2019-12

OPTIMIZATION OF LOGISTIC SUPPLY CHANNELS OF THE GERMAN ARMED FORCES FOR THE TRANSPORTATION OF SMALL-ARMS AMMUNITION

Gramann, Alexander

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZATION OF LOGISTIC SUPPLY CHANNELS OF THE GERMAN ARMED FORCES FOR THE TRANSPORTATION OF SMALL-ARMS AMMUNITION

by

Alexander Gramann

December 2019

Thesis Advisor: Emily M. Craparo
Co-Advisor: Matthew Norton
Second Reader: Armin Fügenschuh, Brandenburgische Technische Universität Cottbus

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2019	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE OPTIMIZATION OF LOGISTIC SUPPLY CHANNELS OF THE GERMAN ARMED FORCES FOR THE TRANSPORTATION OF SMALL-ARMS AMMUNITION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Alexander Gramann	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Today, the responsibilities of the Bundeswehr (German Armed Forces) have become broader and increased in scope. The basic readiness and training requirements for each German service member to be employed domestically (for national defense or a state of emergency) or deployed internationally are focused on training and qualification with small arms.

This thesis describes a two-stage scenario robust integer linear optimization model of logistic supply channels of the German Armed Forces for the transportation of small-arms ammunition. Based on different study cases, we explore how individual units should be optimally assigned to a primary and alternate supply depot. To accomplish this, we optimize the supply routes for each unit by calculating the shortest travel times meeting certain transportation requirements. We consider potential depots to open in the first stage of the model. We wish this decision to be robust to demand uncertainty and adaptability for future supply processes from the perspective of given supply perturbations. Our second-stage decisions reflect day-to-day vehicle routing decisions; these decisions are made after the daily demands are revealed. Finally, we analyze the results for three deterministic cases and a robust case including five demand scenarios.

14. SUBJECT TERMS optimization, MILP, two-stage optimization model, logistic supply channels, German Armed Forces	**15. NUMBER OF PAGES** 73
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

OPTIMIZATION OF LOGISTIC SUPPLY CHANNELS OF THE GERMAN ARMED FORCES FOR THE TRANSPORTATION OF SMALL-ARMS AMMUNITION

Alexander Gramann
Major, German Army
MEng, UniBw Munich Germany, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2019

Approved by: Emily M. Craparo
Advisor

Matthew Norton
Co-Advisor

Armin Fügenschuh
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Today, the responsibilities of the Bundeswehr (German Armed Forces) have become broader and increased in scope. The basic readiness and training requirements for each German service member to be employed domestically (for national defense or a state of emergency) or deployed internationally are focused on training and qualification with small arms.

This thesis describes a two-stage scenario robust integer linear optimization model of logistic supply channels of the German Armed Forces for the transportation of small-arms ammunition. Based on different study cases, we explore how individual units should be optimally assigned to a primary and alternate supply depot. To accomplish this, we optimize the supply routes for each unit by calculating the shortest travel times meeting certain transportation requirements. We consider potential depots to open in the first stage of the model. We wish this decision to be robust to demand uncertainty and adaptability for future supply processes from the perspective of given supply perturbations. Our second-stage decisions reflect day-to-day vehicle routing decisions; these decisions are made after the daily demands are revealed. Finally, we analyze the results for three deterministic cases and a robust case including five demand scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Bundeswehr (German Armed Forces) was originally constructed for the defense of the Federal Republic of Germany, but recently it has evolved to accomplish a wider range of tasks that it can be called upon to do. The responsibilities have become both broader and have increased in depth. Currently, the Bundeswehr is deployed across the globe to fulfill their obligations to North Atlantic Treaty Organization (NATO) and the European Union (EU), requiring at least basic levels of training with small arms for the qualification of each German service member. This training takes place on ranges at military bases throughout the nation. The ammunition that is required by the training units is currently supplied by and drawn from 13 separate supply depots.

This thesis considers the optimization of the logistic supply channels for all units of the German Armed Forces for the transportation of small-arms ammunition. The goal is to increase the efficiency and save time and money for the German Armed Forces.

To increase efficiency in the areas we consider, we formulate a two-stage integer linear program (ILP). The first stage of the model selects which depots to open. We wish this decision to be robust to demand uncertainty and adaptable to future supply perturbations. Our second-stage decisions reflect day-to-day vehicle routing decisions; these decisions are made after the daily demands are revealed. Based on different study cases we explore how individual units should be optimally assigned to a primary and alternate supply depot. We optimize the supply routes for each unit by calculating the shortest travel times given certain restrictions.

Our analysis first considers three deterministic demand cases: the Base-Case, the Extended-Case and the Optimized-Case. These cases are based on historical demand data. The Base-Case represents the status quo, whereby the units draw the ammunition from one of the 13 existing supply depots, and no new depots are opened. The Extended-Case explores the actual future supply scheme of the German Armed Forces. The Ministry of Defense has recently decided to reopen three depots that had been closed. We show that the addition of these three depots decreases total travel time; however, the Optimized-Case

shows us the three depots which should be open from a mathematical perspective (independently of the Ministry of Defense decision). The total travel time for this case decreases significantly. For the deterministic demand cases, we conclude that the difference between the Base-Case and the Extended-Case is not very significant. Only the Optimized-Case results show a noteworthy improvement of the travel times of the customer and the utilization distribution of the depots.

The second part of the analysis considers uncertain demand. To test the model from the perspective of robustness and future supply processes, we create two robust uncertainty cases including five different demand scenarios, labeled US-U1, US-U2, US-T1, US-T2, and US-T3. For the first two scenarios (US-U1 and US-U2) we use a uniform distribution to perturb the historical demand for 2017 and 2018. For the other scenarios (US-T1, US-T2, and US-T3) we generate completely random demands for each unit, ammunition type and day using a triangle distribution. The results show that for all scenarios the customer depot assignments are identical and the same depots are opened. From this we conclude that these decisions are robust to future demand uncertainty. We proceed to do further analysis to underline this conclusion. Therefore, we simulate an outage of the most utilized depot and run the model again. All five scenarios results suggest the reopening of the same depots and only slight differences in the customer assignments. We conclude that this solution is robust to demand uncertainty.

ACKNOWLEDGMENTS

I would like to thank Professor Emily M. Craparo for her outstanding support and guidance during the thesis process. Her help through the process was instrumental to its success. I would also like to thank Professor Matthew D. Norton for his assistance whenever I needed help. Additionally, I would like to thank Professor Armin Fügenschuh for his advice and comments and his support at all times. His know-how in combination with the legwork done by LtCol Björn Oliver Seitner and his team from the German Armed Forces Logistic Center created a beneficial effect for the German Armed Forces. Furthermore, a special thanks goes to my friend and neighbor Major William Muir, who supported me whenever I asked for help or advice.

Finally, I would not be in the position I am without the love and dedication of my wife, Tanja, and my children.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The Bundeswehr (German Armed Forces) is tasked through Article 87a of the German Constitution with the defense of the Federal Republic of Germany. For decades this was interpreted to apply to the defense against an imminent attack against the nation or one of its North Atlantic Treaty Organization (NATO) allies. Today, the Bundeswehr’s responsibilities have become broader and increased in scope. Currently, the Bundeswehr is deployed across the globe in direct support of operations in countries such as Afghanistan, Kosovo, Mali, Syria, and Iraq under either a United Nations mandate or to fulfill their obligations towards NATO and the European Union (EU).

The basic readiness and training requirements for each German service member to be employed domestically (for national defense or a state of emergency) or deployed internationally are focused on the training and qualification with small arms (G36 assault rifle, P8 pistol, etc.). This requires regular training on live-fire ranges for all branches of the Bundeswehr. These trainings take place on military training areas and on ranges throughout the nation. The required ammunition is currently supplied by and drawn from 13 separate supply depots.

The Ministry of Defense has authorized the reactivation of depots in three different locations. This thesis investigates the problem of opening an optimal set of new depots to augment the existing set of depots, while simultaneously assigning training units to depots. Currently, ammunition is typically drawn by each training unit from the supply depot that is geographically closest. However, maximum storage capacities of the supply depots as well as available stock of required resources in storage must be considered and may necessitate longer routes. In allowing longer routes, we assume that the required ammunition for each unit should be picked up the morning of the small-arms training or the day before, while also considering the regulations regarding required rest periods for the driver.

This thesis formulates a scenario-robust integer linear program (ILP) and uses it to analyze several case studies incorporating deterministic and stochastic demand. The three deterministic cases (Base-Case, Extended-Case and Optimized-Case) are analyzed based on historical demand data, while the stochastic robust cases (Robust Extended-Case and Robust Optimized-Case) utilizes five different randomly-generated demand scenarios.

B. LITERATURE REVIEW

Optimization of logistic systems is a wide field, and many different tools and are used to address this class of problems. This literature review places emphasis on two-stage optimization models that include mixed integer linear programming. A two-stage optimization model contains two groups of decision variables: first-stage and second-stage. First-stage variables generally represent "strategic" decisions that are made subject to some uncertainty and that are generally long-term decisions. Second-stage variables represent "operational" decisions that may occur in the short term, after the values of uncertain parameters have been revealed.

In [1], robust two-stage optimization problems are described as an "approach for solving network flow and design problems with uncertain demand." The article goes on to explain:

> Unlike single-stage robust optimization under demand uncertainty, two-stage robust optimization allows one to control conservatism of the solutions by means of an allowed "budget for demand uncertainty." Using a budget of uncertainty, we provide an upper bound on the probability of infeasibility of a robust solution for a random demand vector. [1]

A technical application for a two-stage stochastic optimization model is described in [2]. This research paper uses an approach to optimize a hybrid microgrid system by using ensemble weather forecast. It considers renewable energy sources, besides traditional power sources, to improve energy security and reduce costs for the U.S. military when operating in isolated scenarios. It uses a mixed integer linear program to prescribe an optimal operating power schedule (which will minimize the expected total cost) based on the ensemble weather data. The decision of which generators to use represents the first stage, and is constant across all weather forecast scenarios. Actual weather is revealed and the additional generators are assigned. The other decision variables can vary depending on the different weather scenarios and represent therefore the second stage.

The research in [3] and [4] describes the optimization of an inventory management problem for the German Armed Forces. This research considers how to optimally fill a "warehouse" with spare parts based on a two-stage stochastic programming model under two different scenarios (foreign mission and homeland) of deployment. The goal is to maximize the overall availability of the systems. The first stage decides which parts should be included in the warehouse stock and in the second stage determines which of the parts are assigned to failed systems in order to repair them.

In contrast to the described references, in our model the first-stage decision variables represent the set of depots to be reopened. We wish this decision to be robust to demand uncertainty. Our second-stage decisions reflect day-to-day vehicle type routing decisions; these decisions are made after the daily demands are revealed. This framework allows modelers to find first-stage decisions that are robust to uncertainty in input data, while permitting second-stage decisions to utilize information as it becomes available.

C. OBJECTIVES

This thesis formulates an ILP to optimize the currently executed standard operating procedures and distribution scheme, based on the data received from the German Army Logistic Center. To engage this problem, we consider the following research questions:

1. How should individual units be optimally assigned to a primary and alternate supply depot for the depots currently open (Base-Case) and the set of depots selected for opening by the Ministry of Defense (Extended-Case)?

    - What is the most efficient route for each of the used vehicle types? This calculation considers background information such as weight limits of streets and bridges, height limits for tunnels, and usage restrictions for transporting hazardous material and explosives.

Additionally, the model considers the storage and turnover capacity of each of the supply depots.

2. Which three locations would be the optimal locations to reactivate (Optimized-Case) and how would their reactivation provide the most benefit toward ammunition availability?

   - Based on the findings of Question 1 this thesis provides a recommendation as to which three locations should be reactivated. We compare the results with the decision which was already made by the German Armed Forces for the reopening of three supply depots.

3. Is the new distribution scheme robust and adaptable for future supply processes from the perspective of given supply perturbations and demand uncertainties?

   - Based on the findings of Question 2 this thesis explores the effects of defined uncertainties on the model output.

D. SCOPE, LIMITATIONS AND ASSUMPTIONS

This research models the supply channels of small-arms ammunition and determines the optimal supply routes for all units of the German Armed Forces. To ensure that no conclusions can be made about the defense capability of the German Armed Forces, we use notional values for certain data such as the maximum total storage capacity and the handling capacity of the supply depots. Furthermore, we do not consider the supply of the units deployed on foreign missions. This research is not intended to optimize the inventory of the supply depots. Rather, we focus on the robustness of the small-arms ammunition supply network for the next decade.

E. CONTRIBUTION AND OUTLINE

This thesis aims to improve the overall logistical effectiveness of the German Armed Forces supply network, for the transportation of small-arms ammunition. This benefits the German Armed Forces by saving time and money.

In Chapter II we describe our mathematical model. The goal of the model is to optimize the total travel time of all units of the German Armed Forces.

The model implementation is described in Chapter III. We use Pyomo to solve the developed model in a computational environment [5].

Chapter IV describes our input data, which is required to run and solve the model. The raw data was provided by the German Armed Forces Logistic Center. We cleaned and processed the data. Missing data is identified and then manually determined or calculated by using scripts.

Finally, we analyze the results for all cases and test it from the perspective of robustness and future supply processes. Thereby, the first part of Chapter V explores deterministic demand cases. We determine the optimal solution for the status quo case (Base-Case). In the next step we solve the model for the actual future supply scheme (Extended-Case). Finally, we explore which three locations would optimize the supply chains best, assuming the locations which were closed in the recent years are considered and allowed to reactivate (Optimized-Case). The second part of Chapter V extends the model for uncertainty regarding the annual demand for the customers and compares the results with the deterministic demand cases.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODEL

This chapter describes the model used to test the different cases.

A. DESCRIPTION

The 13 current supply depots are spread across Germany. In 2018 the German Ministry of Defense decided to reopen three additional supply depots that had previously been closed, making a total of 16 depots. Our model considers the question of which depots to open while optimizing the travel times of all units (henceforth called customers). For the stochastic demand cases we incorporate five different demand scenarios. The model incorporates time steps, with one step representing one day. These time steps are necessary to decide whether the ammunition should be picked up on the day of small-arms training, or the day before. The model uses binary decision variables to represent its two main decisions: which depots are reopened, and which depots each customer should be assigned to. The required input data for the model is described in Chapter IV.

B. INDICES AND SETS

Our model contains multiple sets representing relevant entities for the problem. Besides the sets of the depots $\mathcal{D}$ and the set of the customers $\mathcal{K}$, there is a set of different ammunition types $\mathcal{M}$, which are selected based on the different types of arms used in the live-fire training. The vehicle set $\mathcal{V}$ contains the two different vehicle types (regular or heavy). For each vehicle type, the customer can pick up the ammunition within a set of time steps $\mathcal{T}$. One time step is considered to be one day. To assess which depots should be reopened from a mathematical perspective independent from the Ministry of Defense decision, the model considers a set $\mathcal{N}$ which represents the potential supply depots to reopen. The set $\mathcal{S}$ contains the different demand scenarios which we are using for the second stage of our model. To ensure the resupply events of the supply depots we introduce a set of resupply events $\mathcal{R}$. The $\mathcal{P}$ set is used to indicate whether each customer k draws the ammunition for day t on day t or the day before (t'), based on the travel time to each depot d with each vehicle type v. Table 1 shows the indices and sets used for the model.

Table 1. Indices and sets

index $\in$ *set*	set of ...
$d \in \mathcal{D} = \{1,2,3, \ldots, D\}$	depots
$k \in \mathcal{K} = \{1,2,3, \ldots, K\}$	customers
$m \in \mathcal{M} = \{1,2,3, \ldots, M\}$	ammunition types
$n \in \mathcal{N} = \{1,2,3, \ldots, N\}$	potential new depots
$v \in \mathcal{V} = \{1,2\}$	vehicle types (1=regular, 2=heavy)
$t \in \mathcal{T} = \{1,2,3, \ldots, T\}$	discrete time steps (days)
$s \in \mathcal{S} = \{1,2,3, \ldots, S\}$	demand scenarios
$r \in \mathcal{R} = \{1, 2, 3, \ldots, \lceil \frac{365}{\varrho} \rceil\}$	depot resupply events
$(k, d, v, t, t') \in \mathcal{P} \subseteq \mathcal{K} \times \mathcal{D} \times \mathcal{V} \times \mathcal{T} \times \mathcal{T}$	customer k gets ammunition from depot d using vehicle v; day t's demand is picked up on day t' $(t' \leq t)$

C. PARAMETERS

Table 2 shows the parameters used for the model.

Table 2. Parameters

parameter	description
$b_{k,m,t,s} \in \mathbb{R}_+$	demand of customer k for ammunition type m in time step t in scenario s (tons)
$c_{k,d,v} \in \mathbb{R}_+$	travel time for customer k to depot d using vehicle v (minutes)
$g_d \in \mathbb{R}_+$	total storage capacity of depot d (tons)
$h_v \in \mathbb{R}_+$	storage capacity of vehicle type v (tons)
$l_{d,m} \in \mathbb{R}_+$	storage capacity of depot d for ammunition type m (tons)
$w_d \in \mathbb{R}_+$	total handling capacity of depot d (tons)
$\varrho \in \mathbb{Z}_+$	depot resupply interval (days)
$\mu \in \mathbb{Z}_+$	number of new depots to open

The parameter $b_{k,m,t,s}$ represents the demand of a customer $k \in \mathcal{K}$ for the different ammunition types $m \in \mathcal{M}$ in a time step $t \in \mathcal{T}$ for the different scenarios $s \in \mathcal{S}$. The travel time $c_{k,d,v}$ is the parameter used in the objective function which determines the time the customer $k \in \mathcal{K}$ needs to reach a certain supply depot $d \in \mathcal{D}$ by using a regular or heavy transport vehicle $v \in \mathcal{V}$. Each depot has a certain total storage capacity g_d. The maximum capacity of a regular and heavy transport vehicle is assigned to the parameter h_v. The supply depots $d \in \mathcal{D}$ hold various capacities for small-arms ammunition types $m \in \mathcal{M}$ which are represented by the parameter $l_{d,m}$. With the parameter ϱ we can flexibly adjust the resupply events of the depots within the set $\mathcal{R}$. The parameter μ represents the number of depots which we are going to reopen. Lastly, the supply depots have certain handling capacities w_d per day.

D. DECISION VARIABLES

The model uses three families of binary decision variables. The binary decision variable o_d equals one if depot d is reopened and zero otherwise (2.1). The binary decision variable $x_{k,d,v}$ equals one if customer $k \in \mathcal{K}$ is assigned to supply depot $d \in \mathcal{D}$ when using vehicle type $v \in \mathcal{V}$ and equals zero otherwise (2.2).These variables represent our first-stage decisions. Our second-stage decisions reflect day-to-day vehicle type routing decisions; these decisions are made after the daily demands are revealed. Therefore, we use the binary decision variable $u_{k,d,v,t,s}$, to indicate that customer $k \in \mathcal{K}$ sends vehicle type $v \in \mathcal{V}$ to supply depot $d \in \mathcal{D}$ to satisfy the demand for time $t \in \mathcal{T}$ in a scenario $s \in \mathcal{S}$ (2.3).

$$o_d = \begin{cases} 1, \text{ if depot } d \text{ is open} \\ 0, \text{ otherwise} \end{cases} \tag{2.1}$$

$$x_{k,d,v} = \begin{cases} 1, \text{ if customer } k \text{ accesses depot } d \text{ with vehicle } v \\ 0, \text{ otherwise} \end{cases} \tag{2.2}$$

$$u_{k,d,v,t,s} = \begin{cases} 1, \text{ if customer } k \text{ sends vehicle } v \text{ to depot } d \text{ to satisfy demand in a time step } t \text{ in a scenario } s \\ 0, \text{ otherwise} \end{cases} \tag{2.3}$$

E. OBJECTIVE FUNCTION

The objective function (2.4) calculates the total travel time for all customers.

$$\min z = \sum_{k\in\mathcal{K}} \sum_{d\in\mathcal{D}} \sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}} c_{k,d,v}\, u_{k,d,v,t,s} \tag{2.4}$$

F. CONSTRAINTS

The following constraints limit the values of our decision variables. These constraints were developed based on guidance from the German Armed Forces Logistic Center.

1. System State

$$\sum_{d\in\mathcal{D}} x_{k,d,v} \leq 1 \qquad \forall k\in\mathcal{K}, v\in\mathcal{V} \tag{c.1}$$

$$\sum_{d\in\mathcal{D}}\sum_{v\in\mathcal{V}} u_{k,d,v,t,s} \leq 1 \qquad \forall k\in\mathcal{K}, t\in\mathcal{T}, s\in\mathcal{S} \tag{c.2}$$

$$u_{k,d,v,t,s} \leq x_{k,d,v} \qquad \forall k\in\mathcal{K}, d\in\mathcal{D}, v\in\mathcal{V}, s\in\mathcal{S}, t\in\mathcal{T} \tag{c.3}$$

Constraint (c.1) ensures that each customer is assigned to only one supply depot per vehicle type. Constraint (c.2) makes sure that each customer sends one vehicle to an assigned depot per time step per scenario, if there is a demand to satisfy for the customer within the time step and scenario. Constraint (c.3) ensures that the vehicles are deployed based on the supply depot assignment. This ensures that the customers can send the two different vehicle types to different depots to pick up the ammunition.

2. Capacity and Demand

$$\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{V}}\sum_{t'\in\{(r-1)\cdot\varrho+1,\dots,r\cdot\varrho\}}\sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s}u_{k,d,v,t,s} \leq g_d \qquad \forall d\in\mathcal{D}, s\in\mathcal{S}, r\in R \tag{c.4}$$

$$\sum_{k\in\mathcal{K}}\sum_{v\in\mathcal{V}}\sum_{t'\in\{(r-1)\cdot\varrho+1,\dots,r\cdot\varrho\}}\sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s}u_{k,d,v,t,s} \leq l_{d,m} \qquad \forall d\in\mathcal{D}, m\in\mathcal{M}, s\in\mathcal{S}, r\in R \tag{c.5}$$

$$\sum_{m\in\mathcal{M}} b_{k,m,t,s} \leq \sum_{d\in\mathcal{D}}\sum_{v\in\mathcal{V}} h_v\, u_{k,d,v,t,s} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S}, t\in\mathcal{T} \tag{c.6}$$

$$\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{V}}\sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s}u_{k,d,v,t,s} \leq w_d \qquad \forall d\in\mathcal{D}, s\in\mathcal{S}, t'\in T \tag{c.7}$$

Constraint (c.4) implements the total storage capacity of the supply depots. We have to make sure that the total ammunition supplied to customers during a resupply interval does not exceed the actual storage capacity of the depots. The same applies for the storage capacity for the individual small-arms ammunition types, which is ensured by constraint (c.5). The storage capacity for the vehicle types is defined by constraint (c.6). Thereby, the choice of which vehicle to use is based on the demand of the customer, which does not exceed the actual storage capacity of the chosen vehicle type. Constraint (c.7) implements the different handling capacities for all supply depots.

3. Depot Operation

$$x_{k,d,v} \leq o_d \quad \forall d \in \mathcal{D}, k \in \mathcal{K}, v \in \mathcal{V} \tag{c.8}$$

$$\sum_{d \in \mathcal{N}} o_d = \mu \tag{c.9}$$

$$x_{k,d,v}, u_{k,d,v,t,s}, o_d \in \{0,1\} \quad \forall k, d, v, t, s \tag{c.10}$$

Constraint (c.8) ensures that customers are only assigned to an open depot. The number of depots to reopen equals three for our model (c.9). Finally, constraint (c.10) defines the domain of the decision variables.

THIS PAGE INTENTIONALLY LEFT BLANK

III. IMPLEMENTATION

This chapter describes how the developed mathematical model is implemented in a computational environment. We implement the model using the Python software package Pyomo. Pyomo allows us to formulate optimization problems in a manner similar to the notation commonly used in mathematical optimization [5].

The optimization model is initialized with an external Excel file. The data described in Chapter IV is processed and thus made usable for the Pyomo model environment. The sets, indices, parameters, decision variables, constraints, and the objective function described in Chapter II are implemented in Pyomo syntax.

We calculate customer travel times to represent a one-way drive from the customer to each depot. To decide whether a customer picks up the ammunition on the same day or the day before the small-arms training, a threshold variable is necessary. This variable is set to the value 240. If a customer needs more than 240 minutes for a one-way drive, the ammunition must be picked up a day before the small-arms training. If the travel time is less than or equal to 240 minutes the ammunition will be picked up at the same day. Therefore, the maximum driving time per day is 480 minutes (8 hours). If we assume a loading time of an hour, the driver has enough rest time and the daily workload for the transportation of the ammunition does not exceed 9 hours.

The mathematical optimization of the model is done by a solver. The algorithms are based on the procedures described in Appendix A. We use the Gurobi solver to solve the minimization problem.

Depending on the planning horizon (monthly or yearly) and the number of variables and constraints, the runtime results differ. The following table shows the runtime for the deterministic demand cases and a robust stochastic case based on the Optimized Case. A precise description of the cases is given in Chapter V.

Table 3. Model characteristics

	1-months planning horizon			1-year planning horizon		
	# decision variables (all binary)	# constraints	Runtime [min]	# decision variables (all binary)	# constraints	Runtime [min]
Base-Case	3,168,828	3,282,976	38.73	3,168,828	3,288,832	83.30
Extended-Case	3,900,096	4,012,325	48.82	3,900,096	4,018,959	105.58
Optimized-Case	5,362,641	5,477,018	67.36	5,362,641	5,485,253	143.63
Robust Optimized-Case	26,754,561	27,365,535	452.63	26,754,561	27,406,645	965.92

The models were solved using a 2.9 GHz Intel Core i9 CPU with 32 GB RAM. The Pyomo script for the model implementation is shown in Appendix E.

IV. DATA

This chapter describes the data we use to run and solve the model. The raw data was provided by the Logistic Center of the German Armed Forces. To use the data in an appropriate way we cleaned and processed the data. This was mainly done with Microsoft Excel built-in functions (Vlookup, Pivot tables, filter, conditioning, etc.) and the integrated Visual Basic for Application (VBA) environment.

We consider five different cases, where we incorporate demand data of three different deterministic cases (Base-Case, Extended-Case, Optimized-Case) and two robust stochastic cases (Robust Extended-Case, Robust Optimized-Case). The demand data for the deterministic cases is historical and described in Section F. The data for the robust stochastic cases is randomly created and described in Chapter V Section B.

A. CUSTOMER LOCATIONS

The customer data was extracted from an SAP database provided by the German Armed Forces. The addresses are given for some customers. After comparing the customer list with the provided demand lists, it turned out that many customers are mislabeled or are completely missing in the provided customer list. To identify the unknown customers, we use the organization structure chart of the German Armed Forces and other sources. Furthermore, we discard the customers who are not considered (e.g., oversea units) for our optimization model. After finishing the data processing, the customer list contains finally 330 customers distributed all across Germany.

B. SUPPLY DEPOT LOCATIONS

The locations of the 13 existing supply depots for the current situation (Base-Case) and the three future supply depots (Extended-Case) are provided. We research the potential depots to reopen for the Optimized-Case by analyzing historical data and research on the internet. Twenty-seven potential supply depots are identified. The potential depots to reopen are added to a map by using Google Earth Pro. After comparing the distribution of the current supply depots and the results from the Base-Case, six potential depots to reopen are determined. Appendix C shows the map for the depot locations for all cases.

C. COORDINATES

The coordinates for all customers and supply depots are necessary to compute the travel times, but the provided data has no GPS coordinates. We determine the GPS coordinates for all customers and depots manually by using Google Maps.

D. TRAVEL TIME

Based on these GPS coordinates we extend an existing Python script to automate the computation of the travel times. The script sends automated request for travel routes to an OpenStreetMap server located in Germany. This Open Route Service (ORS) allows us to compute the 14,652 travel times quickly. Furthermore, the routing settings can be adjusted easily, enabling the script to compute routes for regular and heavy vehicles. The heavy ground vehicle (HGV) driving profile considers average driving speed for trucks and federal speed limits. Moreover, the script considers infrastructure characteristics (bridges, tunnels, tolls, etc.), and even regulation of hazardous materials transports. The output (travel time and distance) for both vehicle types is saved to an Excel file. The code for the script is shown in Appendix D.

E. SUPPLY DEPOT CAPACITY

The total storage capacity for the 13 supply depots varies and is based on the provided data. As mentioned above, the total storage capacities we use for the model do not correspond with the actual capacities. However, the used values are realistic and lead therefore to a reasonable result. For the three future supply depots (Extended-Case) and the six additional potential depots (Optimized-Case), no data is available and therefore we use the mean of the total storage capacity of the current 13 depots.

The total ammunition storage capacities for all small-arm ammunition types is estimated to be 35 percent of the total storage capacity of the depot. Based on the provided data there are 139 ammunition types for small-arm ammunitions. To allocate the ammunition types effectively, we define four categories for the most common calibers.

Because no data regarding the storage capacities of the different ammunition types of the depots is provided, we assign every category with the same ammunition storage capacity.

The handling capacity for all customers is set to the same value. Our model reflects the fact that every supply depot is equipped with the same equipment and the same manpower to load and unload customer vehicles.

F. DEMAND

Historical customer demand was provided for the years 2017 and 2018. For the model input data we use the outbound delivery date of the supply depots. Filtering the data by order date shows blank entries and many orders exceed the maximum vehicle capacity. This leads to a data validation issue and causes a runtime error in the model. Therefore, we split such demands into multiple demands, each of which is below the maximum vehicle capacity.

G. VEHICLE CAPACITY

The capacities for regular and heavy vehicles are based on the individual maximum load capacity. The model considers two vehicle types. A regular vehicle is allowed to transport a maximum ammunition weight of 0.9 metric tons on public roads. The standard truck used by the most units of the German Armed Forces has a loading capacity of 5 metric tons.

THIS PAGE INTENTIONALLY LEFT BLANK

V. ANALYSIS

Part A of this chapter describes the analysis of the three different deterministic demand cases utilizing historical data from 2018, described in Chapter IV. Part B utilizes the Robust Extended-Case and the Robust Optimized-Case in order to show the results of the stochastic demand cases including five different scenarios.

A. DETERMINISTIC DEMAND

1. Base-Case

The Base-Case is the actual status quo case. The customers draw the ammunition from one of the 13 existing supply depots, and no new depots are opened (μ=0). Our optimization model considers drawing the small-arms ammunition from different depots depending on the vehicle type and distances of the locations. Figure 1 depicts the optimal customer-depot assignment on a map of Germany. The model assigns most customers the same depot for both heavy and light ammunition loads. But for some customers, however, the model finds that optimal assignments would send them to different depot locations for heavy and light loads. The yellow dashed line shows the customers which are assigned to one depot for both vehicle types. Whereas the red dashed line shows assignments for heavy ammunition loads for customers which are assigned to a different depot. The R code we developed for the script is shown in Appendix F. The map is plotted by using the ggmap library in R.

Figure 1. Optimal customer-depot assignments: Base-Case

The total travel time for Base-Case is 4,589.32 hours. Figure 2 shows the utilization of the 13 supply depots for this case.

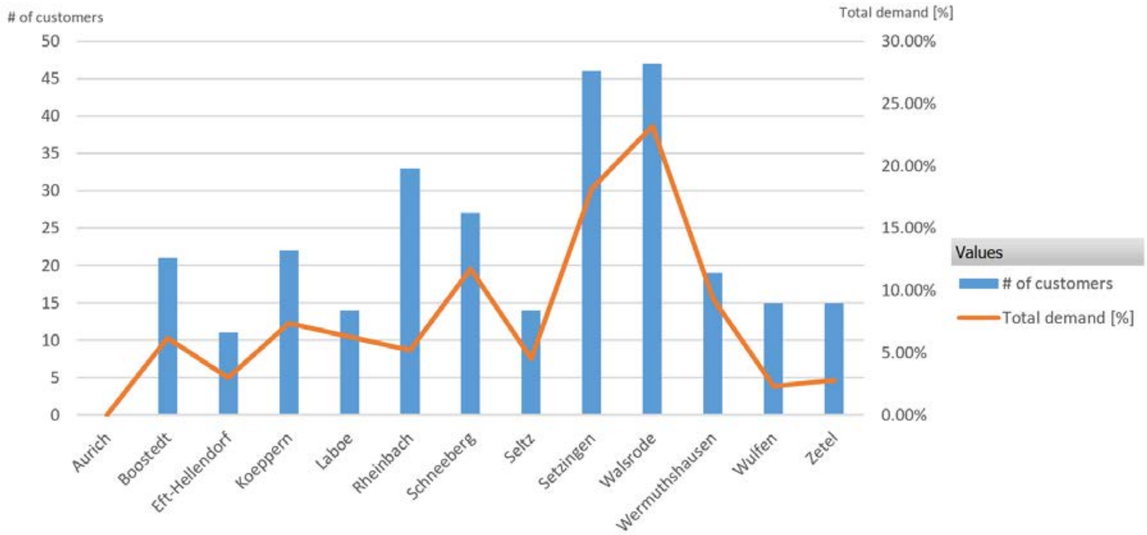


Figure 2. Depot utilization: Base-Case

The horizontal axis represents the depots. The vertical axis represents the number of customers each depot is assigned to and the total demand utilization of the customers as a percentage.

The utilization for the depots “Walsrode” and “Setzingen” is the highest. “Walsrode” handles 23.18% of the total demand, and 47 customers. “Setzingen” handles 18.19% of the total demand and has 46 customers. So, these two depots combined have a workload of 41.37% of the total demand. Figure 1 shows the high utilization of these two depots in the north (near Hanover) and in the south (between Stuttgart and Munich). The supply depot “Aurich” is not used by any customer. The depot “Rheinbach” has 33 customers which is quite high but the total demand (5.18%) is low in comparison to the other depots. However, the depot “Wermuthshausen” has only 19 customers but the workload is quite high (9.32%). So the workload is not necessarily correlated with the number of customer.

2. Extended-Case

The decision of which three locations are reopened was recently made by the German Ministry of Defense. Based on this decision the necessary travel times and capacities were examined. The geographical results for Extended-Case with 16 depots are shown in Figure 3. We can see that adding the three depots does not significantly reduce the utilization of most of the other depots.

Figure 3. Optimal customer-depot assignments: Extended-Case

The total travel time for the Extended-Case decreases to 4,450.10 hours. Figure 4 shows the utilization of the supply depots for the Extended-Case.

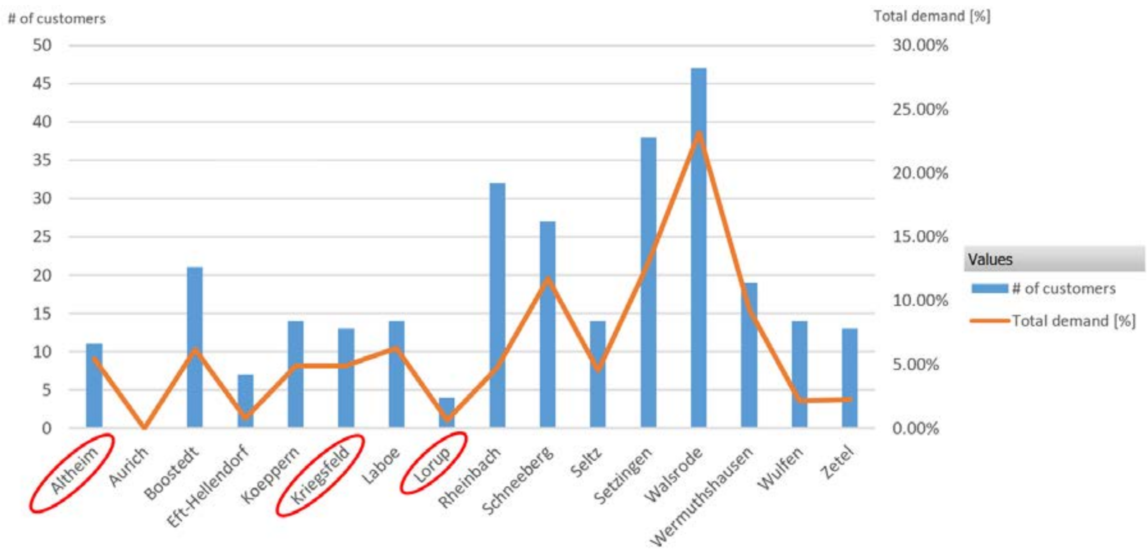


Figure 4. Depot utilization: Extended-Case

The horizontal axis represents the depots. The vertical axis represents the number of customers each depot is assigned to and the total demand utilization of the customers as a percentage.

The utilization of seven depots does not change, including the most utilized depot "Walsrode." The workload for the depot "Setzingen," however, decreases from 18.19% for the Base-Case to 12.95% for the Extended-Case. However, the reopened depots (marked with red circles) are only used by 28 total customers. This corresponds to a total demand rate of 10.99%. The reopened depot "Lorup" is only used by four customers which correspond to a total demand of only 0.64%, which is the lowest value across all depots besides "Aurich." This underlines the fact that the utilization by using 16 depots does not change very much compared to the Base-Case.

3. Optimized-Case

Independently from the Extended-Case, the Optimized-Case seeks to determine which three locations would be the optimal locations to reactivate. Figure 5 shows the geographical customer-depot assignments of the Optimized-Case. We can see that the utilization of the customer-depot assignments is more equally distributed.

Figure 5. Optimal customer-depot assignments: Optimized-Case

The total travel time for the Optimized-Case decreases significantly and is 3,817.78 hours. Figure 6 shows the utilization of the supply depots for the Optimized-Case.

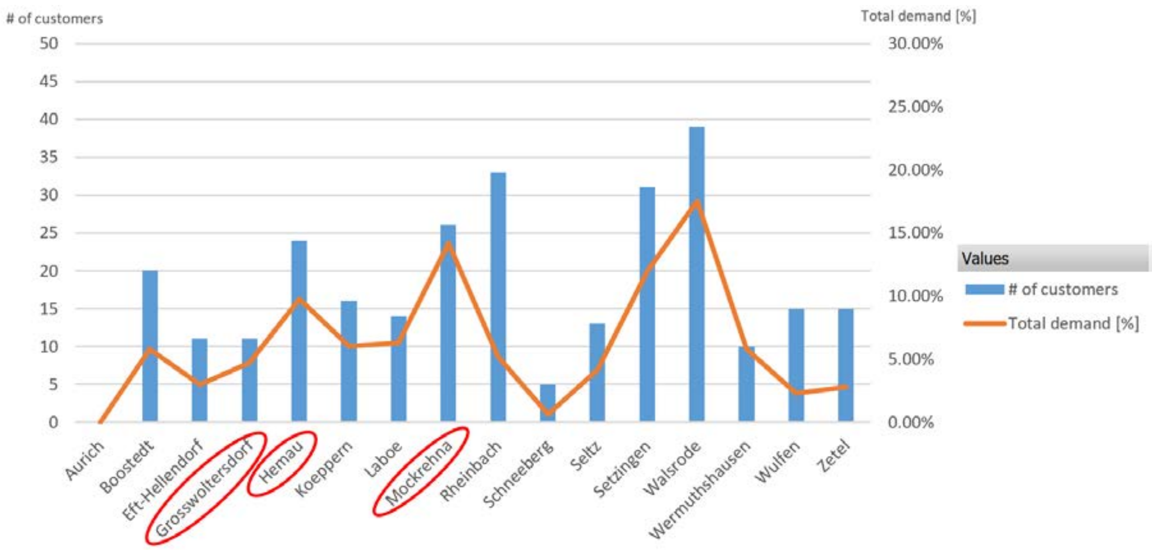


Figure 6. Depot utilization: Optimized-Case

The horizontal axis represents the depots. The vertical axis represents the number of customers each depot is assigned to and the total demand utilization of the customers as a percentage.

The reopening of the optimal depots (marked with red circles) has a big impact on the utilization distribution. These three depots handle 61 customers which correspond to a total demand of 28.67%. This effects the depot "Schneeberg" the most. The number of customers of this depot decreases from 27 for the Base-Case to 5 customers for the Optimized-Case, this is 11.73%, 0.62% of the total demand respectively. The depot with the highest utilization is still "Walsrode" but the workload decreased by 5.68% of the total demand from Base-Case to Optimized-Case. Figure 6 shows the more homogeneous distribution of the utilization of the 16 depots.

4. Summary of Results—Deterministic Demand

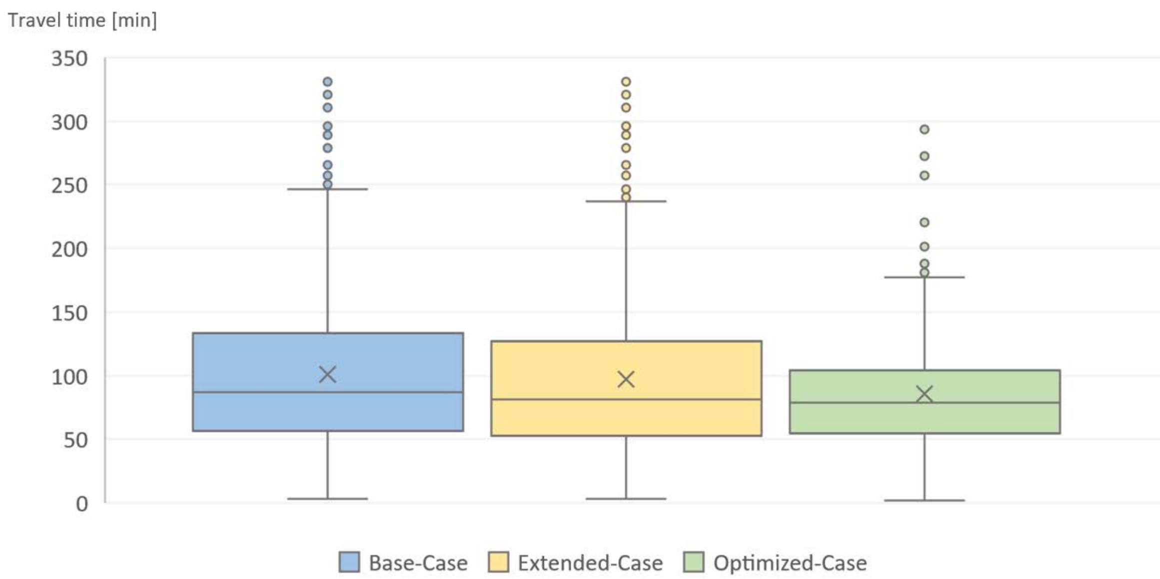
First of all, we can see that the depot "Aurich" is not used in any of the three deterministic cases. This depot is located in the northwest part of Germany. Based on the model output for all three cases all customers in this area should draw the small-arms ammunition from the depot "Zetel."

Table 4 shows the travel times for all three cases. The travel times decrease by adding three depots for both the Extended-Case and the Optimized-Case. However, the difference between these cases is significant. For the Extended-Case the travel time decreases by 139.22 hours, which corresponds to 3.03% less travel time. For the Optimized-Case the travel time decreases by 771.54 hours which means 16.81% less travel time.

Table 4. Travel times for deterministic demand cases

Base-Case	Extended-Case	Optimized-Case
4,589.32 hours	**4,450.10 hours**	**3,817.78 hours**

To get a deeper insight into the cases we examine the results in more detail using a boxplot analysis. Figure 7 shows a boxplot of the customers' travel times for all three cases.



The vertical axis represents the travel times in minutes of the customer to the assigned depots.

Figure 7. Customer travel times

We can see that the average travel times for the customers differ for all three cases. The Base-Case has an average travel time of 101 minutes and the Extended-Case has an average of 98 minutes. For the Optimized-Case the average travel time decreases significantly to 85 minutes. However, the median differs by less. The Base-Case has a median travel time of 87 minutes, the Extended-Case has 82 minutes and the Optimized-Case is slightly lower with a median of 79 minutes. The interquartile range (IQR) and the range (excluding outliers) decrease steadily from the Base-Case to the Optimized-Case. All cases have several outliers. The boxplot shows skewness for the Base-Case and the Extended-Case, the Optimized-Case is almost symmetric.

In summary, we can conclude that the improvement between the Base-Case and the Extended-Case is not very significant. Only the Optimized-Case results show a noteworthy improvement of the travel times of the customer and the utilization distribution of the depots.

B. STOCHASTIC DEMAND

To test the solution for robustness we add uncertainty to the annual demand of the customers. We generate five different scenarios: US-U1, US-U2, US-T1, US-T2, and US-T3. We then incorporate these five demand scenarios into two new cases: the Robust Extended-Case, which models the depot selection made by the Ministry of Defense, and the Robust Optimized-Case, which selects an optimal set of three depots to reopen. We use the Palisade @Risk software package to generate stochastic customer demands. The software allows us to apply different distributions. We first use a uniform distribution to perturb the historical demand for 2017 and 2018. For each customer, ammunition type and time period, we generate a stochastic demand uniformly:

$$b_{k,m,t,s} \sim \mathrm{U}(b_{k,m,t,s=historical},\ 1.2^{*}\ b_{k,m,t,s=historical})$$

The first scenario (US-U1) incorporates the demand for 2017, while the second (US-U2) uses the data for 2018. For the other scenarios (US-T1, US-T2, US-T3) we use a Python random number generator (included in the Pyomo script, Appendix E) which generates demands using a triangle distribution. Therefore, we generate three demand scenarios according to a triangular distribution

$$b_{k,m,t,s} \sim \mathrm{T}(0,\ 0.5,\ 5)$$

The most likely value for the triangle distribution is set to 0.5 metric tons. The minimum value is set to zero and the maximum value is set to five metric tons (the maximum loading capacity of the heavy vehicle). The random number generator creates random demands for each customer, ammunition type and time period.

We first consider the Robust Optimized-Case, in which the model selects three depots to open. For this case, the optimal objective function value (total travel time) is 21,406.78 hours. The same three depots are opened in the Robust Optimized-Case as in the Optimized-Case, indicating that this decision is robust to the demand perturbations we consider.

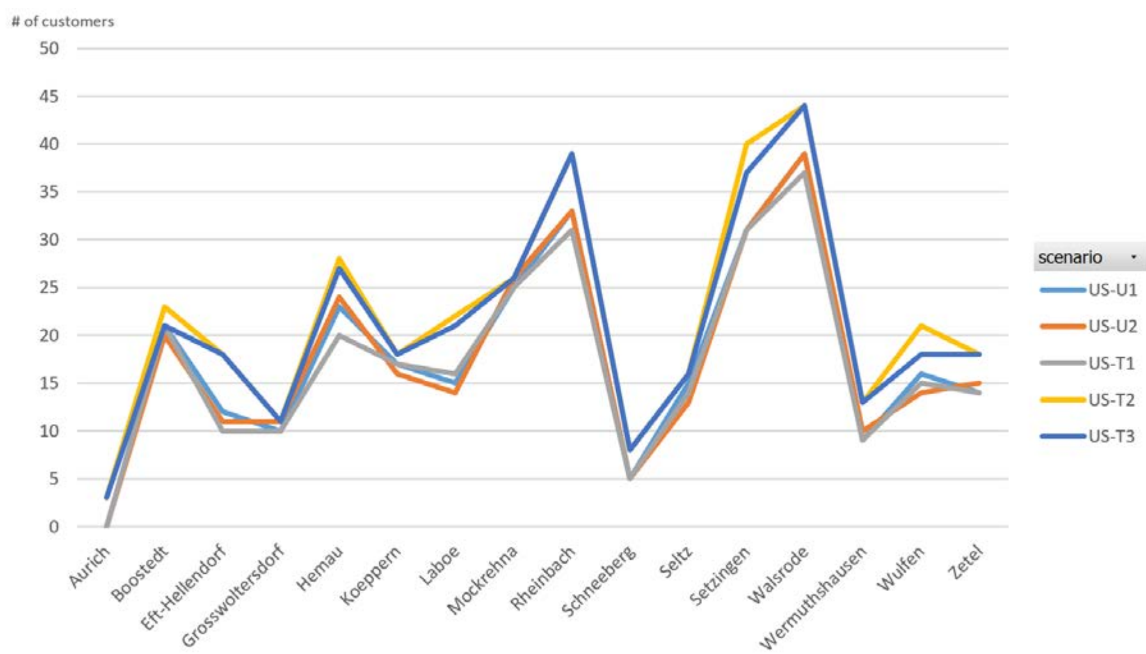


Figure 8. Stochastic demand scenario results

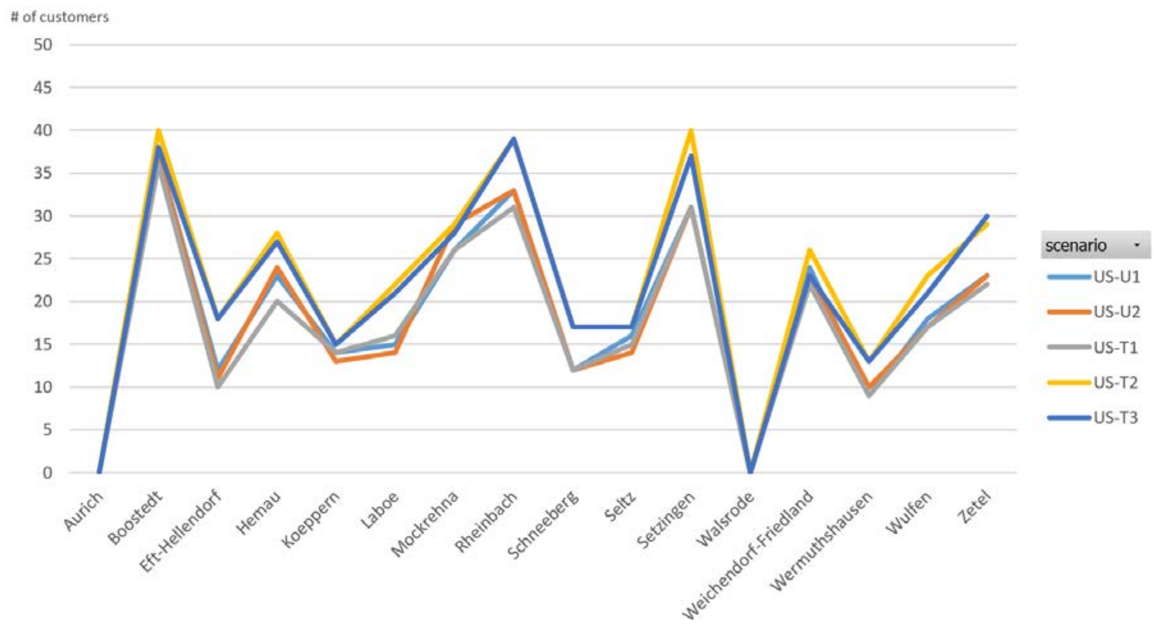
The horizontal axis shows the depots for the Robust Optimized-Case. The vertical axis represents the number of customers accessing each depot at least once.

Figure 8 shows the number of customers accessing each depot at least once in each of the five scenarios. We can see that the number of customers varies only slightly. This is reasonable, since the assignment of customers to depots is a first-stage decision, and thus is constant across all scenarios. The fluctuations observed in Figure 8 are caused by the fact that some customers only visit one of the two depots to which they are assigned.

We next consider the Robust Extended-Case, in which we examine the set of depots selected by the Ministry of Defense with respect to the five demand scenarios described above. The results are not significantly different. For this case, the optimal objective function value (total travel time) is 25,15205 hours. This is reasonable, since the total travel time for the Extended-Case is higher than the total travel time of the Optimized-Case.

The customer-depot assignments are identical to the Extended-Case and the number of customers visiting each depot at least once varies only slightly among the five scenarios.

Finally, we examine the robustness of the supply network to the loss of a depot. Specifically, we assume that the most utilized depot, “Walsrode”, is out of order. We simulate this by setting the total storage capacity g_d for this depot to zero. Rerunning the model based on the Robust Optimized-Case yields a total travel time of 23,986.23 hours; an increase of 12.05% relative to the value when “Walsrode” is available. We note that the other depots adjust to compensate for the outage of “Walsrode”. The decision which depot to open is different for one depot: “Weichendorf-Friedland” is now opened rather than “Grosswoltersdorf”. Figure 9 shows the customer-depot utilization for this model.



The horizontal axis shows the depots for the Robust Optimized-Case. The vertical axis represents the number of customers accessing each depot at least once.

Figure 9. Stochastic demand results: Outage of one depot

Similar to Figure 8, we can see that the number of customers accessing each depot varies only slightly among the scenarios. We can see that several depots have a higher utilization than when "Walsrode" is available; nevertheless, all customers are supplied.

VI. CONCLUSION AND RECOMMENDATIONS

An important aspect of the most effective logistic supply channels is optimizing the supply network. For military applications, this can be very challenging because there are factors to consider beyond profitability. We must consider road capabilities for special equipment, potential attacks on the network, increased demands on timeliness, and the fact that the equipment must get to the end user even if it is not economically feasible. This thesis aims to improve the overall logistical effectiveness of the German Armed Forces supply network, for the transportation of small-arms ammunition.

We have developed a mathematical model and implemented it in a computational environment to derive the optimal total travel time of all units for the German Armed Forces. The raw data was provided by the German Armed Forces Logistic Center and missing required data was determined.

First, we analyzed the deterministic demand cases. The results of these cases give us some insight that helps us to optimize the supply chains. The status quo case (Base-Case) has a quite high utilization for certain depots. We showed that the future supply scheme (Extended-Case) is not optimal from a mathematical perspective. The Optimized-Case shows us the three depots which actually should be open. Such a more homogeneous distribution like in the Optimized-Case would have a significant impact on the total travel time, making ammunition distribution more efficient. However, the optimization model we developed considers only small-arms ammunition. Therefore, the decision which was made by the German Ministry of Defense to reopen the Extended-Case depots is not necessarily wrong. All cases had in common the fact that one depot ("Aurich") is not used. Hence, it should to be checked whether this depot could be closed.

To test the model for future supply robustness we developed demand scenarios that contain uncertain demand data. Applying the stochastic demand to the Robust Optimized-Case and the Robust Extended-Case, we see that both cases are robust for future uncertain demands. Based on the results of the deterministic demand cases we know that the depot handling capacity and the storage capacities are not limiting factors and therefore the model seems very robust for demand uncertainty.

Further analysis could be to explore the monthly utilization and try to optimize over the periods with a high demand.

Furthermore, future work could be to extend the model for all ammunition types. Based on the results we could consider to close depots or reallocate customer depot assignments. This would increase the efficiency and save time and money for the German Armed Forces.

As a next step it is reasonable to feed the model with the exact input data. The results of this research demonstrate a time and respectively money saving potential for the German Armed Forces.

APPENDIX

A. BASICS OF LINEAR OPTIMIZATION

Optimization is a wide field with diverse subcategories. In this appendix we discuss the basics of linear optimization, better known as linear programming (LP). Starting with the mathematical foundations and the explanation of linear programming in operations research, followed by the integer linear programming (ILP) and a practical example given by the transportation problem.

1. Linear Programming in Operations Research

Generally, the model which is created through the operations research process and is described through mathematical methods and solved through mathematical processes. The optimization calculation attempts to solve a mathematically formulated problem. In doing so, there exists mostly a maximization or minimization of a specific objective function, wherein into this function only specific values can be entered, which are in turn regulated by constraints [6].

In LP calculations, the objective function is a linear function of the decision variables and the constraints are linear equations or inequalities of the decision variables. In doing so, the decision variables can assume real numbers. Many economic and technical hypotheses are based on a linear relationship or on objective criteria. The following steps are necessary to achieve a linear optimization equation:

- Specifications of the desired values or decision variables.
- Formulation of all constraints of the problems as linear equations or linear inequalities for decision variables.
- Specification of the objective function in form of a linear function of the decision variables, to minimize or maximize respectively [7].

The following is an example of a minimization problem:

$$\min z = \sum_{j=1}^{n} c_j x_j \qquad (1.1)$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1, ..., m) \qquad (1.2)$$

$$x_j \geq 0 \qquad (j = 1, ..., n) \qquad (1.3)$$

2. Simplex Algorithm

This section covers the basics of the simplex algorithm. It is not intended to cover how to apply and calculate the simplex algorithm in detail.

The simplex algorithm (or simplex method) is a popular algorithm for linear programming. The simplex algorithm requires that the objective function and the constraints are linear and only continuous variables occur. Thereby a polyhedron is built from the objective function and the constraints. The simplex algorithm tries to find an optimal solution by walking along the edges to extreme points. The edge point with the maximum value is the optimal solution. By visiting an unbounded edge, the algorithm concludes that the problem has no optimal solution. Figure 10 illustrates the simplex algorithm.

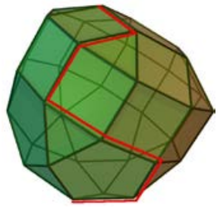


Figure 10. Polyhedron of simplex algorithm in 3D. Source: [8].

3. Integer Linear Programming

For many linear optimization problems, the variables are restricted to take on integer values. Such a problem is known as an integer linear program (ILP). A special case is the binary ILP, in which the unknowns are either 1 or 0. An example is to make a decision, either to add a product to the production process or not. If the decision variable x

= 1, the product should go into production. If x = 0 the product should not be processed to production. If some of the decision variables are continuous the problem is known as a mixed-integer programming problem [9].

The ILP with binary decision variables is used to solve the optimization problem in this thesis. Based on a finite list of customers, binary decision variables are used to assign the customer to an optimal supply location, whereby x = 1 signifies that the customer uses a certain supply depot and x = 0 means that this location is not optimal for the certain customer.

There are different algorithms to solve ILP. The most popular are the Branch-and-Bound and the Cutting-Planes algorithm [10].

4. Transportation Problem

The transportation problem and related problems can be found in practice in a wide variety of applications. The goal of the most transportation problems is to minimize the cost of transportation.

The first algorithm to solve a classic form of transportation problem was formulated by F.L. Hitchcock in 1941 [11]. The LP formulation is also known as the Hitchcock-Koopmans transportation problem and was published in 1949 [6]. We now provide an overview based on the description of S. Dempe and H. Schreier published in 2006 [12]. A homogeneous product is to be transported from m source nodes (supply locations) $A_i = 1, \ldots, m$, to n destination nodes (demand locations) $B_j = 1, \ldots, n$. The supply locations A_i have certain supply capacities (a_i) and the demand locations B_j have certain demands (b_j). We assume that the total supply and demand match. Furthermore, we assume that every supply location is connected to all demand locations and no transport capacities limits exist. The transport costs are proportional to the transporting quantity and proportional to the distance. The goal is to develop a minimum cost transport plan, based on the delivered quantity of goods from the supply locations A_i to the demand locations B_j. Figure 11 illustrates a transportation problem.

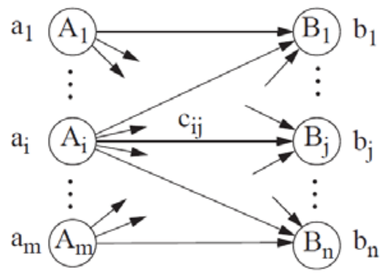


Figure 11. Transportation problem with transport cost vector c_{ij}.
Source: [10].

The formulation for this transportation problem is:

$$\min z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1.4}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad \forall i = 1, ..., m \tag{1.5}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad \forall j = 1, ..., n \tag{1.6}$$

$$x_{ij} \geq 0 \qquad \forall ij \tag{1.7}$$

The multi-level transportation problem is based on the following tasks. A product is produced at different production facilities and following transported to several warehouses. Based on the demand, these warehouses deliver the product to the customer, whereby x_{ij} is the transported amount. Thereby the production facilities and the warehouses cause fixed costs. The transport from the facility to the warehouse and from the warehouse to the customer cause variable costs. The goal is to minimize the total costs *z* by identifying a solution that the production facility and the warehouses select so that customers' demand are satisfied.

The algorithm calculates the saturation for each potential facility location to represent the relative used capacity. The potential facility location with the highest saturation will be opened. In case that the demand of the customer is still higher than the total production capacity, the algorithm finds the next best facility location to open. The algorithm to find the best warehouse follows the same process. This method determines the binary variables.

B. ILP MODEL

Indices and Sets:

$d \in \mathcal{D} = \{1,2,3,\ldots,D\}$: set of depots
$k \in \mathcal{K} = \{1,2,3,\ldots,K\}$: set of customers
$m \in \mathcal{M} = \{1,2,3,\ldots,M\}$: set of ammunition types
$n \in \mathcal{N} = \{1,2,3,\ldots,N\}$: set of potential new depots
$v \in \mathcal{V} = \{1,2\}$: vehicle types (1=regular, 2=heavy)
$t \in \mathcal{T} = \{1,2,3,\ldots,T\}$: discrete time steps (days)
$s \in \mathcal{S} = \{1,2,3,\ldots,S\}$: demand scenarios
$(k,d,v,t,t') \in \mathcal{P} \subseteq \mathcal{K} \times \mathcal{D} \times \mathcal{V} \times \mathcal{T} \times \mathcal{T}$: customer k gets ammunition from depot d using vehicle v; day t's demand is picked up on day t' ($t' \leq t$)

Parameters:

$b_{k,m,t,s} \in \mathbb{R}_+$: demand of customer $k \in \mathcal{K}$ for ammunition type $m \in \mathcal{M}$ in time step $t \in \mathcal{T}$ in scenario $s \in \mathcal{S}$ (tons)
$c_{k,d,v} \in \mathbb{R}_+$: travel time for customer $k \in \mathcal{K}$ to reach depot $d \in \mathcal{D}$ when using vehicle type $v \in \mathcal{V}$ (minutes)
$g_d \in \mathbb{R}_+$: total storage capacity of depot $d \in \mathcal{D}$ (tons)
$h_v \in \mathbb{R}_+$: storage capacity of vehicle type $v \in \mathcal{V}$ (tons)
$l_{d,m} \in \mathbb{R}_+$: storage capacity of depot $d \in \mathcal{D}$ for the ammunition $m \in \mathcal{M}$ (tons)
$w_d \in \mathbb{R}_+$: total handling capacity of depot $d \in \mathcal{D}$ (tons)
$\varrho \in \mathbb{Z}_+$: depot resupply interval (days)
$\mu \in \mathbb{Z}_+$: number of new depots to open; equal to three for our analysis (depots)

Derived Data:

$r \in \mathcal{R} = \{1,2,3,\ldots,\lceil \frac{365}{\varrho} \rceil\}$: depot resupply events

Binary Decision Variables:

o_d : is depot $d \in \mathcal{D}$ open?
$x_{k,d,v}$: does customer $k \in \mathcal{K}$ access depot $d \in \mathcal{D}$ using vehicle type $v \in \mathcal{V}$?
$u_{k,d,v,t,s}$: does customer $k \in \mathcal{K}$ send vehicle type $v \in \mathcal{V}$ to depot $d \in \mathcal{D}$ to satisfy demand for time $t \in \mathcal{T}$ in scenario $s \in \mathcal{S}$?

Objective Function:

$$\min z = \sum_{k\in\mathcal{K}} \sum_{d\in\mathcal{D}} \sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}} c_{k,d,v}\, u_{k,d,v,t,s}$$

Constraints:

$$\sum_{d\in\mathcal{D}} x_{k,d,v} \leq 1 \qquad \forall k \in \mathcal{K}, v \in \mathcal{V}$$
: each customer is assigned to at most one depot per vehicle type

$$\sum_{d\in\mathcal{D}} \sum_{v\in\mathcal{V}} u_{k,d,v,t,s} \leq 1 \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S}$$
: each customer sends at most one vehicle to an assigned depot per time step

$$\sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} \sum_{v\in\mathcal{V}} \sum_{t'\in\{(r-1)\cdot\varrho+1,\ldots,r\cdot\varrho\}} \sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s} u_{k,d,v,t,s} \leq g_d \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, r \in R$$
: total depot storage capacity

$$\sum_{k\in\mathcal{K}}\sum_{v\in\mathcal{V}}\sum_{t'\in\{(r-1)\cdot\varrho+1,\ldots,r\cdot\varrho\}}\sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s}u_{k,d,v,t,s} \leq l_{d,m} \quad \forall d\in\mathcal{D}, m\in\mathcal{M}, s\in\mathcal{S}, r\in R$$

: depot storage capacity by ammo type

$$\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{V}}\sum_{\substack{t\in\mathcal{T}:\\(k,d,v,t,t')\in\mathcal{P}}} b_{k,m,t,s}u_{k,d,v,t,s} \leq w_d \quad \forall d\in\mathcal{D}, s\in\mathcal{S}, t'\in T \qquad \text{: depot's handling capacity}$$

$$u_{k,d,v,t,s} \leq x_{k,d,v} \quad \forall k\in\mathcal{K}, d\in\mathcal{D}, v\in\mathcal{V}, s\in\mathcal{S}, t\in\mathcal{T} \qquad \text{: deploy vehicles based on depot assignments}$$

$$x_{k,d,v} \leq o_d \quad \forall d\in\mathcal{D}, k\in\mathcal{K}, v\in\mathcal{V} \qquad \text{: only assign customers to open depots}$$

$$\sum_{d\in\mathcal{N}} o_d = \mu \qquad \text{: choose new depots to open}$$

$$x_{k,d,v}, u_{k,d,v,t,s}, o_d \in \{0,1\} \quad \forall k,d,v,t,s \qquad \text{: define decision variable domains}$$

C. SUPPLY DEPOT LOCATIONS

D. OPEN STREET MAP—ROUTING SCRIPT

```
#import pyomo.environ as pyo
import pandas as pd
import openrouteservice as ors
import folium

infile = 'Routes_6994-7327_WEI.xlsx.xlsx'

client = ors.Client(key='ABCDE') #API key

###read in start and destination coords
df_SC_coords = pd.read_excel(infile, 'Routes_SC', header=0, index_col=[0])
df_DC_coords = pd.read_excel(infile, 'Routes_DC', header=0, index_col=[0])


SC_list = df_SC_coords.values.tolist()
DC_list = df_DC_coords.values.tolist()

outputList =[]
####
m = folium.Map(location=[49.460116, 11.029364], tiles='cartodbpositron', zoom_start=7)

for f, b in zip(SC_list, DC_list):
    coordinates = [f,b] #loop over coordinates

    #routing process
    for idx, coords in enumerate(coordinates):
        folium.Marker(location=list(reversed(coords)),
                      popup=folium.Popup("ID: {}".format(idx))).add_to(m)

    route = client.directions(
        coordinates=coordinates,
        #profile='driving-car',
        profile='driving-hgv',
        format='geojson',
        preference='fastest',
        units='km',
        instructions=False, #turn instructions off
        validate=False,
        #optimize_waypoints=True
    )

    folium.PolyLine(locations=[list(reversed(coord))
                               for coord in
                               route['features'][0]['geometry']['coordinates']]).add_to(m)

    #print(route['features'][0]['properties']['summary'])

    outputList.append((route['features'][0]['properties']['summary'])) #write results to a list

###export data to excel file
df = pd.DataFrame.from_dict(outputList)
df.to_excel('routing_output_WEI.xlsx')
```

E. PYOMO—SCRIPT

```
################# Pyomo model - Optimization of supply chains #################
# GRAMANN, Alexander, MAJ German Army
##############################################################################
from pyomo.opt import SolverFactory
import pyomo.environ as pyo
import pandas as pd
import time
import math

start = time.time() #start timer first block
model = pyo.ConcreteModel() #create a local variable with instance of Concrete model

############################## Load data ###################################
infile = 'data_UC2.xlsx' #read in file
df0_days = pd.read_excel(infile, '00-Days', header=0, index_col=0)
df1_depots = pd.read_excel(infile, '01-Depots', header=0, index_col=0)
df2_customers = pd.read_excel(infile, '02-Customers', header=0, index_col=0)
df3_ammo = pd.read_excel(infile, '03-Ammo', header=0, index_col=0)
#df4_demand = pd.read_excel(infile, '04-Demand', header=0, index_col=0)
#df5_vehicles = pd.read_excel(infile, '05-Vehicles', header=0, index_col=0)
df6_depots_reopen = pd.read_excel(infile, '06-DepotsReopen', header=0, index_col=0)

df_b = pd.read_excel(infile, '04-Demand', header=0, index_col=[0,1,2,3])
df_c = pd.read_excel(infile, '07-TravelTimes', header=0, index_col=[0,1])
df_l = pd.read_excel(infile, '10-DepotAmmoCap', header=0, index_col=0)
df_h = pd.read_excel(infile, '05-Vehicles', header=0, index_col=0)
df_g = pd.read_excel(infile, '08-TotStorageCap', header=0, index_col=0)
df_w = pd.read_excel(infile, '09-HandlingCap', header=0, index_col=0)
rho = 30 # resupply interval (days)
```

```
############################# Sets and indices ##############################
R = [i+1 for i in range(math.ceil(365/rho))] # set of resupply events
model.R=pyo.Set(initialize=R)

D = list(df1_depots['DepotName']) #set of depots
model.D=pyo.Set(initialize=D)

K = list(df2_customers['CustomerName']) #set of customers
model.K=pyo.Set(initialize=K)

M = list(df3_ammo['AmmoType']) #set of ammo types
model.M=pyo.Set(initialize=M)

N = list(df6_depots_reopen['DepotName']) #set of potential new depots
model.N=pyo.Set(initialize=N, within=model.D)

V = ['regular','heavy'] #vehicle types
model.V=pyo.Set(initialize=V)

T = list(df0_days['Days']) #set of days
model.T=pyo.Set(initialize=T, ordered=True)

#S = ['s1','s2','s3','s4','s5'] #scenario types
S = ['s1']
model.S=pyo.Set(initialize=S)
PickUp = set([]) #customer k gets ammunition from depot d using vehicle v; day t's
                    #demand is picked up on day t'
DepotPickupDays = set([]) #we might possibly have a pickup from depot d on day
                              #tprime, based on demand
```

```
################################ Parameters ##############################
#travel time for customer kK to reach depot dD when using vehicle type vV(minutes)
regular_time = df_c['Regular[min]'].to_dict()
heavy_time = df_c['Heavy[min]'].to_dict()

X = list(df_c.index)
model.c = {(x[0], x[1], 'regular'):regular_time[x] for x in X}
for x in X:
    model.c[x[0], x[1], 'heavy'] = heavy_time[x]

###fill the PickUp set
thresVar = 240 #threshold variable for travel time

#demand of customer kK for ammunition type mM in time step tT in scenario sS(tons)
b = df_b['Demand[tons]'].to_dict()
#print('demand',demand)

bset = set(b.keys())
for k in K:
    for t in T:
        for s in S:
            if sum(b[k,m,t,s] for m in M if (k,m,t,s) in bset)>0:
                for d in D:
                    for v in V:
                        if model.c[k,d,v] <= thresVar:
                            PickUp.add((k,d,v,t,t)) #pick up same day
                            DepotPickupDays.add((d,t))
                        else:
                            PickUp.add((k,d,v,t,int(t)-1)) #pick up day before
                            DepotPickupDays.add((d,t-1))
```

```
################# START - generating random demand data ###################
b = {}
for k in K:
    for t in T:
        for s in S:
            for m in M:
                temp = random.random() #generate a random number between 0 and 1.
                if temp <= 0.033: #If that number is less than 0.033, you have an order today.
                    new_temp = random.triangular(0,5,0.5)
                    b[k,m,t,s] = new_temp #add demand values to dict
bset = set(b.keys())
for k in K:
    for t in T:
        for s in S:
            if sum(b[k,m,t,s] for m in M if (k,m,t,s) in bset)>0:
                for d in D:
                    for v in V:
                        if model.c[k,d,v] <= thresVar:
                            PickUp.add((k,d,v,t,t)) #pick up same day
                            DepotPickupDays.add((d,t))
                        else:
                            PickUp.add((k,d,v,t,int(t)-1)) #pick up day before
                            DepotPickupDays.add((d,t-1))

################## END - generating random demand data ####################

#storage capacity of depot dD(tons) for the ammunition mM(tons)
X = list(df_l.index)
Y = list(df_l.columns)
l = {(x,y):df_l.at[x,y] for x in X for y in Y}
```

```
#total storage capacity of depot dD(tons)
X = list(df_g.index)
Y = list(df_g.columns)
g = {x:df_g.at[x,y] for x in X for y in Y}

#loading capacity of vehicle type vV(tons)
X = list(df_h.index)
Y = list(df_h.columns)
h = {x:df_h.at[x,y] for x in X for y in Y}

#total handling capacity of depot dD(tons)
X = list(df_w.index)
Y = list(df_w.columns)
w = {x:df_w.at[x,y] for x in X for y in Y}

######################### Binary decision variables #########################
#stage one decision variables
model.o = pyo.Var(N, within=pyo.Binary) #is depot dD open?
model.x = pyo.Var(K,D,V, within=pyo.Binary) #does customer kK access depot dD
                                            # using vehicle type vV?

#stage two decision variables
model.u = pyo.Var(K,D,V,T,S, within=pyo.Binary) #does customer kK send vehicle type
                        #vV to depot dD to satisfy demand for time tT in scenario sS?

############################# Objective function ############################
def obj_fct(model):
   return sum(model.c[k,d,v]*model.u[k,d,v,t,s] for k in K for d in D for v in V for t in T
for s in S)
```

```
model.obj = pyo.Objective(rule=obj_fct, sense=pyo.minimize)

############################### Constraints ###############################
#each customer is assigned to at most one depot per vehicle type
def const1(model,k,v):
    return sum(model.x[k,d,v] for d in D) <= 1
model.const1 = pyo.Constraint(K,V, rule=const1)

#each customer sends at most one vehicle to an assigned depot on a certain day
def const2(model,t,k,s):
    return sum(model.u[k,d,v,t,s] for d in D for v in V) <= 1
model.const2 = pyo.Constraint([(t,k,s) for t in model.T for k in model.K for s in model.S
if sum(b[k,m,t,s] for m in M if (k,m,t,s) in bset)>0], rule=const2)

#total depot storage capacity EMC
def const3(model,d,r,s):
    kmvttp_list = [(k,m,v,t,tprime) for v in V for (k,m,t,s) in bset for tprime in T if
(tprime>=(r-1)*rho+1 and tprime<=r*rho) if (d,tprime) in DepotPickupDays if
(k,d,v,t,tprime) in PickUp]
    if len(kmvttp_list)==0:
        return pyo.Constraint.Skip
    return sum(b[k,m,t,s]*model.u[k,d,v,t,s] for (k,m,v,t,tprime) in kmvttp_list) - g[d] <= 0
model.const3 = pyo.Constraint([(d,r,s) for d in D for s in S for r in R], rule=const3)

#depot storage capacity by ammo type EMC
def const4(model,d,m,r,s):
    ktvtprime_list = [(k,t,v,tprime) for k in K for t in T for tprime in T if (k,m,t,s) in bset
for v in V if (k,d,v,t,tprime) in PickUp if (d,tprime) in DepotPickupDays]
    if len(ktvtprime_list)==0:
        return pyo.Constraint.Skip
```

```
    return sum(b[k,m,t,s]*model.u[k,d,v,t,s] for (k,t,v,tprime) in ktvtprime_list) <= l[d,m]
model.const4 = pyo.Constraint([(d,m,r,s) for d in D for m in M for s in S for r in R if
r>1], rule=const4)

#storage capacity by vehicle type
def const5(model,k,t,s):
    return sum(b[k,m,t,s] for m in M if (k,m,t,s) in bset) - sum(h[v]*model.u[k,d,v,t,s] for
d in D for v in V) <= 0
model.const5 = pyo.Constraint(K,T,S, rule=const5)

#depot's handling capacity
def const6(model,d,tprime,s):
    kmtv_list = [(k,m,t,v) for k in K for t in T for m in M if (k,m,t,s) in bset for v in V if
(k,d,v,t,tprime) in PickUp]
    if len(kmtv_list)==0:
        return pyo.Constraint.Skip
    return sum(b[k,m,t,s]*model.u[k,d,v,t,s] for (k,m,t,v) in kmtv_list) <= w[d]
model.const6 = pyo.Constraint([(d,tprime,s) for (d,tprime) in DepotPickupDays for s in
S], rule=const6)

#deploy vehicles based on depot assignments
def const7(model,k,d,v,t,s):
    return model.u[k,d,v,t,s] <= model.x[k,d,v]
model.const7 = pyo.Constraint(K,D,V,T,S, rule=const7)

#only assign customers to open depots
def const8(model,n,k,v):
    return model.x[k,n,v] <= model.o[n]
model.const8 = pyo.Constraint(N,K,V, rule=const8)
```

```
#choose new depots to open
def const9(model):
    return sum(model.o[n] for n in N) == 3
model.const9 = pyo.Constraint(rule=const9)

print("time of block 1:", time.time() - start) #print time first block
start = time.time() #start timer second block

######################### SOLVE & PRINT ##############################
opt = pyo.SolverFactory('gurobi')
#opt = pyo.SolverFactory("cbc",executable='cbc.exe') #solver
results = opt.solve(model)
#print(results)

#model.pprint() #print everything
print("Total travel time: ",model.obj(),"min") #print obj val
#model.display() #print var, obj and constraint

#results = opt.solve(model, tee=True) #print solver output
#model.x.pprint() #print x-var
#model.u.pprint()

print("time of block 2:", time.time() - start) #print time second block
```

##

############################## CREATE OUTPUT ##############################

```
#which customer access which depot with vehicle type
for n in model.x:
    #print(n)
    if model.x[n] == 1:
        print('\t',n)
```

```
#which customer access which depot with what vehicle type at what day
uSet = set()

for n in model.u:
    #print(n)
    if model.u[n] == 1:
        #print('\t',n)
        uSet.add(n)

#type(model.u)
#type(uSet)
uSet
```

```
#print('depot pickup set=',DepotPickupDays)
```

```
#print('pickup set=',PickUp)
#for val in PickUp:
#    print (val)
sorted(PickUp)
```

```
sorted(uSet)
```

```
print(len(uSet))
print(len(PickUp))
```

```
#how to find the appropriate items from uSet in PickUp?
#we use lists to merge the 2 sets

listUset = list(sorted(uSet))
#listUset
listPickUp = list(sorted(PickUp))
#listPickUp
c = listUset + listPickUp
c.sort()
#type(c)
c
```

```
#now we want to know the pickup days
#therefore we look for the items from the original uSet with length 4 in the new list c and then print the next item

indexList = []
for item in c:
    #print(len(item))
    if len(item) == 4:
        index = c.index(item)
        indexList.append(index)
        #print(index)

for item in indexList:
    print(c[item+1])

###export data to excel file
df = pd.DataFrame.from_dict(indexList)
df.to_excel('routing_output_ALL_2017.xlsx')
```

F. R—SCRIPT FOR GRAPHICAL RESULTS

```
#import libraries
library(ggmap)
library(ggplot2)

register_google(key = “xxx”) #API key; valid until 20th of April 2020

#read in csv
#depots.old <- read.csv(“/tmp/depots.csv”)
data1  <- read.csv(“/tmp/final_results_R_map1.csv”)
data2  <- read.csv(“/tmp/final_results_R_map2.csv”)

###regular vehicle
#data1
depots1 <- data1[which(data1$color.group==“ASP”),]  ### THESE ARE THE DEPOTS
customers1 <- data1[which(data1$color.group==“Customer”),]
customers1$id <- NA

for (i in 1:nrow(customers1)) {
 id = paste(customers1[i,]$pair, i, sep=““)
 customers1[i,]$id <- id
 customers1 <- rbind(
   customers1,
   cbind(depots1[which(depots1$pair==customers1[i,]$pair),], id=id)
 )
}

###heavy vehicle
#data2
```

```
depots2 <- data2[which(data2$color.group==“ASP”),]  ### THESE ARE THE DEPOTS
customers2 <- data2[which(data2$color.group==“Customer”),]
customers2$id <- NA

for (i in 1:nrow(customers2)) {
 id = paste(customers2[i,]$pair, i, sep=““)
 customers2[i,]$id <- id
 customers2 <- rbind(
   customers2,
   cbind(depots2[which(depots2$pair==customers2[i,]$pair),], id=id)
 )
}

data_temp1 <- customers1
data_temp2 <- customers2

# getting the map
mapgilbert <- get_map (location = c(lon = mean(depots1$lon), lat = mean(depots1$lat)), zoom = 6, maptype = “hybrid,” scale = 2, API_console_key = Sys.getenv(“xxx”))

# plotting the map with the depot/customer connections
ggmap(mapgilbert) +
 geom_point(data = data_temp1, aes(x = data_temp1$lon, y = data_temp1$lat, color = data_temp1$color.group)) +
 geom_line(data = data_temp1, color=‘gold’, size=0.5, linetype=‘longdash’, aes(x = data_temp1$lon, y = data_temp1$lat, group = data_temp1$id))+
 geom_line(data = data_temp2, color=‘red’, size=0.5, linetype=‘longdash’, aes(x = data_temp2$lon, y = data_temp2$lat, group = data_temp2$id))
```

PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] A. Atamturk and M. Zhang, "Two-stage robust network flow and design under demand uncertainty." *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007.

[2] E. Craparo, M. Karatas, and D. I. Singham, "A robust optimization approach to hybrid microgrid operation using ensemble weather forecasts," *Journal of Applied Energy*, vol. 201, pp. 135–147, Sept. 2017.

[3] L. M. Johannsmann, "Optimized spare parts inventory management for military deployment." Master's thesis, Angewandte Mathematik und Optimierung Schriftenreihe AMOS #63, 2017

[4] L. M. Johannsmann, E. M. Craparo, T. L. Dieken, A. R. Fuegenschuh, and B. O. Seitner, "Stochastic mixed-Integer programming for a spare parts inventory management problem," working paper.

[5] "Pyomo," *Wikipedia.* Accessed May 05, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Simplex_algorithm

[6] T. Ellinger, *Operations Research: Eine Einführung*. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 1990.

[7] K. Neumann, M. Morlock, *Operations Research*. Munich, Germany: Carl Hanser Verlag, München, 1993.

[8] "Simplex algorithm," *Wikipedia*. Accessed July 16, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Simplex_algorithm

[9] B. Chachuat, "Mixed-integer linear programming (MILP): Model formulation," McMaster University Department of Chemical Engineering. Accessed July 17, 2019. [Online]. Available: http://macc.mcmaster.ca/maccfiles/chachuatnotes/07-MILP-I_handout.pdf

[10] W. Domschke, A. Drexl, R. Klein, A. Scholl, *Einführung in Operations Research*. Berlin and Heidelberg, Germany: Springer-Verlag, Berlin Heidelberg, 2015.

[11] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *MIT Journal of Mathematics and Physics*, vol. 20, pp. 224–230, April 1941.

[12] S. Dempe, H. Schreier, *Operation Research Deterministische Modelle und Methoden*. Wiesbaden, Germany: Teubner Verlag, Wiesbaden, 2006.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California