour solution (as described in Subsections 5.3.1 to 5.3.5) is stored. In *PossibleProducts*, the possible products $g' \neq g \in G$ (if any) which the truck can also carry are stored. In *PossibleSuppliers*, the possible supplier locations $i' \neq i \in I_N$ (if any) from which the truck can also originate are stored. In *PossibleCustomers*, the possible customer locations $j' \neq j \in I_M$ (if any) to which the truck can also deliver are stored. In Step 2, we randomly select a combination of $i \in I_N$, $j \in I_M$ and $g \in G$ by randomly picking from the set Π (which contains all necessary combinations of (i, j, g)). Furthermore, we select a random time period $t \in T$.

Steps 4 to 7 describe the FIND Z PROCEDURE, which we use more often in this algorithm. First, we check if there are no trucks scheduled on the input (of the procedure) route (i, j) at the input time period $t \in T$. If there are no trucks scheduled, we set $z = 0$. If there are scheduled trucks on the input route (i, j) at the input time period $t \in T$, then we set $z \in T$ to that number of scheduled trucks.

In Step 8 to 39, we check which changes are available for the selected (i, j, g, t, z) combination. First, we check whether the demand of customer location $j \in I_M$ for product $g \in G$ allows for another scheduled truck for all time periods $t^* \geq t \in T$, as adding a scheduled truck on the route (i, j) affects all these time periods. Furthermore, we check whether it is possible to schedule a higher number of trucks, i.e., the probability of adding an extra truck being available is not zero. If both of these conditions are true, then we add the option to add a truck to the list *PossibleChanges*.

All the other changes are only possible if $\sigma_{ijgt}^z > 0$, as these changes depend on deleting or substituting one scheduled truck of σ_{ijgt}^z . If $\sigma_{ijgt}^z > 0$, then we have the option to delete a scheduled truck. For the other options, we need to check other conditions as well.

If the selected time period $t \in T$ is not the latest time period and it is possible to schedule a higher number of trucks at time period $t + 1 \in T$, then we can add the option to move the scheduled truck to a later time to the list *PossibleChanges*.

If the selected time period $t \in T$ is not the earliest time period, the demand until time period $t - 1 \in T$ allows for another scheduled truck in time period $t - 1 \in T$, and it is possible to schedule a higher number of trucks

at the time period $t - 1 \in T$, then we can add the option to move the scheduled truck to an earlier time period to the list *PossibleChanges*.

In order to check whether it is an option to change the product $g \in G$ the scheduled truck is carrying, we add all the products $g' \neq g \in G$ for which the customer location $j \in I_M$ has a remaining demand $\forall t^* \geq t \in T$ to the list *PossibleProducts*. If this list is not empty, then we add the option to change the product of the scheduled truck to the list *PossibleChanges*.

Similarly, we consider each supplier location $i' \neq i \in I_N$ which has the product $g \in G$ available at time period $t \in T$, i.e., $C_{igt} < \epsilon 10000$. Then, we check if it is possible to add another scheduled truck to the route (i', j) (by checking the condition $z' < Z^{i'j^t}$). If this is true, then we add the supplier location $i' \neq i \in I_N$ to the list *PossibleSuppliers*. If this list is not empty, then we add the option to change the supplier location of the scheduled truck to the list *PossibleChanges*.

Lastly, if a customer $j' \neq j \in I_M$ has a remaining demand for the product $g \in G$ for all time periods $t^* \geq t \in T$ and it is possible to add another scheduled truck on the route (i, j') (by obtaining z' and checking $z' < Z^{ij't}$), then we add the customer location $j' \neq j \in I_M$ to the list *PossibleCustomers*. If this list is not empty, then we add the option to change the customer location of the scheduled truck to the list *PossibleChanges*.

In Step 40, we pick a random *option* from the list *PossibleChanges*. Depending on which option is picked, we yield the output of ζ_{ijgt}^z and/or $\zeta_{ijgt}^{(z+1)}$. If the *option* is to add a truck, the output $\zeta_{ijgt}^{(z+1)}$ is given the value of $\sigma_{ijgt}^z + 1$. This step of the algorithm is based on the first equation (equation 5.7) of Subsection 5.3.1. All the changed output variables are based on the first equation of each corresponding subsection (from Subsection 5.3.1 to Subsection 5.3.5). It is also possible that both variables are not given any value. For example, if an (i, j, g, t, z) combination is picked with $z = 0$ (such that no truck can be deleted), but the customer location $j \in I_M$ has no remaining demand for product $g \in G$ at time period $t \in T$, then none of the options are added to the empty list *PossibleChanges*. If that is the case, the algorithm gives no values to the output variables. If an output variable is not given a value in the algorithm, then the variable will be interpreted as a *null* value for the processing algorithm. The processing of potential output is done in Algorithm 4.

5.3.7. Processing algorithm

Now that we have stated the different changes that can be made to the current solution and how we can obtain the change from our current solution, we need to construct a feasible solution from these possible changes. To do this, Algorithm 4 is used. In this algorithm, the values of the current solution, i.e., $\omega, v_{igt}, \phi_{igt}, l_{jgt}, \theta_{jgt}, \sigma_{ijgt}^z$, are updated to the corresponding values for the change made, yielding the (feasible) neighbour. The algorithm itself is found in Appendix A.

We start the algorithm with the ζ_{ijgt}^z . The following steps described in this paragraph will be named the NEIGHBOURHOOD PROCEDURE. This updated variable ζ_{ijgt}^z may have a variable which value has been decreased compared to the solution it originates from. If that is the case, we know that there is at least one $\sigma_{ijgt'}^{z'} \neq \text{null}$ (with $z' > z$) in the current solution. If there is just one variable, then $g' = g$, otherwise there are more products on the route. For all those products $g \in G$, we update the objective value ω by first deleting the original contribution of $\sigma_{ijgt'}^{z'}$ of the expected profit, then setting the value of the new $\sigma_{ijgt'}^z$, and then adding the new contribution of $\sigma_{ijgt'}^z$ to the expected profit. Finally, we set the value of the old $\sigma_{ijgt'}^z$ to zero, unless no change was made to the product of the scheduled truck. In that case, we did not change number of scheduled trucks z , so no values need to be set to zero. The last two steps in the NEIGHBOURHOOD PROCEDURE are the previously defined CUSTOMER PROCEDURE and SUPPLIER PROCEDURE.

The next step in the algorithm is to repeat the NEIGHBOURHOOD PROCEDURE for ζ_{ijgt}^z , unless no change was made to the product of the scheduled truck. In that case, we did not change number of scheduled trucks z and the processing already happened. However, there is not necessarily a nonzero value of $\sigma_{ijgt}^{(z-1)}$. If $\zeta_{ijgt}^z = 1$, we know that $\sigma_{ijgt}^{(z-1)} = 0$, and thus, ζ_{ijgt}^z was not processed in the NEIGHBOURHOOD PROCEDURE and some extra steps need to be taken. Therefore, if $\zeta_{ijgt}^z = 1$, we need to update the value of σ_{ijgt}^z , add the expected profit ω of σ_{ijgt}^z , set the contract y_{igt} to ϵ , update the expected supplier penalty θ_{jgt} , and subtract this from

the objective value ω . We end the algorithm with the CUSTOMER PROCEDURE.

5.4. Changing the step size of the changes

Now that we have stated the process of obtaining a neighbour, we make an addition to the simulated annealing method. In this section, we change the step increase the step size of a change. For example, instead of deleting one scheduled truck from a route, we delete five scheduled trucks. In this example, the step size of a change is set to five.

Note that the step size of a change is a maximum, i.e., the step size taking in a change can be lower. For example, if there are only three scheduled trucks on a route, it is not possible to delete five scheduled trucks from this route. Therefore, the actual step size depends on both the input step size of a change, and the number of scheduled trucks the change can make without the solution becoming infeasible.

With this addition, our simulated annealing method is completed. This method and the exact solution method are tested in Chapter 8. However, to test these methods, we first need to construct a test instance. This is done in Chapters 6 and 7.

Problem Data

In Chapters 4 and 5, we constructed methods to solve our problem. To test these methods, we construct a test instance. To do this, all data from ██████████ related to the problem is described and analysed in this chapter. From this analysis, it follows that some data will be disregarded for our model, as it increases the problem size and complexity of the problem too much. From the remaining data, a test instance is constructed.

In Section 6.1, the available data is described. The different locations and routes are analysed in Section 6.2. The transportation costs and capacity the routes are described in Section 6.3. Then, different aspects of the sold and purchased contracts are analysed in Section 6.4. The different products and their prices are discussed in Section 6.5.

Furthermore, the input that is used for the results in Chapter 8 is described in Section 6.6. Finally, the results of the method to reduce the number of variables and constraints are described in Section 6.7 (the method itself is described in Section 4.3).

6.1. Data description

[illegible]

(b) (7)(C), (b) (7)(D)

[illegible]

_____ €

_____. _____

_____ €
_____. _____.

[illegible][illegible][illegible][illegible]

Now that we have described the available data, we can start with the analysis. We start with an analysis of the different locations and routes.

6.2. Locations and routes

[illegible][illegible]



Figure 6.1: Delivered orders (all goods) per month per year

[REDACTED]

[REDACTED]

[REDACTED]

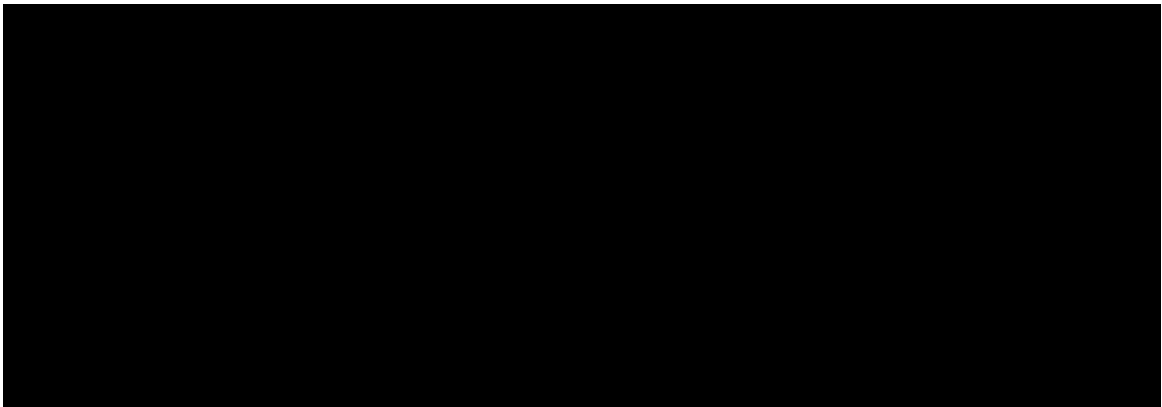


Figure 6.2: Delivered truckloads (all products) per month per country in 2017

[REDACTED]

[REDACTED]

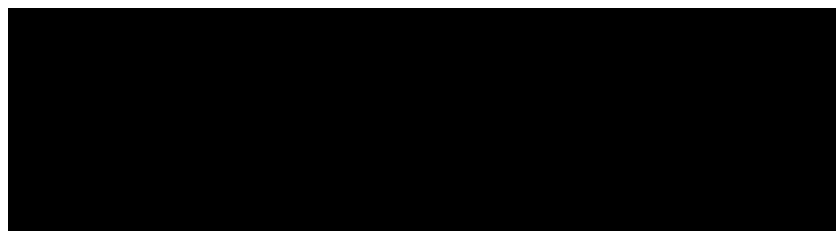


Table 6.1: Number of supplier location in 2015-2017 depending on location indicator and number of pick ups

Now that we have analysed the locations of the supplier and customer location, we move on to analysing the transportation costs and capacities between the supplier and customer locations.

6.3. Transportation costs and capacity

In this section, we analyse the different transportation costs and capacities for the routes that the trucks can be scheduled on.

[REDACTED]

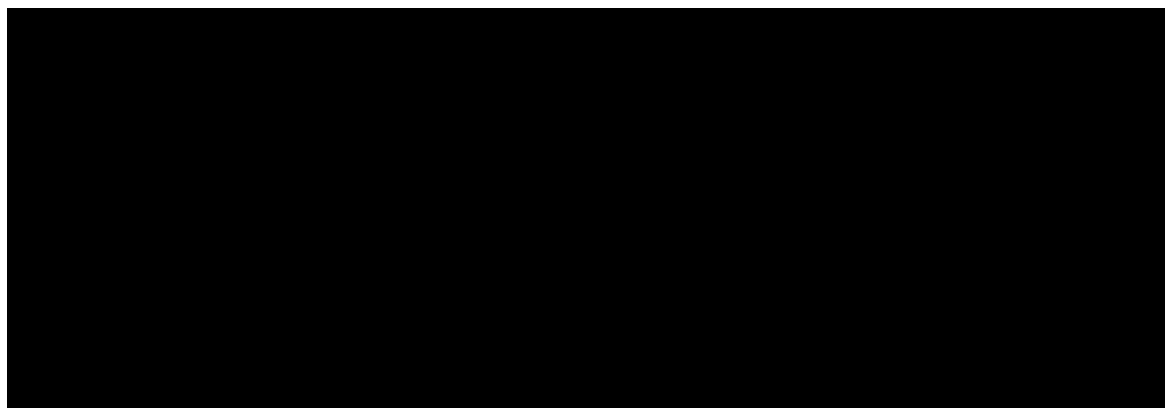


Figure 6.3: Picked up truckloads (all goods) per month per country in 2017



Figure 6.4: Postal zones in France

[illegible][illegible]

Table 6.2: The average transportation price, number of orders and the average (Avg.) and maximum (Max.) truckload weight between supplier and customer countries from 2015-2017

[REDACTED]

Now that we have analysed the data considering transportation, we move on to the data regarding contracts.

6.4. Contracts and lateness

In this section, we analyse the contracts [REDACTED] purchased and sold. We analyse different aspects, such as number of contracts, purchase/sell date and lateness.

[REDACTED]

[REDACTED]

[REDACTED]

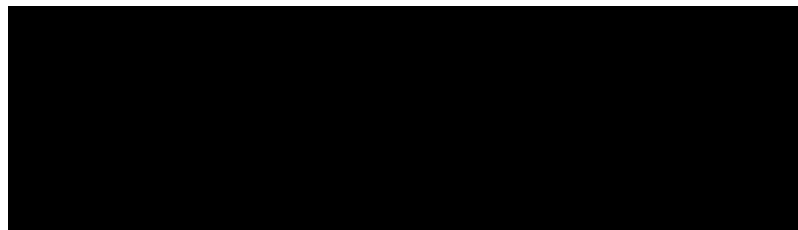


Table 6.3: Lateness of orders

[REDACTED]

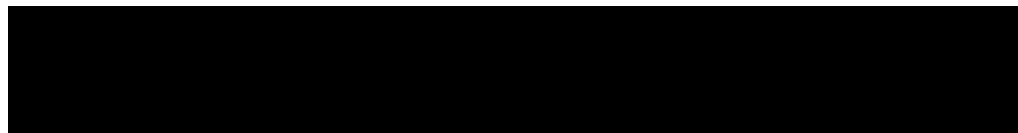


Table 6.4: Percentage of contracts purchased based on the number of days before the contract starts

[REDACTED]

Now that we have analysed the different aspects of contracts, we move on to analysing the different products [REDACTED] has traded in.

Table 6.5: Percentage of contracts sold based on the number of days before the contract starts

6.5. Products and prices

In this section, we analyse the different products [REDACTED] imports. [REDACTED] [REDACTED] [REDACTED]
[REDACTED]

[REDACTED]

[REDACTED] [REDACTED] [REDACTED] [REDACTED]. [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED]. [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED], [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED].

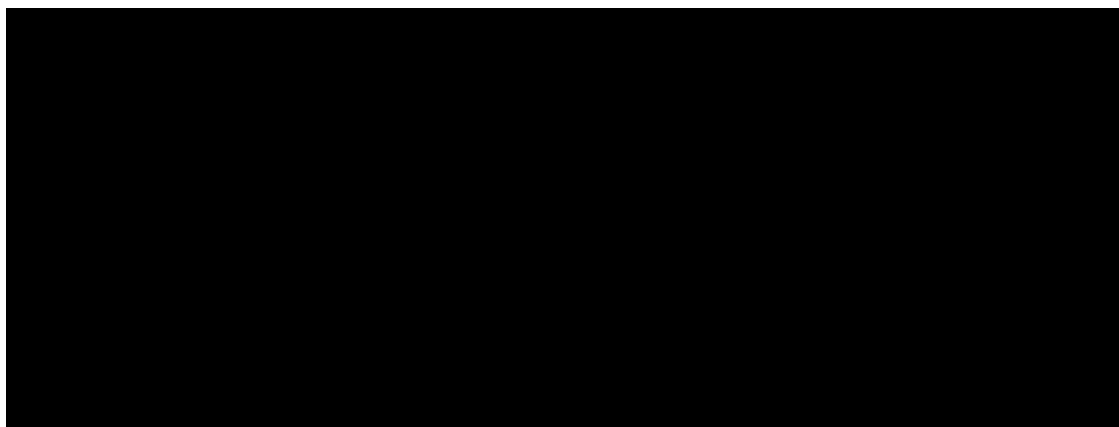
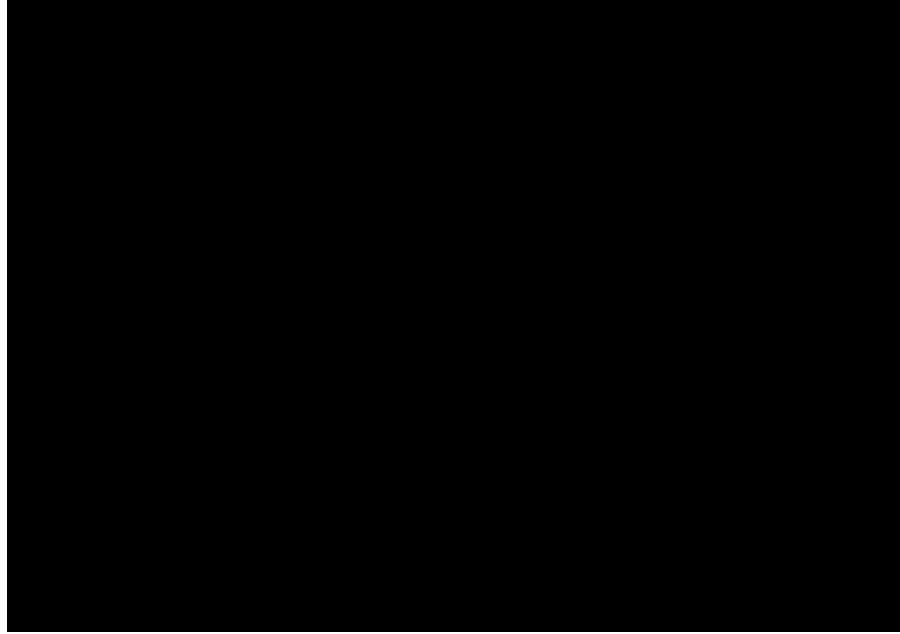


Figure 6.5: The price difference between selling and purchasing price per product and the average transport price

[illegible]

(b) (7)(C), (b) (7)(D)



Now that we have analysed the different types of data, we can define the input for our model.

6.6. Defining the test instance

To generate results for the solution methods defined in Chapters 4 and 5, we need to define the input for our model. In this section, we give the input of the test instance, obtained by using the data described in the previous sections. The discussions regarding which data to use for the input are given in the previous sections.

wants to have a sourcing strategy for at least 6 months in the future. We will optimise the first six months of 2018 as a test instance. A time period represents a month. Thus, T contains the months January 2018 up to and including June 2018. For the supplier and customer location indicators, we have taken every supplier and customer postal zones that has had more than 50 deliveries or pickups between 2015 and 2017, as discussed in Section 6.2 In these three years, used .

All 23 products are included in the calculations. Furthermore, we need to calculate Z^{ijt} for each route (i, j) and time period $t \in T$. Originally, Z was defined as follows in Section 3.2:

$$Z = \max_{\substack{t \in T \\ i \in I_N \\ j \in I_M}} \left(\frac{\sum_{g \in G} D_{jgt}}{Q_{ij}} \right).$$

This yielded the maximum number of trucks that could be scheduled on a route (i, j) for a time period $t \in T$: 350. Z^{ijt} is defined as the number such that for all $z \leq Z$ for which $p_{ijt}^z > 0$ we have $z \leq Z^{ijt}$, and if $p_{ijt}^z = 0$ then $z > Z^{ijt}$ in Section. The average value of Z^{ijt} is 3.1.

The demand of customer locations $j \in I_M$ for the product $g \in G$ at time period $t \in T$, D_{jgt} , is the total number of tons delivered to customer location $j \in I_M$ of product $g \in G$ for the corresponding month of 2018, i.e., we have modelled the demand of the customer in each month as the number of tons of products they received that month.

The average transportation price of the last three years between a supplier location $i \in I_N$ and customer location $j \in I_M$ is the value of M_{ij} , as explained in Section 6.3. The capacity for the trucks, Q_{ij} , is the maximum number of tons a truck is allowed to transport according to trade legislation.

Furthermore, the price regions described in Section 6.5 are implemented. If a supplier location $i \in I_N$ does not have a known purchase price C_{igt} for product $g \in G$ at time period $t \in T$, the purchase price C_{igt} is the average price of that product at that time period of the price region. If the price region also does not have a known purchase price the product at time period, it is assumed that this price region did not have the product $g \in G$ available. To model this unavailability, unavailable products are given a purchase price C_{igt} of €10000, i.e., a price so high such that the model does not purchase contracts in that price region.

Lastly, for the supplier and customer penalties, Θ_{jgt} and Φ_{igt} , various values are taken. As there is no consensus within concerning the values of these penalties, the model is computed for values of Θ_{jgt} and Φ_{igt} between €1 and €5. There is a consensus within that the values should be in that region. The results for the different values are given in Section 8.1.1.

Now that we have defined the input for our model, i.e., created a test instance, we give the number of (necessary) variables and constraints associated with this test instance.

6.7. Results of reducing the number of variables and constraints

Some variables and constraints are considered unnecessary, so these are excluded in the model. The necessity of variables and constraints are based on certain characteristics. These characteristics are described in Section 4.3. In this section, the number of necessary variables and constraints is given.

Using all possible variables, i.e., including unnecessary variables, yields a total number of = 522,799,200 variables for σ_{ijgt}^z . Furthermore, there are = 22,730,400 variables x_{ijt}^z . For l_{jgt} , θ_{jgt} , η_{jgt} , we have a maximum of = 11,178 variables each. For a_{igt} , e_{igt} , δ_{igt} , v_{igt} , h_{igt} and ϕ_{igt} , we have a maximum of = 18,216 variables each. For b_{igt} and f_{igt} we have a maximum of = 15,180 variables for the both. Finally, we have a maximum of = 54,648 contract variables $y_{igt_1t_2}$.

Excluding unnecessary variables as in Section 4.3, we have 290,536 variables left, instead of 545,742,258. The number of variables of σ_{ijgt}^z has decreased the most, from 522,799,200 to 184,838. An overview of the number of reduced variables is given in Table 6.6. It is expected that this reduced number of variables results a faster computing time.

Number of variables	σ_{ijgt}^z	x_{ijt}^z	l_{jgt}	v_{igt}	b_{igt}	$y_{igt_1t_2}$
Original	522,799,200	22,730,400	11,178	18,216	15,180	54,648
Reduced	184,838	74,435	3,012	1,710	1,425	10,542

Table 6.6: Original number of variables compared to reduced number of variables.

Furthermore, the method to decrease the number of constraints, as described in Section 4.3, results in 171,195 constraints, instead of 24,603,877 constraints.

In this chapter, we have analysed the data given by . From this data, we have defined the input for a test instance of our model. However, one parameter is left undefined: the truck availability. The construction of this parameter is described in the next chapter.

Truck Availability

In addition to the data described in Chapter 6, the truck availability, P_{ijt}^z , needs to be calculated to obtain a test instance for our model. This chapter describes the method used to obtain the truck availability and the subsequent results.

The truck availability is defined as follows: given a z number of scheduled trucks on the route (i, j) at time period $t \in T$, we expect $P_{ijt}^z z$ trucks of the z scheduled trucks to complete the route (i, j) at time period $t \in T$. Thus, $P_{ijt}^z z$ is a conditional expected value, i.e., the expected value of realised trucks (\bar{z}) on route (i, j) at time period $t \in T$ given that z trucks are scheduled: $P_{ijt}^z z = \mathbb{E} \left[\bar{z} | x_{ijt}^z = 1 \right]$.

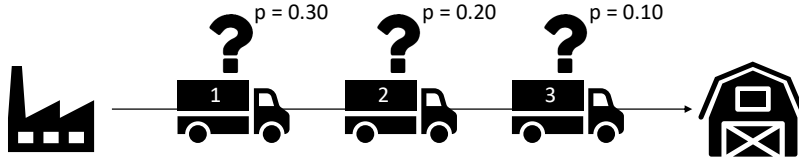


Figure 7.1: Example of truck availability

The conditional expected value (for $z^* \geq 1$, where z^* is the scheduled number of trucks; $z, z^* = 0$ does not exist) is computed as follows: $\mathbb{E} \left[\bar{z} | x_{ijt}^{z^*} \right] = \frac{\sum_{z \leq z^*} z P_{ijt}^z}{z^*}$, where P_{ijt}^z is the probability that the z^{th} truck can traverse the route (i, j) at time period t . Figure 7.1 gives an example for a supplier location $i \in I_N$, customer location $j \in I_M$ and time period $t \in T$. In this example, $p_{ijt}^1 = 0.30$, $p_{ijt}^2 = 0.20$ and $p_{ijt}^3 = 0.10$. If we were to schedule one truck, i.e., $x_{ijt}^1 = 1$, then the expected number of realised trucks, $P_{ijt}^z \cdot 1 = p_{ijt}^1 = 0.30$. Thus, $P_{ijt}^1 = \frac{p_{ijt}^1}{1} = 0.30$. If we were to schedule two trucks, i.e., $x_{ijt}^2 = 1$, then the expected number of realised trucks is $p_{ijt}^1 + p_{ijt}^2 = 0.50$. Thus, $P_{ijt}^2 \cdot 2 = p_{ijt}^1 + p_{ijt}^2 = 0.50$, which results in $P_{ijt}^2 = 0.25$. If we were to schedule three trucks, i.e., $x_{ijt}^3 = 1$, then the expected number of realised trucks, $P_{ijt}^3 \cdot 3 = p_{ijt}^1 + p_{ijt}^2 + p_{ijt}^3 = 0.60$. Thus, $P_{ijt}^3 = \frac{p_{ijt}^1 + p_{ijt}^2 + p_{ijt}^3}{3} = 0.20$.

Thus, the expected value of realised trucks (\bar{z}) on route (i, j) at time period $t \in T$ given that z^* trucks are scheduled, is the sum of all probabilities that $z \leq z^*$ trucks traverse the route (i, j) at time period t , divided by the number of scheduled trucks z^* . So, in order to define P_{ijt}^z , we need to calculate the probability p_{ijt}^z .

In Section 7.1, we define this probability using state space models. The subsequent results are presented in Section 7.2.

7.1. State space models

The only data that is available from [redacted] regarding truck availability is the number of realised trucks on route (i, j) at time period $t \in T$. The trucks that were realised on a route (i, j) for a realised time period \bar{t} can be modelled as a time series. Each entry in this time series is the number of trucks that carried truckloads from supplier location $i \in I_N$ to customer location $j \in I_M$ for a realised time period \bar{t} . From this data, we need to model a prediction interval for the number of trucks that are available on route (i, j) for all the time periods $t \in T$. This needs to be done for each route (i, j) .

In this section, we detail the state space models based on exponential smoothing. This model yields prediction intervals, i.e., the probabilities p_{ijt}^z . The state space models were proposed in Hyndman et al. [9]. First, we introduce exponential smoothing.

7.1.1. Exponential smoothing

Some of the most successful forecasting methods are based on exponential smoothing [9]. The basis of an exponential smoothing method is that it produces forecasts based on weighted averages of past observations. The weights decay exponentially: the older the observation, the lower the weight of the observation. There is a wide variety of methods based upon exponential smoothing.

Trend	Seasonal		
	N	A	M
N	$\hat{y}_{t+h t} = \ell_t$ $\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$	$\hat{y}_{t+h t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1 - \gamma)s_{t-m}$
A	$\hat{y}_{t+h t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$	$\hat{y}_{t+h t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$
A _d	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1 - \gamma)s_{t-m}$

Figure 7.2: Various kinds of exponential smoothing methods, from Hyndman et al. [9]. The highlighted method is the method we use.

An exponential smoothing method generates point forecasts, i.e., a forecast for the next point(s) in the time series. There are many different exponential smoothing methods to generate point forecasts for time series, as shown in Figure 7.2. The difference between these methods are based on whether a time series has a seasonal and/or trend component. These components are added if a time series has a season and/or trend in its data. Adding these components yield extra equations for the methods. For instance, in Figure 7.2, in the top left corner of the table, the equations are given for generating the next point (point h after the most recent observation t) forecast $\hat{y}_{t+h|t}$, given no trend or seasonal smoothing factors. In that case, the point forecast is based only on the level, ℓ_t , of the time series, based on the data points of the time series and the smoothing factor α , $0 < \alpha \leq 1$. Note that if $\alpha = 1$ the output series is just the most recent observation.

Furthermore, there are multiple variations of incorporating the seasonal and/or trend component in the forecast. For instance, one may use a method using multiplicative seasonality, additive seasonality, or no seasonality. For example, in the middle of the table of Figure 7.2, the equations are given for generating the next point forecast $\hat{y}_{t+h|t}$, given additive trend and additive seasonal smoothing (where m is the number of time periods which constitute a season). The point forecast for point h (which is h time periods ahead from the most recent time period t) is given by the current level, ℓ_t , then adding the current trend, b_t (the most recent change in level) h times and the seasonal component $s_{t+h-m(k+1)}$, where k is the integer part of $\frac{h-1}{m}$, which ensures that the estimates of the seasonal component come from the last season of the sample. Essentially,

we add up the level, trend and seasonal components of the upcoming point h to arrive at the point forecast $\hat{y}_{t+h|t}$.

If one uses a multiplicative seasonality method, one needs to be sure that all entries in the data are nonzero. As the time series of truck availability are not always nonzero, we can only use additive models.

For the trend, b_t , and seasonal component, s_t , there are also smoothing coefficients, namely β and γ , respectively. These coefficients are restricted to lie between 0 and 1, just as α , the smoothing coefficient for the level of the trend. These smoothing coefficients need to be selected, which can be done in a subjective manner, i.e., based on experience, or in a more objective manner, by estimating them from the data in the time series.

Two methods for estimating these coefficients are minimising the sum of squared errors and maximising the likelihood of the probability of the observed data from the model. A small sum of squared errors or a large likelihood is associated with a good model. For an additive error model, maximising the likelihood (assuming normally distributed errors) gives the same results as minimising the sum of squared errors [9]. The errors are the residuals of the time series, which are specified as $e_t = y_t - \hat{y}_{t|t-1}$. Hence, we wish find the smoothing parameters such that we minimise:

$$\sum_{t=1}^T (y_t - \hat{y}_{t|t-1})^2. \quad (7.1)$$

This involves a nonlinear optimisation problem. We can solve this problem with the method described by Nelder and Mead [14]. This method determines the minimum of a function with multiple variables.

7.1.2. State space methods

The statistical models described in Hyndman et al. [9] generate the same point forecasts as the corresponding exponential smoothing methods, but can also generate prediction intervals. This method is described in this subsection.

Each model is made up from two types of equations. The measurement equations describe the entries in the time series, while the state equations describe how the level, trend and season change over time.

The statistical models have an extra component next to the trend and seasonal components: the error component. Similar to these first two components, there exist two variants for the error component: models with additive errors and with multiplicative errors. The point forecasts produced by the two variants of the error models are identical, if they use the same smoothing parameter values. They will, however, generate different prediction intervals. These statistical models are called (E,T,S) models, for their **E**rror, **T**rend and **S**easonal components. The different types of (E,T,S) models are given in Hyndman et al. [9]. We have an (A,A,A) model, as we have additive error, trend and seasonal components, for which the formulas are as follows:

$$y_t = l_{t-1} + b_{t-1} + s_{t-m} + e_t \quad (7.2)$$

$$l_t = l_{t-1} + b_{t-1} + \alpha e_t \quad (7.3)$$

$$b_t = b_{t-1} + \beta e_t \quad (7.4)$$

$$s_t = s_{t-m} + \gamma e_t \quad (7.5)$$

The points forecasts that are given by the (E,T,S) models are equal to the medians of the forecast distributions. Furthermore, the forecast distributions are normal distributions if the (E,T,S) model only has additive components [9]. As our model only has additive components, we have a normal (forecast) distribution, so the medians and means of the forecast distributions are equal.

Now that we have the medians of the forecast distributions, we need the variances for the forecast distribution. The forecast variances, σ_h^2 , for our (A,A,A) model is, as given by Hyndman et al. [9]:

$$\sigma_h^2 = \sigma^2 \left(1 + (h-1) \left(\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1) \right) + \gamma k(2\alpha + \gamma + \beta m(k+1)) \right), \quad (7.6)$$

where σ^2 is the residual variance, m the number of time periods in a season, k the integer part of $\frac{h-1}{m}$, and α , β and γ the smoothing coefficients for the level, trend, and seasonal components, respectively. With the

mean and the variance of the forecast distribution, we can calculate the probability of every point forecast as follows:

$$p_{ijt}^z = \frac{1}{\sqrt{2\pi\sigma_{ijt}^2}} e^{-\frac{(z-\mu_{ijt})^2}{2\sigma_{ijt}^2}}, \quad (7.7)$$

where μ_{ijt} is the point forecast and σ_{ijt}^2 is the forecast variance for route (i, j) for time $t \in T$.

7.2. Results of the state space models

For each of the routes (i, j) between a supplier location $i \in I_N$ (with 50 truckloads or more picked up in 2015-2017) and customer location $j \in I_M$ (with 50 truckloads or more delivered in 2015-2017) for which a truck has travelled in the years 2015-2017 (as noted in Section 6.3, a lot of routes (i, j) were not used in the years 2015-2017), a time series has been constructed. It may have been possible that there was no active contract at the supplier location $i \in I_N$ or customer location $j \in I_M$. In such a case, the entry in the time series has not been set to zero, but left blank. For the minimisation of the sum of squared residuals, we do not consider (the residual of) these blank observations. We minimise the sum of squared residuals for all the routes together, not for all routes individually.

Running the Nelder-Mead method for all the routes (████ in total), yields the smoothing coefficients $\alpha = 0.26$, $\beta = 0.023$ and $\gamma = 0.97$. The minimised sum of squared residuals for all routes is 274,385 for the smoothing coefficients from the optimisation for all routes.

From these smoothing coefficients, we can generate results, as in Figure 7.3. In this figure, we see two lines: the orange line represents the realised number of trucks and the blue line represents the forecast. Furthermore, we see for the last six points a blue area. This area represents the 95% confidence interval, i.e., if an infinite amount of numbers were drawn from the modelled normal distribution (i.e., the forecast distribution), 95 percent of the drawn numbers will be in this interval.

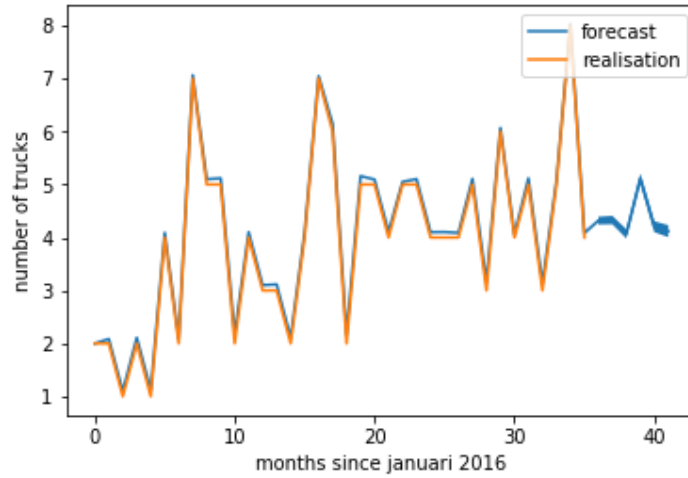


Figure 7.3: Number of trucks per month from *censored* to *censored*

The route from Figure 7.3 is one of the routes which has an accurate prediction, i.e., a low variance in the forecast distribution. However, there are also routes for which the prediction is not accurate. An example of this is given in Figure 7.4. In this figure, the forecasted number of trucks is █████ for all months, with a high variance which leads to a probability for less than zero trucks realised. As this is not possible, we treat these forecasts as zeros, i.e., the forecast for that month on that route is zero scheduled trucks.

Now that we have calculated p_{ijt}^z for every route and time in our problem, we can use these values to calculate P_{ijt}^z for every route and time in our problem, as $P_{ijt}^z = \frac{\sum_{z \leq z^*} p_{ijt}^z}{z^*}$. So, with the truck availability calculated

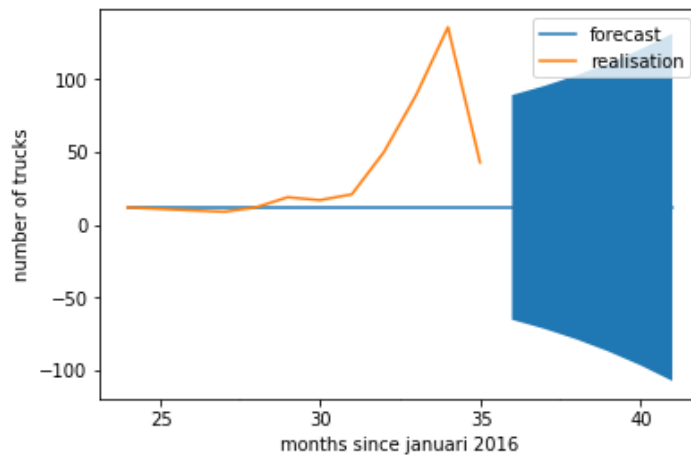
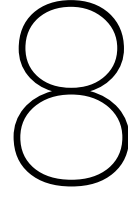


Figure 7.4: Number of trucks per month from *censored* to *censored*

in this chapter and the data from Chapter 6, we have a test instance for our model. We can use the solution methods defined in Chapters 4 and 5 on this test instance. The results of these solution methods are described in the next chapter.



Results

In this chapter, we describe the results of solving the test instance described in Chapters 6 and 7 with the exact solution method and the simulated annealing method proposed in Chapters 4 and 5. The methods are implemented in Python and all computations are done on a computer with an Intel® Xeon® Platinum 8164 CPU 2.00 GHz with 32 GB RAM memory.

We describe and analyse the results of the exact solution method and the simulated annealing method in Section 8.1 and 8.2, respectively. The results of these methods are compared in Section 8.3.

8.1. Exact solution method

The mixed integer linear program proposed Chapter 4 is solved by the CPLEX Optimizer [10] which is called by Python. The input parameters for the model are described in Section 6.6. The results of this exact solution method are given in this section.

In the following subsections, different types of results are generated. In Subsection 8.1.1, we determine the values for the penalty function. The analysis of our model with these penalty function values is done in Subsection 8.1.2. Then, we describe the results of the simulations of the model in Subsection 8.1.3. In Subsection 8.1.4, we analyse the run time of the model. Finally, we describe the results of the model constrained with a limited run time in Subsection 8.1.5.

We also ran our model with the open source COIN-OR Optimizer. However, the COIN-OR Optimizer was not able to yield a feasible solution within one hour. Therefore, using such an optimizer is not an interesting option for [REDACTED].

8.1.1. Determining the penalty function values

In this section, we compare the results (based on various values for the supplier and customer penalties) of the exact model with the results of [REDACTED] to determine the values for the supplier and customer penalties per ton. [REDACTED] determined that the penalty value per ton unfulfilled supplier contract and under delivered customer demand should be between the €1 and €5 as stated in Section 6.6. Thus, the model is run for all combinations of the different integer values for Θ_{igt} and Φ_{jgt} .

The different optimal objective function values for the model, run with different values of penalty function values, are given in Figure 8.1. In this figure, the different values for the customer penalty value (per ton under delivered customer demand) Φ_{jgt} are plotted on the x-axis, while the different supplier penalty values (per ton unfulfilled supplier contract) Θ_{igt} are represented by the different lines on the graph.

The objective function value for the trucks [REDACTED] has scheduled, i.e., if [REDACTED] were to schedule these trucks, is -390355 (given $\Theta_{igt} = 1$ and $\Phi_{jgt} = 1$). The value of the objective function value of [REDACTED] is not close to other values in Figure 8.1. Note that this value is not the actual profit [REDACTED] made in the first six

months of 2018.

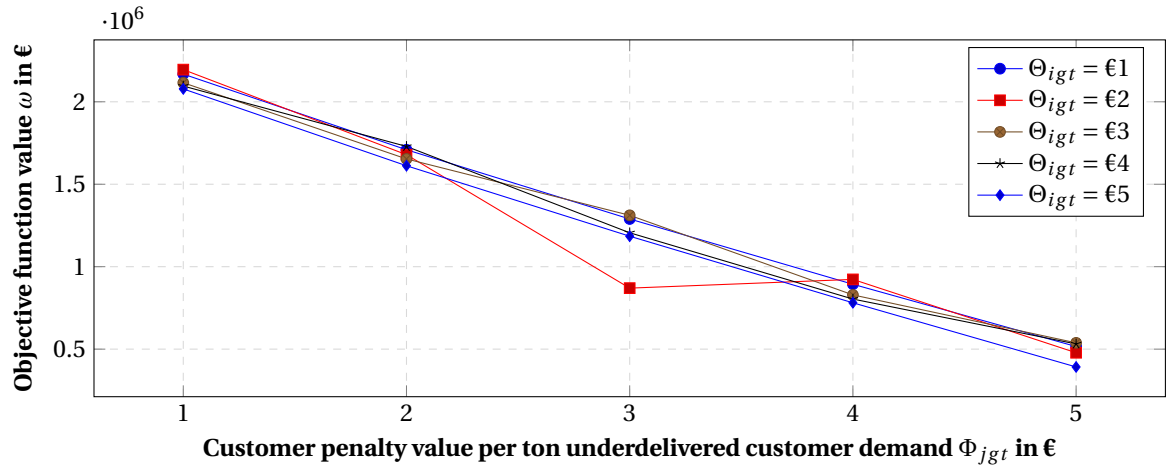


Figure 8.1: Exact solution objective function values for different penalty functions

From the figure, we can conclude that the different supplier penalties have less impact than the different customer penalties. Note that the value for $\Theta_{igt} = 2$ and $\Phi_{jgt} = 3$ does not seem to correspond with the curve of the graph, i.e., the value appears to be incorrect. It is possible something went wrong with the CPLEX Optimizer. The optimal solution for $\Theta_{igt} = 2$ and $\Phi_{jgt} = 4$ is also feasible for $\Theta_{igt} = 2$ and $\Phi_{jgt} = 3$ and results in a higher objective function value. Therefore, the solution for $\Theta_{igt} = 2$ and $\Phi_{jgt} = 4$ can not be optimal. Due to time limitations, it is not possible to run the model again for $\Theta_{igt} = 2$ and $\Phi_{jgt} = 3$.

Based on Figure 8.1, it is not clear what the values for Θ_{igt} and Φ_{jgt} should be. These values are an indicator for how important [REDACTED] deems the under delivered customer demand and unfulfilled supplier contracts, and in what ratio. Therefore, we compare the total amount of expected under delivered customer demand ($\sum_{j,g,t} l_{jgt}$) to the total amount of expected unfulfilled supplier contract ($\sum_{i,g,t} v_{igt}$). Thus, we look at the value $\frac{\sum_{j,g,t} l_{jgt}}{\sum_{i,g,t} v_{igt}}$. The idea is that a high value indicates that [REDACTED] deem customers penalties more important, a low value indicates otherwise. These values are given in Table 8.1.

		Φ_{jgt}				
		€1	€2	€3	€4	€5
Θ_{igt}	€1	14.5	12.2	10.9	9.9	9.2
	€2	97.8	15.9	75.1	69.0	10.9
	€3	22.6	18.8	94.1	14.3	65.4
	€4	26.3	117.9	18.7	16.3	98.0
	€5	31.5	24.7	20.8	18.6	15.9

Table 8.1: Ratio of $\frac{\sum_{j,g,t} l_{jgt}}{\sum_{i,g,t} v_{igt}}$, resulting from the optimal solution of our model for different values Φ_{jgt} and Θ_{igt}

The ratio of total expected under delivered customer demand to the total expected unfulfilled supplier contracts $\left(\frac{\sum_{j,g,t} l_{jgt}}{\sum_{i,g,t} v_{igt}}\right)$ for the trucks that [REDACTED] scheduled was 5.6, i.e., with the trucks [REDACTED] scheduled, it is expected [REDACTED] had 5.6 more under delivered customer demand than unfulfilled supplier contracts. The ratio from the mixed integer linear program of $\Phi_{jgt} = 5$ and $\Theta_{igt} = 1$ is closest to this value: 9.2. It is possible that the exact solution with a higher penalty function value for Φ_{jgt} gives a ratio that is even closer to the ratio of the scheduled trucks of [REDACTED]. However, [REDACTED] determined the penalty values should be between the €1 and €5, as stated in Section 6.6. Therefore, the values for the penalty functions are defined as follows: $\Phi_{jgt} = 5$ and $\Theta_{igt} = 1$. For these penalty function values, the optimal objective function value is €553435.

8.1.2. Result analysis

Now that we have established the values for the penalty values, Θ_{igt} and Φ_{jgt} , we analyse the results of the exact solution method for these values. In this subsection, we analyse the differences in expected revenue, expected costs and the expected penalties. Furthermore, we analyse the scheduled trucks and the purchased contracts.

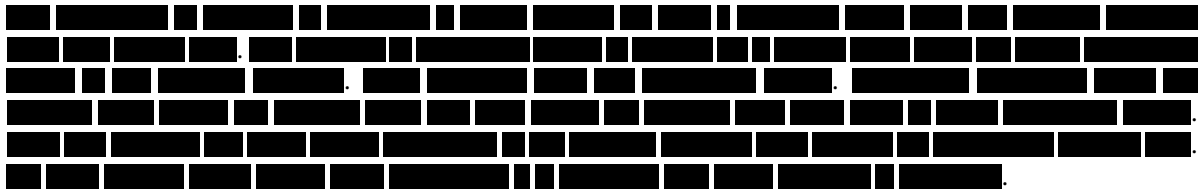
In this subsection, we compare the optimal solution of the model with the results of [REDACTED]. The results of [REDACTED] are expected values, not their realised results. As the demand is based on what [REDACTED] delivered in these months, the realised under delivered customer demand is zero for the actions of [REDACTED]. This is an unfair benchmark, and therefore, we compare the expected values of [REDACTED] actions to the expected values of our optimal solution.

In Table 8.2, the results are given for the different aspects of the objective function value, i.e., the selling result (total expected revenue minus total expected costs) and the total expected penalties. [REDACTED]

[REDACTED]



Table 8.2: Comparison between the exact solution method and company's actions





8.1.3. Simulating the time horizon

In this subsection, we give the results of simulating the realisation of trucks. Simulating the realisation of the trucks entails running our model and then realise time period $t = 0$ such that $t = 0$ becomes \bar{t} . This process is repeated until no time periods are left in the time horizon. These results gives us an indication how robust our model is, i.e., it gives an indication of how much the objective function value can fluctuate based on different realisations of the scheduled trucks.

For the realisation of time period $t = 0$, we realise the trucks for route (i, j) scheduled at $t = 0$ by drawing from the forecast distribution defined in Chapter 7. This drawn number is the realised number of trucks, unless the drawn number is higher than the z scheduled trucks. In that case, z trucks are realised. Furthermore, we realise all contracts starting on time $t = 0$. For contracts starting at later moments, they have a probability to be realised. These probabilities depend on the time $t \in T$ and are based on Table 6.4. This table indicates how many days in advance [REDACTED] purchases their contracts.

Unfortunately, due the time limitations, it was only possible to run these simulations three times. In Table 8.3 the results are shown. The three results are within 0.5% of the objective function value. On average, the

results are slightly better than the expected objective function value of the optimal solution of the exact solution method. A possible reason for this is that running a model after a realisation gives the opportunity to deal with the new situation. However, too few iterations have been done to be certain.

Case	Obj. value [€]	Obj. val. diff. [%]
Worst	██████	0.23
Average	██████	-0.14
Best	██████	-0.40

Table 8.3: Different cases for the simulations of the exact solution method

We compare these results with ten realisations of the optimal solution of the exact solution method. Thus, in this case, no new model is run for the remaining time periods after a realisation. The results are shown in Table 8.4. We see that the average of these results is worse than the expected value of the original solution. Furthermore, the worst and best case scenario are worse than in Table 8.3. A possible reason for this, is that the method used for Table 8.3 has the possibility to deal with different realisations, leading to better objective function values, though this does not explain why the best case scenario is worse than the expected value of optimal solution. This can be due to the fact that only ten realisations of the optimal solution were run.

Case	Obj. value [€]	Obj. val. diff. [%]
Worst	██████	2.30
Average	██████	1.44
Best	██████	0.59

Table 8.4: Different cases for the realisation of the optimal solution

8.1.4. Run time

In this subsection, we analyse the run time for the exact solution method. The run time consists of the time to build the model in Python (build time) and the time to solve the model in the CPLEX Optimizer (computing time).

Unfortunately, we cannot give an estimate of the run time of the problem with both the necessary and unnecessary variables. The number of variables for the problem including the unnecessary variables (that means 545,757,438 variables) was too large. It was not possible to build the problem, as building the problem resulted in memory errors. Therefore, only results of the run time of the model with the reduced number of variables (291,961 variables) are given.

So, we can give the run time of the problem with the reduced number of variables. To determine the penalty function values, the exact solution was run 25 times. The results of the run time are given in Figure 8.2. The average run time for all the different penalty function values was 4.8 hours. The average build time was 0.3 hours. Thus, the average computing time was 4.5 hours. The minimum run time was 1.0 hours and the maximum run time was 11.7 hours. Most run times were between the 2.0 and 3.0 hours. The run time for the penalty function values that were chosen ($\Phi_{jgt} = 5$ and $\Theta_{igt} = 1$) was 2.3 hours.

8.1.5. Limited run time

██████ desires a good solution within an hour. As seen from Figure 8.2, not a single of the 25 runs took less than one hour. In this subsection, we run the CPLEX Optimizer for different run times (build time plus computing time) in order to determine the trade off between optimality and run time for the exact solution method.

The results are shown in Table 8.5 and in Figure 8.3. The relative optimality gap in the table is the difference between the given solution and the tightest upper bound of the CPLEX Optimizer has found. The results of

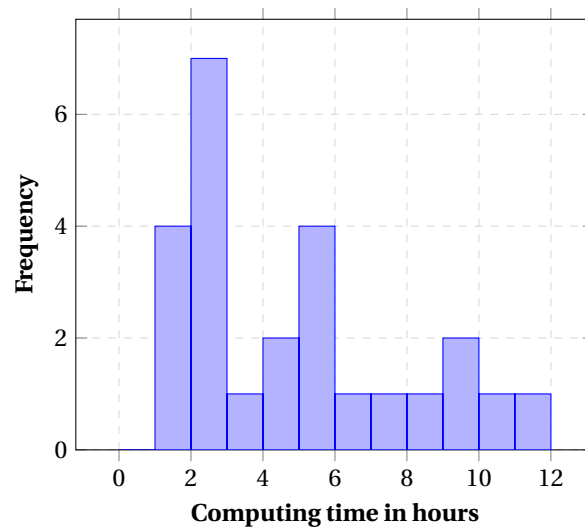


Figure 8.2: Histogram of the computing time of the exact solution

this method are also a benchmark for the simulated annealing method. The first results of CPLEX Optimizer were returned after 17.1 minutes (including 15 minutes build time). However, these are not plotted in Figure 8.3 to preserve the readability of the figure.

Limited run time [min.]	Computing time [min.]	Obj. val. [€]	Relative opt. gap [%]
17.1	2.1	<div style="width: 100%; height: 10px; background-color: black;"></div>	958.23
20.0	5.8	<div style="width: 100%; height: 10px; background-color: black;"></div>	3.41
30.0	15.6	<div style="width: 100%; height: 10px; background-color: black;"></div>	2.10
40.0	25.7	<div style="width: 100%; height: 10px; background-color: black;"></div>	1.92
50.0	35.8	<div style="width: 100%; height: 10px; background-color: black;"></div>	1.79
60.0	47.4	<div style="width: 100%; height: 10px; background-color: black;"></div>	1.78

Table 8.5: Comparison between the exact solver with different run time limits

As can be seen in the figure, the CPLEX Optimizer quickly approaches a gap of less than 3.00%. However, to converge to an solution with a gap of (less than) 0.01% (which is the default setting for the CPLEX Optimizer) takes a relative long time. An explanation for the slow convergence is the NP-hardness of (mixed) integer linear programming.

8.2. Simulated annealing

In this section, we give the results of the simulated annealing method described in Chapter 5. In each of the different subsections, we vary a parameter of the simulated annealing method to obtain a better value for that parameter.

Note that this method does not necessarily yield the best parameters for the simulated annealing. However, running the simulated annealing method for every combination of the different parameters is not possible due to time limitations.

In Subsection 8.2.1, we discuss the results of the two different initial solutions. Then, in Subsections 8.2.2, 8.2.3, and 8.2.4, we discuss the results of varying different parameters of the simulated annealing method, the starting temperature, the parameters of the cooling schedule, and the step size of a change (per iteration), respectively. Finally, in Subsection 8.2.5, we give the final parameters and results of the simulated annealing method.

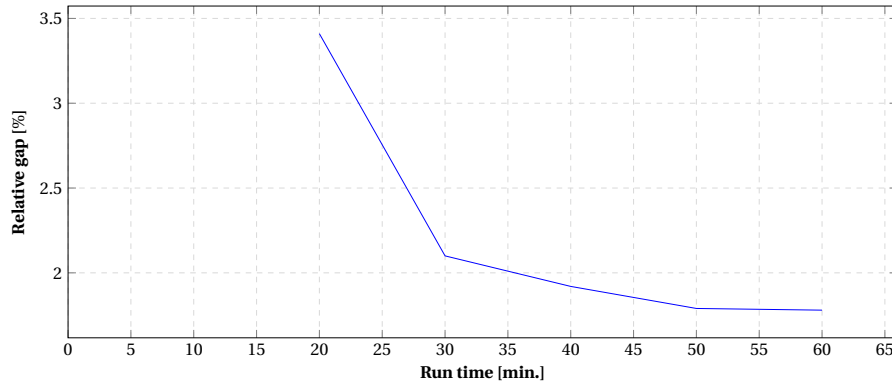


Figure 8.3: Convergence of CPLEX Optimizer

8.2.1. Initial solution

In this subsection, we give the results of the initial solution proposed in Section 5.2. In that section, two different initial solutions were described. Both initial solutions are constructed using the solution of a mixed integer linear program which does not consider the customer and supplier results, yielding a simpler model. The difference between the two initial solutions is the objective function of this mixed integer linear program. One objective function approximates the optimal objective function value of the model in Chapter 4 the best (named the approximate (approx.) initial solution), the other yields the most potentially profitable trucks (named the truck initial solution).

The results of the two initial solutions are given in Table 8.6. The run time in the table for the initial solutions consists of two parts. The first part consists of the time to build and solve the mixed integer linear program (in Table 8.6: Build time). The second part consists of the time to construct a feasible solution for the model defined in Chapter 4 from the solution of the mixed integer linear program of the first part (in Table 8.6: Computing time).

Initial solution	Build time [min.]	Computing time [min.]	Obj. val. [€]	Gap exact [%]
Approx.	3.9	6.1	██████	4.07
Trucks	3.6	5.6	██████	4.45

Table 8.6: Comparison between the initial solutions

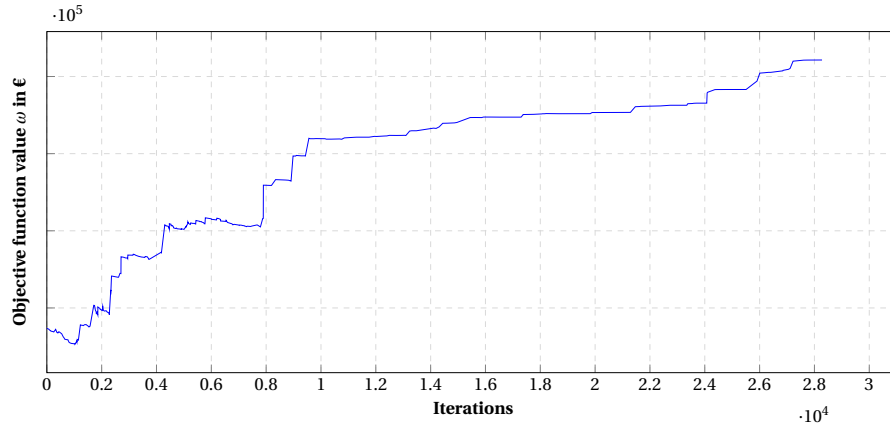
As expected, the approximate initial solution yields a higher objective function value than the truck initial solution. However, in order to properly compare the two initial solutions, we need to run the simulated annealing algorithm, as the truck initial solution was constructed such that the simulated annealing method could be more effective. To obtain a feasible solution using this method takes at most 10.0 minutes. Therefore, the remaining run time for the simulated annealing method is at least 50.0 minutes.

Note that using run time as stopping criterium for the simulated annealing method can lead to a different number of iterations of the method for different parameters.

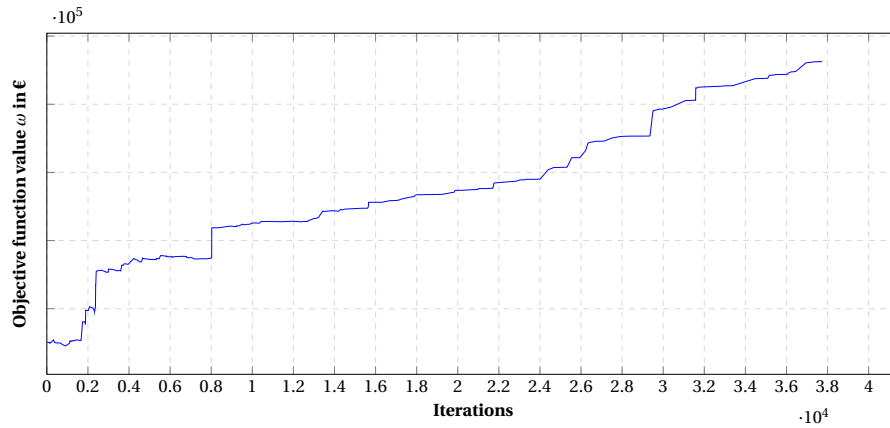
Parameter	Value
T_0	142
α	0.99
M_{\max}	50
Step	5

Table 8.7: Initial values of the simulated annealing parameters

To test which initial solution is best, the other parameters of the simulated annealing method need a value. These parameters are given in Table 8.7. The start temperature is T_0 and α is the reduction factor of the temperature. This reduction happens after M_{\max} iterations. The maximum step size of a change (per iteration) is set to 5, i.e., no more than 5 trucks can be added or deleted per iteration.



(a) Simulated annealing method with approximation initial solution



(b) Simulated annealing method with truck initial solution

Figure 8.4: Simulated annealing method for the two different initial solutions

The results of the simulated annealing method with the parameters of Table 8.6 are given in Figure 8.4. In Figure 8.4a, the results with the approximate initial solution are given. Figure 8.4b describes the results with the truck initial solution. The simulated annealing method starting from the trucks initial solution improves its objective function value more than the simulated annealing method starting from the approximate initial solution (€130,000 compared to €142,000). However, the objective function value of approximate initial solution is €142,000 higher. Therefore, we select the approximate initial solution as our initial solution.

8.2.2. Start temperature

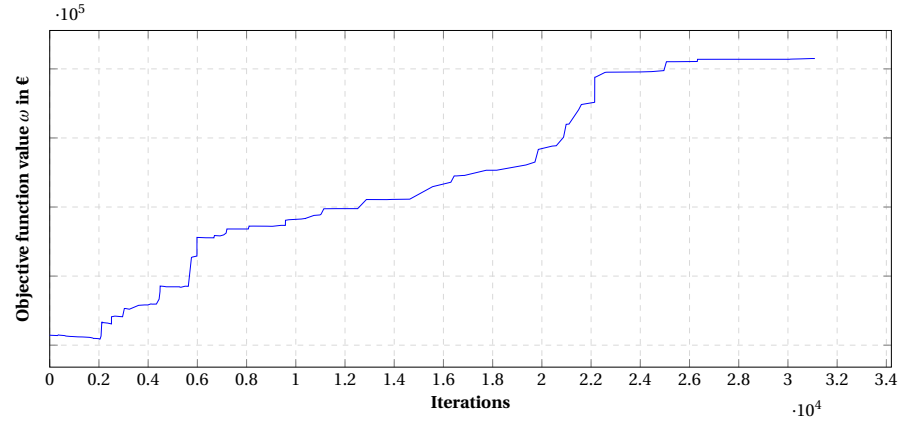
In this subsection, we discuss the results of varying the start temperature of the simulated annealing method. The start temperature T_0 is determined by:

$$T_0 = \frac{\Delta^+}{\ln(a_0^{-1})} \quad (5.2)$$

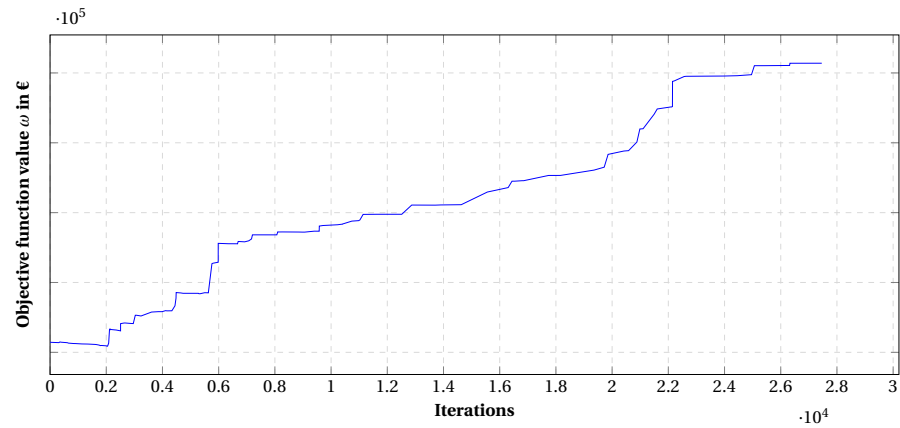
Based on preliminary experiments, $\Delta^+ = 130$. The acceptance ratio, a_0 , is set to 0.4. This yields $T_0 = 142$. However, it can be seen from Figure 8.4a, that during the first 2,000 iterations, the simulated annealing method accepts solutions which are too bad. Therefore, we vary with different acceptance ratios a_0 in order to determine a better start temperature.

However, setting the value of a_0 to 0.3, 0.2, and 0.1 (giving a temperature of 108, 81, and 56, respectively), yielded the same problem: the simulated annealing method accepts solutions which are too bad. Therefore, we set a_0 to 0.1, 0.04 and 0.01. This yields three different T_0 -values: 56, 41 and 28. The progress of the objective function values for the last two different start temperatures is given in Figure 8.5. The comparison between the results of all the different start temperatures are given in Table 8.8.

We select $T_0 = 41$, as the objective function value for this start temperature is the highest.



(a) Simulated annealing method with $T_0 = 41$



(b) Simulated annealing method with $T_0 = 28$

Figure 8.5: Simulated annealing method for the different start temperatures T_0

T_0	Obj. val. [€]	Gap exact [%]
141	██████████	3.39
108	██████████	3.38
81	██████████	3.38
56	██████████	3.38
41	██████████	3.38
27	██████████	3.38

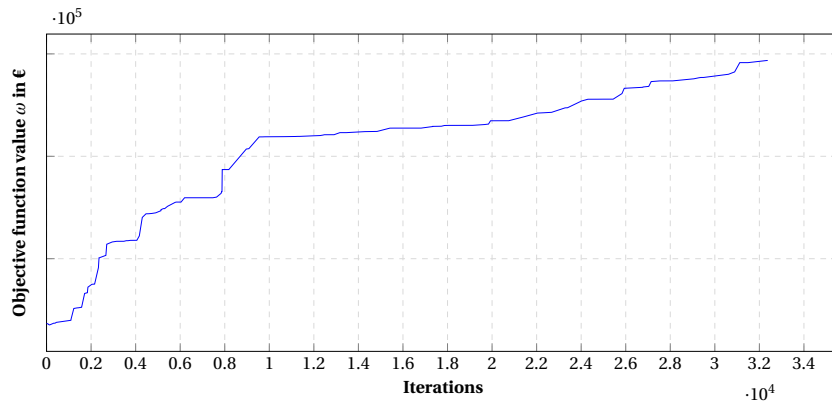
Table 8.8: Results of the simulated annealing method for the different initial temperatures

8.2.3. Cooling schedule

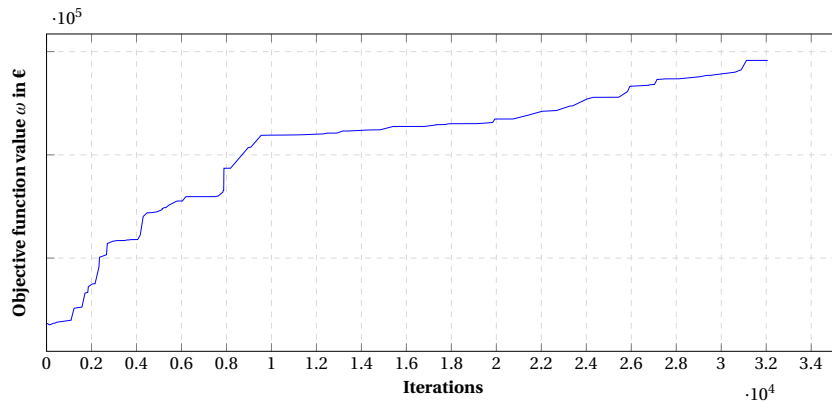
In this subsection, we vary the different parameters related to the cooling schedule. We have the reduction factor α , originally set to 0.99. After M_{\max} iterations, this reduction factor is applied to the temperature. We start with varying the reduction factor α .

We set the value to 0.95, 0.90, 0.85, and 0.80. The progress of the simulated annealing method with the last 2 values of α are shown in Figure 8.6. Note that the figure for $\alpha = 0.99$ is Figure 8.5a. In Table 8.9, the results are compared to the results with the results of $\alpha = 0.99$.

We select $\alpha = 0.85$, as the objective function value for this value of α is the highest.



(a) Simulated annealing method with $\alpha = 0.85$



(b) Simulated annealing method with $\alpha = 0.80$

Figure 8.6: Simulated annealing method for the best 2 values of α

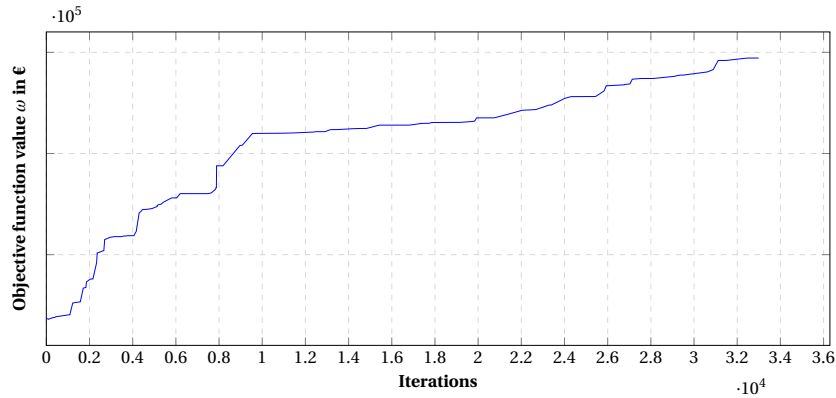
α	Obj. val. [€]	Gap exact [%]
0.99	████████	3.38
0.95	████████	3.38
0.9	████████	3.38
0.85	████████	3.37
0.8	████████	3.37

Table 8.9: Results of simulated annealing method for the different values of α

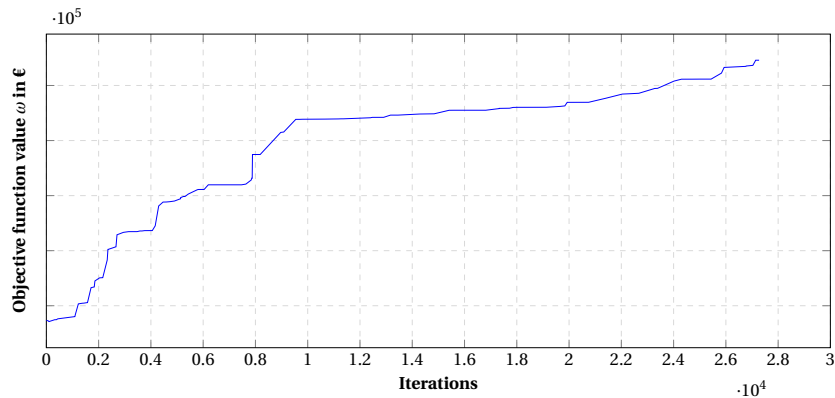
Now that we have established the best reduction factor α , we vary over M_{\max} , originally set to 50, to obtain the best value for the number of iterations after which the reduction factor is applied.

We set the value of M_{\max} to 40 and 30. The progress of the simulated annealing method with these values of M_{\max} is shown in Figure 8.7. Note that the figure for $M_{\max} = 50$ is Figure 8.6a. In Table 8.10, the results are compared to the results with the results of $M_{\max} = 50$.

We select $M_{\max} = 40$, as the objective function value for this value of M_{\max} is the highest.



(a) Simulated annealing method with $M_{\max} = 40$



(b) Simulated annealing method with $M_{\max} = 30$

Figure 8.7: Simulated annealing method for the different values of M_{\max}

M_{\max}	Obj. val. [€]	Obj. val. exact [%]
50	████████	3.37
40	████████	3.37
30	████████	3.37

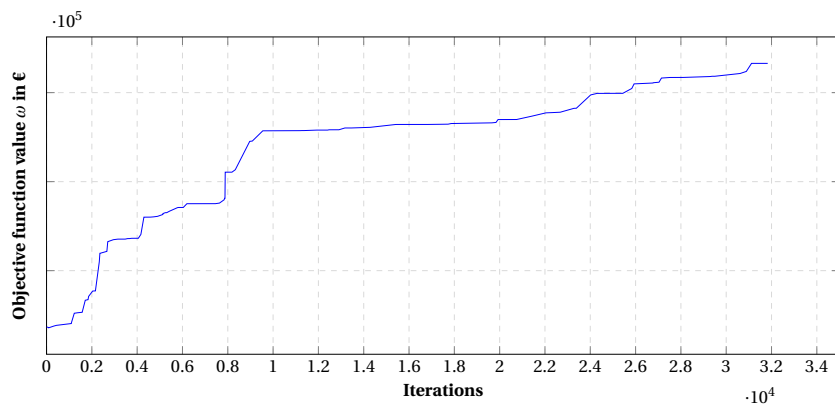
Table 8.10: Results of simulated annealing method for different values of M_{\max}

8.2.4. Step size

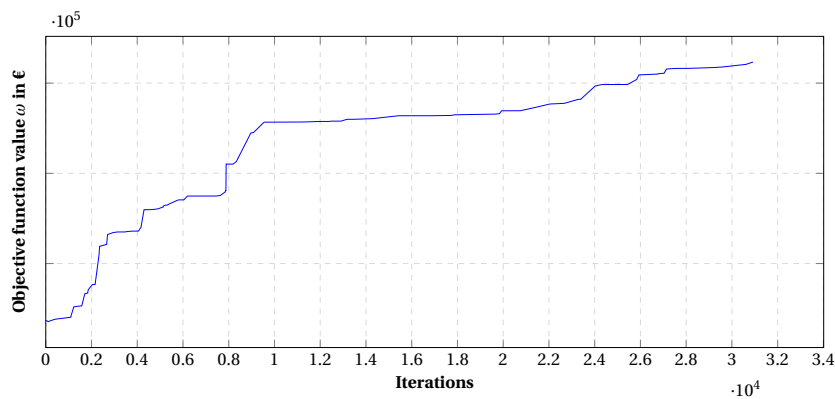
In this subsection, we vary the parameter related to the step size of a change per iteration. The maximum step size (of a change) per iteration was originally set to 5, i.e., no more than 5 trucks can be added or deleted per iteration.

We set the value of the step size to 10, 15, 20, and 25. The progress of the simulated annealing method with these last 2 values of the step size is shown in Figure 8.8. Note that the figure for a step size of 5 is Figure 8.6a. In Table 8.11, the results are compared to the results with the results of original step size.

We select a step size of 20, as the the objective function value for this value of the step size is the highest.



(a) Simulated annealing method with step = 20



(b) Simulated annealing method with step = 25

Figure 8.8: Simulated annealing method for the different step sizes

Step	Obj. val. [€]	Gap exact [%]
5	████████	3.37
10	████████	3.37
15	████████	3.37
20	████████	3.37
25	████████	3.37

Table 8.11: Results of simulated annealing method for the different step sizes

8.2.5. Final selection of parameters

From all the previous subsections, we obtained the final parameters for the simulated annealing method. In this subsection, we gather these parameters and determine the final simulated annealing method for our problem. The parameters are given in Table 8.12.

Parameter	Value
T_0	41
α	0.85
M_{\max}	40
step	20
initial solution	approx

Table 8.12: Final selection of values of the simulated annealing parameters

The simulated annealing method has a gap of 3.37% with the optimal solution. This is largely due to the initial solution which has a gap of 4.07%, which took 10 minutes to run. The iterations of the simulated annealing method took 50 minutes. With the initial parameters, the simulated annealing method decreased the gap with 0.68%. Varying the different parameters, this gap was decreased another 0.02%.

The simulated annealing method with the final parameters, shown in Figure 8.8a, almost does not accept any worse solution. A few worse solutions are accepted, but these decrease the objective function value with €5 to €50. This is negligible. Therefore, our simulated annealing method behaves a lot like a local search method, i.e., only accepting better solutions.

8.3. Comparison between methods

We have analysed the different solution methods in Sections 8.1 and 8.2. In this section, we compare these methods in terms of objective function value and run time.

The CPLEX Optimizer achieved optimal results in 2.3 hours. However, the maximum run time was 1.0 hours. In Figure 8.9, we see the simulated annealing method plotted over time and the results of the limited run time from Table 8.5. From this figure, we can conclude which method is better for a given run time.

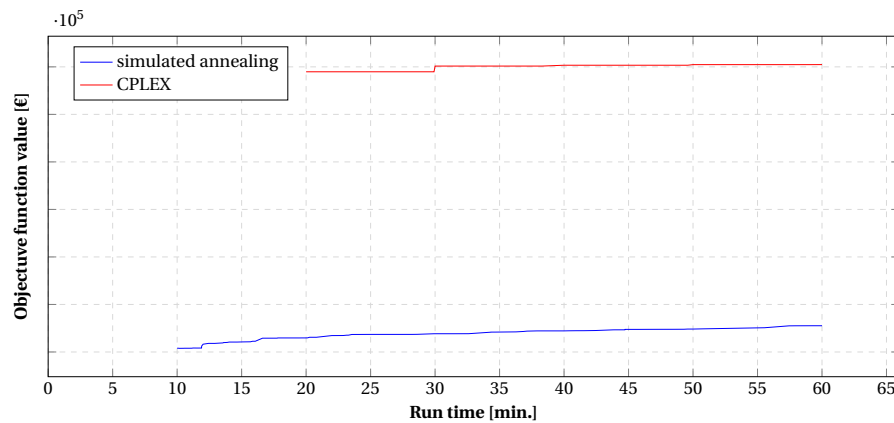


Figure 8.9: Objective function values for the different solution methods for a given run time

We conclude that using the CPLEX Optimizer is the best solution, if the run time may be 20 minutes or longer. For a run time between 10 and 20 minutes, using the simulated annealing method is better. Neither method obtains a feasible solution within 10 minutes.

9

Conclusions

The research question stated at the beginning of this thesis was:

How can we optimise ██████'s sourcing strategy pertaining to its backhaul availability in relation to profit maximisation?

In Chapter 1, we described four characteristics to which our model should adhere:

- The model maximises profits
- The model minimises unfulfilled contracts and under delivered customer demands
- The model incorporates the backhaul availability
- The model has a computing time of less than one hour

In Section 9.1, we draw our conclusions based on these characteristics. Furthermore, we give suggestions for future research in Section 9.2. Finally, we give some recommendations to ██████.

9.1. Conclusions

In Chapter 2, we conducted a literature review to gather information about related mathematical problems. The transportation problem was the closest match to our formulation. However, to account for the backhaul availability, we also adopted the framework of stochastic programming with recourse.

Based on the literature review, we constructed a mixed integer linear program based on the transportation problem and stochastic programming with recourse in Chapter 3. This model has as objective to maximise the total expected profits and minimise the total expected penalties given by unfulfilled contracts and under delivered customer demand.

We proposed two different solution methods to solve our model: one exact solution method and one meta-heuristic, i.e., simulated annealing. In Chapter 4, we linearised the nonlinearities in the constraints and objective function of the model defined in Chapter 3. This yielded a linearised model which can be optimised by a mixed integer optimiser. The other solution method, simulated annealing, was introduced in Chapter 5. In that chapter, the method, initial solution, and potential neighbours are described.

To test the solution methods, we constructed an input instance. This input instance is based on the sourcing strategy based of ██████ on the first six months of 2018. In Chapter 6, ██████'s data of the first six months of 2018 is analysed. All input parameters are based on that analysis. The backhaul availability is discussed in Chapter 7. In that chapter, state space models are introduced to provide the truck availability parameter.

Finally, the results are discussed in Chapter 8. The exact solution method took 135 minutes to obtain an optimal solution. As this solution time is longer than the maximum run time of 60 minutes required, the method was executed with different run times. A feasible solution was found after 17 minutes. After 20 minutes, the CPLEX Optimizer yielded solutions within 3.41% of its upper bound.

Furthermore, the results of the metaheuristic were discussed. The different parameters of the simulated annealing method were varied to obtain the highest objective function value. After 10 minutes, an initial solution was obtained, which has a gap of 4.07% with the exact solution method. Furthermore, the simulated annealing method decreased the gap to 3.37%.

Regarding the run time, it is concluded that using the exact solution is most desirable, if the run time may be 20 minutes or longer. For a run time between 10 and 20 minutes, using the simulated annealing method is better. Within 10 minutes, neither method obtains a feasible solution.

The optimal solution of our model yields that the total expected selling result (expected revenue minus expected costs) of can be improved by %. Furthermore, the optimal solution of our model expects only % of the expected supplier penalties of and % of the expected customer penalties.

9.2. Future Research

In this section, we describe recommendations regarding future research.

One of the goals of this model was to minimise unfulfilled supplier contracts and under delivered customer demands. The total amount of penalties did decrease compared to 's actions. However, the total expected supplier penalties decreased relatively more than the total expected customer penalties. This was due to the constraints on the values per ton of under delivered customer demand and unfulfilled supplier contract. Possibly, the original ratio of under delivered customer demand and unfulfilled supplier contract can be restored by allowing for higher (or lower) penalty values per ton of under delivered customer demand (or unfulfilled supplier contract).

Furthermore, the quality of the current truck availability results can be improved. Some forecasts, such as in Figure 9.1, are bad. In the figure, we see that the variance of the forecast distribution is very high and the forecast also gives a probability to a negative number of realised trucks, which is impossible. To improve the forecasts, it is recommended to use a more advanced model which makes use of more relevant (but currently unavailable) data, such as the total available backhauls per months.

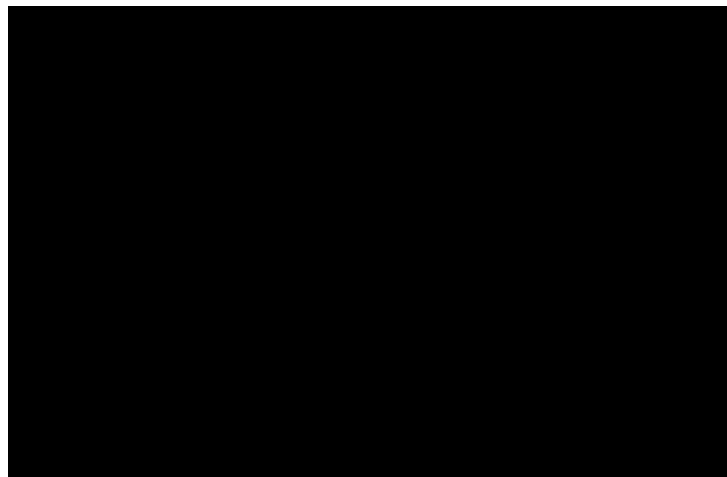


Figure 9.1: Number of trucks per month from postal zone *censored* to *censored*

Finally, we expect that the performance of the simulated annealing method can be improved. As the simu-

lated annealing method behaves as a local search method, it is perhaps an indication that such a method is more promising. Furthermore, when creating neighbour solutions, six changes are possible, and currently, during each iteration it is tested whether each change is available. This is time consuming and it may be faster to select a change and check whether it is available. If it is available, apply the change, and if not, then move to another iteration. Or, perhaps only using the possible changes of deleting and adding a scheduled truck yields better results, as these are computed very fast.

9.3. Recommendations

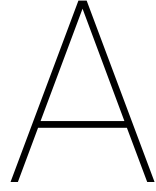
We discuss the general recommendations to ████████ in this section. All recommendations are based on the acquiring and quality of data.

We recommend ████████ to gather more data regarding price, contract and truck data. Currently, none of the prices are logged, unless a contract is purchased. It may be beneficial to keep track of the prices of products, such that more accurate testing can be done with the model.

Furthermore, contracts are not all logged in a consistent way. For example, there exist contracts which include a storage cost for each month. However, such a cost is only made apparent in an irregular way in a comment on the data. Moreover, the intent with which a contract is purchased or sold is also unknown. For example, if a contract is purchased to be resold later, or if the product is to be transported to a silo, or if the product is to be used for direct transport is unknown. Creating more clarity in this data yields better approximations whether contracts were good deals or not. It has been declared that this was unclear by many different people.

Moreover, truck data was rather consistent. However, it may be insightful to log the missed opportunities to realise a truck. It is common practice that a third party logistics provider calls ████████ to ask whether they need a truck for a certain location and that ████████ does not have a contract in that given location. To improve the data for the truck availability, ████████ can log these opportunities.

Appendices



Appendix A: Algorithms

Algorithm 1 Simulated annealing algorithm

```
1: input  $g(T)$ 
2: set  $time \leftarrow 00:00$ 
3: run Algorithm 2: Initial solution ( $\omega_1 / \omega_2$ )
4: while  $time < 60:00$  do (algorithm has duration of 60 minutes)
5:    $M = 0$ 
6:   while  $M < M_{max}$  do
7:     procedure GENERATE NEIGHBOUR
8:       run Algorithm 3: Neighbour search
9:       run Algorithm 4: Processing algorithm
10:    if  $\omega' - \omega \geq 0$  then
11:      accept solution
12:    else
13:       $q \sim U(0, 1)$ 
14:      if  $q < e^{-\frac{|\omega' - \omega|}{T}}$  then
15:        accept solution
16:      else
17:        do not accept solution
18:     $M \leftarrow M + 1$ 
19:    update  $time$ 
20:   $T \leftarrow g(T)$ 
```

Algorithm 2 Initial solution

```
1: Redacted
```

Algorithm 3 Neighbour search

```
1: Redacted
```

Algorithm 4 Processsing algorithm

1: **Redacted**

Bibliography

- [1] S. Arunapuram, K. Mathur, and D. Solow. Vehicle routing and scheduling with full truckloads. *Transportation Science*, 37(2):170–182, 2003.
- [2] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.
- [3] G. Clarke and J. Wright. Scheduling of vehicles from central depot to number of delivery points. *Operations Research*, 12(4):568–581, 1964.
- [4] A. Dekkers and E. Aarts. Global optimization and simulated annealing. *Mathematical programming*, 50(1-3):367–393, 1991.
- [5] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.
- [6] M. Gendreau, O. Jabali, and W. Rei. *Chapter 8: Stochastic Vehicle Routing Problems*, pages 213–239. 2014.
- [7] B. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.
- [8] K. Haley. New methods in mathematical programming—the solid transportation problem. *Operations Research*, 10(4):448–463, 1962.
- [9] R. Hyndman, A. Koehler, K. Ord, and R. D Snyder. *Forecasting with exponential smoothing. The state space approach*. 2008.
- [10] IBM. CPLEX optimizer. URL <https://www.ibm.com/us-en/marketplace/ibm-ilogcplex>.
- [11] P. Kall and S. Wallace. *Stochastic programming*. Springer, 1994.
- [12] J. Kim and Y. Kim. A decomposition approach to a multi-period vehicle scheduling problem. *Omega*, 27(4):421–430, 1999.
- [13] P. J. M. Laarhoven and E. H. L. Aarts, editors. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1987.
- [14] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [15] J. Oyola, H. Arntzen, and D. Woodruff. The stochastic vehicle routing problem, a literature review, part II: solution methods. *EURO Journal on Transportation and Logistics*, 6(4):349–388, 2017.
- [16] J. Potvin, Y. Xu, and I. Benyahia. Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 33(4):1129–1137, 2006.
- [17] D. Sariklis and S. Powell. A heuristic method for the open vehicle routing problem. *Journal of the Operational Research Society*, 51(5):564–573, 2000.
- [18] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.
- [19] F. Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3):192–204, 1969.