



universität
wien

MAGISTERARBEIT / MASTER'S THESIS

Titel der Magisterarbeit / Title of the Master's Thesis

Bikriterielle ungleichheits-averse Standortoptimierung für Hilfseinrichtungen in Erdbebenregionen

verfasst von / submitted by

Raphael Rath, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

**Magister der Sozial- und Wirtschaftswissenschaften
(Mag.rer.soc.oec.)**

Wien, 2018 / Vienna, 2018

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 951

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Magisterstudium Statistik

Betreut von / Supervisor:

ao. Univ.-Prof. Dr. Walter Gutjahr

Abstract

Im Falle einer Naturkatastrophe hat die Bereitstellung von medizinischer Versorgung, Nahrungsmitteln und einer sicheren Unterkunft höchste Priorität. Um diese Grundversorgung gewährleisten zu können, müssen Katastrophenpläne erstellt und Standorte für Notversorgungseinrichtungen ausgewählt werden. In dieser Arbeit wird ein Modell zur optimalen Standortbestimmung in Erdbebenregionen entwickelt. Dabei soll einerseits die Effizienz maximiert und gleichzeitig die Ungleichheit minimiert werden. Durch die Berücksichtigung mehrerer Erdbebenszenarien mit bestimmten Eintrittswahrscheinlichkeiten können diese beiden Gesichtspunkte nun sowohl ex ante, als auch ex post, d.h. vor bzw. nach dem Eintreten des Erdbebens, betrachtet werden. Das entwickelte Modell besteht aus einer ex ante, sowie einer ex post Zielfunktion, welche Linearkombinationen aus der mittleren zurückzulegenden Distanz (dadurch wird Effizienz garantiert) und der „Gini’s mean absolute difference“ (ein Maß für Ungleichheit) sind. Beide Zielfunktionen werden anhand des adaptive epsilon constraint Algorithmus verknüpft, wodurch man alle pareto optimalen Lösungen für die vorliegende Problemstellung erhält. In zwei Anwendungsbeispielen werden die Ergebnisse des Modells für simulierte Daten ausführlich diskutiert. Anschließend wird eine Laufzeitanalyse durchgeführt und es werden Ergebnisse für verschiedene Parameter-settings verglichen. Die Entscheidungsfindung bei der Standortwahl wird durch die Auswahlmöglichkeit aus mehreren pareto optimalen Lösungen erleichtert und bleibt gleichzeitig flexibel. Je nach Präferenzen und Gegebenheiten kann somit für eine vorliegende Problemstellung die beste Lösung aus dem Set aller optimalen Lösungen gewählt werden.

Abstract

The top priority in the case of a natural disaster is the provision of medical care, food and a safe shelter. Disaster plans have to be set up and locations for shelters have to be chosen in order to be able to guarantee the provision of those basic supplies. In this work a model for the optimal selection of shelter locations in earthquake areas is developed. For this purpose, the efficiency is maximised and at the same time inequality is minimised. Furthermore, different earthquake scenarios are taken into consideration and therefore efficiency and inequality can be measured ex ante and ex post, which means before or after the event of an earthquake. The model consists of an ex ante and an ex post objective function, which are linear combinations of the mean distance (which measures efficiency) and the gini's mean absolute difference (which measures inequality). Both objective functions are combined using the adaptive epsilon constraint algorithm, yielding all pareto optimal solutions. Using simulated data the results of two sample applications are discussed. Additionally, a runtime analysis is conducted and the results for different parameter settings are compared. The decision for the best shelter location is supported by the possibility to be able to choose between multiple pareto optimal solutions while remaining flexible. Depending on preferences and the given circumstances, the best fitting solution can be chosen amongst the set of all optimal solutions.

Inhaltsverzeichnis

1	Einleitung	2
2	Theorie	6
2.1	Ungleichheit und Unsicherheit	6
2.2	Fragestellung	8
2.2.1	Notation	8
2.3	Modell	10
2.3.1	Zusammensetzung der Zielfunktionen	10
2.3.2	Ex ante Zielfunktion und MIP Formulierung	11
2.3.3	Ex post Zielfunktion und MIP Formulierung	14
2.3.4	Adaptive Epsilon Constraint Algorithmus	16
3	Simulation	19
3.1	Datengenerierung und Parameter	19
3.2	Implementierung	21
3.3	Anwendungsbeispiel 1	23
3.3.1	Einführung	23
3.3.2	Ergebnisse	24
3.4	Anwendungsbeispiel 2	31
3.5	Laufzeitanalyse	35
3.6	Variation des λ Parameters	37
4	Fazit	40
	Appendices	41
A	Detaillerggebnisse für verschiedene λ-Parameter	42

Abbildungsverzeichnis

2.1	Gini Koeffizient für zwei Einrichtungen	7
3.1	Zufallsgenerierte Daten mit 40 PK und 7 Szenarien	20
3.2	Pareto optimale Platzierungen der Einrichtungen bei 40 PK und 7 Szenarien	26
3.3	Pareto-Front für Anwendungsbeispiel 1 (von rechts unten nach links oben)	28
3.4	Tabelle mit Informationen zu jeder Lösung	29
3.5	Veranschaulichung der Werte aus Abb. 3.4	30
3.6	Zufallsgenerierte Daten mit 30 PK und 20 Szenarien	32
3.7	Pareto-optimale Platzierungen der Einrichtungen bei 30 PK und 20 Szenarien	33
3.8	Funktionswerte von jeder Lösung (von rechts unten nach links oben) . . .	34
3.9	Tabelle mit Informationen zu jeder Lösung	34
3.10	Laufzeit für versch. Kombinationen von Szenarien und Knoten. Die Werte auf der x-Achse geben die Anzahl der Szenarien (1. Zahl) und Anzahl der PK und KS (immer gleich hoch, 2. Zahl) an	36
3.11	Variablenanzahl vs. Laufzeit	37
3.12	Vergleich bei Variation des λ Parameters nach der ersten Iteration	39
A.1	Ex ante und ex post Zielfunktionswerte	43
A.2	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0$	44
A.3	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,25$	45
A.4	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,27$	46
A.5	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,28$	47
A.6	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,75$	48
A.7	Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 1$	49
A.8	Ergebnisse für $\lambda = 0$	50
A.9	Ergebnisse für $\lambda = 0,25$	51

A.10 Ergebnisse für $\lambda = 0,27$	52
A.11 Ergebnisse für $\lambda = 0,28$	53
A.12 Ergebnisse für $\lambda = 0,75$	54
A.13 Ergebnisse für $\lambda = 1$	55

Abkürzungsverzeichnis

ADTS	Average distance to shelter ($= \mu$)
GMAD	Ginis mean absolute difference
KS	Kandidaten Standort
MIP	Mixed integer programming
NB	Nebenbedingung
PK	Populationsknoten
ZF	Zielfunktion

1. Einleitung

In dieser Arbeit wird ein Modell vorgestellt, welches für eine gegebene Problemstellung mehrere optimale Lösungen zur Platzierung von Notversorgungseinrichtungen in einem Erdbebengebiet liefert, bei denen alle betroffenen Bewohnerinnen und Bewohner Zuflucht suchen können. Neben dem Ziel die Wege für alle Personen möglichst kurz zu halten, sollen die Standorte auch gerecht verteilt werden um der gesamten Bevölkerung einen guten Zugang zu den Hilfseinrichtungen zu ermöglichen. Die Fragestellung ist ähnlich wie in der Arbeit von Mostajabdaveh et al. [11], wobei in dieser Arbeit ein alternatives Modell zur Optimierung verwendet wird. Statt einer gewichteten Zielfunktion werden zwei Zielfunktionen definiert, wodurch mithilfe des von Laumanns et al. [8] vorgestellten adaptive epsilon-constraint Algorithmus alle pareto-optimalen Lösungen gefunden werden können. Dies bietet für einen Entscheidungsträger den großen Vorteil, dass eine große Menge von realisierbaren Varianten auf ein viel kleineres Set von Alternativen eingeschränkt werden kann, sodass aus diesen Lösungen dann je nach Präferenz die Beste für das vorliegende Problem gewählt werden kann. Dabei gilt für jede pareto-optimale Lösung, dass keine Zieleigenschaft verbessert werden kann, ohne dabei eine andere zu verschlechtern (vgl. Gutjahr und Nolz [6]). Die Zieleigenschaften in der vorliegenden Fragestellung sind dabei eine sowohl effiziente als auch gerechte Platzierung der Hilfseinrichtungen.

Wie bereits erwähnt, hat das Modell eine optimale Platzierung von Notversorgungsstellen in einer Erdbebenregion zum Ziel, wobei das Modell natürlich auch adaptiert werden könnte, um für die Entscheidungsfindung bei der Platzierung von verschiedensten Arten von Einrichtungen unter Unsicherheit Verwendung zu finden. Mit Unsicherheit sind dabei Ereignisse gemeint, für die es mehrere Eintrittsszenarien mit bestimmten Eintrittswahrscheinlichkeiten gibt, wie z.B. Erdbeben, Dürren, Hochwasser oder auch andere Ereignisse wie Kriegsszenarien oder großflächige Stromausfälle.

Während bei den meisten ökonomischen Fragestellungen der Profit maximiert oder Kosten minimiert werden sollen, gilt es bei humanitären Fragestellungen jedoch nicht nur die Effizienz zu optimieren, sondern es sollen auch faire Bedingungen hergestellt werden (vgl. Mostajabdaveh et al. [11]). D.h. es soll sichergestellt werden, dass keine der betroffenen Personen einen zu großen Nachteil - z.B. in Form von zu erhaltenden Ressourcen oder zurückzulegenden Distanzen - erleiden. Daher müssen Modelle, welche diese Anforderungen erfüllen die Effizienz optimieren und gleichzeitig eine gewisse Gleichheit garantieren. In der Literatur werden solche Modelle als 'inequity-averse', also ungleichheits-avers oder 'equitable', also gerecht oder fair bezeichnet. Einen Überblick über verschiedene Methoden und Ansätze, die neben der Effizienz auch Gleichheit miteinbeziehen, bieten z.B. Karsu und Morton [7], oder Marsch und Schilling [9].

In der vorliegenden Fragestellung soll neben einer möglichst effizienten Platzierung - d.h. die zurückzulegende Distanz für alle Betroffenen soll möglichst gering sein - auch eine gewisse Gleichheit bzw. Fairness sichergestellt werden. Außerdem muss zusätzlich die stochastische Komponente von verschiedenen Erdbebenszenarien, die mit gewissen Wahrscheinlichkeiten auftreten, berücksichtigt werden. Die Ungewissheit (hier die verschiedenen Erdbebenszenarien) in Kombination mit der Partizipation der Ungleichheit führt zu dem Problem, dass diese nun ex ante oder ex post gemessen werden kann (vgl. Mostajabdaveh et al. [11]). Konkret bedeutet das für die vorliegende Fragestellung, dass die Ungleichheit entweder vor dem Eintreten des Ereignisses eines Erdbebens (ex ante) oder danach (ex post), wenn also keine Unsicherheit mehr besteht, gemessen werden kann (genauer wird dies in Kapitel 2 erläutert). Auf die Problematik der Messung von Ungleichheit unter Unsicherheit geht Ben-Porath et al. [1] genauer ein.

Es gibt nun verschiedene Varianten um beide Aspekte der ex ante und ex post Ungleichheit zu berücksichtigen, wie z.B. die Verwendung eines gewichteten Mittels (dieser Ansatz wurde von Mostajabdaveh et al. [11] umgesetzt). In dieser Arbeit wird statt eines gewichteten Mittels eine ex ante und eine ex post Zielfunktion definiert und diese werden mit dem von Laumanns et al. [8] vorgestellten adaptive epsilon-constraint Algorithmus verknüpft. Um sowohl die Effizienz (der zurückzulegende Weg soll möglichst kurz sein) als auch Gleichheit (die Distanzen sollen für alle Betroffenen ähnlich groß sein) zu berücksichtigen, bestehen beide Zielfunktionen jeweils aus zwei Teilen. Der erste Teil dient zur Optimierung der Effizienz und fließt als Mittelwert aller zurückzulegenden Wege in das Modell ein. Der zweite Teil soll eine gewisse Gleichheit sicherstellen und basiert auf dem Gini-Koeffizienten (s. Gini [5]) in einer etwas abgewandelten Form. Die Wahl der beiden Komponenten wird im Kapitel 2 genauer erläutert und begründet.

Die epsilon constraint Methode kann grundsätzlich für beliebig viele Zielfunktionen (ZF) angewendet werden. Im vorliegenden Fall von zwei ZF wird die erste ZF minimiert, während die andere als zusätzliche Nebenbedingung (NB) integriert wird. Zu Beginn ist diese NB noch unbeschränkt, d.h. die erste ZF wird ohne Einschränkungen minimiert. Nach dem ersten Schritt wird diese zusätzliche Nebenbedingung jedoch bei jeder Iteration strenger, sodass nach und nach immer neue Lösungen gefunden werden. Dies wird so lange fortgesetzt bis die NB so streng ist, dass keine Lösung mehr gefunden werden kann (s. Kapitel 2.3.4 für eine genaue Erklärung des Algorithmus). Mit dieser Methode erhält man die gesamte Pareto-Front, d.h. alle Lösungen, bei denen keine Zieleigenschaft verbessert werden kann, ohne eine andere zu verschlechtern. Konkret kann also bei einer pareto optimalen Lösung die Lage keiner Person verbessert werden (z.B. indem die zurückzulegende Distanz verringert wird), ohne eine andere Person schlechter zu stellen. Dies schränkt die Auswahlmöglichkeiten für einen Entscheidungsträger auf alle optimalen Lösungen ein und es kann nun je nach Präferenz die beste Lösung gewählt werden. Bei der Verwendung eines gewichteten Mittels aus beiden Zielfunktionen erhält man hingegen nur eine Lösung. Durch Variation der Gewichtungsfaktoren können zwar weitere Lösungen gefunden werden, diese liegen aber alle in einer konvexen Region des Zielfunktionsraumes (vgl. Laumanns et al. [8]). Dadurch wird die Anzahl der gefundenen Lösungen stark eingeschränkt (s. Kapitel 3.3, wo der Großteil der gefundenen Lösungen nicht in einer konvexen Region liegt).

Im folgenden Kapitel wird zuerst allgemein auf die Problemstellung, sowie die Begriffe Ungleichheit und Unsicherheit im Kontext der humanitären Optimierung und Standortbestimmung eingegangen. Anschließend wird zuerst die im Modell verwendete Notation genauer erklärt und anschließend der theoretische Aufbau des Modells diskutiert. Dabei wird zuerst die ex ante und anschließend die ex post Zielfunktion definiert und erklärt. Im darauffolgenden Kapitel wird die Verknüpfung der beiden Zielfunktionen mittels des adaptive epsilon-constraint Algorithmus erläutert und das finale Modell spezifiziert. Nach der vollständigen Beschreibung des theoretischen Settings und der Zielfunktionen wird in jedem Kapitel eine mixed integer programming (MIP) Formulierung des Modells eingeführt, welche in weiterer Folge zur Implementierung in dem Programm Matlab [10] dient. Zur Optimierung wird die IBM ILOG CPLEX [4] Optimierungs-Toolbox für Matlab verwendet. Eine genauere Beschreibung der Implementierung wird im Kapitel 3.2 als Vorbereitung für die Anwendungsbeispiele gegeben.

Anschließend werden mehrere Simulationen mit verschiedenen Parametern in Bezug auf die Anzahl der zu berücksichtigenden Populationsknoten, geplanter Notversorgungsstel-

len, sowie Erdbebenszenarien durchgeführt. Bei der Analyse der Ergebnisse wird neben der Qualität der Lösungen in Bezug auf Effizienz und Gleichheit außerdem die Laufzeit bei verschiedenen Zusammensetzungen der Parameter diskutiert. Im letzten Kapitel werden die Parameter so verändert, dass entweder die Effizienz- oder die Ungleichheitskomponenten der Zielfunktionen höher bzw. niedriger gewichtet werden. Die daraus resultierenden Verteilungen der Standorte und die pareto optimalen Ergebnisse werden dann sowohl untereinander als auch mit den Ergebnissen aus den vorigen Kapiteln verglichen.

2. Theorie

2.1 Ungleichheit und Unsicherheit

In der Literatur gibt es für verschiedenste operations-research Problemstellungen Lösungsvorschläge zur Optimierung der Gleichheit (im Sinne von Fairness). Einen Überblick über verschiedene Methoden zur Messung von Ungleichheit sowie die Anwendung in unterschiedlichen Fragestellungen geben Karsu und Morton [7]. Eine Zusammenfassung der wichtigsten Methoden zur Messung von Ungleichheit bieten Marsh und Schilling [9]. Zu den am häufigsten verwendeten Methoden zählen dabei neben dem Ansatz von Rawls [13] - bei dem z.B. bei der Platzierung von Einrichtungen die größte Distanz minimiert wird - vor allem der Gini-Koeffizient bzw. Gini-Index (s. Gini [5]), für den es eine Vielzahl von verschiedenen Repräsentationsarten gibt (s. z.B. Kapitel 2 in Yitzhaki et al. [14]).

Wenn man nun zur Minimierung der Ungleichheit - welche sich in der vorliegenden Fragestellung auf die zurückzulegende Distanz zur nächsten Notversorgungsstelle bezieht - den Gini-Koeffizienten verwendet, erhält man in vielen Fällen sehr ineffiziente Lösungen. Konkret bekommt man dann oft Ergebnisse, bei denen die zu eröffnende Hilfseinrichtung sehr weit von allen Betroffenen entfernt sind. Dadurch ist die Distanz zwar sehr groß, dies gilt aber für alle Betroffenen, wodurch die Ungleichheit und somit der Gini-Index minimal sind. Abb. 2.1 veranschaulicht den oben beschriebenen Effekt anhand eines Beispiels mit zwei möglichen Standorten für eine Einrichtung, sowie zwei Populationsknoten (z.B. Dörfer oder Häuser). Man erkennt sofort, dass die Einrichtung im Punkt $[2, 2]$ gebaut werden sollte, da bei dieser Option beide Nachfrageknoten eine kürzere Distanz überwinden müssen, als wenn die Einrichtung im Punkt $[2, 10]$ errichtet werden würde. Minimiert man jedoch den Gini-Koeffizienten, erhält man letzteren Punkt als Lösung, da in diesem Punkt die Ungleichheit bzgl. der zurückzulegenden Distanz geringer ist. Auf diese Problematik gehen z.B. Drezner et al [3] genauer ein.

Um solche ineffizienten Lösungen zu vermeiden, muss also eine Funktion verwendet wer-

den, welche neben der Gleichheit auch die Effizienz (z.B. in Form von einer mittleren zurückzulegenden Distanz) miteinbezieht. Eine genaue Beschreibung und Begründung der Wahl der Zielfunktion für die vorliegende Fragestellung wird im nachfolgenden Kapitel angegeben.

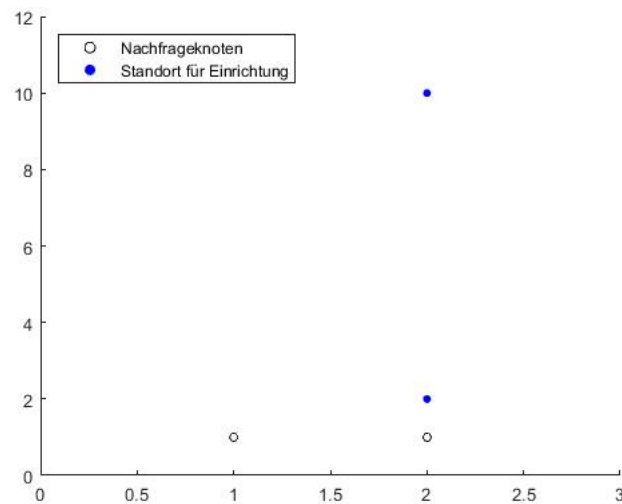


Abbildung 2.1: Gini Koeffizient für zwei Einrichtungen

Neben der Herausforderung eine Zielfunktion zu definieren, welche sowohl Ungleichheit als auch Effizienz miteinbezieht, muss für die vorliegende Fragestellung außerdem noch die Unsicherheit in Form der verschiedenen Erdbebenszenarien berücksichtigt werden. Die Problemstellung kann also nicht mehr in einem deterministischen Kontext betrachtet werden, sondern es müssen zusätzlich noch die Szenarien mit ihren Eintrittswahrscheinlichkeiten miteinbezogen werden. Dadurch gilt es wiederum weitere Aspekte bei der Definition der Zielfunktion zu berücksichtigen. Genauer gesagt läuft es in diesem Fall darauf hinaus, dass zwei Zielfunktionen definiert werden, die dann später mit dem adaptive epsilon constraint Algorithmus wieder miteinander verknüpft werden (s. Kapitel 2.3.1).

Dies ist der Problematik geschuldet, dass man in einem stochastischen Kontext die Ungleichheit sowohl vor dem Eintreten des Ereignisses, als auch danach - wenn also keine Unsicherheit mehr besteht - messen kann. Wie bereits in der Einleitung erwähnt, bezeichnet man dies als ex ante bzw. ex post Ungleichheit. Ben-Porath et al. [1] gehen auf die konzeptuelle Schwierigkeit bei der Messung von Ungleichheit unter Unsicherheit ein

und schlagen als Lösungsansatz sogenannte min-of means functionals vor. Dieses Konzept wurde jedoch aus einer ökonomischen bzw. entscheidungstheoretischen Sichtweise heraus entwickelt, ohne dabei auf mögliche Fragestellung bzgl. der Optimierung (im humanitären Bereich) einzugehen. Dieser Schritt wird von Mostajabdaveh et al. [11] sowohl theoretisch, als auch praktisch anhand eines Anwendungsbeispiels zur Optimierung eines 'shelter-location'-Problems dargestellt. Im folgenden Abschnitt wird noch einmal genauer auf die Fragestellung eingegangen und die für den nachfolgenden Theorieteil benötigte Notation spezifiziert.

2.2 Fragestellung

Bei der vorliegenden Problemstellung sollen optimale Standorte für Notversorgungseinrichtungen gefunden werden, welche im Falle eines Erdbebens eine Anlaufstelle für die Bevölkerung der umliegenden Dörfer darstellen. Die Einrichtungen sollen sowohl als Zufluchtsort dienen, als auch Hilfsgüter sowie medizinische Versorgung bereitstellen um so im Katastrophenfall eine Grundversorgung für die Bevölkerung zu garantieren. Neben einer möglichst effizienten Platzierung soll auch darauf geachtet werden, dass auch abgelegene Dörfer eine Hilfseinrichtung in einer annehmbaren Entfernung erreichen können. Zusätzlich soll sowohl die ex ante, als auch die ex post Betrachtungsweise berücksichtigt werden und in das Modell mit einfließen. Um besser auf die konkreten Gegebenheiten eingehen zu können und einen Entscheidungsträger bei der Standortbestimmung optimal zu unterstützen, sollen mehrere Lösungsvarianten angeboten werden, welche alle pareto optimal sind. So kann zwischen Lösungen mit möglichst hoher Effizienz, oder geringer Ungleichheit aus ex ante oder ex post Sicht die beste Variante ausgewählt werden.

2.2.1 Notation

Als Ausgangspunkt für das Modell sei eine Menge K gegeben, welche die Dörfer bzw. Populationsknoten (PK) widerspiegelt, wobei die Anzahl der PK $|K| = n$ ist.¹ Jeder PK, also jedes Element $k \in K$ hat wiederum w_k Einwohner (insgesamt gibt es also $N = \sum_{k \in K} w_k$ Bewohner in der vorliegenden Problemstellung). Um die Unsicherheit in Form von Erdbebenszenarien widerzuspiegeln, sei S eine Menge von Szenarien, wobei jedes Szenario $s \in S$ eine Eintrittswahrscheinlichkeit p_s besitzt. Weiters seien in Szenario

¹Die Notation und Modellspezifikation orientiert sich aufgrund der ähnlichen Fragestellung an der Arbeit von Mostajabdaveh et al. [11]

s und PK k genau b_k^s Personen von den Auswirkungen des eingetretenen Erdbebens betroffen, wobei natürlich $0 \leq b_k^s \leq w_k$ gelten muss.

J sei die Menge aller möglichen Kandidaten-Standorte (KS) für eine Notversorgungseinrichtung, wobei $|J| = m$ ist. Die zurückzulegende Distanz von PK k zur Einrichtung j wird in einer Distanzmatrix $D = (d_{kj})$ angegeben, wobei hier angenommen wird, dass die Distanz in jedem Szenario gleich bleibt. Das Modell könnte aber auch leicht dahingehend adaptiert werden, dass für verschiedene Szenarien unterschiedliche Distanzen (z.B. in Folge von zerstörten Straßen) zu überwinden sind. Außerdem sei C das Budget, welches zur Errichtung der Notversorgungseinrichtungen zur Verfügung steht und somit die Anzahl der zu eröffnenden Einrichtungen beschränkt.

$d^{(i)}$ (für $i = 1, \dots, N$) sei die zurückzulegende Distanz von Individuum i . Dieser Ausdruck ist allgemein zu verstehen und soll die Kosten (hier in Form von Distanz) widerspiegeln, die einer Person entstehen. Weiters sei $d_s^{(i)}$ die von Individuum i zurückzulegende Distanz in Szenario s , welche gleich d_{kj} ist, wenn die Person betroffen ist (vorausgesetzt i befindet sich im Knoten k und j ist die nächstgelegene geöffnete Einrichtung). Wenn die Person nicht von dem in Szenario s auftretendem Erdbeben betroffen ist, muss sie keine Hilfseinrichtung aufsuchen und die zurückzulegende Distanz $d_s^{(i)}$ ist in diesem Fall gleich 0.

Ein binärer Vektor x der Länge m gibt in der Lösung die ausgewählten KS an, in welchen eine Einrichtung eröffnet werden soll. Für alle Standorte $j \in J$ ist $x_j = 1$, wenn der jeweilige Standort ausgewählt wird und 0, wenn an dem KS keine Einrichtung eröffnet werden soll. Weiters sei $Y = (y_{kj})$ eine binäre Matrix, in der das $i - j'$ te Element gleich 1 ist, wenn die Bewohner von PK k im Katastrophenfall die Einrichtung am Standort j aufsuchen. Mit anderen Worten ist y_{kj} gleich 1, wenn KS j der am nächsten gelegene geöffnete Zufluchtsort zum PK k ist. Klarerweise kann y_{kj} nur 1 sein, wenn die jeweilige Einrichtung geöffnet ist, d.h. wenn x_j gleich 1 ist. Da die Zeilen von Y die PK und die Spalten die KS's widerspiegeln und es für jeden PK nur eine nächste geöffnete Einrichtung geben kann, muss die Zeilensumme der Matrix Y immer gleich 1 sein. Die Spaltensumme kann zwischen 0 (wenn die j -te Einrichtung nicht geöffnet ist, bzw. wenn sie für keinen PK die nächstgelegene geöffnete ist) und n liegen (wenn die Bewohner aller PK bei dieser Einrichtung Zuflucht suchen). Dies setzt natürlich voraus, dass die Einrichtungen genügend Kapazitäten für alle Personen haben und niemand von einer Einrichtung abgewiesen wird.

2.3 Modell

2.3.1 Zusammensetzung der Zielfunktionen

Wie bereits erwähnt, würde man bei einer reinen Betrachtung von Ungleichheit in vielen Situationen sehr ineffiziente Lösungen erhalten. Daher wird in diesem Abschnitt eine Ungleichheits-Averse Funktion mittels einer Linearkombination definiert, sodass die beiden Aspekte der Effizienz und der Gleichheit berücksichtigt werden. Später wird diese Funktion in zwei Varianten (ex ante und ex post) in das Modell einfließen. Der erste Teil dieser Funktion - welcher Effizienz garantiert - besteht aus der mittleren zurückzulegenden Distanz $\mu = \frac{1}{N} \sum_{i=1}^N d^{(i)}$. Die Ungleichheit fließt in einer abgewandelten Form des Gini Index, nämlich als „Gini’s Mean Absolute Difference“ (GMAD) mit einem Faktor λ in die Funktion ein. Dieses Maß ist als $\Delta = 2\mu G$ definiert, wobei G den Gini Index darstellt.

$$G = \frac{1}{2N} \frac{\sum_{i=1}^N \sum_{j=1}^N |d^{(i)} - d^{(j)}|}{\sum_{i=1}^N d^{(i)}} \quad (2.1)$$

Zusammengefasst ergibt sich daraus die Funktion $\mathcal{I} = \mu + \lambda\Delta = \mu + \lambda 2\mu G$, wobei der Faktor λ zwischen 0 und $\frac{1}{2}$ gewählt werden sollte ². Die Wahl von $\mathcal{I}$ und die Einschränkung bei der Wahl des λ Parameters wird von Mostajabdavah et al. [11] begründet und gerechtfertigt, wobei sich diese auf die in Ben-Porath et al. [1] beschriebene Theorie der sogenannten „symmetric comonotonically linear functionals“ berufen und beweisen, dass die Funktion $\mathcal{I}$ in diese Gruppe fällt.

Nachdem nun eine Ungleichheits-Averse Funktion definiert wurde, muss bei der Definition der Zielfunktion außerdem noch die Unsicherheit in Betracht gezogen werden. Es kann entweder die Ungleichheit des Erwartungswertes über alle Szenarien (ex ante) berechnet werden, oder der Erwartungswert der Ungleichheit aller Individuen (ex post). Der Unterschied liegt darin, dass bei der ex ante Betrachtung zuerst der Erwartungswert der zurückzulegenden Distanzen (bzgl. der Szenarien) für jedes Individuum berechnet wird und dann die Ungleichheit für diese erwarteten Distanzen bestimmt wird. Dahingegen wird bei der ex post Betrachtung zuerst für jedes Szenario (d.h. es besteht keine Unsicherheit mehr bzgl. der Distanzen) die Ungleichheit für jedes Individuum bestimmt und anschließend wird davon der Erwartungswert gebildet. Formal dargestellt erhält

²Würde man λ größer wählen, wird klarerweise mehr Wert auf die Minimierung der Ungleichheit als auf die Optimierung der Effizienz gelegt. Dadurch kann es passieren, dass ein PK einem Standort zugewiesen wird, der nicht der am nächsten gelegene ist, nur um die Ungleichheit möglichst gering zu halten. Solche Lösungen sollen natürlich vermieden werden, was durch die Wahl von $\lambda \leq 0.5$ garantiert wird.

man somit

$$\text{ex ante Ungleichheit:} \quad \mathcal{I}(E[d(s, i)]) \quad (2.2)$$

$$\text{ex post Ungleichheit:} \quad E[\mathcal{I}(d(s, i))] \quad (2.3)$$

wobei $E[\dots]$ der Erwartungswert und $d(s, i)$ ein Vektor mit den Elementen $(d_s^{(i)})_{i \in N}$ ist. Der Erwartungswert wird immer bzgl. der Szenarien gebildet und die Ungleichheit $\mathcal{I}$ wird bzgl. der Individuen $i \in N$ berechnet. Vertauscht wird bei den zwei Ansätzen lediglich die Reihenfolge, in der die Berechnung erfolgt.

Mithilfe der in (2.2) und (2.3) spezifizierten Formeln kann nun eine Zielfunktion zur Minimierung der ex ante sowie der ex post Ungleichheit definiert werden, welche dann mit dem adaptive epsilon-constraint Algorithmus zusammengesetzt und in ein Modell gegossen werden können.

2.3.2 Ex ante Zielfunktion und MIP Formulierung

Die ex ante Zielfunktion erhält man durch die Anwendung der Ungleichheits-Aversen Funktion $\mathcal{I}$ auf den Erwartungswert der zurückzulegenden Distanz aller Personen (s. Gleichung 2.2). Da in Knoten k und Szenario s nur b_k^s (von insgesamt w_k) Personen betroffen sind, wird der Erwartungswert der Distanz für eine beliebige Person aus Knoten k unter Zuhilfenahme des Faktums, dass für gegebenes s und k der Anteil der betroffenen Bevölkerung genau gleich $\frac{b_k^s}{w_k}$ ist, direkt berechnet. Somit muss dieser Anteil die Distanz d_{kj} zurücklegen, während die restlichen Personen nicht betroffen sind und somit keinen Weg zurücklegen müssen. Die erwartete zurückzulegende Distanz für eine beliebige Person aus Knoten k kann somit mit folgende Formel berechnet werden:

$$E[d_k] = \sum_{s \in S} \sum_{j \in J} \frac{b_k^s}{w_k} p_s y_{kj} d_{kj}$$

Zu beachten ist dabei, dass man die Gesamtdistanz aller Bewohnerinnen und Bewohner von Knoten k durch die Multiplikation mit w_k erhält. Der oben definierte Erwartungswert muss nun nur noch in die Formel $\mathcal{I}$ eingesetzt werden, wobei immer die Populationsgrößen der PK durch die Multiplikation mit w_k berücksichtigt werden müssen. Dazu wird der Mittelwerts-Teil sowie der Ungleichheits-Teil (GMAD) von $\mathcal{I}$ separat betrachtet. Durch Einsetzen und einfache Rechenoperationen ergibt sich:

$$\begin{aligned}
\mu &= \frac{1}{N} \sum_{k \in K} \sum_{s \in S} \sum_{j \in J} b_k^s p_s y_{kj} d_{kj} \\
G &= \frac{1}{2N} \frac{\sum_{k \in K} \sum_{k' \in K} w_k w_{k'} |E[d_k] - E[d'_k]|}{\sum_{k \in K} w_k E[d_k]} = \\
&= \frac{1}{2N} \frac{\sum_{k \in K} \sum_{k' \in K} \left| \sum_{s \in S} \sum_{j \in J} p_s (w_{k'} b_k^s y_{kj} d_{kj} - w_k b_{k'}^s y_{k'j} d_{k'j}) \right|}{\sum_{k \in K} \sum_{s \in S} \sum_{j \in J} b_k^s p_s y_{kj} d_{kj}}
\end{aligned}$$

Die zusammengesetzte ex ante Zielfunktion sieht somit wie folgt aus:

$$\begin{aligned}
F_{ex_ante} &= \mu + \lambda \Delta = \mu + \lambda 2\mu G = \\
&= \frac{1}{N} \sum_{s \in S} \sum_{k \in K} \sum_{j \in J} p_s b_k^s y_{kj} d_{kj} + \\
&+ \frac{\lambda}{N^2} \sum_{k \in K} \sum_{k' \in K} \left| \sum_{s \in S} \sum_{j \in J} p_s (w_{k'} b_k^s y_{kj} d_{kj} - w_k b_{k'}^s y_{k'j} d_{k'j}) \right|
\end{aligned} \tag{2.4}$$

Um die Zielfunktion in Matlab [10] implementieren zu können und mit der CPLEX-Toolbox für Matlab des IBM ILOG CPLEX Optimierers [4] in weiterer Folge optimale Lösungen zu finden, muss die Zielfunktion zuerst in eine geeignete Form gebracht werden und es müssen alle nötigen Nebenbedingungen spezifiziert werden, um das gewünschte Resultat zu erhalten. Dazu wird das Problem im folgenden Kapitel als MIP-Modell (mixed integer program) dargestellt. Die Implementierung in Matlab wird später im Kapitel 3.2 noch genauer erläutert.

Es gibt zwei Haupt-Entscheidungsvariablen, nämlich x und y , welche die zu wählenden Standorte bzw. die Zuteilung der PK zu den Standorten angeben. Wie bereits im Kapitel 2.2.1 erwähnt, sind beide Variablen binär, wobei folgendes gilt:

$$\begin{aligned}
x_j &= \begin{cases} 1 & , \text{ wenn KS } j \text{ geöffnet wird} \\ 0 & , \text{ sonst} \end{cases} \\
y_{kj} &= \begin{cases} 1 & , \text{ wenn KS } j \text{ der nächste geöffnete Standort zum PK } k \text{ ist} \\ 0 & , \text{ sonst} \end{cases}
\end{aligned}$$

Um das Modell in der MIP Form darstellen zu können, werden zusätzlich noch die Hilfs-Entscheidungsvariablen $\tau_{k,k'}^+$ und $\tau_{k,k'}^-$ eingeführt, um die Zielfunktion zu linearisieren (dabei bildet die erste Hilfsvariable alle positiven und die zweite alle negativen Summanden ab, sodass man durch Addition der Beiden den Absolutbetrag aus der Gleichung 2.4 erhält). Sei also $\tau_{k,k'}^+ = \max(0, \tau_{k,k'})$ und $\tau_{k,k'}^- = \max(0, -\tau_{k,k'})$, wobei

$$\tau_{k,k'} = \sum_{s \in S} \sum_{j \in J} p_s \left(w_{k'} b_k^s y_{kj} d_{kj} - w_k b_{k'}^s y_{k'j} d_{k'j} \right) \quad k, k' \in K$$

ist. Das ex ante Problem in der MIP-Form sieht dann folgendermaßen aus:

$$\min \mu + \lambda 2\mu G = \frac{1}{N} \sum_{s \in S} \sum_{k \in K} \sum_{j \in J} p_s b_k^s y_{kj} d_{kj} + \frac{\lambda}{N^2} \sum_{k \in K} \sum_{k' \in K} \left(\tau_{k,k'}^+ + \tau_{k,k'}^- \right) \quad \text{s.t.} \quad (2.5)$$

$$\tau_{k,k'}^+ - \tau_{k,k'}^- - \sum_{s \in S} \sum_{j \in J} p_s \left(w_{k'} b_k^s y_{kj} d_{kj} - w_k b_{k'}^s y_{k'j} d_{k'j} \right) = 0 \quad \forall k, k' \in K \quad (2.6)$$

$$\tau_{k,k'}^+, \tau_{k,k'}^- \geq 0 \quad \forall k, k' \in K \quad (2.7)$$

$$y_{kj} - x_j \leq 0 \quad \forall k \in K, j \in J \quad (2.8)$$

$$\sum_{j \in J} y_{kj} = 1 \quad \forall k \in K \quad (2.9)$$

$$\sum_{j \in J} x_j \leq C \quad (2.10)$$

$$y_{kj} \in \{0, 1\} \quad \forall k \in K, j \in J \quad (2.11)$$

$$x_j \in \{0, 1\} \quad \forall j \in J \quad (2.12)$$

Die beiden Nebenbedingungen 2.6 und 2.7 stellen sicher, dass die in der Zielfunktion vorkommende Summe der beiden Hilfsvariablen genau dem Absolutbetrag aus Gleichung 2.4 entspricht. Da sowohl x_j als auch y_{kj} binäre Variablen sind, garantiert NB 2.8, dass KS j nur dann der nächstgelegene geöffnete Standort zum PK k sein kann, wenn KS j geöffnet ist (d.h. y_{kj} kann nur dann gleich 1 sein, wenn auch x_j gleich 1 ist). Außerdem kann es für jeden PK k nur exakt einen nächsten geöffneten KS geben, was durch die NB 2.9 gewährleistet wird. In NB 2.10 wird die Anzahl der maximal zu eröffnenden Standorte durch die Kostenbeschränkung der insgesamt zur Verfügung stehenden Mittel (C) angegeben. Dabei wird C gleich der Anzahl der zu eröffnenden Einrichtungen gewählt

und es wird angenommen, dass die Eröffnung einer Einrichtung an jedem Standort gleich viel kostet.

2.3.3 Ex post Zielfunktion und MIP Formulierung

Im Gegensatz zur ex ante Zielfunktion wird nun zuerst die Ungleichheit $\mathcal{I}$ für jeden PK in jedem Szenario einzeln berechnet und danach wird davon der Erwartungswert gebildet (s. 2.3). Da in diesem Setting nach dem Eintreten eines Szenarios bereits bekannt ist, welche Personen von den Folgen des Erdbebens betroffen sind und welche nicht, werden bei den nachfolgenden Berechnungen nur die betroffenen Personen in Betracht gezogen. Da alle nicht betroffenen Personen keine Distanz überwinden müssen und auch keine Hilfsseinrichtung in Anspruch nehmen werden, die Ungleichheit aber für all jene berechnet werden soll, welche tatsächlich eine Einrichtung aufsuchen, wird für die Berechnung der Ungleichheit hier nur mehr die Gruppe der betroffenen Personen in Betracht gezogen. Bei der ex ante Ungleichheit war dies nicht möglich, da vor dem Eintreten des Ereignisses noch nicht abzusehen ist, welche Personen betroffen sein werden. Für die Definition der ZF bedeutet das, dass in jedem gegebenen Szenario s und Knoten k genau b_k^s Personen betroffen sind und somit in die Berechnungen einfließen (die restlichen $w_k - b_k^s$ Personen werden nicht mehr miteinbezogen). Insgesamt sind in Szenario s also $N_s = \sum_{k \in K} b_k^s$ Personen betroffen. Um die Ungleichheit $\mathcal{I}$ für die Betroffenen in Szenario s berechnen zu können, werden die mittleren zurückzulegenden Distanzn μ_s , sowie die Gini-Koeffizienten G_s benötigt :

$$\mu_s = \frac{1}{N_s} \sum_{k \in K} \sum_{j \in J} b_k^s y_{kj} d_{kj} \quad , s \in S$$

$$G_s = \frac{1}{2N_s} \frac{\sum_{k \in K} \sum_{k' \in K} b_k^s b_{k'}^s \left| \sum_{j \in J} (y_{kj} d_{kj} - y_{k'j} d_{k'j}) \right|}{\sum_{k \in K} \sum_{j \in J} b_k^s d_{kj}} \quad , s \in S$$

Um nun die ex post Zielfunktion zu erhalten, muss nur mehr der Erwartungswert $E[\mu_s + \lambda 2\mu G_s]$ gebildet werden. Nach einigen einfachen Umformungen erhält man somit

$$\begin{aligned}
F_{ex_post} &= E[\mu_s + \lambda \Delta_s] = E[\mu_s + \lambda 2\mu G_s] = \\
&= \sum_{s \in S} p_s \frac{1}{N_s} \sum_{k \in K} \sum_{j \in J} b_k^s y_{kj} d_{kj} + \\
&+ \lambda \sum_{s \in S} p_s \frac{1}{2N_s^2} \sum_{k \in K} \sum_{k' \in K} b_k^s b_{k'}^s \left| \sum_{j \in J} (y_{kj} d_{kj} - y_{k'j} d_{k'j}) \right|.
\end{aligned} \tag{2.13}$$

Ähnlich wie im ex ante Kapitel, kann nun das ex post Problem in der MIP Formulierung angegeben werden. Um eine lineare Zielfunktion zu erhalten, wird folgende Hilfsvariable definiert

$$\rho_{kk's} = \sum_{j \in J} (y_{kj} d_{kj} - y_{k'j} d_{k'j}) \quad k, k' \in K, s \in S$$

und fließt ähnlich wie im ex ante Problem als $\rho_{kk's}^+ = \max(0, \rho_{kk's})$ und $\rho_{kk's}^- = \max(0, -\rho_{kk's})$ in die Zielfunktion ein. Das ex post Problem in der MIP-Darstellung ist dann gegeben durch

$$\min \sum_{s \in S} \sum_{k \in K} \sum_{j \in J} \frac{1}{N_s} p_s b_k^s y_{kj} d_{kj} + \sum_{k \in K} \sum_{k' \in K} b_k^s b_{k'}^s (\rho_{kk's}^+ + \rho_{kk's}^-) \quad \text{s.t.} \tag{2.14}$$

$$\rho_{kk's}^+ - \rho_{kk's}^- - \sum_{j \in J} (y_{kj} d_{kj} - y_{k'j} d_{k'j}) = 0 \quad \forall k, k' \in K, s \in S \tag{2.15}$$

$$\rho_{kk's}^+, \rho_{kk's}^- \geq 0 \quad \forall k, k' \in K, s \in S \tag{2.16}$$

$$y_{kj} - x_j \leq 0 \quad \forall k \in K, j \in J \tag{2.17}$$

$$\sum_{j \in J} y_{kj} = 1 \quad \forall k \in K \tag{2.18}$$

$$\sum_{j \in J} x_j \leq C \tag{2.19}$$

$$y_{kj} \in \{0, 1\} \quad \forall k \in K, j \in J \tag{2.20}$$

$$x_j \in \{0, 1\} \quad \forall j \in J. \tag{2.21}$$

Wie auch schon beim ex ante Problem stellen die beiden Nebenbedingungen 2.15 und 2.16 sicher, dass die in der Zielfunktion vorkommende Summe der beiden Hilfsvariablen genau dem Absolutbetrag aus Gleichung 2.13 entspricht. Die restlichen NB sind identisch mit jenen aus dem ex ante Problem.

2.3.4 Adaptive Epsilon Constraint Algorithmus

Wie bereits eingangs erwähnt, wird der von Laumanns et al. [8] vorgestellte adaptive epsilon-constraint Algorithmus verwendet, um bei der Bestimmung der optimalen Platzierung der KS sowohl die ex ante, als auch die ex post ZF miteinbeziehen zu können und somit alle pareto-optimalen Lösungen zu erhalten. In der vorliegenden Fragestellung wird der Algorithmus auf die in den vorigen Kapiteln definierten Zielfunktionen angewendet, wobei die ex ante Zielfunktion minimiert wird und die ex post ZF als zusätzliche NB einfließt, welche in jeder Iteration strenger wird. Da hier nur zwei Zielfunktionen vorliegen und somit nur eine ZF als zusätzliche NB verwendet wird, kann der Algorithmus in einer vereinfachten Form dargestellt werden. Ein Pseudocode für den Algorithmus sieht wie folgt aus:

```

Set  $P = \emptyset$  ,  $\epsilon = \infty$  , continue = true and choose  $\delta > 0$ 
while continue == true {
    res = solution of lexmin ( $F_{ex\_ante}(x)$  s.t.  $F_{ex\_post}(x) \leq \epsilon - \delta$ )
    if (res found) {
        set  $P = P \cup \{res\}$ 
        set  $\epsilon = F_{ex\_post}(res)$ 
    }
    else {
        set continue = false
    }
}
Output:  $P$ 

```

Um den Algorithmus anwenden zu können, muss also die ex post Zielfunktion als Nebenbedingung in das ex ante Problem integriert werden. Zusätzlich müssen dann auch

noch die Nebenbedingungen 2.15 und 2.16 hinzugefügt werden. Weiters ist zu beachten, dass die restlichen NB aus der ex post MIP-Formulierung bereits im ex ante Problem aufscheinen.

Da in der ersten Iteration $\epsilon = \infty$ ist, wird hier die optimale Lösung bzgl. der ex ante ZF gefunden. Wichtig ist dabei, dass immer das lexikographische Minimum gesucht wird um pareto-optimale Lösungen zu finden. Das lexikographische Minimum entspricht dabei jener Lösung, für die beide Zielfunktionen unter gegebenen Nebenbedingungen minimal sind, d.h. keine ZF kann weiter verbessert werden, ohne die jeweils andere zu verschlechtern. Ein kurzes Beispiel soll dies verdeutlichen: Angenommen in der ersten Iteration (wo F_{ex_ante} ohne Einschränkung bzgl. F_{ex_post} minimiert wird), wird eine Lösung gefunden. Im nächsten Schritt wird $\epsilon = F_{ex_post}$ (Wert der ex post ZF aus der 1. Iteration) gesetzt und F_{ex_ante} wird nun unter der Bedingung $F_{ex_post}(x) \leq \epsilon - \delta$ minimiert (wobei der Parameter $\delta > 0$ ist). Nun kann es passieren, dass trotz der strengeren NB wieder dieselbe Lösung für die ex ante ZF gefunden wird. Das lexikographische Minimum besagt nun, dass nur die zweite Lösung akzeptiert wird, da diese für das selbe ex ante Niveau eine bessere Lösung bzgl. der ex post Zielfunktion bietet. Das kombinierte Problem, welches dann für den adaptive epsilon constraint Algorithmus verwendet wird, sieht also in der MIP-Notation wie folgt aus:

$$\min \mu + \lambda 2\mu G = \frac{1}{N} \sum_{s \in S} \sum_{k \in K} \sum_{j \in J} p_s b_k^s y_{kj} d_{kj} + \frac{\lambda}{N^2} \sum_{k \in K} \sum_{k' \in K} (\tau_{k,k'}^+ + \tau_{k,k'}^-) \quad \text{s.t.} \quad (2.22)$$

$$\sum_{s \in S} \sum_{k \in K} \sum_{j \in J} \frac{1}{N_s} p_s b_k^s y_{kj} d_{kj} + \sum_{k \in K} \sum_{k' \in K} b_k^s b_{k'}^s (\rho_{kk's}^+ + \rho_{kk's}^-) \leq \epsilon - \delta \quad (2.23)$$

$$\rho_{kk's}^+ - \rho_{kk's}^- - \sum_{j \in J} (y_{kj} d_{kj} - y_{k'j} d_{k'j}) = 0 \quad \forall k, k' \in K, s \in S \quad (2.24)$$

$$\rho_{kk's}^+, \rho_{kk's}^- \geq 0 \quad \forall k, k' \in K, s \in S \quad (2.25)$$

$$\tau_{k,k'}^+ - \tau_{k,k'}^- - \sum_{s \in S} \sum_{j \in J} p_s (w_{k'} b_k^s y_{kj} d_{kj} - w_k b_{k'}^s y_{k'j} d_{k'j}) = 0 \quad \forall k, k' \in K \quad (2.26)$$

$$\tau_{k,k'}^+, \tau_{k,k'}^- \geq 0 \quad \forall k, k' \in K \quad (2.27)$$

$$y_{kj} - x_j \leq 0 \quad \forall k \in K, j \in J \quad (2.28)$$

$$\sum_{j \in J} y_{kj} = 1 \quad \forall k \in K \quad (2.29)$$

$$\sum_{j \in J} x_j \leq C \quad (2.30)$$

$$y_{kj} \in \{0, 1\} \quad \forall k \in K, j \in J \quad (2.31)$$

$$x_j \in \{0, 1\} \quad \forall j \in J \quad (2.32)$$

Dieses MIP-Problem wurde in Matlab implementiert und unter Verwendung der CPLEX-Toolbox schrittweise, wie im oben angegebenen Pseudocode, für verschiedene Parameter-Kombinationen minimiert. Im folgenden Kapitel wird zuerst die Datengenerierung und die Wahl der Parameter diskutiert. Anschließend folgen zwei Anwendungsbeispiele. Im ersten Anwendungsbeispiel werden alle pareto optimalen Lösungen für eine kleine Anzahl an Szenarien gesucht, im zweiten Beispiel wird die Anzahl der Szenarien erhöht und im Gegenzug die Knoten-, sowie Standortzahl verringert. Nachdem alle Ergebnisse im Detail analysiert wurden, wird ein Kapitel der Laufzeitanalyse gewidmet. Anschließend werden noch weitere Ergebnisse präsentiert, die für verschiedene λ Parameters erreicht werden.

3. Simulation

3.1 Datengenerierung und Parameter

Um den im vorigen Kapitel beschriebenen Algorithmus anwenden zu können, wurden mehrere Datensätze verschiedener Größe generiert. Als Inputs werden dafür die Szenarien inklusive deren Eintrittswahrscheinlichkeiten, die Knoten, Anzahl der Bewohner und Anzahl der Betroffenen je Szenario, sowie mögliche Orte für die Einrichtungen und die zugehörigen Distanzen zwischen den Knoten und Einrichtungen benötigt. Außerdem müssen die Parameter λ und δ , sowie die Anzahl der maximal zu eröffnenden Einrichtungen in Form einer Kostenobergrenze angegeben werden.

Zuerst werden zufällige Erdbebenszenarien erstellt, die als überlappende Kreise verschiedener Größen dargestellt werden und anschließend werden innerhalb dieser Regionen zufällige Punkte gezogen (s. Abb 3.1 für ein Beispiel mit sieben Szenarien und 40 Knoten). Zur Simulation der Erdbebenszenarien soll eine tektonische Bruchlinie nachempfunden werden. Solche Bruchlinien oder Verwerfungslinien befinden sich an den Grenzen zwischen zwei tektonischen Platten. Bei einer Verschiebung der Platten kommt es zu großen Spannungen, wodurch an einer Stelle entlang der Bruchlinie ein Erdbeben ausgelöst werden kann (vgl. [12], S. 39 f. und S. 181 f.). Um eine solche Bruchlinie zu simulieren und gleichzeitig eine gewisse Überlappung der Szenarien zu erhalten, werden die Epizentren der Erdbeben aus einer bivariaten Normalverteilung mit Erwartungswert $\mu = [0.5, 0.5]$ und Kovarianzmatrix $\Sigma = 0.1 * \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$ erzeugt. Die Radien der betroffenen Gebiete - ausgehend von den Epizentren - werden aus einer Gleichverteilung auf dem Intervall $[0.1, 0.5]$ gezogen. Aus den gegebenen Epizentren und Radien ergeben sich dann die Erdbebenszenarien, wobei angenommen wird, dass ein Knoten von einem Erdbeben betroffen ist, sobald er innerhalb des jeweiligen Radius liegt. Nachdem nun die Szenarien feststehen, werden innerhalb der Erdbebenzonen zufällige Punkte gezogen, welche die Nachfrageknoten (z.B. Dörfer oder Häuser) darstellen. Um eine gleichmäßige Verteilung

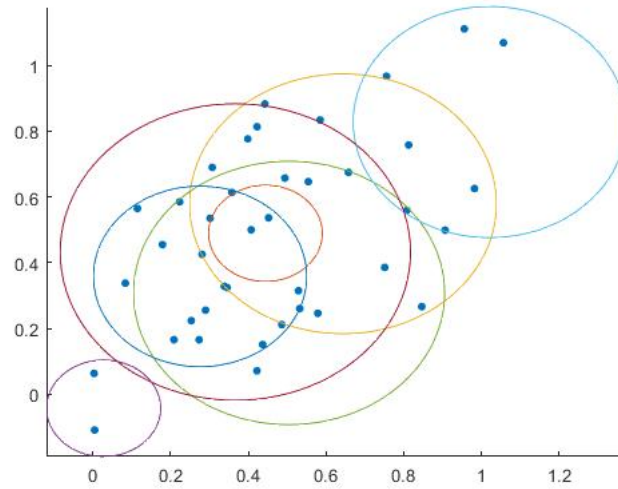


Abbildung 3.1: Zufallsgenerierte Daten mit 40 PK und 7 Szenarien

innerhalb der Zonen zu erhalten, wird hierbei zuerst zufällig festgelegt wie viele Punkte in einer Zone liegen sollen, wobei die Wahrscheinlichkeit von der Größe der Zone abhängt. Danach werden die Punkte innerhalb der Erdbebenzonen zufällig verteilt. Durch diese Vorgehensweise - zuerst Festlegung der Szenarien, danach zufällige Verteilung der Knoten innerhalb der Szenarien - wird sichergestellt, dass jeder Knoten zumindest in einem Erdbebenszenario betroffen ist, falls dieses eintritt. Würde man zuerst zufällige Knoten (z.B. gleichverteilt auf $[0, 1] \times [0, 1]$) ziehen und dann die Erdbebenszenarien zufällig erzeugen, würden meist einige Knoten außerhalb der Gefahrenzone liegen und in keinem der Szenarien betroffen sein, was die Zuweisung zu einer Noteinrichtung obsolet macht und das Ergebnis verzerren würde, da diese PK nie eine Distanz überwinden müssten.

Weiters wird angenommen, dass sich in jedem Knoten gleich viele Bewohner befinden, daher wird diese Zahl für alle Knoten gleich eins gesetzt. Die Anzahl der betroffenen Bewohner je Knoten und Szenario wird gleich 1 gewählt, wenn der PK innerhalb des zum entsprechenden Szenario gehörenden Kreises liegt und 0, wenn er außerhalb liegt. D.h. es ist entweder jeder, oder kein Bewohner betroffen. Natürlich könnte man die Werte auch zwischen 0 und 1 wählen, um einen relativen Anteil an Betroffenen widerzuspiegeln, oder größer oder gleich 1 um die absolute Anzahl der Betroffenen anzugeben.

Neben den PK müssen außerdem noch die möglichen Standorte für die Notversorgungseinrichtungen festgelegt werden. Da in der Praxis solche Einrichtungen meistens in, oder

nahe an einem Dorf gebaut werden, da dort bereits eine bestehende Infrastruktur an Zufahrtsstraßen und Verbindungsstrecken zu anderen Dörfern in der Umgebung besteht, wird jeder PK gleichzeitig als möglicher Standort für eine Einrichtung festgelegt.

Der Parameter λ wird im Folgenden gleich 0.5 gewählt, um eine gute Balance zwischen Effizienz (mittlere Distanzen minimieren) und Gleichheit (Ginis mean absolute difference) zu garantieren. Der Parameter δ , welcher nach jeder Iteration im epsilon constraint Algorithmus verwendet wird, um die Grenze für die ex post Nebenbedingung neu zu berechnen, wird als 0.0001 gewählt. Bei der Anwendung des Algorithmus hat sich gezeigt, dass manche Lösungen übersprungen und nicht gefunden werden, da die ex post Nebenbedingung zu schnell zu streng wird, wenn der Parameter δ zu groß gewählt wird. Wird der Parameter dahingegen sehr klein gewählt, verlängert dies die Laufzeit, da eine größere Anzahl an Iterationen durchgeführt werden muss.

Im folgenden Kapitel wird zuerst genauer auf die Implementierung in Matlab und die verwendete Funktion aus der CPLEX Toolbox eingegangen. Danach werden die Ergebnisse der Optimierung unter Verwendung des im Kapitel 2.3.4 vorgestellten adaptive epsilon constraint Algorithmus, sowie der kombinierten Zielfunktion (s. Gleichung 2.22 mit den zugehörigen Nebenbedingungen), präsentiert.

3.2 Implementierung

Wie bereits erwähnt, wurden für das Beispiel sieben Szenarien, sowie 40 Populationsknoten - welche auch gleichzeitig die möglichen Standorte für die Hilfseinrichtungen darstellen - zufällig gewählt. Im ersten Schritt wird nun eine Distanzmatrix d erstellt, wobei im i, j -ten Eintrag die Distanz von PK i zu KS j eingetragen wird, wobei als Distanzen die Länge der Geraden zwischen den zwei Punkten verwendet wird. Man erhält somit eine symmetrische Matrix und da jeder PK auch gleichzeitig KS ist, sind alle Einträge auf der Hauptdiagonalen der Matrix gleich 0. Als Nächstes wird eine 'Betroffenen-Matrix' b erstellt, deren i, k -tes Element 1 ist, wenn in Szenario i der PK k betroffen ist und 0 sonst. Multipliziert man nun die i -te Spalte von $b_{(7 \times 40)}$ mit der i -ten Zeile von $d_{(40 \times 40)}$, erhält man eine (7×40) Matrix, in deren k, l -ten Element die zu überwindende Distanz von Knoten i zu KS l im Szenario k eingetragen ist.

Diese Matrizen werden nun bei der Implementierung verwendet, um das im Kapitel 2.3.4 definierte lineare Optimierungsproblem in die richtige Form zu bringen und dann mittels Matlab und der CPLEX Toolbox unter Anwendung des adaptive epsilon constraint Algorithmus alle pareto-optimalen Lösungen für die Standorte der Einrichtungen zu fin-

den. Da in der zu minimierenden Funktion (s. Gleichung 2.22) sowohl binäre (die x und y -Variablen), als auch reelle Entscheidungsvariablen ($\tau_{kk'}^+$ und $\tau_{kk'}^-$) vorkommen, wird zur Minimierung die Funktion 'cplexmilp' aus der CPLEX Toolbox verwendet (s. [2]). Dieser Funktion müssen nun die Vektoren f , $bineq$ und beq , sowie Matrizen $Aineq$ und Aeq übergeben werden. Außerdem müssen Unter- und Obergrenzen (lb und ub) für die Entscheidungsvariablen angegeben werden. Dann löst die Funktion das folgende lineare Minimierungsproblem:

$$\begin{aligned}
\min \quad & f * x \\
s.t. \quad & Aineq * x \leq bineq \\
& Aeq * x \leq beq \\
& lb \leq x \leq ub
\end{aligned}$$

Diese Form wird sofort unter Verwendung der im Kapitel 2.3.4 dargestellten MIP-Formulierung und einigen einfachen Umformungen erreicht. Dabei müssen nur alle Gleichheits-, sowie Ungleichheits-Nebenbedingungen in Matrixform zusammengefasst werden, wobei jede Zeile einer Nebenbedingung entspricht und die Anzahl der Spalten die Variablenanzahl ist. Alle Ungleichheits-Nebenbedingungen werden in der Matrix $Aineq$, die zugehörigen Schranken im Vektor $bineq$ gespeichert. Dasselbe wird für die Gleichheits-NB mit der Matrix Aeq und dem Vektor beq gemacht. Die primären Entscheidungsvariablen sind die $x's$ und $y's$, welche die zu öffnenden Einrichtungen bzw. die von jedem Knoten aufgesuchte Einrichtung angeben. Weiters werden zur Berechnung des Gini-Teils die Hilfsvariablen $\tau_{kk'}^+$ und $\tau_{kk'}^-$ für die ex ante Ungleichheit, sowie $\rho_{kk's}^+$, sowie $\rho_{kk's}^-$ für die ex Post Ungleichheit benötigt. Somit werden insgesamt $s + ns + 2n^2(1 + p)$ Variablen definiert, wobei s die Anzahl der KS, n die Anzahl der PK und p die Anzahl der Szenarien ist. Vor allem durch die Hilfsvariablen wird diese Zahl schnell sehr groß. So hat der Vektor der Entscheidungsvariablen bei 40 PKs und KSs, sowie 7 Szenarien bereits eine Länge von 27240.

Durch die gegebenen computationalen Bedingungen (Windows 10, mit 8GB RAM und einem Intel Core i7-2670QM Prozessor) liegt die Grenze der maximal verarbeitbaren Datenmengen bei ca. 30000 Entscheidungsvariablen (wobei dies von der konkreten Problemstellung und der Zusammensetzung der Variablen abhängt - s. Kapitel 3.5). Der limitierende Faktor ist dabei aber nicht die Laufzeit, sondern der freie Arbeitsspeicher,

der durch die Größe der oben genannten Matrizen schnell verbraucht ist.

Nachdem nun die Zielfunktion sowie alle Nebenbedingungen definiert sind, muss an die Funktion 'cplexmilp' nur noch der Typ der Zielvariablen übergeben werden. Für die x und y Variablen wird binär und für die restlichen Hilfsvariablen reell gewählt. Somit ist das im Kapitel 2.3.4 definierte Problem vollständig implementiert und findet für gegebene Daten die optimale Platzierung der Einrichtungen. Um nun alle pareto-optimalen Lösungen (also Kombinationen der ex ante und ex post Zielfunktionen) zu finden, muss nun nur noch nach jeder Iteration die Obergrenze der Ungleichheitsbedingung 2.23 - basierend auf dem Ergebnis der letzten Iteration - neu gewählt werden. Die Zielfunktion sowie alle anderen Nebenbedingungen ändern sich dabei nicht.

Die Matlab-Funktion, welche die oben beschriebenen Schritte durchführt und als Ergebnis alle pareto optimalen Lösungen, sowie Plots und Tabellen mit diversen Informationen ausgibt, ist im Appendix B aufgeführt.

3.3 Anwendungsbeispiel 1

3.3.1 Einführung

Für das erste Anwendungsbeispiel wurden zuerst 7 Szenarien zufällig erstellt und dann, wie im Kapitel 3.1 erläutert, mit 40 Populationsknoten besiedelt. Abb. 3.1 zeigt das Ergebnis dieser zufälligen Zuweisung, die als Ausgangspunkt für die Anwendung des Modells und die Interpretation der Ergebnisse dienen soll. Weiters wird angenommen, dass die Eintrittswahrscheinlichkeit für alle Szenarien gleich groß ist und jeweils $1/7$ beträgt. Außerdem steht ein Budget für maximal vier Einrichtungen zur Verfügung, wobei angenommen wird, dass die Kosten für jede Einrichtung und jeden Standort gleich hoch sind und dass es keine Kapazitätsgrenzen gibt. D.h. die Einwohner jedes Knotens können im Katastrophenfall zu der am nächsten gelegenen Versorgungsstelle flüchten, ohne befürchten zu müssen, dass sie abgewiesen werden.

In dem zufällig erstellten Beispiel liegen die Epizentren von fünf der insgesamt sieben Szenarien relativ zentral, wobei die Größe der betroffenen Gebiete variiert. Es gibt also ein Gebiet mit hoher tektonischer Aktivität, wobei die verschiedenen großen Radien der Szenarien auf die Schwere des jeweiligen Bebens hindeuten könnten. In diesem zentral gelegenen Gebiet liegt auch der Großteil der Populationsknoten, welche durch die Überlappung der Szenarien einer Mehrfachbedrohung ausgesetzt sind. Weiters gibt es noch zwei etwas abgelegenere Regionen (der kleine Kreis im süd-westlichen, sowie der größere Kreis im nord-östlichen Gebiet), in denen nur wenige PK liegen, welche auch einem geringeren Risiko ausgesetzt sind, da sie nur beim Eintreten von einem Szenario betrof-

fen wären. Intuitiv würde man die meisten Einrichtungen relativ zentral platzieren, da hier das Risiko am größten ist und auch die meisten Knoten eher zentral gelegen sind. Um jedoch eine gewisse Gleichheit sicherzustellen und die abgelegenen Dörfer nicht zu sehr zu benachteiligen, muss man einen Mittelweg finden, um auch für diese abgelegenen Knoten eine annehmbare Lösung zu erhalten.

3.3.2 Ergebnisse

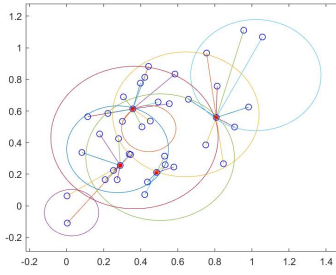
Unter Anwendung des adaptive epsilon constraint Algorithmus wird in der ersten Iteration die ex ante Zielfunktion global, d.h. ohne Einschränkung durch die ex post Zielfunktion, minimiert. Aus dem Ergebnis wird dann der Funktionswert der ex post ZF an dieser Stelle bestimmt. Für die nächste Iteration wird wieder die ex ante ZF minimiert, nun aber unter der zusätzlichen Nebenbedingung, dass die ex post ZF kleiner oder gleich dem aus der ersten Iteration erhaltenen Wert minus δ sein muss. Die ex ante ZF wird also etwas verschlechtert, um einen besseren Wert für die ex post ZF zu erreichen. Dies wird nun so lange fortgesetzt, bis keine Lösung mehr gefunden werden kann. Die Anzahl der pareto optimalen Lösungen, die mit diesem Algorithmus gefunden werden, hängt stark von den vorliegenden Daten ab. So findet man bei derselben Anzahl an Szenarien und Knoten, je nachdem wie diese verteilt sind, manchmal nur einige wenige Lösungen, während man bei einer anderen Verteilung der Knoten eine große Zahl an pareto optimalen Lösungen erhält. Durch die bereits erwähnte Konzentration der Knoten im Zentrum, wodurch die Zielfunktionen durch kleinere Verschiebungen der zu eröffnenden Einrichtungen gut variiert werden kann, erhält man insgesamt elf pareto optimale Lösungen. In Abb. 3.2 sind die Ergebnisse aller Iterationen grafisch dargestellt. Dabei stellen die großen Kreise wiederum die Szenarien dar und jeder der kleinen Kreise steht für einen Populationsknoten. In jeder Lösung werden die Standorte der ausgewählten Einrichtungen als ausgefüllte Kreise dargestellt und es wird von jedem Knoten eine Linie zu demjenigen Standort eingezeichnet, zu dem die Bewohner zugeteilt werden bzw. bei dem sie Schutz suchen können. Durch den Einfluss der mittleren zurückzulegenden Distanz (Effizienz) und die Wahl von $\lambda = 0,5$ ist der zugeteilte Standort auch der am nächsten gelegene. Würde man jedoch den Ungleichheits-aversen Teil der Zielfunktion höher gewichten, könnte durchaus auch die Situation auftreten, dass manche Knoten zu weit entfernten Standorten zugeteilt werden, um die Ungleichheit auf Kosten der Effizienz gering zu halten. Im vorliegenden Beispiel werden die Knoten immer zu dem geöffneten Standort mit der kürzesten Entfernung zum PK zugeteilt.

In Abb. 3.2 ist gut zu erkennen, dass bei einer reinen ex ante Betrachtung (Lösung 1)

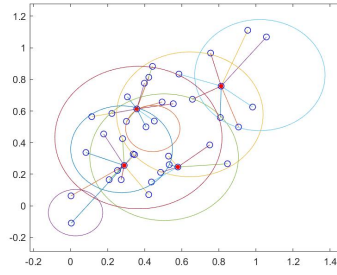
alle Standorte sehr zentral gelegen sind, wodurch alle im Zentrum gelegenen PK, welche von mehreren Szenarien bedroht werden, eine sehr kurze Strecke zurücklegen müssen. Die abgelegeneren Knoten müssen jedoch einen weiten Weg auf sich nehmen, um zu einer Einrichtung zu gelangen. In der nächsten Lösung verschiebt sich der Standort des am östlichsten gelegenen Standorts deutlich in Richtung Norden, wodurch die abgelegenen Knoten im nordöstlichen Teil einen deutlich kürzeren Weg zurücklegen müssen. Außerdem verschiebt sich einer der südlichen Standorte etwas in Richtung Osten, wodurch die Verschiebung des anderen Standortes etwas kompensiert wird und einige Knoten nun zu einem Standort in einer anderen Region zugeteilt werden (s. Verbindungslinien zwischen Knoten und Standorten).

In den Lösungen drei bis sieben lassen sich nur kleinere Veränderungen der Standorte feststellen, wodurch sich die Zuteilung der Knoten und deren zurückzulegende Distanzen nur geringfügig verändern. In der achten Lösung kann eine deutliche Verschiebung aller Standorte beobachtet werden, da ein Standort an den südlichsten Punkt gesetzt wird ¹, wodurch auch diese weit abgelegenen Knoten einen guten Zugang zu einer Notversorgungseinrichtung erhalten. Um diese Verschiebung zu kompensieren, werden jedoch auch die anderen Standorte verschoben, wodurch sich vor allem für die nördlichen Knoten eine deutliche Verschlechterung feststellen lässt. In den letzten beiden Lösungen wird die Lage der nördlichen Knoten durch eine Verschiebung des nord-östlichen Standortes wieder deutlich verbessert. Außerdem werden die zentral gelegenen Standorte etwas verschoben und zwei weitere Knoten werden daraufhin dem südlichen Standort zugewiesen.

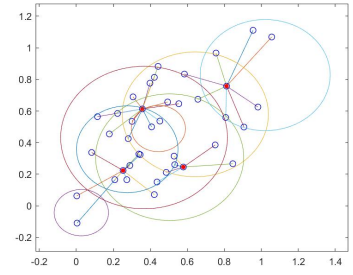
¹Anm.: Da dem südlichsten Standort nur zwei Knoten zugewiesen werden, nämlich der PK am Standort selbst sowie ein weiterer, ist es vorerst egal welcher Knoten als Standort gewählt wird, da man für beide dieselbe mittlere zurückzulegende Distanz als auch Ungleichheit erhält. Wenn weitere Knoten zugewiesen werden sollen, wird dieser Standort an dem anderen Knoten platziert, da sich dieser näher an allen anderen Knoten befindet (s. Abb. 3.2, Lösung 10).



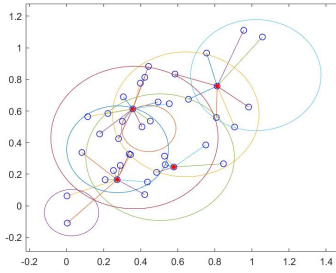
(a) Lösung 1



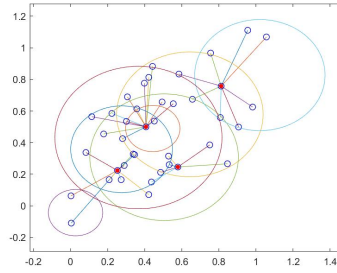
(b) Lösung 2



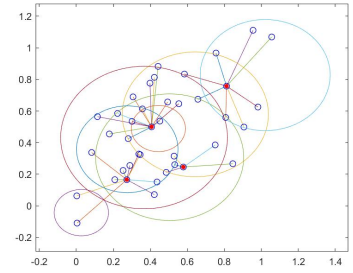
(c) Lösung 3



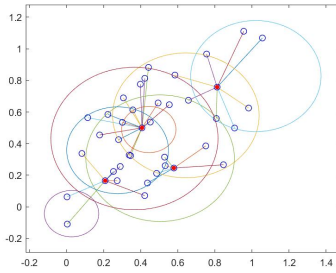
(d) Lösung 4



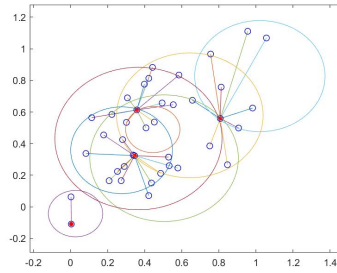
(e) Lösung 5



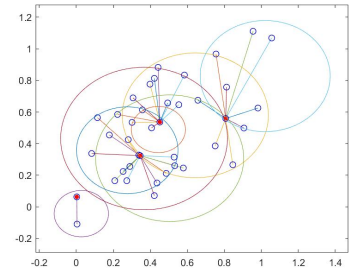
(f) Lösung 6



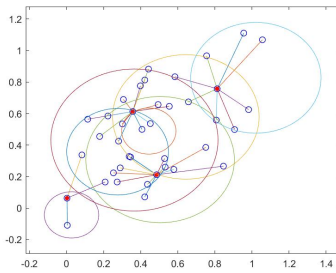
(g) Lösung 7



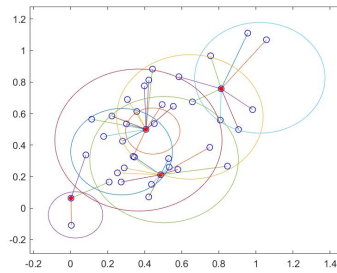
(h) Lösung 8



(i) Lösung 9



(j) Lösung 10



(k) Lösung 11

Abbildung 3.2: Pareto optimale Platzierungen der Einrichtungen bei 40 PK und 7 Szenarien

Die Funktionswerte beider Zielfunktionen sind in Abb. 3.3 dargestellt. Die Funktionswerte der ex post ZF sind auf der x-Achse und die der ex ante ZF auf der y-Achse aufgetragen. Außerdem werden in einer Tabelle (Abb. 3.4) alle Funktionswerte, sowie weitere Informationen zu den Lösungen angegeben. In der zweiten Spalte der Tabelle wird die mittlere zurückzulegende Distanz (aus ex ante bzw. ex post Sicht) angegeben. Diese Werte entsprechen dem Mittelwerts-Teil der ex ante bzw. ex post Zielfunktionen (s. Gleichungen 2.4 und 2.13). μ_{exante} ist also die mittlere zurückzulegende Distanz aus ex ante Sicht, d.h. es wird zuerst die erwartete Distanz für jedes Individuum berechnet und anschließend über alle Personen gemittelt. Bei μ_{expost} wird hingegen zuerst die mittlere Distanz je Szenario berechnet und davon der Erwartungswert gebildet. Wie bereits in Kapitel 2.3.3 erwähnt, werden bei der Berechnung der mittleren Distanz pro Szenario nur die von dem jeweiligen Szenario betroffenen Personen in Betracht gezogen. Dies erklärt die Differenz bei den Mittelwerten μ_{exante} und μ_{expost} .

Die in der Spalte GMAD (Ginis mean absolute difference) angegebenen Werte erhält man, wenn man vom Wert der ZF μ abzieht und durch λ dividiert (s. Gl. 2.4 und 2.13). Da Ginis mean absolute difference gleich $2\mu G$ ist, wobei G für den Gini-Koeffizienten steht, lässt sich dieser aus der zweiten und dritten Spalte berechnen und wird als Zusatzinformation angegeben. Um die Veränderungen zwischen den Lösungen zu veranschaulichen, ist in Abb. 3.5 jeweils aus ex ante bzw. ex post Sicht die Entwicklung der mittleren zurückzulegenden Distanz (μ), Ginis mean absolute difference (GMAD) und der Gini-Koeffizient grafisch dargestellt.

Von der ersten zur zweiten Lösung kann die ex post ZF bei einer moderaten Erhöhung der ex ante ZF deutlich verbessert werden. Auch bei dem Schritt von der vierten auf die fünfte Lösung wird der ex post Funktionswert deutlich kleiner, wobei die ex ante ZF nur minimal (in der sechsten Nachkommastelle) verschlechtert werden muss. Bei den Lösungen sieben bis zehn ist der Preis für eine Verbesserung der ex post ZF relativ hoch, da die ex ante ZF hier deutlich verschlechtert werden muss. Der Gini Koeffizient in der ex ante Betrachtung ist über alle Lösungen gesehen relativ konstant mit Werten nahe 0,3, was auf eine gute Gleichheit der zurückzulegenden Distanzen aus ex ante Sicht hindeutet. Der beste Gini-Wert aus ex ante Sicht wird erst in der neunten Lösung erreicht, wobei dies durch einen starken Anstieg der mittleren Distanz von Lösung 8 zu 9 und einer gleichzeitigen leichten Verbesserung der GMAD verursacht wird. Außerdem ist bei dieser Lösung der Gini-Koeffizient aus ex post Sicht am schlechtesten, da hier die mittlere Distanz zwar sehr gering, GMAD aber sehr hoch ist (s. Grafiken in Abb. 3.5). Aus ex post Sicht ist der Gini-Koeffizient in den ersten vier Lösungen mit Werten nahe 0,2

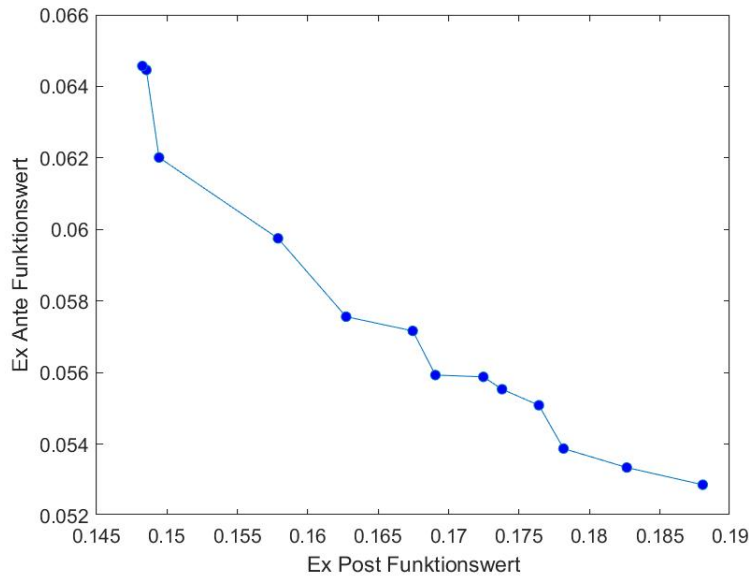


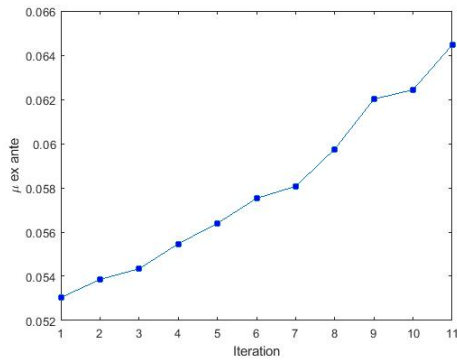
Abbildung 3.3: Pareto-Front für Anwendungsbeispiel 1 (von rechts unten nach links oben)

sehr gut, ab der fünften Lösung steigt er jedoch deutlich an. Mit Werten zwischen 0,25 und 0,35 ist die ex post Ungleichheit aber weiterhin auf einem niedrigen Niveau. Auf Kosten einer schlechteren Effizienz könnte man durch eine andere Wahl des λ -Parameters auch eine bessere Gleichheit erreichen. Weiters kann für die fortschreitenden Lösungen ein geringer aber kontinuierlicher Anstieg der mittleren zurückzulegenden Distanz μ aus ex ante Sicht festgehalten werden, während diese Werte aus ex post Sicht bis auf eine Ausnahme kontinuierlich abfallen (s. Abb. 3.5 a und b). Ginis mean absolute difference hingegen steigt zuerst an, fällt dann wieder leicht und erreicht den höchsten Wert in der zehnten Lösung, wo folglich auch der Gini Koeffizient für die ex ante Betrachtung den schlechtesten Wert erreicht. Da zuerst die ex ante Zielfunktion minimiert und dann graduell verschlechtert wird, um bessere ex post Funktionswerte zu erreichen, ist die mittlere Distanz aus ex post Sicht in den ersten Lösungen relativ hoch und verringert sich dann, während Ginis mean absolute difference bereits zu Beginn sehr niedrige Werte annimmt und sich in den letzten Lösungen sogar noch deutlich verschlechtert. Diese Tendenz wird jedoch durch die Verbesserung der Effizienz (d.h. Verringerung der mittleren zurückzulegenden Distanz) kompensiert.

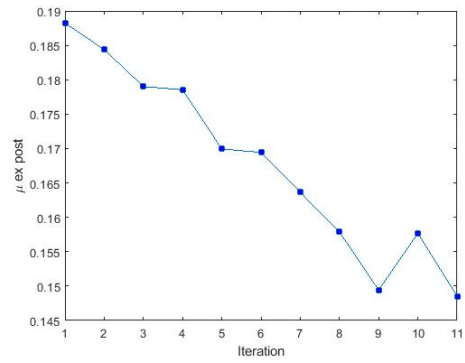
	μ	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05304	0,02959	0,27900	0,067834	1
Ex Post	0,18824	0,08826	0,23445	0,232370	1
Ex Ante	0,05385	0,03036	0,28183	0,069031	2
Ex Post	0,18440	0,07553	0,20481	0,222168	2
Ex Ante	0,05434	0,03226	0,29684	0,070470	3
Ex Post	0,17902	0,07691	0,21480	0,217473	3
Ex Ante	0,05547	0,03302	0,29761	0,071976	4
Ex Post	0,17854	0,07554	0,21154	0,216311	4
Ex Ante	0,05639	0,03117	0,27631	0,071977	5
Ex Post	0,16993	0,08750	0,25746	0,213676	5
Ex Ante	0,05752	0,03179	0,27637	0,073420	6
Ex Post	0,16945	0,08596	0,25365	0,212429	6
Ex Ante	0,05807	0,03201	0,27560	0,074072	7
Ex Post	0,16365	0,08684	0,26532	0,207070	7
Ex Ante	0,05975	0,03255	0,27240	0,076030	8
Ex Post	0,15788	0,09534	0,30196	0,205549	8
Ex Ante	0,06201	0,03245	0,26161	0,078236	9
Ex Post	0,14943	0,10539	0,35266	0,202127	9
Ex Ante	0,06244	0,03794	0,30381	0,081406	10
Ex Post	0,15768	0,08398	0,26632	0,199669	10
Ex Ante	0,06446	0,03594	0,27878	0,082433	11
Ex Post	0,14853	0,09342	0,31447	0,195242	11

Abbildung 3.4: Tabelle mit Informationen zu jeder Lösung

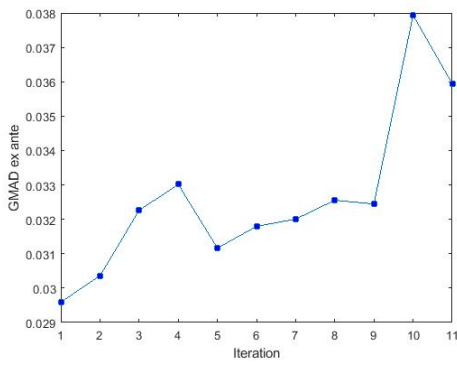
Ein Entscheidungsträger kann nun aus den elf optimalen Lösungen, die für die vorliegende Problemstellung aus seiner Sicht am besten passende auswählen. Dabei hängt diese Entscheidung von mehreren Faktoren ab. Zum einen muss zwischen Effizienz und Ungleichheit abgewogen werden, da manche Lösungen eine bessere Effizienz bieten, gleichzeitig aber eine größere Ungleichheit bei den zurückzulegenden Distanzen zur Folge haben. Außerdem muss man sich zwischen einer aus ex ante oder ex post Sicht optimalen Platzierung entscheiden, wobei eine Lösung, welche beide Aspekte berücksichtigt (das wären dann Lösungen aus den 'mittleren' Iterationen, wo die ex ante ZF schon zugunsten der ex post ZF verschlechtert wurde, aber trotzdem noch einen möglichst niedrigen Wert hat) in den meisten Fällen vorzuziehen ist. Außerdem sind bei diesen Lösungen (im vorliegenden Bsp. Lösung fünf bis sieben) die Werte aller Maße (s. Abb. 3.5) in einem mittleren bis guten Bereich. Durch die große Auswahl an optimalen Lösungen kann auch besser auf die lokalen Gegebenheiten eingegangen werden und die Flexibilität bei der Entscheidungsfindung ist deutlich höher als wenn nur eine bzw. einige wenige Lösungen



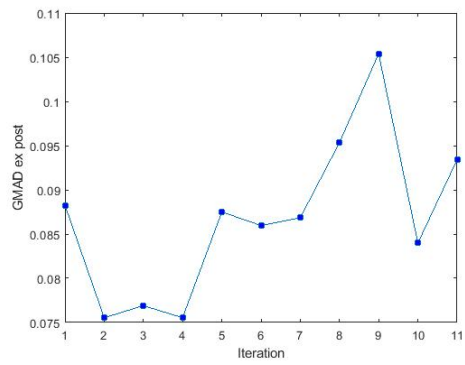
(a) Mittlere Distanz ex ante



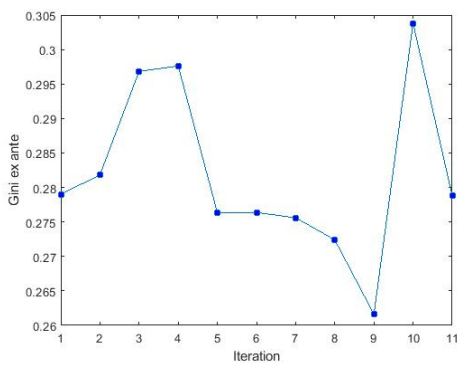
(b) Mittlere Distanz ex post



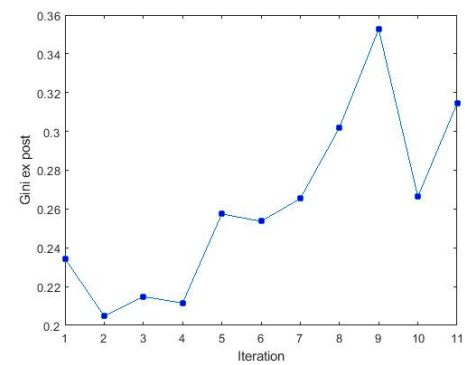
(c) GMAD ex ante



(d) GMAD ex post



(e) Gini-Koeffizient ex ante



(f) Gini-Koeffizient ex post

Abbildung 3.5: Veranschaulichung der Werte aus Abb. 3.4

zur Verfügung stehen würden.

In der Garantie alle pareto optimalen Lösungen zu finden, liegt auch der große Vorteil des adaptive epsilon constraint Algorithmus im Gegensatz zu anderen Methoden wie z.B. die Verwendung eines gewichteten Mittels aus den beiden Zielfunktionen. Um bei zweiterem Ansatz mehrere Lösungen zu erhalten, muss man die verwendeten Gewichte variieren, was computationally sehr aufwendig ist, wenn man die Gewichte nur geringfügig ändert um keine Lösungen zu übersehen. Außerdem kann man mit dieser Methode nur jene pareto optimalen Lösungen finden, welche in einer konvexen Region des Zielfunktionsraumes liegen (vgl. Laumanns et al. [8]). Im vorliegenden Beispiel würden man also den Großteil der pareto optimalen Lösungen nicht finden (s. Abb 3.3) und ist dadurch bei der Entscheidungsfindung deutlich stärker eingeschränkt.

Zusammenfassend kann also festgehalten werden, dass das vorliegende Modell gut funktioniert und genügend Lösungen gefunden werden, um eine gute und fundierte Entscheidung bzgl. der Wahl der Standorte für die Notversorgungseinrichtungen treffen zu können. Durch das Modell wird dabei für alle Lösungen garantiert, dass sowohl eine gute Effizienz bzgl. der mittleren zurückzulegenden Distanzen als auch eine befriedigende Ungleichheits-Aversion gegeben ist. Dadurch müssen alle Personen sowohl einen möglichst kurzen Weg zur nächsten Einrichtung zurücklegen und gleichzeitig erleidet niemand einen zu großen Nachteil.

3.4 Anwendungsbeispiel 2

In dem nächsten Anwendungsbeispiel wurden alle optimalen Lösungen für eine größere Anzahl an Szenarien gesucht. Aufgrund der bereits erwähnten computationalen Einschränkungen wurden 20 Szenarien sowie 30 Populationsknoten - welche wiederum gleichzeitig die möglichen Standorte sind - zufällig gewählt. Einen Überblick über die verwendeten Daten bietet Abb. 3.6. Wie auch bereits im ersten Anwendungsbeispiel sind die meisten Epizentren zentral gelegen und auch der Großteil der Knoten liegen in diesem Gebiet. Die Eintrittswahrscheinlichkeit wurde wieder für jedes Szenario gleich groß, also mit $1/20$ gewählt.

Alle pareto optimalen Lösungen sind in Abb. 3.7 abgebildet. Die Funktionswerte sind in Abb. 3.8 dargestellt und für jede Lösung wird sowohl für die ex ante, als auch die ex post Betrachtungsweise wieder die mittlere Distanz μ , Ginis mean absolute difference

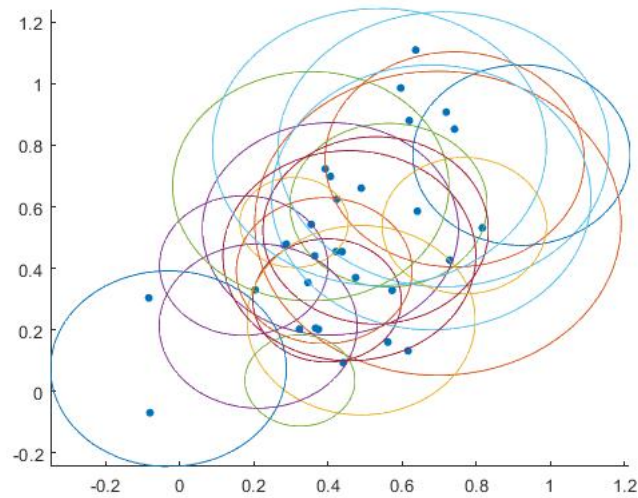
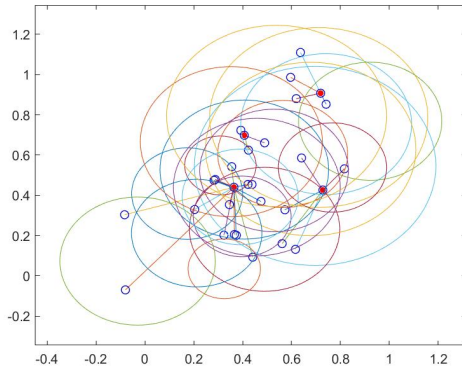


Abbildung 3.6: Zufallsgenerierte Daten mit 30 PK und 20 Szenarien

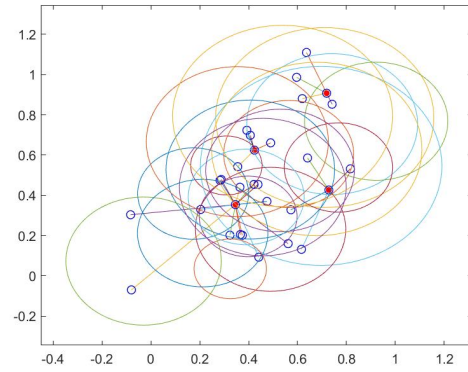
(GMAD), sowie der Gini Koeffizient angegeben (s. Abb 3.9).

Die im Zentrum gelegenen Knoten sind im vorliegenden Beispiel von bis zu 13 Szenarien gleichzeitig bedroht, während die abgelegeneren Knoten im Süd-Westen nur in der Risikozone eines Erdbebens liegen. Die nord-östlich gelegenen Knoten sind mit zwei bis fünf Szenarien etwas stärker bedroht, jedoch ist auch hier die Gefahr deutlich geringer als im Zentrum. Daher sind in allen Lösungen drei der vier Knoten sehr zentral gelegen und bieten allen stark bedrohten Populationsknoten einen guten Zugang zu den Notversorgungseinrichtungen.

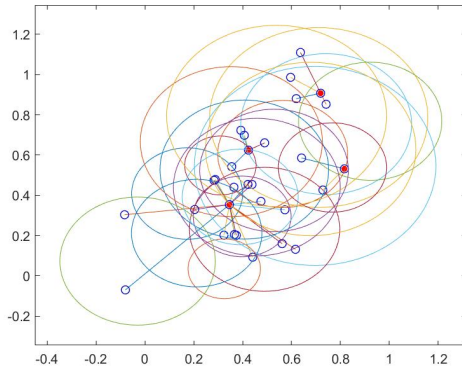
Durch das geringe Risiko, welchem die beiden süd-westlichen gelegenen Knoten ausgesetzt sind, müssen diese stets eine große Distanz zur nächsten Einrichtung zurücklegen. Da im Nord-Osten gleich fünf Knoten relativ nahe beieinander liegen, wird in allen Lösungen ein Standort in diesem Gebiet gewählt, obwohl das Erdbebenrisiko in der Region deutlich geringer als im Zentrum ist (diese Knoten liegen im Auswirkungsbereich von 2-6 Beben, während die Knoten im Zentrum von 8-13 Erdbebenszenarien bedroht werden). Durch die Tatsache, dass man im Zentrum bereits mit drei Standorten allen Knoten eine kurze Distanz ermöglichen kann, ist es möglich einen Standort in der weniger bedrohten Region zu platzieren.



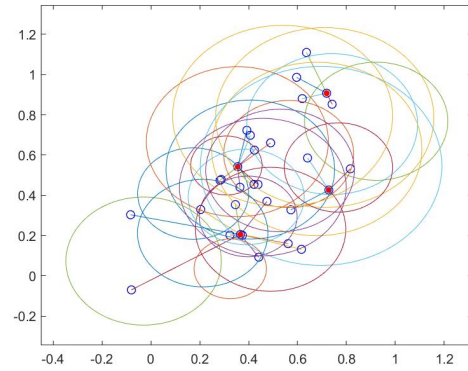
(a) Lösung 1



(b) Lösung 2



(c) Lösung 3



(d) Lösung 4

Abbildung 3.7: Pareto-optimale Platzierungen der Einrichtungen bei 30 PK und 20 Szenarien

Bei der Betrachtung der erreichten ex ante bzw. ex post Zielfunktionswerte ist es vor allem auffallend, dass von der ersten zur zweiten Lösung durch eine minimale Verschlechterung der ex ante ZF die ex post ZF deutlich verbessert werden kann. Nach der zweiten Lösung muss hingegen für eine kleine Verbesserung in der ex post ZF jedes Mal eine starke Verschlechterung der ex ante ZF in Kauf genommen werden. Um also für beide Betrachtungsweisen einen guten Wert zu erreichen, ist in dem vorliegenden Beispiel die Lösung aus Iteration 2 die am besten geeignetste, da hier im Vergleich zu den anderen Lösungen sowohl für die ex ante als auch ex post ZF gute Werte erreicht werden. Außerdem sollte, wie bereits erwähnt, eine diversifizierte Lösung in den meisten Fällen einer einseitigen Lösung (d.h. wo nur eine der Zielfunktionen einen sehr guten Wert aufweist) vorgezogen werden. Zusätzlich wird mit der Lösung aus Iteration 2 sowohl ex ante als

auch ex post der geringste Gini-Koeffizient erreicht, d.h. die Ungleichheit ist in dieser Lösung am geringsten (s. Abb. 3.9).

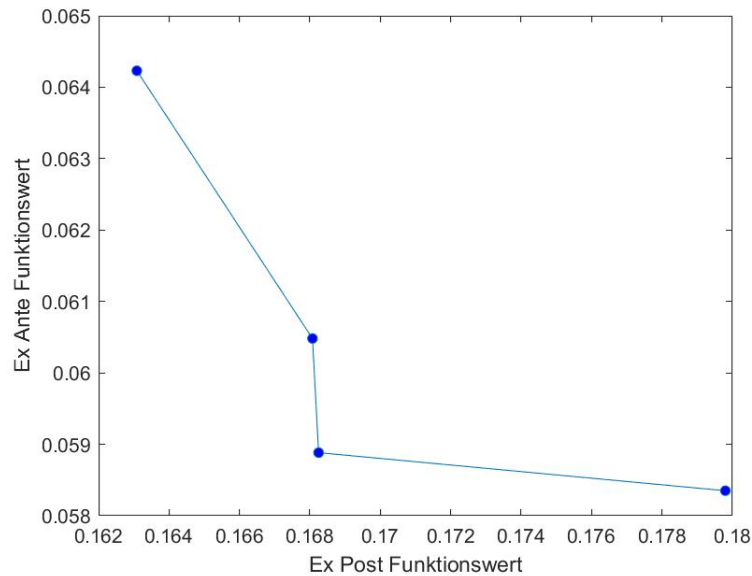


Abbildung 3.8: Funktionswerte von jeder Lösung (von rechts unten nach links oben)

Name	ADTS	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,04322	0,03026	0,35001	0,058351	1
Ex Post	0,14336	0,07287	0,25416	0,179799	1
Ex Ante	0,04434	0,02909	0,32807	0,058884	2
Ex Post	0,13727	0,06196	0,22567	0,168246	2
Ex Ante	0,04532	0,03033	0,33463	0,060482	3
Ex Post	0,13713	0,06189	0,22568	0,168074	3
Ex Ante	0,04565	0,03717	0,40715	0,064231	4
Ex Post	0,13012	0,06774	0,26030	0,163990	4
Ex Ante	0,04565	0,03717	0,40715	0,064231	5
Ex Post	0,13012	0,06593	0,25334	0,163086	5

Abbildung 3.9: Tabelle mit Informationen zu jeder Lösung

Interessant ist außerdem, dass für die vorliegenden Daten im Vergleich zum ersten Anwendungsbeispiel deutlich weniger pareto optimale Lösungen gefunden werden konnten. Für eine andere zufällige Verteilung der Szenarien und Knoten gäbe es möglicherweise mehr Lösungen, aber durch die geringere Anzahl der Knoten - was die Variation der Standorte stärker einschränkt - und das hohe Risiko im Zentrum - was aus Effizienzgrün-

den eine stärkere Konzentration der Standorte im Zentrum erfordert - ist es durchaus plausibel, dass für diese Kombination aus Szenarien und Knoten weniger pareto optimale Lösungen existieren.

3.5 Laufzeitanalyse

Um die Performance des Modells im Hinblick auf die Laufzeit zu messen, wurden für verschiedene Kombinationen aus der Anzahl der Knoten und Szenarien die durchschnittliche Laufzeit gemessen. Jeder PK ist wieder gleichzeitig ein möglicher Standort für eine Einrichtung. Dabei wurde für jedes vorliegende Problem unter Verwendung des adaptive epsilon constraint Algorithmus für jede Iteration die Zeit gemessen, die der Optimierungsvorgang (d.h. die Ausführung der 'cplexmilp'-Funktion) benötigte. Die Laufzeit der Vorbereitungen, wie die Erstellung der Nebenbedingungs-Matrizen und andere Zuweisungen von Variablen etc., wurde dabei nicht berücksichtigt. Nachdem alle pareto-optimalen Lösungen gefunden wurden, wurde der Mittelwert aller Iterationen pro Knoten-Szenarien-Kombination berechnet. Gemessen wurden dabei die Laufzeiten für alle Kombinationen aus 3,4,10,15,20,25 und 30 Szenarien, sowie 20,25,30,40, bzw. 50 Populationsknoten. Aufgrund der bereits angesprochenen Problematik der Speicherkapazität und der damit einhergehenden Beschränkung der Problemgrößen konnte die Laufzeit für einige Kombinationen nicht gemessen werden, da nicht genügend Speicher für die Verarbeitung der benötigten Matrizen zur Verfügung stand. In Abb. 3.10 ist die Laufzeit in Sekunden für alle Kombinationen, für die sie gemessen werden konnte als dunkle ausgefüllte Punkte abgebildet. Alle Kombinationen, wo die Laufzeit wegen der zu geringen Speicherkapazitäten nicht gemessen werden konnte, sind als helle Kreise mit einem Laufzeitwert von 0 abgebildet. In der Abb. 3.10 sieht man einen deutlichen Anstieg der Laufzeit, je größer die Anzahl der Szenarien oder Knoten wird. Bis zu einer Größe von 10 Szenarien und 40 Knoten (und 40 möglichen Standorten) kann die Laufzeit für alle Kombinationen berechnet werden. Ab dann ist die Laufzeitmessung bei jeder Steigerung der Szenarienzahl durch die Speicherbeschränkung für immer weniger Knoten möglich. Die Laufzeit ist selbst für die größten möglichen Berechnungen mit weniger als 150 Sekunden pro Iteration akzeptabel. Es muss jedoch bei der Ausführung des adaptive epsilon constraint Algorithmus beachtet werden, dass die Laufzeit zwischen den einzelnen Iterationen schwanken kann und dass sich die Laufzeit für die vollständige Durchführung des Algorithmus je nach der Anzahl der Iterationen deutlich erhöhen kann.

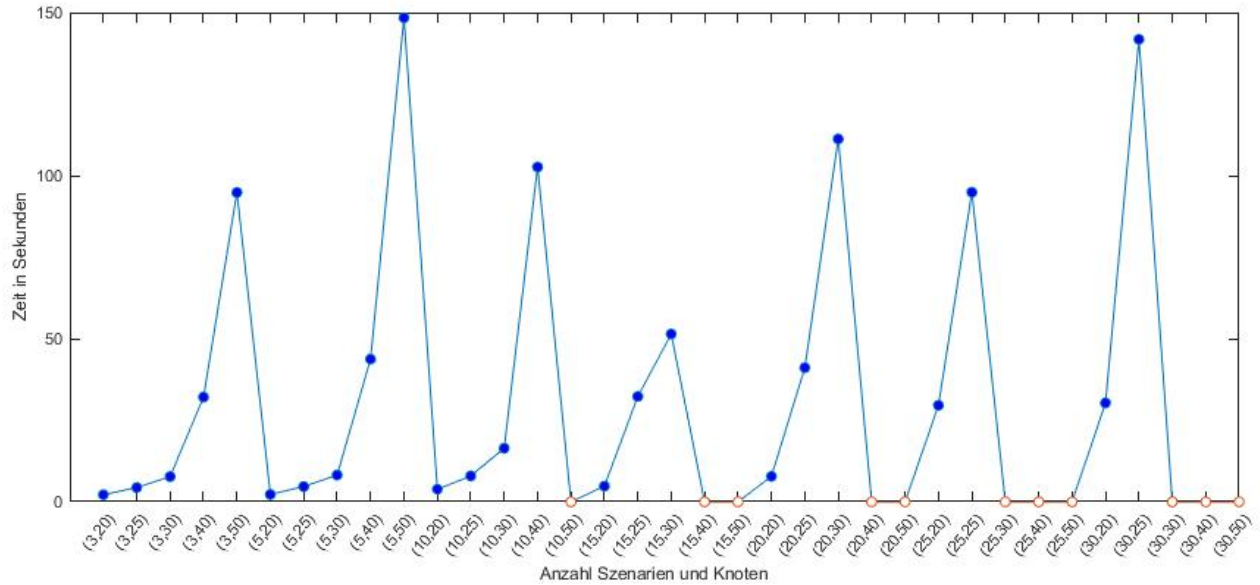


Abbildung 3.10: Laufzeit für versch. Kombinationen von Szenarien und Knoten.
Die Werte auf der x-Achse geben die Anzahl der Szenarien (1. Zahl) und Anzahl der PK und KS (immer gleich hoch, 2. Zahl) an

Zusätzlich wurde für jede Kombination auch die Anzahl der benötigten Entscheidungsvariablen gespeichert und mit der Laufzeit verglichen. Insgesamt gibt es für eine Kombination aus n Knoten, s möglichen Standorten (hier gilt immer $s = n$) und p Szenarien $s + ns + 2n^2 + 2n^2p$ Entscheidungsvariablen (s. Kapitel 3.2). Die benötigte Laufzeit für jede Variablenanzahl, die sich aus dem vorliegenden Problem ergibt, ist in Abb. 3.11 abgebildet. Klarerweise steigt die Laufzeit stark an, je mehr Variablen vorliegen. Neben der absoluten Variablenanzahl spielt natürlich auch deren Zusammensetzung (Anzahl Szenarien vs. Anzahl Knoten) und auch viele andere Faktoren eine Rolle, welche die Laufzeit positiv oder negativ beeinflussen können.

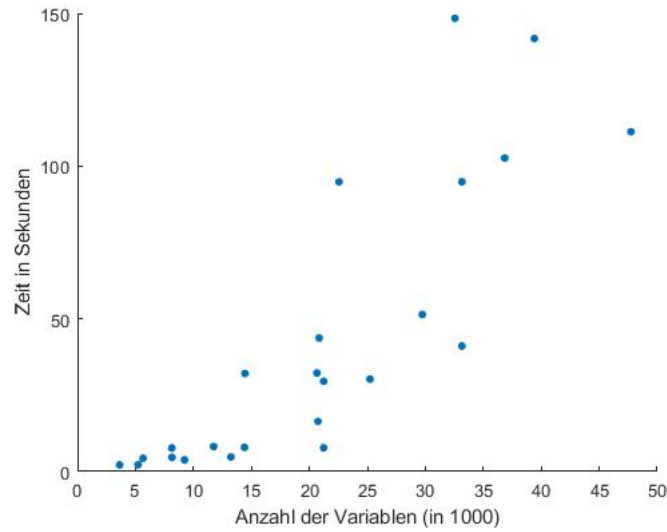


Abbildung 3.11: Variablenanzahl vs. Laufzeit

Abschließend lässt sich festhalten, dass für das vorliegende Modell weniger die Laufzeit, sondern der benötigte Arbeitsspeicher einen limitierenden Faktor für die Durchführung der Optimierung darstellt. Mit einer besseren Hardware könnte man jedoch deutlich größere Probleme mit mehr Knoten und Szenarien innerhalb einer akzeptablen Zeit lösen.

3.6 Variation des λ Parameters

Um entweder eine bessere Effizienz oder aber weniger Ungleichheit in den Lösungen zu erreichen, kann der λ Parameter verändert werden. Wird $\lambda = 0$ gewählt, wird bei der Optimierung der ZF nur der Mittelwerts-Teil μ beachtet und Ginis mean absolute difference, welche mit dem Parameter λ multipliziert wird, spielt keine Rolle mehr (s. Kapitel 2.3). Je größer der Parameter gewählt wird, desto höher wird die Ungleichheits-Komponente in Form von Ginis mean absolute difference gewichtet, was natürlich wieder einen Verlust an Effizienz zur Folge hat. Wie bereits im Theorieteil erwähnt, sollte der Parameter zwischen 0 und 0,5 gewählt werden. In diesem Kapitel werden jedoch auch Lösungen analysiert, welche man durch eine höhere Gewichtung erhält. Dieselbe Problemstellung wird im folgenden Abschnitt unter Verwendung verschiedener λ -Parameter analysiert und die Lösungen werden sowohl in Bezug auf die erreichte Effizienz, als auch die Ungleichheit untersucht und miteinander verglichen. Für diese Analyse werden dieselben Daten wie im Anwendungsbeispiel 1 verwendet. Es gibt also 40 Populationsknoten,

welche gleichzeitig die möglichen Standorte für die Einrichtungen darstellen, sowie 7 Szenarien.

Es werden alle Ergebnisse des epsilon constraint Algorithmus verglichen, wobei $\lambda = 0, 0,25, 0,27, 0,28, 0,5, 0,75$ und 1 gewählt wird. Die Vergleichbarkeit der Ergebnisse ist insofern schwierig, da durch die verschiedenen Anzahlen an Iterationen und die unterschiedlichen Ausgangssituationen in jeder Iteration durch die Beschränkung der ex post ZF die erreichten Werte (μ , GMAD und Gini) nicht aus derselben Ausgangssituation berechnet werden können. Lediglich bei der ersten Iteration ist die Ausgangslage für alle Varianten identisch. Aus diesem Grund werden im Folgenden vorwiegend die Ergebnisse aus der ersten Iteration miteinander verglichen. In den vier Grafiken (Abb. 3.12) werden die Effizienzwerte (μ) sowie die Ungleichheitswerte (GMAD) für alle sieben λ -Varianten sowohl ex ante als auch ex post verglichen. Wie erwartet steigt bei einer Erhöhung des Parameters λ die mittlere zu überwindende Distanz für die Betroffenen (s. Grafiken (a) und (b) in Abb. 3.12). Gleichzeitig sinkt die Ungleichheit und somit Gini's mean absolute difference (s. Grafiken (c) und (d)). Wie in den Grafiken gut erkennbar ist, gibt es keine graduelle Veränderung bei den gefundenen Lösungen, sondern eine Sprungstelle. Im vorliegenden Beispiel ist die Lösung der ersten Iteration bei einer Wahl von $0 \leq \lambda \leq 0,26$ immer gleich. Bei einer Wahl von $0,27 \leq \lambda \leq 1$ erhält man auch immer dieselbe Lösung, welche sich aber von der Lösung bei einer Wahl kleiner $0,26$ unterscheidet. Da es nur eine Sprungstelle und nicht kleinere graduelle Veränderungen in den ersten Lösungen gibt, liegt vermutlich an der moderaten Beispielgröße mit 40 PK's und 7 Szenarien und der damit einhergehenden eingeschränkten Variationsmöglichkeit der Standorte, sowie der Konzentration der Populationsknoten und Szenarien im Zentrum. Durch diese starke Konzentration im Zentrum muss man bei einer Verbesserung der Gleichheit durch eine Verschiebung der Einrichtungen in periphere Gebiete eine starke Verschlechterung der mittleren zurückzulegenden Distanz in Kauf nehmen.

Alle Ergebnisse, die Tabellen und die Plots der gewählten Standorte für alle Lösungen sind im Appendix A angegeben.

Wie bereits erwähnt, sind die Ergebnisse für die Verschiedenen λ Werte ab der zweiten Iteration nicht mehr miteinander vergleichbar, da die Ausgangslagen unterschiedlich sind. Da die Nebenbedingung, welche ein gewisses Niveau der ex post ZF garantiert nach jeder Iteration individuell angepasst wird, entstehen hier unterschiedliche Ausgangssituationen.

Grundsätzlich ist aber die Wahl von $\lambda = 0,5$, welche für die Anwendungsbeispiele verwendet wurde wohl in den meisten Fällen am sinnvollsten. Aufgrund der Theorie sollte λ zwischen 0 und 0,5 gewählt werden (s. Kapitel 2.3.1). Aus praktischen Überlegungen und

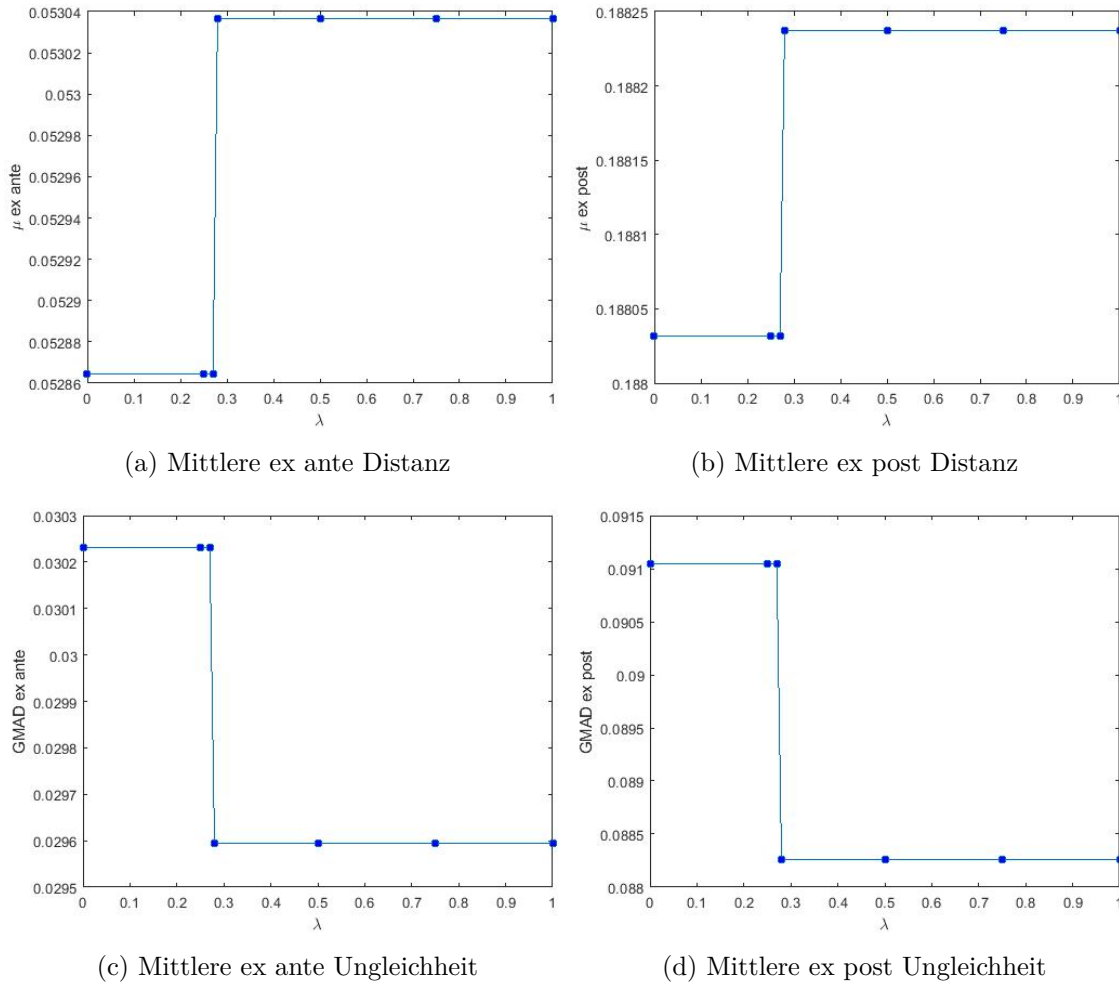


Abbildung 3.12: Vergleich bei Variation des λ Parameters nach der ersten Iteration

um eine gute Balance zwischen der Effizienz und ungleichheits-Aversion zu erhalten wurde daher der aus theoretischer Sicht maximal zulässige Wert von 0,5 gewählt. Aufgrund der Problemgröße und der Auswahlmöglichkeiten bei den Standorten sieht man jedoch bei genauerer Betrachtung aller verschiedenen Pareto-Fronten (s. Kapitel 3.3 und Appendix A), dass die Unterschiede der gewählten Standorte nicht besonders groß sind. Bei anderen Problemstellungen könnte aber eine Wahl von λ größer als 0,5 zu dem oben beschriebenen Problem führen, dass Knoten nicht zu dem am nächsten gelegenen Standort zugewiesen werden. Wählt man λ kleiner 0,5, kann dieses Problem zwar nicht auftreten, jedoch wird dann die Ungleichheit nur mehr in sehr geringem Maße berücksichtigt und man erhält zwar effiziente aber oftmals auch sehr ungerechte Lösungen.

4. Fazit

Das vorgestellte Modell bietet für Problemstellungen moderater Größenordnungen einen vielversprechenden Ansatz um mehrere optimale Lösungen zu finden und damit die Entscheidungsfindung bei der Standortbestimmung für Hilfseinrichtungen in Erdbebengebieten zu unterstützen. Als optimal werden in diesem Kontext alle Lösungen angesehen, welche die folgenden Kriterien erfüllen: Erstens ist für alle Lösungen pareto Optimalität garantiert, dass heißt es gibt keine weiteren Lösungen, welche eine der Zieleigenschaften verbessern ohne dabei eine andere zu verschlechtern. Zweitens sind alle Lösungen sowohl effizient als auch fair bzw. gerecht. Je nach Wahl der Parameter kann ein größerer Fokus auf Effizienz (d.h. die mittlere zu überwindende Distanz wird minimiert) oder Gleichheit (d.h. man erhält Lösungen, bei denen die zu überwindenden Distanzen für alle Betroffenen möglichst gleich groß sind) gelegt werden.

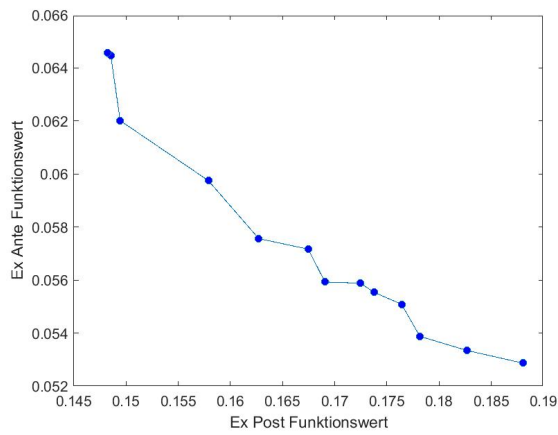
Außerdem können je nach den Präferenzen eines Entscheidungsträgers Lösungen gewählt werden, die entweder einen stärkeren Fokus auf die ex ante oder ex post Sicht legen, oder aber Lösungen, bei denen beide Aspekte ausgewogen sind. Durch diese vielfältigen Auswahlmöglichkeiten und die zusätzliche Option das Modell und die daraus resultierenden Lösungen durch eine Anpassung der Parameter noch genauer auf die vorliegende Problemstellung zuzuschneiden, wird eine gute Ausgangslage für die Entscheidungsfindung geschaffen.

Durch die für die Implementierung des Modells notwendige, aber sehr speicherintensive Erstellung der Zielfunktionen und Nebenbedingungen ist das Anwendungsgebiet auf überschaubare Problemgrößen beschränkt. Um Lösungen auch für größere Problemstellungen finden zu können, könnte dieses Modell jedoch als Ausgangspunkt zur Erforschung von neuen Algorithmen verwendet werden, die dann möglicherweise nicht alle pareto optimalen Lösungen finden, dafür aber deutlich größere Datenmengen verarbeiten können.

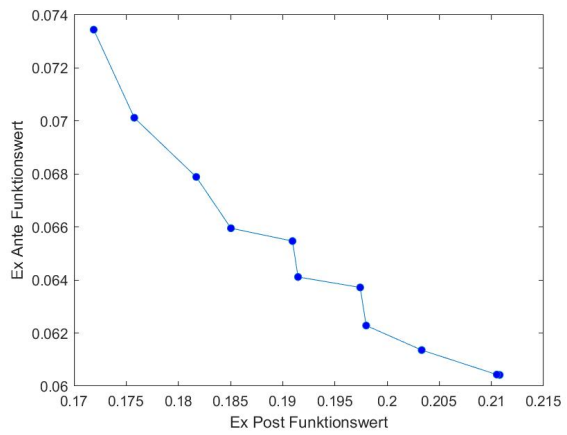
Appendices

A. Detailergebnisse für verschiedene λ -Parameter

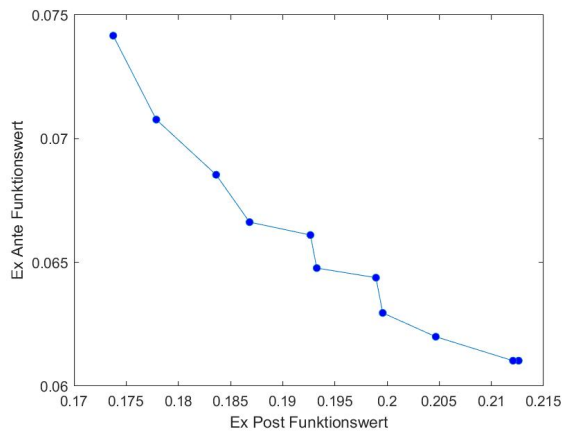
Die verwendeten Daten in allen folgenden Ergebnissen werden im Kapitel 3.3 erläutert. Es werden 40 PK, 40 KS und 7 Szenarien verwendet. Hier werden nur die Ergebnisse für $\lambda = 0, 0,25, 0,27, 0,28, 0,75$ und 1 präsentiert. Die Ergebnisse für $\lambda = 0,5$ wurden bereits ausführlich im Kapitel 3.3 diskutiert.



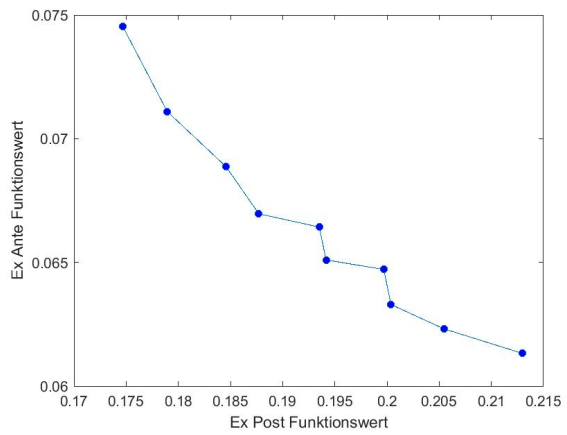
(a) $\lambda = 0$



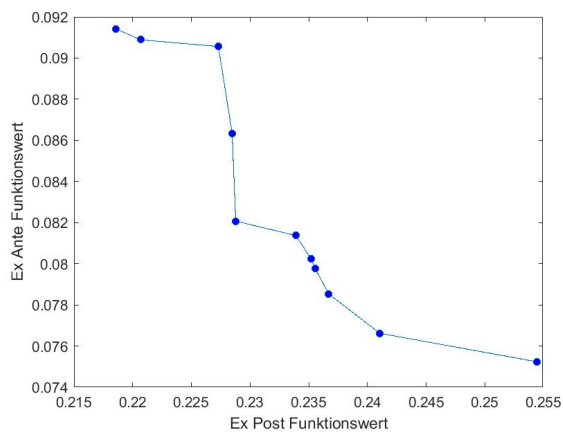
(b) $\lambda = 0,25$



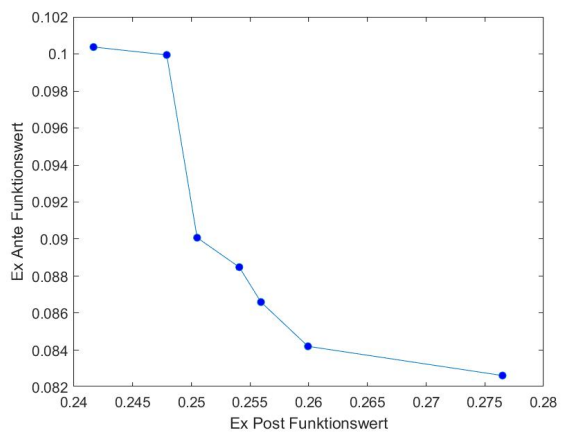
(c) $\lambda = 0,27$



(d) $\lambda = 0,28$

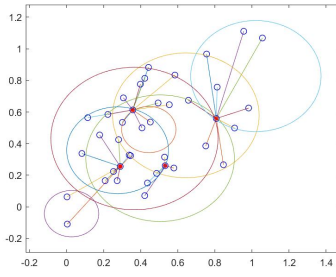


(e) $\lambda = 0,75$

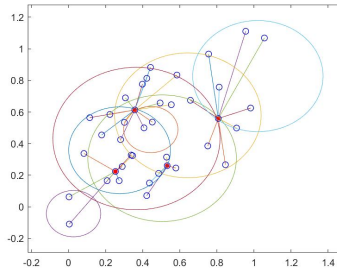


(f) $\lambda = 1$

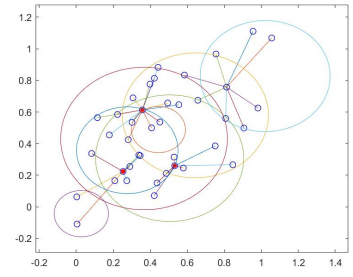
Abbildung A.1: Ex ante und ex post Zielfunktionswerte



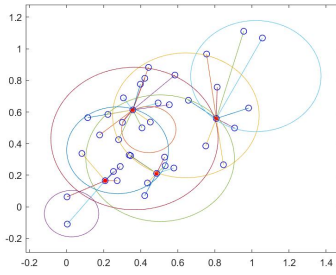
(a) Lösung 1



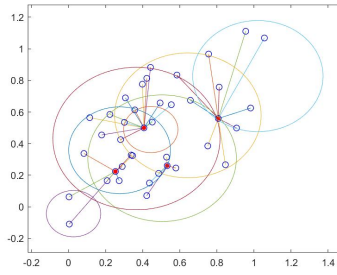
(b) Lösung 2



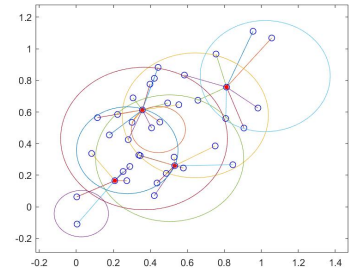
(c) Lösung 3



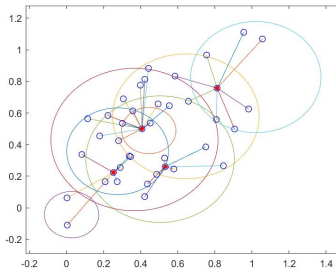
(d) Lösung 4



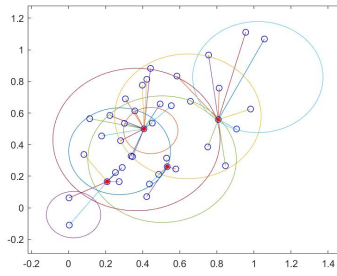
(e) Lösung 5



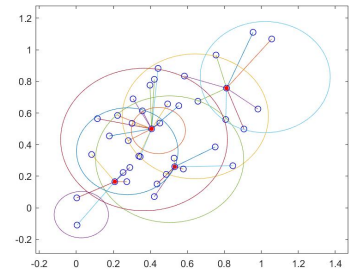
(f) Lösung 6



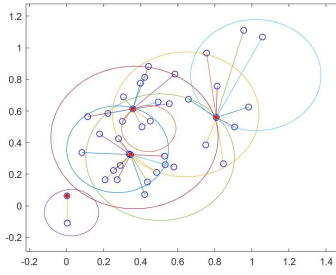
(g) Lösung 7



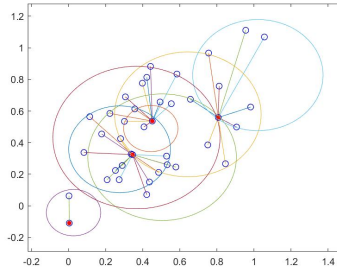
(h) Lösung 8



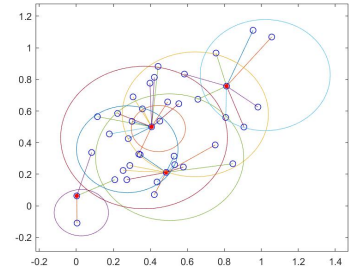
(i) Lösung 9



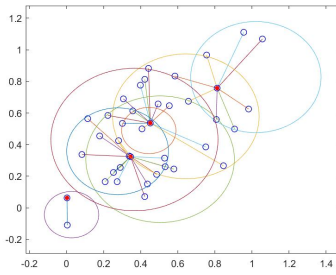
(j) Lösung 10



(k) Lösung 11

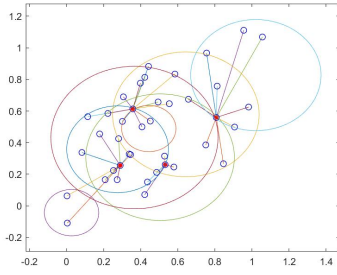


(l) Lösung 12

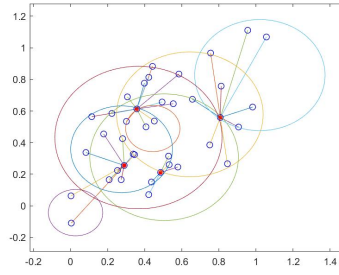


(m) Lösung 13

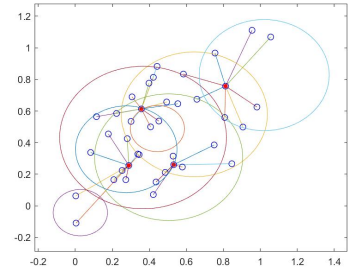
Abbildung A.2: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0$



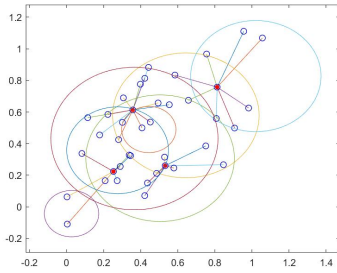
(a) Lösung 1



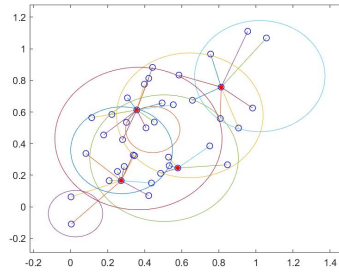
(b) Lösung 2



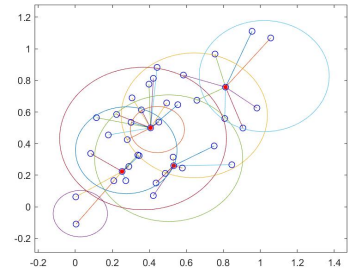
(c) Lösung 3



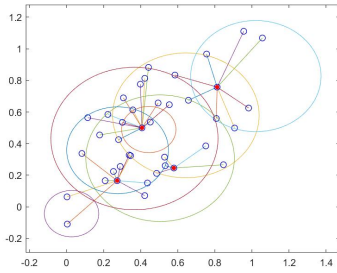
(d) Lösung 4



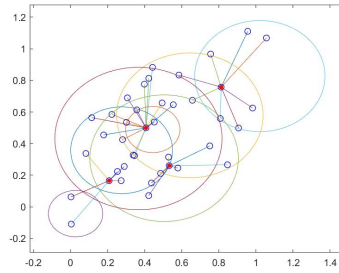
(e) Lösung 5



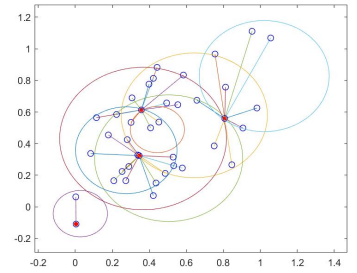
(f) Lösung 6



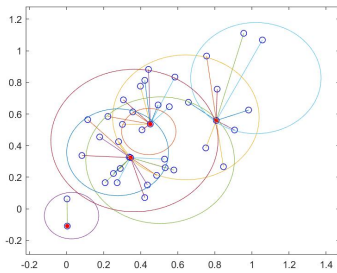
(g) Lösung 7



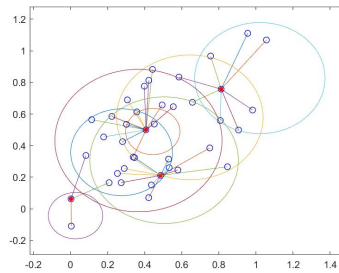
(h) Lösung 8



(i) Lösung 9

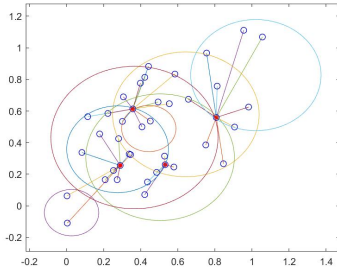


(j) Lösung 10

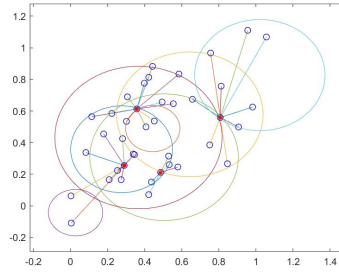


(k) Lösung 11

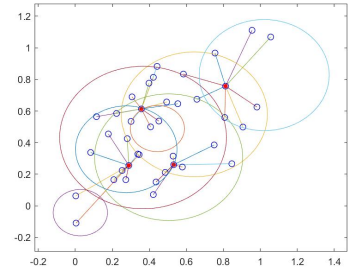
Abbildung A.3: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,25$



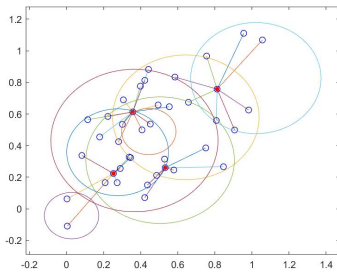
(a) Lösung 1



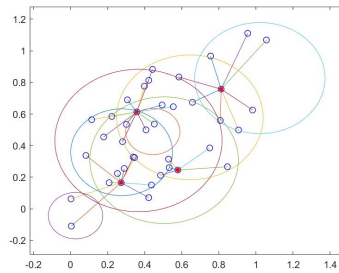
(b) Lösung 2



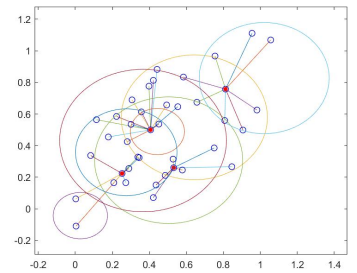
(c) Lösung 3



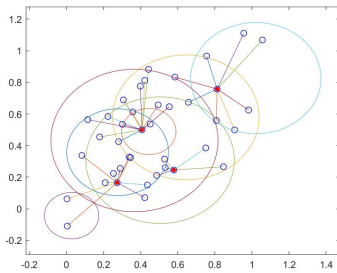
(d) Lösung 4



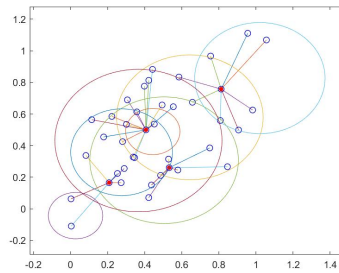
(e) Lösung 5



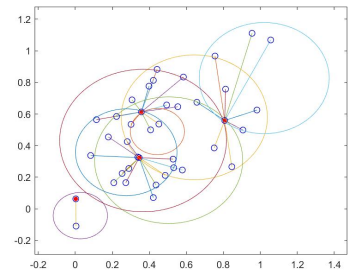
(f) Lösung 6



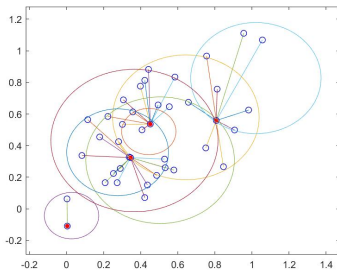
(g) Lösung 7



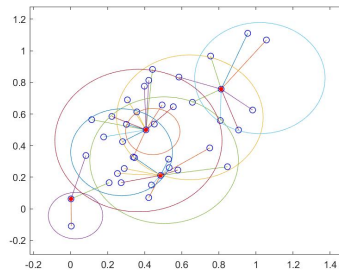
(h) Lösung 8



(i) Lösung 9

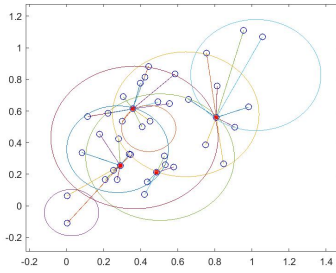


(j) Lösung 10

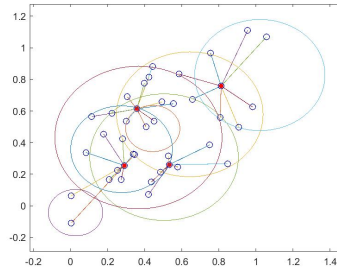


(k) Lösung 11

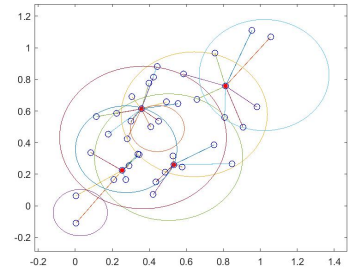
Abbildung A.4: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,27$



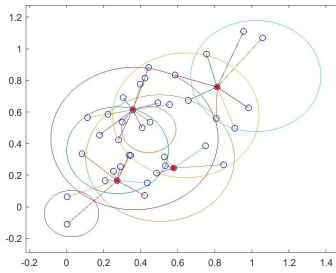
(a) Lösung 1



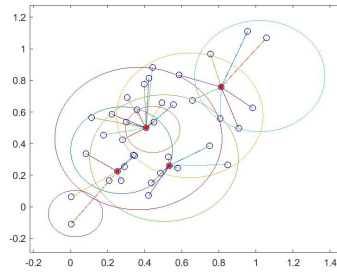
(b) Lösung 2



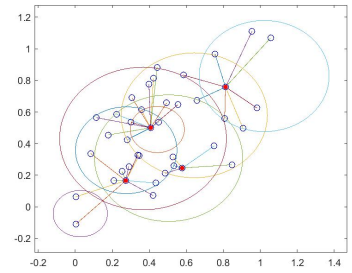
(c) Lösung 3



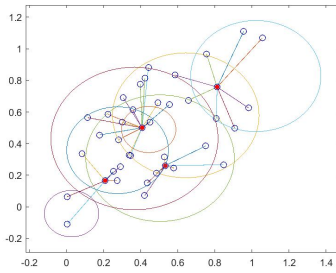
(d) Lösung 4



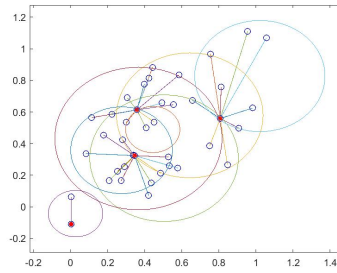
(e) Lösung 5



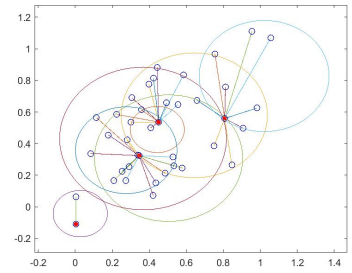
(f) Lösung 6



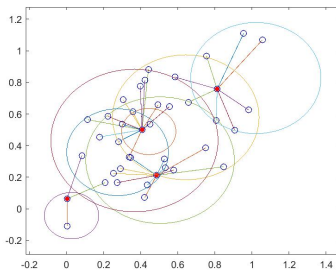
(g) Lösung 7



(h) Lösung 8

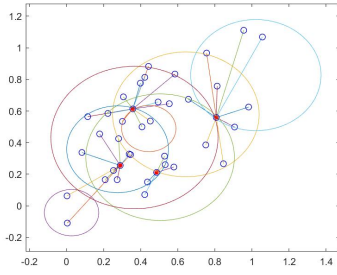


(i) Lösung 9

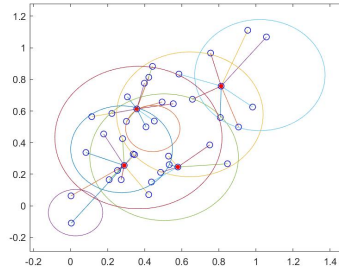


(j) Lösung 10

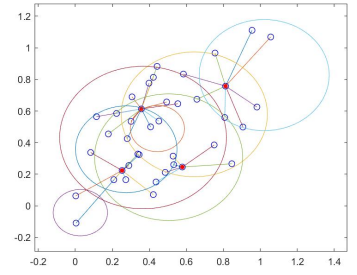
Abbildung A.5: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,28$



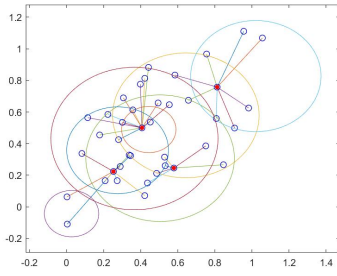
(a) Lösung 1



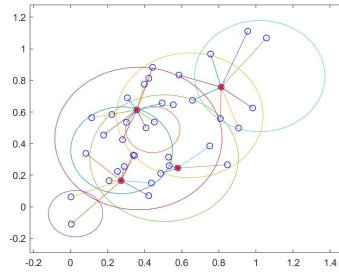
(b) Lösung 2



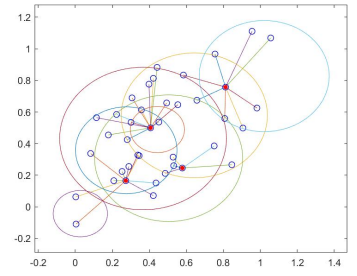
(c) Lösung 3



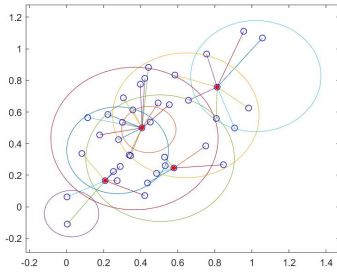
(d) Lösung 4



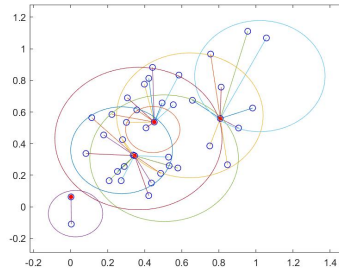
(e) Lösung 5



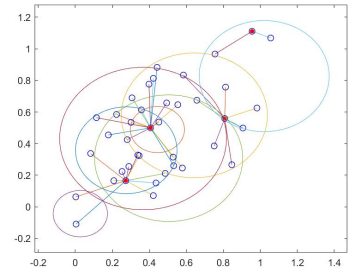
(f) Lösung 6



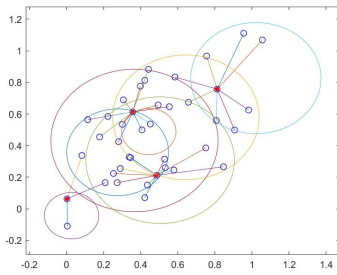
(g) Lösung 7



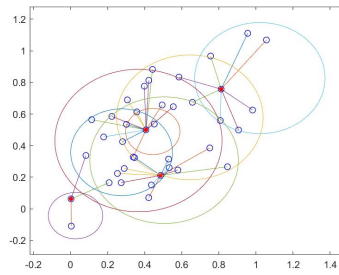
(h) Lösung 8



(i) Lösung 9

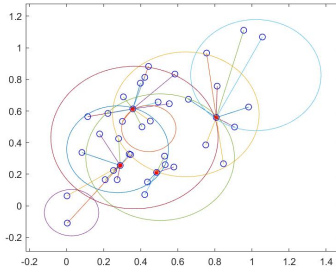


(j) Lösung 10

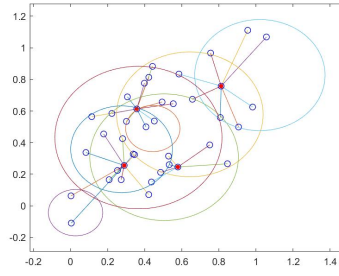


(k) Lösung 11

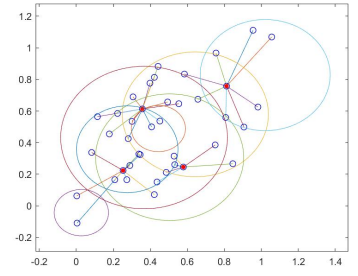
Abbildung A.6: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 0,75$



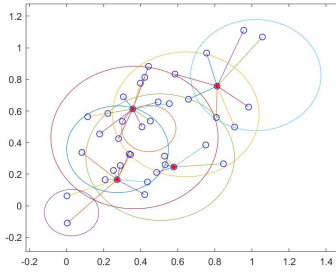
(a) Lösung 1



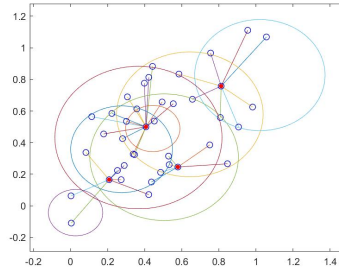
(b) Lösung 2



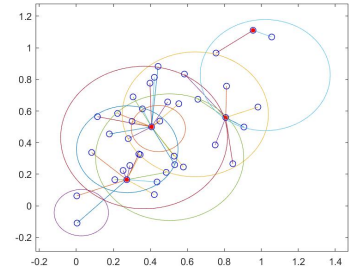
(c) Lösung 3



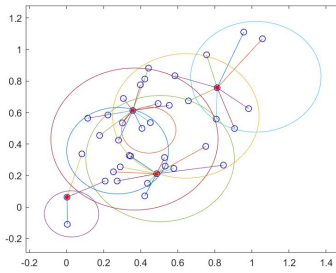
(d) Lösung 4



(e) Lösung 5



(f) Lösung 6



(g) Lösung 7

Abbildung A.7: Pareto-optimale Platzierungen der Einrichtungen für $\lambda = 1$

Name	μ	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05286	0,03023	0,28593	0,052864	1
Ex Post	0,18803	0,09105	0,24212	0,188031	1
Ex Ante	0,05335	0,03208	0,30068	0,053346	2
Ex Post	0,18264	0,09232	0,25274	0,182642	2
Ex Ante	0,05388	0,03364	0,31217	0,053876	3
Ex Post	0,17815	0,07941	0,22288	0,178146	3
Ex Ante	0,05509	0,03239	0,29402	0,055087	4
Ex Post	0,17640	0,08985	0,25468	0,176398	4
Ex Ante	0,05553	0,03103	0,27938	0,055534	5
Ex Post	0,17378	0,10340	0,29751	0,173782	5
Ex Ante	0,05588	0,03515	0,31449	0,055881	6
Ex Post	0,17246	0,07978	0,23130	0,172457	6
Ex Ante	0,05593	0,03275	0,29273	0,055931	7
Ex Post	0,16905	0,08967	0,26521	0,169053	7
Ex Ante	0,05717	0,03192	0,27916	0,057166	8
Ex Post	0,16744	0,10324	0,30828	0,167442	8
Ex Ante	0,05756	0,03358	0,29164	0,057563	9
Ex Post	0,16271	0,08931	0,27445	0,162712	9
Ex Ante	0,05975	0,03255	0,27240	0,059753	10
Ex Post	0,15788	0,09534	0,30196	0,157877	10
Ex Ante	0,06201	0,03245	0,26161	0,062013	11
Ex Post	0,14943	0,10539	0,35266	0,149430	11
Ex Ante	0,06446	0,03594	0,27878	0,064462	12
Ex Post	0,14853	0,09342	0,31447	0,148533	12
Ex Ante	0,06457	0,03778	0,29257	0,064572	13
Ex Post	0,14825	0,09694	0,32696	0,148247	13

Abbildung A.8: Ergebnisse für $\lambda = 0$

Name	μ	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05286	0,03023	0,28593	0,060422	1
Ex Post	0,18803	0,09105	0,24212	0,210794	1
Ex Ante	0,05304	0,02959	0,27900	0,060435	2
Ex Post	0,18824	0,08826	0,23445	0,210304	2
Ex Ante	0,05339	0,03186	0,29830	0,061359	3
Ex Post	0,18354	0,07828	0,21326	0,203106	3
Ex Ante	0,05388	0,03364	0,31217	0,062286	4
Ex Post	0,17815	0,07941	0,22288	0,197999	4
Ex Ante	0,05547	0,03302	0,29761	0,063723	5
Ex Post	0,17854	0,07554	0,21154	0,197426	5
Ex Ante	0,05593	0,03275	0,29273	0,064117	6
Ex Post	0,16905	0,08967	0,26521	0,191471	6
Ex Ante	0,05752	0,03179	0,27637	0,065472	7
Ex Post	0,16945	0,08596	0,25365	0,190939	7
Ex Ante	0,05756	0,03358	0,29164	0,065957	8
Ex Post	0,16271	0,08931	0,27445	0,185041	8
Ex Ante	0,05975	0,03255	0,27240	0,067892	9
Ex Post	0,15788	0,09534	0,30196	0,181713	9
Ex Ante	0,06201	0,03245	0,26161	0,070125	10
Ex Post	0,14943	0,10539	0,35266	0,175779	10
Ex Ante	0,06446	0,03594	0,27878	0,073448	11
Ex Post	0,14853	0,09342	0,31447	0,171888	11

Abbildung A.9: Ergebnisse für $\lambda = 0,25$

Name	μ	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05286	0,03023	0,28593	0,0610268	1
Ex Post	0,18803	0,09105	0,24212	0,2126151	1
Ex Ante	0,05304	0,02959	0,27900	0,0610270	2
Ex Post	0,18824	0,08826	0,23445	0,2120689	2
Ex Ante	0,05339	0,03186	0,29830	0,0619959	3
Ex Post	0,18354	0,07828	0,21326	0,2046716	3
Ex Ante	0,05388	0,03364	0,31217	0,0629583	4
Ex Post	0,17815	0,07941	0,22288	0,1995874	4
Ex Ante	0,05547	0,03302	0,29761	0,0643829	5
Ex Post	0,17854	0,07554	0,21154	0,1989370	5
Ex Ante	0,05593	0,03275	0,29273	0,0647720	6
Ex Post	0,16905	0,08967	0,26521	0,1932641	6
Ex Ante	0,05752	0,03179	0,27637	0,0661077	7
Ex Post	0,16945	0,08596	0,25365	0,1926582	7
Ex Ante	0,05756	0,03358	0,29164	0,0666280	8
Ex Post	0,16271	0,08931	0,27445	0,1868269	8
Ex Ante	0,05975	0,03255	0,27240	0,0685429	9
Ex Post	0,15788	0,09534	0,30196	0,1836195	9
Ex Ante	0,06201	0,03245	0,26161	0,0707734	10
Ex Post	0,14943	0,10539	0,35266	0,1778867	10
Ex Ante	0,06446	0,03594	0,27878	0,0741664	11
Ex Post	0,14853	0,09342	0,31447	0,1737562	11

Abbildung A.10: Ergebnisse für $\lambda = 0,27$

Name	ADTS	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05304	0,02959	0,27900	0,0613230	1
Ex Post	0,18824	0,08826	0,23445	0,2129516	1
Ex Ante	0,05339	0,03186	0,29830	0,0623145	2
Ex Post	0,18354	0,07828	0,21326	0,2054544	2
Ex Ante	0,05388	0,03364	0,31217	0,0632947	3
Ex Post	0,17815	0,07941	0,22288	0,2003816	3
Ex Ante	0,05547	0,03302	0,29761	0,0647130	4
Ex Post	0,17854	0,07554	0,21154	0,1996924	4
Ex Ante	0,05593	0,03275	0,29273	0,0650995	5
Ex Post	0,16905	0,08967	0,26521	0,1941608	5
Ex Ante	0,05752	0,03179	0,27637	0,0664256	6
Ex Post	0,16945	0,08596	0,25365	0,1935178	6
Ex Ante	0,05756	0,03358	0,29164	0,0669638	7
Ex Post	0,16271	0,08931	0,27445	0,1877200	7
Ex Ante	0,05975	0,03255	0,27240	0,0688684	8
Ex Post	0,15788	0,09534	0,30196	0,1845730	8
Ex Ante	0,06201	0,03245	0,26161	0,0710979	9
Ex Post	0,14943	0,10539	0,35266	0,1789407	9
Ex Ante	0,06446	0,03594	0,27878	0,0745258	10
Ex Post	0,14853	0,09342	0,31447	0,1746904	10

Abbildung A.11: Ergebnisse für $\lambda = 0,28$

Name	ADTS	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05304	0,02959	0,27900	0,075232	1
Ex Post	0,18824	0,08826	0,23445	0,254436	1
Ex Ante	0,05385	0,03036	0,28183	0,076619	2
Ex Post	0,18440	0,07553	0,20481	0,241052	2
Ex Ante	0,05434	0,03226	0,29684	0,078535	3
Ex Post	0,17902	0,07691	0,21480	0,236700	3
Ex Ante	0,05639	0,03117	0,27631	0,079768	4
Ex Post	0,16993	0,08750	0,25746	0,235550	4
Ex Ante	0,05547	0,03302	0,29761	0,080230	5
Ex Post	0,17854	0,07554	0,21154	0,235196	5
Ex Ante	0,05752	0,03179	0,27637	0,081369	6
Ex Post	0,16945	0,08596	0,25365	0,233920	6
Ex Ante	0,05807	0,03201	0,27560	0,082074	7
Ex Post	0,16365	0,08684	0,26532	0,228780	7
Ex Ante	0,06201	0,03245	0,26161	0,086347	8
Ex Post	0,14943	0,10539	0,35266	0,228476	8
Ex Ante	0,06240	0,03755	0,30088	0,090561	9
Ex Post	0,16551	0,08239	0,24889	0,227302	9
Ex Ante	0,06244	0,03794	0,30381	0,090890	10
Ex Post	0,15768	0,08398	0,26632	0,220665	10
Ex Ante	0,06446	0,03594	0,27878	0,091418	11
Ex Post	0,14853	0,09342	0,31447	0,218597	11

Abbildung A.12: Ergebnisse für $\lambda = 0,75$

Name	μ	GMAD	Gini	Funktionswert	Iteration
Ex Ante	0,05304	0,02959	0,27900	0,082631	1
Ex Post	0,18824	0,08826	0,23445	0,276502	1
Ex Ante	0,05385	0,03036	0,28183	0,084208	2
Ex Post	0,18440	0,07553	0,20481	0,259935	2
Ex Ante	0,05434	0,03226	0,29684	0,086600	3
Ex Post	0,17902	0,07691	0,21480	0,255927	3
Ex Ante	0,05547	0,03302	0,29761	0,088484	4
Ex Post	0,17854	0,07554	0,21154	0,254080	4
Ex Ante	0,05807	0,03201	0,27560	0,090076	5
Ex Post	0,16365	0,08684	0,26532	0,250490	5
Ex Ante	0,06240	0,03755	0,30088	0,099948	6
Ex Post	0,16551	0,08239	0,24889	0,247899	6
Ex Ante	0,06244	0,03794	0,30381	0,100375	7
Ex Post	0,15768	0,08398	0,26632	0,241661	7

Abbildung A.13: Ergebnisse für $\lambda = 1$

B. Matlab Code - Adaptive Epsilon Constraint Algorithmus

```
1 function [All_Info,avtime,times, open_shelter,node_to_shelter,
    fval_exante_all, fval_expost_all] = epsilon_algorithm_new(nodecoord,
    sheltercoord,maxshelter,nscen,epi,radii,p,lambda,delta,plotnum,
    plotlines, savefile, folder)
2
3 %Loest das Minimierungsproblem der zu oeffnenden Einrichtungen iterativ
    nach dem epsilon constraint Algorithmus
4
5 % Input:
6
7 % nodecoord..... Koordinaten der Knoten
8 % sheltercoord.... Koordinaten der Einrichtungen
9 % maxshelter.....Max zu oeffnende Einrichtungen
10 % epi..... Epizentren der Erdbeben
11 % radii..... Radien der betr. Gebiete
12 % p..... Wahrscheinlichkeiten fuer Erdbeben
13 % lambda..... Lambda Parameter fuer Gini Funktion (zw 0 und 0.5
    waehlen)
14 %           Wenn Lambda=0 wird nur Mittelwertspart geloest
15 % delta..... gibt an um wie viel strenger die NB fuer die ex post ZF
16 %           nach jeder Iteration gewaehlt werden soll
17 % nscen..... Anzahl der Szenarien
18 % plotnum..... logical, ob Knoten mit oder ohne Nummer geplottet werden
19 % plotlines..... logical, ob Linien geplottet werden sollen
20 % savefile..... logical, ob plots gespeichert werden sollen
21 % folder..... ordner, in dem die files gespeichert werden sollen
22 % Output:
23 % All_Info..... Infos zu jeder Iteration (ADTS GMAD GINI und
24 %           Funktionswerte
25 % avtime..... Mittlere Laufzeit der Iterationen
```

```

26 %times..... Laufzeiten der einzelnen Iterationen
27 %open_shelter..... Geoeffnete Einrichtungen (binaer)
28 %node_to_shelter..1 Wenn PK i zu KS j geht, 0 sonst
29 %fval_exante_all.. Alle ex ante ZF-Werte
30 %fval_expost_all.. Alle ex post ZF-Werte
31
32 %%%Vorbereitungen
33 exitflag = 1;
34 fval_expost=9999;
35 lowbound=0;
36 upbound=1;
37
38 n = size(nodecoord,1);
39 s = size(sheltercoord,1);
40
41 distmat = pdist2(sheltercoord,nodecoord);
42 %Betroffenen-Matrix erstellen
43 b = zeros(nscen,n); %b-matrix: row i...scenario i, column j...node j
44 for i = 1:nscen
45     idx = rangesearch(nodecoord,epi(i,:),radii(i));
46     idx = cell2mat(idx); %index of cells within respective area
47     b(i,idx) = 1;
48 end
49
50 %
51 % NUN ZUM MINIMIERUNGSPROBLEM
52 %
53 %hier kommen zuerst tau+, dann tau- dann y's dann x's
54 f1 = (lambda/n^2)*ones(1,2*n^2); %all tau+ and tau-
55
56 %Berechnung der y's (Mittelwerte)
57 d=distmat;
58 %nscen = anzahl der scenarien
59 AB = zeros(n*nscen,n*s);
60 for i=1:n
61     bd = b(:,i)*d(i,:);
62     AB(i*nscen-nscen+1:i*nscen,i*s-s+1:i*s) = bd;
63 end
64 pn = repmat(p,1,n);
65 f2 = (1/n)*pn*AB;
66
67 f3 = zeros(1,s); %x's werden in ZF 0 gesetzt.
68
69 %Minimierungsfunktion

```

```

70 f = [f1,f2,f3];
71
72 %GLEICHHEITS NB's
73 %in NB-Matrizen kommen zuerst alle tau+ und tau- (jeweils n^2 spalten)
74 %dann alle y's (n*s spalten)
75 %dann alle x's (s Spalten)
76
77 A = zeros(n^2,length(f));
78 for k = 1:n
79     for m = 1:n %m ist k'
80         A(n*(k-1)+m,n*(k-1)+m) = 1; %set tau+ to 1
81         A(n*(k-1)+m,n^2+n*(k-1)+m) = -1; %set tau- to -1
82         %if k = m then tau+ + tau- = 0 -> next
83         if k==m
84             continue
85         end
86         for j = 1:s
87             A(n*(k-1)+m,2*n^2+s*(m-1)+j) = +p*b(:,m)*distmat(m,j);
88             A(n*(k-1)+m,2*n^2+s*(k-1)+j) = -p*b(:,k)*distmat(k,j);
89         end
90     end
91 end
92
93 Aeq1 = A;
94 beq1 = zeros(n^2,1);
95
96 %sum j=1,...,s von(yij)=1 fuer alle i , d.h. es gibt nur einen naechsten
97 %geoeffneten shelter pro knoten
98 Aeq2= zeros(n,length(f));
99 for i = 1:n
100     Aeq2(i,i*s-s+1+2*n^2:i*s+2*n^2)=1;%+2*n^2...ueberspringe alle tau's
101 end
102 beq2= 1*ones(n,1);
103
104 Aeq=[Aeq1;Aeq2];
105 beq=[beq1;beq2];
106
107 %UNGLEICHHEITS NB's
108
109 Aineq2= [zeros(1,2*n^2+n*s),ones(1,s)];
110 bineq2 = maxshelter; %anzahl der maximal zu eroeffnenden shelter
111
112 %yij <= xj for all i,j (d.h. wenn xj=0 ist , also der j-te shelter
113 %geschlossen ist , dann kann nicht yij=1, also shelter j der naechste

```

```

114 %geoeffnete shelter von knoten i sein
115 %verw.  $y_{ij}-x_j \leq 0$ 
116  $y_{ij} = \text{eye}(n*s);$ 
117  $x_j = -\text{eye}(s);$ 
118  $x = \text{repmat}(x_j, n, 1);$ 
119  $\text{Aineq3} = [\text{zeros}(n*s, 2*n^2), y_{ij}, x];$ 
120  $\text{bineq3} = \text{zeros}(n*s, 1);$ 
121
122  $\text{Aineq}=[\text{Aineq2}; \text{Aineq3}];$ 
123  $\text{bineq}=[; \text{bineq2}; \text{bineq3}];$ 
124
125
126 %Bis hier war es das normale ExAnte Problem. Ab hier wird nun die
127 %Minimierungsfunktion fuer das ExPost Problem als NB hinzugefuegt.
128 %Muessen zus. Hilfsvariablen (rho) dazunehmen
129 %D.h. aber auch obige NB's muessen am Ende noch fuer alle rhos mit
130 %Nullen aufgefuellt werden!
131
132 %Reihenfolge der variablen: zuerst kommen alle  $\tau^+$  &  $\tau^-$ , dann y's, dann
    alle x's
133 %danach kommen dann alle  $\rho^+$  und  $\rho^-$ 
134 %erster teil - Mittelwertspart
135  $Ns = \text{sum}(b, 2);$  %Summe aller betroffenen pro Scenario (sum(b) weil wir
    annehmen dass entweder alle oder keiner betroffen ist(binaer))
136  $Ns1 = 1./Ns;$ 
137  $\text{Aineqexpost} = \text{zeros}(1, 2*n^2+n*s+s+2*n*n*nscen);$  %n*n*nscen...anzahl der
    spalten fuer  $\rho^+$  und  $\rho^-$ 
138 for k=1:n
139     for j=1:s
140          $\text{btmp} = \text{diag}(b(:, k));$ 
141          $\text{Aineqexpost}(1, 2*n^2+(k-1)*n+j) = \text{p}*\text{btmp}*Ns1*\text{distmat}(k, j);$  %trage
            ergebnis fuer  $y_{kj}$  ein (ueberspringe dazu alle taus)
142     end
143 end
144 %nun der zweite teil - Gini-berechnung
145  $Ns2 = 1./Ns.^2;$ 
146 for k=1:n
147     for m=1:n %m...k'
148         for scen=1:nscen
149             %Eintragen fuer  $\rho^+$ : ueberspringe  $2*n^2+n*s+s$  (i.e. alle taus
                ys und xs, dann beginnen die rhos)
150              $\text{Aineqexpost}(1, 2*n^2+n*s+s+n*nscen*(k-1)+nscen*(m-1)+scen) =$ 
                 $\text{lambda}*\text{p}(\text{scen})*Ns2(\text{scen})*b(\text{scen}, k)*b(\text{scen}, m) ;$ 

```

```

151         %Eintragen fuer rho-: ueberspringe  $2*n^2+n*s+s+n*n*nscen$  (i.e.
           alle taus ys und xs und alle rho+, dann beginnen die rhos-)
152         Aineqexpost(1,  $2*n^2+n*s+s+n*n*nscen+n*nscen*(k-1)+nscen*(m-1)+$ 
           scen) = lambda* p(scen)*Ns2(scen)*b(scen,k)*b(scen,m);
153     end
154 end
155 end
156
157 bineqexpost = fval_expost; %hier fval von vorigem schritt im epsilon
           constraint algorithmus einfuegen!!!
158
159 %jetzt noch die gleichheits und ungleichheitsbed fuer alle rho+ und rho-
160 Aeqexpost = zeros(n*n*nscen,  $2*n^2+n*s+s+2*n*n*nscen$ );
161 for k=1:n
162     for m=1:n %m...k'
163         for scen=1:nscen
164             %eine zeile pro k,m und scen
165             Aeqexpost(n*nscen*(k-1)+nscen*(m-1)+scen,  $2*n^2+n*s+s+n*nscen*(k$ 
               -1)+nscen*(m-1)+scen) = 1; %rho+_kmscen in k_m_scen-ter
               Spalte
166             Aeqexpost(n*nscen*(k-1)+nscen*(m-1)+scen,  $2*n^2+n*s+s+n*n*nscen+$ 
               n*nscen*(k-1)+nscen*(m-1)+scen) = -1;%rho-_kmscen in
               k_m_scen-ter Spalte
167             if k==m %wenn k=k' ist alles andere 0
168                 continue
169             end
170             for j=1:s %alle shelter
171                 Aeqexpost(n*nscen*(k-1)+nscen*(m-1)+scen,  $2*n^2+s*(k-1)+j$ ) =
                   -distmat(k,j);
172                 Aeqexpost(n*nscen*(k-1)+nscen*(m-1)+scen,  $2*n^2+s*(m-1)+j$ ) =
                   distmat(m,j);
173             end
174         end
175     end
176 end
177 beqexpost = zeros(n*n*nscen,1);
178 %Alle Indizes fuer die rhos wo k = k' ist -> diese Spalten werden
179 %entfernt da sie nicht benoetigt werden, weil die Werte dafuer immer 0 sind
180 idx_rm = 1: $2*n^2+n*s+s+2*n*n*nscen$ ;
181 rm1 = zeros(1,n*nscen);
182 rm2 = zeros(1,n*nscen);
183 for k=1:n
184     tmp = idx_rm( $2*n^2+n*s+s+ n*nscen*(k-1)+nscen*(k-1) + 1: 2*n^2+n*s+s+$ 
               n*nscen*(k-1)+nscen*(k-1) + nscen );

```

```

185     rm1(1,nscen*(k-1)+1:nscen*(k-1)+nscen) = tmp;
186     tmp = idx_rm(2*n^2+n*s+s+ n*n*nscen+ n*nscen*(k-1)+nscen*(k-1) + 1: 2*n
        ^2+n*s+s+ n*n*nscen+ n*nscen*(k-1)+nscen*(k-1) + nscen );
187     rm2(1,nscen*(k-1)+1:nscen*(k-1)+nscen) = tmp;
188 end
189 cols2remove1 = [rm1 , rm2];
190 %cols2remove = cols2remove1;
191 %Alle Indizes fuer die taus wo k = k' ist -> diese Spalten werden
192 %entfernt da sie nicht benoetigt werden, weil die Werte dafuer immer 0 sind
193 rm1 = zeros(1,n);
194 rm2 = zeros(1,n);
195 for k=1:n
196     rm1(k) = n*(k-1)+k;
197     rm2(k) = n^2+n*(k-1)+k;
198 end
199 cols2remove2 = [rm1 , rm2];
200 cols2remove = [cols2remove1 , cols2remove2];
201
202 %Num alle NB's entsprechend mit Nullen auffuellen und zusammensetzten
203
204 %0 fuer alle rho+ und rho- in f da diese nur Hilfsvariablen sind
205 f = [f, zeros(1,2*n*n*nscen)];
206 f(cols2remove) = [];
207 f1(cols2remove2) = [];
208 Aineqsize = size(Aineq,1);
209 Aineq=[Aineq, zeros(Aineqsize,2*n*n*nscen)];
210 Aineq=[Aineq; Aineqexpst];
211 %Entferne alle Spalten wo k=k' fuer rho- or rho+
212 %Da die werte dafuer immer 0 sind kann man das weglassen!
213 %Und Entferne alle Spalten wo k=k' ist fuer tau- und tau+ aus demselben
214 %Grund
215 Aineq(:,cols2remove) = [];
216 Aineqexpst(:,cols2remove) = [];
217 Aeqsize = size(Aeq,1);
218 Aeq = [Aeq, zeros(Aeqsize,2*n*n*nscen)];
219 Aeq = [Aeq; Aeqexpst];
220 Aeq(:,cols2remove) = [];
221 len =length(bineq);
222 bineq = [bineq; bineqexpst];
223 beq = [beq; beqexpst];
224
225 %spezifiziere Variablen-Typen (B... binary , C... continuous)
226 %alle tau und rho continuous, alle y und x binaer
227 binary = repmat('B',1,n*s+s);

```

```

228 %continuous = repmat('C',1,2*n^2);%-2*n);
229 continuous = repmat('C',1,2*n^2-2*n);
230 %continuous1 = repmat('C',1,2*n*n*nscen);%
231 continuous1 = repmat('C',1,2*n*n*nscen-(n*nscen*2));
232 ctype = [continuous,binary,continuous1];
233 %size(ctype);
234
235 %set lower and upper bounds
236 lb = repmat(lowbound,1,length(f));
237 ub = repmat(upbound,1,length(f));
238 sostype=[];
239 sosind=[] ;
240 soswt=[];
241
242 %Minimierungsfunktion
243 open_shelter = [1:n]';
244 node_to_shelter = [1:n]';
245 fval_exante_all = [];
246 fval_expost_all = [];
247
248 %Initialisiere Zaehler und leere Listen fuer Ergebnisse;
249 iter=0;
250 times =[];
251 All_Info=[];
252 while exitflag==1 | exitflag==5
253     iter= iter+1;
254
255     %update NB fuer ex post funktionswert fval_expost
256     bineq(len+1)= fval_expost;
257     tic;
258     [minres, fval, exitflag, output] = cplexmilp(f,Aineq,bineq, Aeq, beq,
        sostype,sosind,soswt,lb,ub,ctype);
259     time=toc;
260     disp(output)
261     if exitflag==1 | exitflag==5
262         times(iter) = time;
263         avtime = sum(times)/iter;
264         res=zeros(s,n+1);
265         for i=1:n+1
266             res(:,i)=minres(2*n^2-2*n+i*s-s+1:2*n^2-2*n+i*s);
267         end
268         shelter = res(:,n+1); %1 if i'th shelter is open
269         nodetosshelter = res(:,1:n)';%1 if node i goes to shelter j (node in
            column, shelter in row)

```

```

270
271     fvalexpost = Aineqexpost*minres;
272
273     if plotlines == true
274         [row,col]=find(round(nodetoshelter));
275         x=[nodecoord(row,1),sheltercoord(col,1)];
276         y=[nodecoord(row,2),sheltercoord(col,2)];
277         figure();
278         plot(x',y')
279         hold on
280     end
281     x = nodecoord(:,1);
282     y = nodecoord(:,2);
283     scatter(x,y,'black');
284     xs = sheltercoord(:,1);
285     ys = sheltercoord(:,2);
286     scatter(xs,ys,40,'b');
287     scatter(xs(shelter==1),ys(shelter==1),20,'r','filled');%open
288     shelters are plotted filled out and red
289     xlim([min(epi(:,1)-radii(:))-0.1 max(epi(:,1)+radii(:))+0.1]);ylim
290     ([min(epi(:,2)-radii(:))-0.1 max(epi(:,2)+radii(:))+0.1])
291     if plotnum == true
292         a = [1:n]'; astring = num2str(a); c = cellstr(astring);
293         dx = 0.01; dy = 0.01; % displacement so the text does not
294         overlay the data points
295         text(x+dx, y+dy, c);
296     end
297
298     for i=1:nscen
299         circle(epi(i,1),epi(i,2),radii(i));
300     end
301
302     if savefile == true
303         filename = sprintf('n%d_scen%d_max%d_iter%d_lambda%d.jpg',n,
304             nscen,maxshelter,iter,lambda);
305         saveto = fullfile(folder,filename);
306         saveas(gcf,saveto);
307     end
308
309     hold off
310
311     %berechnung der ADTS-ex ante (average distance each individuum has
312     to travel to
313     %reach assigned shelter mit ex ante mittelwertsformel)

```

```

309     pb = p*b;
310     nd = nodetoshelter.*distmat;
311     ADTSante = mean(pb*nd);
312     %alternative berechnung:
313     %     ADTSante = f2*minres(2*n^2-2*n+1:2*n^2-2*n+n*s);
314
315     %GMADante (Ginis mean absolute difference);
316     N = n;
317     yd = nodetoshelter.*d;
318     res=0;
319     for k = 1:n
320         for m = 1:n %entspricht k'
321             pbyd = zeros(1,nscen);
322             for scen = 1:nscen
323                 pbyd(scen) = p(scen)*(b(scen,k)*sum(yd(k,:))-b(scen,m)*
324                     sum(yd(m,:)));
325             end
326             res = res + abs(sum(pbyd));
327         end
328     GMADante = res/N^2;
329     %Alternative berechnung
330
331     %     GMADante=f1*minres(1:2*n^2-2*n)/lambda;
332
333     %ADTSpost:
334     Ns = sum(b,2); %summe der betroffenen pro szenario
335     pNs = p./Ns';
336     pNsb = pNs*b;
337     ADTSpost = sum(pNsb*nd);
338     %ADTSpost alternative berechnung:
339     %     ADTSpost=Aineqexpost(2*n^2-2*n+1:2*n^2-2*n+n*s)*minres(2*n^2-2*n
340     +1:2*n^2-2*n+n*s);
341
342     %GMADpost
343     yd = nodetoshelter.*d;
344     res= zeros(1,nscen);
345     for scen=1:nscen
346         pN = p(scen)/sum(b(scen,:))^2;
347         byd=zeros(n,n);
348         for k=1:n
349             for m=1:n
350                 byd(k,m) = b(scen,k)*b(scen,m)*abs(sum(yd(k,:))-sum(yd(
351                     m,:)));

```

```

350         end
351     end
352     res(scen)=pN*sum(sum(byd));
353 end
354 GMADpost = sum(res);
355 %Alternative Berechnung:
356 % GMADpost=Aineqexpost(2*n^2-2*n+n*s+n+1:length(minres))*minres(2*n
^2-2*n+n*s+n+1:length(minres))/lambda;
357
358 %Alle Masse zusammengefasst:
359 Name = [ 'Ex Ante'; 'Ex Post' ];
360 ADTS = [ ADTSante; ADTSpost ];
361 GMAD = [ GMADante; GMADpost ];
362 Gini = [ GMADante/(2*ADTSante); GMADpost/(2*ADTSpost) ];
363 Funktionswert = [ fval; fvalexport ];
364 Iteration = [ iter; iter ];
365 Info = table(Name, ADTS, GMAD, Gini, Funktionswert, Iteration);
366 All_Info = [ All_Info; Info ];
367 disp(Info)
368
369 %Funktionswert neu = Funktionswert alt - delta fuer naechste
Iteration
370 fval_expost = fvalexport-delta;
371 open_shelter = [ open_shelter, shelter ];
372 node_to_shelter = [ node_to_shelter, nodetoshelter ];
373 fval_exante_all = [ fval_exante_all, fval ];
374 fval_expost_all = [ fval_expost_all, fvalexport ];
375
376 else
377     disp('no feasible solution found')
378 end
379 end
380
381 end

```

Literatur

- [1] Elchanan Ben-Porath, Itzhak Gilboa und David Schmeidler. “On the measurement of inequality under uncertainty”. In: *journal of economic theory* 75.1 (1997), S. 194–204.
- [2] *cplexmilp Function Reference*. https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.cplex.help/refmatlabcp/plex/html/cplexmilp-m.html. Zugegriffen: 20.06.2018.
- [3] Tammy Drezner, Zvi Drezner und Jeffery Guyse. “Equitable service by a facility: Minimizing the Gini coefficient”. In: *Computers & Operations Research* 36.12 (2009), S. 3240–3246.
- [4] M Gay. “IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual”. In: *International Business Machines Corporation* 12 (2012).
- [5] Corrado Gini. “Variabilità e mutabilità”. In: *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi* (1912).
- [6] Walter J Gutjahr und Pamela C Nolz. “Multicriteria optimization in humanitarian aid”. In: *European Journal of Operational Research* 252.2 (2016), S. 351–366.
- [7] Özlem Karsu und Alec Morton. “Inequity averse optimization in operational research”. In: *European journal of operational research* 245.2 (2015), S. 343–359.
- [8] Marco Laumanns, Lothar Thiele und Eckart Zitzler. “An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method”. In: *European Journal of Operational Research* 169.3 (2006), S. 932–942.
- [9] Michael T Marsh und David A Schilling. “Equity measurement in facility location analysis: A review and framework”. In: *European Journal of Operational Research* 74.1 (1994), S. 1–17.

- [10] MATLAB. *version 9.3.0 (R2017b)*. Natick, Massachusetts: The MathWorks Inc., 2017.
- [11] Mahdi Mostajabdaveh, Walter J. Gutjahr und Sibel F. Salman. “Inequity-averse shelter location for disaster preparedness”. accepted for publication in: IISE Transaction. 2018.
- [12] Hans Murawski und Wilhelm Meyer. *Geologisches Wörterbuch, 12. überarbeitete und erweiterte Auflage*. Springer Spektrum, 2010.
- [13] John Rawls. “A theory of justice (cambridge”. In: *Mass.: Harvard University* (1971).
- [14] Shlomo Yitzhaki und Edna Schechtman. *The Gini methodology: A primer on a statistical methodology*. Bd. 272. Springer Science & Business Media, 2012.