



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

**Micha Pikora**

**Lösungsstrategien von Fluchtkantenreduzierungen -  
Implementierung eines dezentralen Sukey-Netzes**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Micha Pikora

**Lösungsstrategien von Fluchtkantenreduzierungen -  
Implementierung eines dezentralen Sukey-Netzes**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ing. Julia Padberg  
Zweitgutachter: Prof. Dr. rer. nat. Christoph Klauck

Eingereicht am: 16. Dezember 2011

**Micha Pikora**

**Thema der Arbeit**

Lösungsstrategien von Fluchtkantenreduzierungen - Implementierung eines dezentralen Sukey-Netzes

**Stichworte**

Sukey, Sukey-Netz, Sukey-Algorithmus, Kessel, Einkesselung, Kanten, Fluchtkanten, Knoten, Fluchtwege

**Zusammenfassung**

Zur Lösung von Fluchtkantenreduzierungen - in der realen Welt spricht man hier vom Suchen und Finden von Auswegen bei einer möglichen Fluchtwegreduzierung, z. B. bei Einkessel-Situationen - sind verschiedene Strategien bekannt. Die vorgelegte Arbeit konzentriert sich auf eine dieser Strategien, nämlich auf die Untersuchung und Entwicklung einer bestimmten aussichtsreich scheinenden Methode, die Entwicklung des Algorithmus eines dezentralen Sukey-Netzes und seine Implementierung in einer Simulation. Folgende Inhalte werden dabei bearbeitet: Beschreibung eines Sukey-Netzes, Voraussetzungen eines dezentralen Sukey-Netzes, Entwicklung eines Sukey-Algorithmus, Simulation des Algorithmus, Evaluation.

**Micha Pikora**

**Title of the paper**

Solving strategies to identify reduction of arcs in an urban evacuation area - implementation of a decentralized sukey network

**Keywords**

sukey, sukey-net, algorithm, kettle, kettling, arc, edge, escape, node

**Abstract**

There are several strategies to identify the reduction of arcs (e.g. kettling) in an urban evacuation area. This document describes how to build a sukey-net with a decentral structure in developing the algorithm to warn for cetteling by identify the reduction of leaving arcs for escape. The steps are: description and assumptions of the sukey-net, formal description and simulation of the algorithm, evaluation.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziel der Arbeit . . . . .	2
1.2	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Voraussetzungen eines Sukey-Netzes</b>	<b>4</b>
2.1	Eigenschaften des Sukey-Netzes . . . . .	4
2.2	Betriebsmodi . . . . .	5
2.2.1	Der Infrastruktur-Modus . . . . .	5
2.2.2	Der Ad-Hoc-Modus . . . . .	6
2.2.3	Der Mischbetrieb . . . . .	7
2.2.4	Gegenüberstellung - Betriebsmodi . . . . .	8
2.3	Organisation der beteiligten Personen . . . . .	8
2.3.1	Informationen aus dem Stadtplan . . . . .	9
2.3.2	Informationen über Hindernisse . . . . .	9
2.3.3	Anzahl der Personen am Standort . . . . .	10
2.3.4	Mobilität der Personen am Standort . . . . .	10
2.4	Zusammenfassung - Voraussetzungen eines Sukey-Netzes . . . . .	11
2.4.1	Parameter des Sukey-Algorithmus . . . . .	11
<b>3</b>	<b>Der Sukey-Algorithmus</b>	<b>13</b>
3.1	Darstellung des A*-Algorithmus und seiner Varianten . . . . .	13
3.1.1	Algorithmen für den Sukey-Algorithmus . . . . .	14
3.2	Adaption des Sukey-Algorithmus . . . . .	16
3.3	Der Fluchtkantengraph . . . . .	18
3.4	Hindernisregeln . . . . .	21
3.4.1	Die direkte Hindernisregel . . . . .	22
3.4.2	Die indirekte Hindernisregel . . . . .	22
3.5	Die Fluchtweggüte . . . . .	24
3.6	Die heuristische Fluchtweggüte . . . . .	29
3.7	Herleitung des Sukey-Algorithmus . . . . .	30
3.8	Status . . . . .	31
<b>4</b>	<b>Implementierung des Sukey-Algorithmus</b>	<b>33</b>
4.1	Aufbau des Simulators . . . . .	33
4.2	Die Grammatik des Fluchtkantengraphen . . . . .	34

4.3	Die Klassen des Fluchtkantengraphen . . . . .	36
4.3.1	Die Position . . . . .	38
4.3.2	Die Knoten . . . . .	38
4.3.3	Die Fluchtkanten . . . . .	39
4.3.4	Die Hindernisse . . . . .	40
4.3.5	Der Standort . . . . .	41
4.4	Ausblick - Kurvenführung . . . . .	42
4.4.1	Beschreibung von Kurven . . . . .	42
4.4.2	Auswirkung von Kurven auf die maximale Länge . . . . .	43
<b>5</b>	<b>Evaluation</b>	<b>45</b>
5.1	Die Interviews . . . . .	45
5.1.1	Vorstellung der Ergebnisse . . . . .	46
5.1.2	Zusammenfassung - Interviews . . . . .	56
5.2	Parametereinstellungen . . . . .	57
5.2.1	Die Anzahl der Personen am Standort . . . . .	57
5.2.2	Die Mobilität der Personen am Standort . . . . .	57
5.2.3	Die Performanz . . . . .	58
<b>6</b>	<b>Zusammenfassung</b>	<b>59</b>
6.1	Aufbau der Arbeit . . . . .	59
6.2	Ergebnisse . . . . .	60
6.3	Ausblicke und Visionen . . . . .	60

# Tabellenverzeichnis

2.1	Gegenüberstellung von Eigenschaften und Betriebsmodi . . . . .	8
3.1	Übersicht - Dateneingabe des Fluchtkantengraphen . . . . .	18
3.2	Beispiele für Einflüsse von $dhr(a_{ij})$ und $ihr(a_{ij})$ auf die Güte einer Flucht- kante und ihre Bedeutung . . . . .	22
3.3	Die Funktionen des A*-Algorithmus nach [Hart u. a. (1968)] . . . . .	30
3.4	Die Funktionen des Sukey-Algorithmus . . . . .	31
3.5	Vergleich von A*- und Sukey-Algorithmus. Alle weiteren Regeln des A* bleiben für den Sukey-Algorithmus unberührt . . . . .	31
4.1	Ableitungsschritte der Grammatik für die Beschreibung des Fluchtkantengraphen	35
4.2	Beispiele für die Beschreibung eines Fluchtkantengraphen mithilfe der vorge- stellten Grammatik . . . . .	36
4.3	Das Modul Fluchtkantengraph . . . . .	37
4.4	Die abstrakte Klasse APosition . . . . .	38
4.5	Die Klasse Node . . . . .	39
4.6	Die Klasse Arc . . . . .	40
4.7	Die Klasse Obstacle . . . . .	40
4.8	Die Klasse Device . . . . .	41
5.1	Interviews - Einschätzung der Variable <i>STATUS</i> . . . . .	47
5.2	Interview - Fluchtkantengraph 1 . . . . .	48
5.3	Interview - Fluchtkantengraph 2 . . . . .	49
5.4	Interview - Fluchtkantengraph 3 . . . . .	50
5.5	Interview - Fluchtkantengraph 4 . . . . .	51
5.6	Interview - Fluchtkantengraph 5 . . . . .	52
5.7	Interview - Fluchtkantengraph 6 . . . . .	53
5.8	Interview - Fluchtkantengraph 7 . . . . .	54
5.9	Interview - Fluchtkantengraph 8 . . . . .	55
5.10	Interview - Fluchtkantengraph 9 . . . . .	56

# Abbildungsverzeichnis

3.1	Die Umkehrfunktion $k(x) = -\ln x$ . . . . .	17
3.2	Die Umkehrfunktion $w(x) = e^{-x}$ . . . . .	17
3.3	Diagonale des Fluchtkantengraphen <i>maxlen</i> . . . . .	20
3.4	Wirkungsweise der indirekten Hindernisregel (ihr) . . . . .	23
3.5	Berücksichtigung der Bewegungsrichtung durch die indirekte Hindernisregel (ihr) . . . . .	24
3.6	Auswirkung eines Engpasses auf die Sicherheit des Fluchtweges . . . . .	25
3.7	Bevorzugung eines Fluchtweges mit einer zusätzlichen Ausweichmöglichkeit . . . . .	26
3.8	Einfluss der Länge auf $FW_n$ bei hoher Mobilität von Personen . . . . .	27
3.9	Einfluss der Länge auf $FW_n$ bei geringer Mobilität von Personen . . . . .	28
3.10	Fluchtkantengraph mit 12 unterschiedlichen Fluchtwegen, die gleiche Fluchtkanten nutzen . . . . .	28
4.1	Zusammenspiel der Komponenten des Simulators . . . . .	34
4.2	Die Komponente Fluchtkantengraph. . . . .	38
4.3	Ungewünschte Aufwertung des Fluchtweges mit einem zusätzlichen Knoten mit einem Knotengrad gleich 2 . . . . .	42
4.4	Berechnung von $qual(FW_C)$ um eine Kurve . . . . .	43
4.5	Darstellung des Falls $len(a_{curve}) > 1$ . . . . .	44
5.1	Interview - Legende des Fluchtkantengraphen für die Interview-Partner . . . . .	46

# 1 Einführung

Diese Arbeit entsteht unter dem Eindruck der sich im Jahre 2011 häufenden Kundgebungen für bürgerliche Freiheiten in der westlichen Welt und den Volksaufständen für parlamentarische Demokratien in der arabischen Welt. Im Vordergrund steht der Gedanke, mit modernen Kommunikationsmitteln, die jedermann zur Verfügung stehen, Gefahren gemeinsam zu erfassen, zu teilen und für jeden Einzelnen am Standort auswert- und interpretierbar zu machen. Ein solches ziviles Netzwerk wird in dieser Arbeit *Sukey-Netz*<sup>1</sup> genannt.

Eine Lösungsstrategie bei Fluchtkantenreduzierungen ist zum Beispiel am 28. Januar 2011 mit einem zentral aufgebauten Sukey-Netz das erste Mal erfolgreich eingesetzt worden<sup>2</sup>. Das in London entwickelte System beschränkt sich zur Zeit (Dezember 2011) ausschließlich auf den Einsatz auf Demonstrationen gegen Polizei-Kessel und führt die Einschätzung der Situationen an den Standorten der Teilnehmer manuell durch freiwillige Helfer in einer Zentrale durch.

Andere Beispiele für Lösungsstrategien bei Fluchtkantenreduzierungen sind Navigationssysteme, die auch auf Engpässe, Staus usw. reagieren müssen und auf über Sprechfunk laufende Kommunikationswege, wie sie bei der Bekämpfung von großen Bränden üblich sind.

In der vorgelegten Arbeit wird *Sukey* als der Oberbegriff für Software verstanden, die es Gefährdeten ermöglicht mit Mobiltelefonen ihre Positionen in Echtzeit in ihre Geräte einzugeben, eine Reduzierung möglicher Fluchtkanten festzustellen und Informationen über mögliche Auswege über die Geräte zu erhalten. Ziel eines solchen Netzes ist die schnelle Information der Teilnehmer über die Zustände sich verändernder Fluchtkanten und die Ausgabe einer Fluchtempfehlung bei einer zu starken Reduzierung der Fluchtkanten.

---

<sup>1</sup>Die Namensgebung *Sukey* stammt von den Londoner Studenten Sam Gaus und Sam Carlisle im Dezember 2010 mit Blick auf das britische Kinderlied über zwei Schwestern: »Polly, put the kettle on, (...) And let's drink tea. Sukey, take it off again. (...) They've all gone away.« Dies nimmt implizit Bezug auf Kesselsituationen (*kettle*) durch Polly und erwähnt Sukey die diese Situation beendet, weil alle schon gegangen sind. (vgl. <http://sukey.org/metminutes>)

<sup>2</sup>vgl. [Wikipedia \(2011b\)](#)



### 1.1 Ziel der Arbeit

In dieser Arbeit wird der Fokus auf den *Sukey-Algorithmus* als kombinierte Lösungsstrategie für die Berechnung geeigneter Fluchtwege in einer Kesselsituation im zivilen Umfeld gelegt. Dabei werden die möglichen Fluchtwege als Kanten in einem Graphen aufgefasst und eine Reduktion der Möglichkeiten verfolgt. Die Reduktion der Fluchtkanten soll die möglichst beste Wahl des Fluchtweges liefern und erfordert dadurch einen qualitativen Algorithmus.

Die Motivation dieser Arbeit ist die Erkenntnis, dass der Schwachpunkt eines zentral aufgebauten Sukey-Netzes gerade diese Zentrale ist. Um die Zentrale zu umgehen, muss ein *dezentrales Sukey-Netz* entwickelt werden. Durch den Wegfall der Zentrale muss jeder Teilnehmer an einem Sukey-Netz selber eine Bewertung seiner Situation vornehmen.

Ziel dieser Arbeit ist es Lösungsstrategien zur Reduktion von Fluchtkanten zu einem Algorithmus - dem Sukey-Algorithmus - zusammenzufassen, der der intuitiven Wahl eines Fluchtweges bei bekannten Randbedingungen entspricht. Die vorgelegte Arbeit rückt also den Sukey-Algorithmus in den Mittelpunkt, der die eigenständige Einschätzung an Stelle des Teilnehmers übernimmt. Die Aufgabe des Sukey-Algorithmus ist es, Fluchtkantenreduzierungen automatisch zu erkennen, zu bewerten und vor Einkesselungen zu warnen. Der Sukey-Algorithmus soll dabei menschliches Verhalten und menschliche Intuition nachstellen.

Die Evaluation des Sukey-Algorithmus beruht auf dem Vergleich mit der Auswahl von Fluchtkanten durch die Teilnehmer eines Interviews. Dieser Vergleich wird durch die Implementierung eines Werkzeugs zur Simulation von Sukey-Netzen ermöglicht.

### 1.2 Aufbau der Arbeit

Die vorliegende Arbeit verfolgt diese Aufgabenstellung in folgenden Schritten: Zunächst werden in [Kapitel 2] die Voraussetzungen des dezentralen Sukey-Netzes vorgestellt. Dabei werden einerseits die technischen Randbedingungen, wie die Betriebsmodi, untersucht und andererseits die Faktoren, die die Bewertung einer Fluchtkante ermöglichen. Dadurch ergeben sich wesentliche Parameter, die in den Algorithmus einfließen.

Im Folgenden wird in [Kapitel 3] der Sukey-Algorithmus beschrieben, der auf einer Adaption des A\*-Algorithmus beruht. Wie gezeigt werden wird, arbeitet der Sukey-Algorithmus gewinnorientiert und verwendet relative Güten von Fluchtkanten. Der A\*-Algorithmus hingegen arbeitet kostenorientiert und verwendet absolute Längen. Die Adaption ist mithilfe einer Umkehrfunktion möglich.

In [Kapitel 4] wird die Implementierung des Sukey-Algorithmus als Simulationswerkzeug vorgestellt.

Schließlich werden in [Kapitel 5] die Ergebnisse mittels Online-Interviews an den Gegebenheiten der Realität verglichen.

## 2 Voraussetzungen eines Sukey-Netzes

In diesem Kapitel wird das Sukey-Netz beschrieben. Dieses Kapitel wird zeigen, dass ein dezentraler Aufbau des Sukey-Netzes die meisten Vorteile bietet. Zunächst werden die wichtigen Eigenschaften eines Sukey-Netzes vorgestellt und erläutert. Anschließend werden die Modi für den Betrieb eines Sukey-Netzes vorgestellt und anhand der Eigenschaften in einer Gegenüberstellung verglichen. Dabei wird gezeigt, dass die Voraussetzung für ein dezentrales Sukey-Netz ein Algorithmus zum Erkennen von Fluchtkantenreduzierungen (möglicher Einkesselungen) ist. Ein solcher Algorithmus wird in der vorgelegten Arbeit *Sukey-Algorithmus* genannt.

### 2.1 Eigenschaften des Sukey-Netzes

Das Sukey-Netz ist ein Netzwerk mit dem Ziel, Informationen über mögliche Gefahren zu verbreiten und das Erkennen von Einkesselsituationen zu ermöglichen. Um dieses Ziel zu erreichen, müssen sich die Teilnehmer in einem Sukey-Netz organisieren. Dabei braucht nicht jeder Teilnehmer Teil des Sukey-Netzes zu sein. Eine kleine Gruppe Informierter reicht aus, an denen sich Nicht-Informierte orientieren können (Mundpropaganda).

Wegen des Einsatzes in Gefahrensituationen treten folgende gewünschte Eigenschaften eines Sukey-Netzes nach Meinung des Autors besonders in den Vordergrund:

1. Unabhängigkeit von ziviler Infrastruktur

Eine gegebene örtliche Infrastruktur (z.B. das Mobilfunknetz) ist nicht für eine unbegrenzte Anzahl an Nutzern geeignet. Kommt es zu einem Ausfall durch Überlastung, fällt das Sukey-Netz aus. Dadurch werden unbeteiligte Dritte (Anwohner oder Passanten) gefährdet, die auf ein intaktes Mobilfunknetz angewiesen sind (z.B. Herzpatienten).

2. Unabhängigkeit von einer Operationszentrale

In einer Operationszentrale laufen die Informationen zusammen, werden verarbeitet und die Ergebnisse an die Teilnehmer geschickt. Fällt diese Zentrale aus, funktioniert das Sukey-Netz nicht mehr. Gerade in Gefahrengebieten sind die Gründe für einen solchen Ausfall vielfältiger Natur.

### 3. Geringer Energieverbrauch der Mobiltelefone

Der Blick auf die Performanz ist begründet, doch spielt in naher Zukunft der Energieverbrauch von Mobiltelefonen eine immer geringere Rolle. Bei voranschreitender Technik wird die Leistung von Mobiltelefonen immer weiter ansteigen - bei gleichem oder gering erhöhtem Energieverbrauch. Die Entwicklung immer stärkerer Akkus mit längeren Laufzeiten ist ebenfalls abzusehen. Aus Umweltschutzgründen ist ein geringer Energieverbrauch dennoch wichtig.

### 4. Problemlose Kompensation des Wegfalls eines Teilnehmers

Der Wegfall eines Teilnehmers ist im besten Fall durch Funkschatten gegeben. Ist ein Sukey-Netz unmittelbar abhängig von einzelnen Teilnehmern, müssen Verfahren zur Verminderung dieser Abhängigkeit getroffen werden, sei es durch Redundanzen oder durch eine aufwändigere Verwaltung.

## 2.2 Betriebsmodi

Wie jedes drahtlose Netzwerk kann ein Sukey-Netz im *Infrastruktur-Modus*, im *Ad-Hoc-Modus* oder im *Mischbetrieb* betrieben werden. Vor- und Nachteile der Modi beziehen sich auf die in [Abschnitt 2.1] vorgestellten, wichtigen Eigenschaften eines Sukey-Netzes.

### 2.2.1 Der Infrastruktur-Modus

Die Kommunikation verläuft nach [Wikipedia (2011c)] über Hotspots oder Access-Points eines bestehenden Funknetzes, wobei die Geräte ausschließlich mit diesen Punkten in Verbindung stehen. Der Infrastruktur-Modus wird von jedem Mobiltelefon unterstützt, da diese auf die Kommunikation in Funkzellen ausgelegt sind. Es gibt eine Vielzahl von Standard-Sätzen die von unterschiedlichen Organisationen und Instituten entwickelt und verbreitet werden. Der Betrieb eines zentralen Sukey-Netzes ist beispielsweise nach dem GSM-Standard [ETSI (2011a)] oder nach dem UMTS-Standard [ETSI (2011b)] vorstellbar.

Die im [Abschnitt 2.1] vorgestellten Eigenschaften werden für den Betrieb eines Sukey-Netzes im Infrastruktur-Modus wie folgt in Vor- und Nachteile unterteilt:

Vorteile des Infrastruktur-Modus:

- Vergleichsweise geringer Energieverbrauch der Mobiltelefone
- Problemlose Kompensation des Wegfalls eines Teilnehmers

Nachteile des Infrastruktur-Modus:

- Die Abhängigkeit von ziviler Infrastruktur
- Die Abhängigkeit von einer Operationszentrale

Eine zentrale Implementierung eines Sukey-Netzes im Infrastruktur-Modus mit der Nutzung von GSM und UMTS wurde bereits im Dezember 2010 in London entwickelt und wird im Jahr 2011 nach [Sukey.org (2011a)] noch immer weiterentwickelt.

### 2.2.2 Der Ad-Hoc-Modus

Die Kommunikation verläuft über ein oder mehrere mitgeführte Geräte, wobei Informationen nach [Wikipedia (2011d)] nur über *mitgebrachte* Infrastruktur per *Peer-To-Peer* und über sogenannte *Hopps*<sup>1</sup> verbreitet werden. Für den Betrieb eines dezentralen Sukey-Netzes im Ad-Hoc-Modus<sup>2</sup> mit WiFi<sup>3</sup>, bauen die Teilnehmer ein MANET<sup>4</sup> auf und senden sich die Informationen bezüglich der Gefahren gegenseitig zu. Jeder Teilnehmer zieht dann seine eigenen Schlüsse aus den Daten.

Die im [Abschnitt 2.1] vorgestellten Eigenschaften werden für den Betrieb eines Sukey-Netzes im Ad-Hoc-Modus wie folgt in Vor- und Nachteile unterteilt:

Vorteile des Ad-Hoc-Modus:

- Unabhängigkeit von ziviler Infrastruktur
- Unabhängigkeit von einer Operationszentrale
- Problemlose Kompensation des Wegfalls eines Teilnehmers

Nachteil des Ad-Hoc-Modus:

- Vergleichsweise hoher Energieverbrauch der Mobiltelefone

Problem des Ad-Hoc-Modus:

- Die eigenständige Beurteilung von Fluchtkantenreduzierungen

---

<sup>1</sup>»Hopps« ist die Datenvermittlung (Routing) über einzelne Teilnehmer als Vermittler zwischen Start und Ziel. Ein Hopp ist dabei ein Schritt zwischen den Vermittlern.

<sup>2</sup>Von der Entwicklung eines dezentralen Sukey-Netzes im Ad-Hoc-Modus mit der Nutzung von WiFi ist dem Autor im Jahr 2011 noch nichts bekannt.

<sup>3</sup>Der Wireless-Standard [IEEE Standard for Information Technology (2011)] WiFi, wurde nach [Jin-Shyan Lee und Shen (2007)] für *Peer-To-Peer-Netze als Erweiterung oder Ersatz von Kabelnetzwerken entwickelt*. Das Protokoll wird vom WiFi-Firmen-Konsortium [Wi-Fi Alliance, 2011 (2011) und Wikipedia (2011a)] eingehalten. Das Institut für Ingenieure der Elektrik und Elektronik (IEEE) gibt die Standards für die physikalische- und die MAC-Schicht vor, welche von Firmenkonsortien um die darauf basierenden Protokoll-Standards erweitert werden. Dank der weltweiten Einigungen von Firmen ist dabei eine Kommunikation über Modell-Grenzen hinaus möglich.

<sup>4</sup>MANET bedeutet Multiple-Ad-Hoc-Network, ist also eine Abkürzung für ein Netzwerk im Ad-Hoc-Modus mit Hopps.

Der Nachteil *hoher Energieverbrauch der Mobiltelefone* entsteht durch das im Ad-Hoc-Modus notwendige Routing. Das Problem der *eigenständigen Beurteilung von Fluchtkantenreduzierungen* durch den Teilnehmer kann durch eine Software gelöst werden, ohne die die Lagebeurteilung im Ad-Hoc-Modus für jeden Teilnehmer schwierig ist. Die Verwendung eines Algorithmus zur automatischen Einschätzung von Fluchtkantenreduzierungen und der Warnung vor Einkesselungen erhöht allerdings den Energieverbrauch zusätzlich.

### 2.2.3 Der Mischbetrieb

Wie von [Balasubramanian u. a. (2010)] beschrieben, ist »Wiffler« eine Kombination von Ad-Hoc- und Infrastruktur-Modus. Durch den Mischbetrieb können Energieeinsparungen durch teilweisen Verzicht auf in Ad-Hoc-Netzen notwendiges Routing erzielt werden. Der Mischbetrieb schaltet - so oft es geht - vom Ad-Hoc-Modus auf den Infrastruktur-Modus um, um Energie durch Vermeiden des Routings im Ad-Hoc-Modus zu sparen. Die im [Abschnitt 2.1] vorgestellten Eigenschaften werden für den Betrieb eines Sukey-Netzes im Mischbetrieb wie folgt in Vor- und Nachteile unterteilt:

Vorteile des Mischbetriebes:

- Problemlose Kompensation des Wegfalls eines Teilnehmers
- Unabhängigkeit von ziviler Infrastruktur im Ad-Hoc-Modus
- Unabhängigkeit von einer Operationszentrale im Ad-Hoc-Modus
- Geringer Energieverbrauch der Mobiltelefone im Infrastruktur-Modus

Nachteil des Mischbetriebes:

- Hoher Energieverbrauch der Mobiltelefone im Ad-Hoc-Modus
- Die Abhängigkeit von ziviler Infrastruktur im Infrastruktur-Modus
- Die Abhängigkeit von einer Operationszentrale im Infrastruktur-Modus

Problem des Mischbetriebes im Ad-Hoc-Modus:

- Die eigenständige Beurteilung von Fluchtkantenreduzierungen im Ad-Hoc-Modus

Der Mischbetrieb ist ein Entweder-Oder-Betrieb. Sein Reiz liegt dabei in der Flexibilität, wenn möglich den Vorteil des geringeren Energieverbrauchs im Infrastruktur-Modus zu nutzen und erst wenn es notwendig ist in den Ad-Hoc-Modus umzuschalten. Der Nachteil des Energieverbrauchs im Ad-Hoc-Modus wird im Mischbetrieb durch seltenere Nutzung des

Ad-Hoc-Modus verbessert. Was bleibt, ist das Problem einer eigenständigen Beurteilung von Fluchtkantenreduzierungen durch den Teilnehmer, welches durch einen Sukey-Algorithmus gelöst werden kann.

### 2.2.4 Gegenüberstellung - Betriebsmodi

Um zu einer hinreichenden Einschätzung zu gelangen, werden Infrastruktur-Modus, Ad-Hoc-Modus und Mischbetrieb miteinander verglichen. Um die Vor- und Nachteile übersichtlich darzustellen, werden die Eigenschaften des Sukey-Netzes und ihr Bezug zu den Modi in tabellarischer Form aufgestellt. Dabei wird für das Erfüllen einer Eigenschaft einem Modus ein Punkt gegeben. Der Modus mit den meisten Punkten erfüllt die vorgestellten Eigenschaften eines Sukey-Netzes am besten. Diese Methode und Darstellungsform basiert auf Schätzungen und soll die Entscheidungsfindung erleichtern und transparenter machen. Von einer möglichen unterschiedlichen Gewichtung der angeführten vier Eigenschaften des Sukey-Netzes wird in der vorliegenden Arbeit bewusst abgesehen.

Eigenschaft	Infrastruktur	Ad-Hoc	Mischbetrieb
Keine zivile Infrastruktur	0	1	1
Keine Operationszentrale	0	1	1
Geringer Energieverbrauch	1	0	1
Kompensation von Verlusten	1	1	1
Gesamtpunkte	2	3	4

Tabelle 2.1: Gegenüberstellung von Eigenschaften und Betriebsmodi

Die Schlussfolgerung aus dieser Bewertung ist, dass für ein Sukey-Netz ein Mischbetrieb am besten geeignet ist. Da für den Mischbetrieb wie für den Ad-Hoc-Modus ein Sukey-Algorithmus zur automatischen Beurteilung von Fluchtkantenreduzierungen - wie im [Abschnitt 2.2] gezeigt - Voraussetzung ist, konzentriert sich die vorgelegte Arbeit auf die Entwicklung eines Sukey-Algorithmus als Voraussetzung für die Implementierung eines dezentralen Sukey-Netzes.

## 2.3 Organisation der beteiligten Personen

Im folgenden Teil der vorliegenden Arbeit wird untersucht, wie sich die Teilnehmer im Vorfeld und während des Einsatzes organisieren müssen. Aus den Resultaten werden wichtige Parameter des zu entwickelnden Sukey-Algorithmus gewonnen.

Die korrekte Einstellung des Sukey-Algorithmus ist für den Erfolg des Sukey-Netzes - und somit für den Schutz vor Fluchtkantenreduzierungen - von großer Bedeutung. Dieser Abschnitt stellt dar, welche Informationen für den Sukey-Algorithmus zum Lösen von Fluchtkantenreduzierungen aus Stadt- und Umgebungsplänen gewonnen werden, welche Parameter von den beteiligten Personen selbst eingestellt werden müssen und welchen Einfluss das Verhalten der Personen auf die Fluchtwegsuche hat.

### 2.3.1 Informationen aus dem Stadtplan

Die Grundlage für die Wegsuche und Analyse der Situation durch einen Sukey-Algorithmus ist der in der vorliegenden Arbeit sogenannte *Fluchtkantengraph*. Vor dem Einsatz eines Sukey-Netzes wird der Fluchtkantengraph aus einem Stadtplan generiert und um eigene Erfahrungen über die städtebaulichen Gegebenheiten erweitert. Aus den Geodaten und dem Wissen der Teilnehmer über die Umgebung muss der Fluchtkantengraph abgeleitet werden, welcher die *Sicherheit* der Fluchtkanten im Stadtplan, die Positionen von Kreuzungen und Abzweigungen, den Straßenverlauf und somit auch die *Längen der Fluchtkanten* beinhaltet. Während des Einsatzes des Sukey-Netzes erhält der Fluchtkantengraph laufend Informationen über Art und Position von *Hindernissen* im Stadtplan. Der Fluchtkantengraph muss also auf sich dynamisch verändernde Objekte eingerichtet sein.

Die Generierung eines Fluchtkantengraphen aus einem Stadtplan wird in der vorliegenden Arbeit nicht behandelt.

### 2.3.2 Informationen über Hindernisse

Gefahren durch Hindernisse, neben oder auf Fluchtkanten, werden in zwei Phasen eingestellt:

In der *Vorbereitungsphase* werden mögliche bestehende Hindernisse anhand der Beobachtungen vor Ort (Baustellen etc.) berücksichtigt und Fluchtwegen ein entsprechender Sicherheitswert zugewiesen.

In der *Durchführungsphase* werden während des Einsatzes beobachtete, bewegliche Gefahren und Hindernisse (im weiteren Verlauf dieser Arbeit wird nur noch von Hindernissen gesprochen) hinzugefügt. In einen digitalen Stadtplan werden über die Benutzerschnittstelle die Koordinaten von Hindernissen eingegeben<sup>5</sup>. Dabei entscheidet der Teilnehmer, um welchen Schweregrad es sich bei dem Hindernis handelt, z.B. ob die Straße unpassierbar, nur beeinträchtigt oder eventuell wieder passierbar ist. Diese Hindernis-Informationen werden

---

<sup>5</sup>vgl. [Sukey.org](http://Sukey.org) (2011b)



durch die Teilnehmer im laufenden Betrieb in ihre Geräte eingegeben und über das Sukey-Netz verbreitet.

Der Sukey-Algorithmus muss die freie Bewegung, also die Mobilität von Hindernissen berücksichtigen. So ist Feuer eine Art von Gefahr, die sich unberechenbar und unabhängig von Straßen und Wegen fortbewegen kann.

### 2.3.3 Anzahl der Personen am Standort

Es gibt kritische Stellen, wie Brücken, Unterführungen und andere städtebaulichen Gegebenheiten, die einem Stadtplan entweder nicht entnommen werden können oder in diesem erst gar nicht verzeichnet sind. Solche Engpässe können ab einer bestimmten Menge von Personen, die zeitgleich hindurch wollen, unpassierbar werden. Der Sukey-Algorithmus muss also die *Anzahl der Personen* am Standort berücksichtigen und sie in Bezug zur *Sicherheit* der Fluchtkanten eines Fluchtweges setzen. Der in dieser Arbeit entwickelte Sukey-Algorithmus gibt keine Fluchtweg-Empfehlung ab, wenn nicht ein Mindestmaß an Sicherheit gewährleistet ist. Es gilt der Grundsatz, dass eine Einkesselung bei unbefriedigender Sicherheit der verbleibenden Fluchtwege ungefährlicher ist, als eine Flucht in einen Engpass.

Simulationen von Fußgängergruppen in normalen Situationen und im Evakuierungsfall<sup>6</sup> auf der einen Seite, Kritik an der Panikforschung<sup>7</sup> auf der anderen Seite, zeichnen gegenteilige Bilder vom menschlichen Verhalten in Krisensituationen, die sich aber in einem Punkt einig sind: Immer sind die Orte von Katastrophen bei großen Menschenansammlungen die Flaschenhälse (Engpässe).

Aus den Standort-Informationen müssen aus der *Sicherheit* der Teilnehmer zum einen und aus der *Anzahl der Personen*, die zeitgleich eine Fluchtkante durchlaufen wollen, zum anderen Flaschenhälse erkannt und als Fluchtwege verworfen werden.

### 2.3.4 Mobilität der Personen am Standort

Die Größe des Gebietes, in dem ein Sukey-Netz zum Einsatz kommt, kann stark variieren. Eine Voraussetzung, die zur Empfehlung eines Fluchtweges durch den Sukey-Algorithmus angenommen wird, ist, dass jede Person mobil genug sei, um den längsten, eventuell letzten Fluchtweg in einer *der Gefahr angemessenen Zeit* hinter sich zu bringen. Je länger diese *der Gefahr angemessene Zeit* ist, desto höher ist das Risiko der Person vor plötzlichen Fluchtkantenreduzierungen auf dem Fluchtweg. Die Qualität eines Fluchtweges sinkt mit zunehmender Länge oder sinkender Mobilität der Personen.

---

<sup>6</sup>vgl. Dirk Helbing

<sup>7</sup>vgl. F. G. Pajonk (2006)

Die *Länge des Fluchtweges* und die *Mobilität der Personen am Standort* stehen also in einem Zusammenhang, den der Sukey-Algorithmus berücksichtigen muss.

### 2.4 Zusammenfassung - Voraussetzungen eines Sukey-Netzes

Die Abhängigkeiten zwischen der *Sicherheit* einer Fluchtkante und der *Anzahl an Personen* am Standort, sowie der *Länge von Fluchtkanten* und der *Mobilität der Personen* am Standort muss vom Sukey-Algorithmus bei der Fluchtwegsuche berücksichtigt werden. Die Sicherheit der Fluchtkanten muss im Vorfeld aus dem Stadtplan und dem Wissen der Teilnehmer, die Längen der Fluchtkanten aus dem Stadtplan gewonnen werden. Die Mobilität der Personen und die Anzahl der Personen am Standort sowie die Performanz müssen vom Teilnehmer am Standort in Eigenverantwortung eingestellt werden.

Wie gezeigt sollte ein Sukey-Netz dezentral aufgebaut werden. Für einen dezentralen Aufbau empfiehlt sich der Mischbetrieb von Infrastruktur- und Ad-Hoc-Modus. Um ein Sukey-Netz im Ad-Hoc-Modus des Mischbetriebes überhaupt betreiben zu können, muss das *Problem der eigenständigen Beurteilung von Fluchtkantenreduzierungen* mithilfe eines Sukey-Algorithmus gelöst werden. Ein Sukey-Algorithmus ist ein Vielziel-Vielwege-Suchalgorithmus, der als Input den Fluchtkantengraphen verwendet, um die aktuelle Situation am Standort einzuschätzen und im normalen Fall eine Fluchtwegempfehlung ausgibt. Eine Einkesselung ist entstanden, wenn eine Gruppe von Personen keine Möglichkeit mehr hat aus einer Gefahrensituation zu fliehen. Die Entstehung einer Einkesselung zu erkennen, ist von der korrekten Interpretation sich reduzierender Fluchtkanten abhängig. Fluchtkantenreduzierungen automatisch zu erkennen, zu bewerten und vor Einkesselungen zu warnen, ist Aufgabe des Sukey-Algorithmus.

#### 2.4.1 Parameter des Sukey-Algorithmus

Die Parameter, mit denen der Sukey-Algorithmus arbeitet, werden durch die Fluchtkanten und die Einstellungen am Standort vorgegeben. Da der Sukey-Algorithmus im Einsatz auf einem batteriebetriebenen Gerät laufen soll, wird als zusätzlicher Parameter die Performanz des Gerätes vorgegeben.

- Sicherheit der Fluchtkante *sav*

Die Sicherheit der Fluchtkante wird in der Vorbereitungsphase fest eingestellt.

- Länge der Fluchtkante *len*

Die Länge der Fluchtkante wird durch die Koordinaten ihrer Knoten im Fluchtkantengraphen vorgegeben.

- Anzahl an Personen am Standort  $p$

Die Anzahl an Personen am Standort wird in der Durchführungsphase vom Teilnehmer individuell geschätzt und als Parameter in sein Gerät eingegeben.

- Mobilität der Personen am Standort  $m$

Die Mobilität der Personen am Standort wird wie die Anzahl an Personen in der Durchführungsphase geschätzt und als Parameter eingegeben.

- Performanz des Gerätes  $e$

Die Performanz des Gerätes am Standort wird in der Durchführungsphase vom Teilnehmer für sein Gerät geschätzt. Der Sukey-Algorithmus bekommt so die Möglichkeit abhängig vom Gerät seine Rechenzeit (auf Kosten der Genauigkeit) abzukürzen.

## 3 Der Sukey-Algorithmus

In diesem Kapitel wird der Sukey-Algorithmus vorgestellt, der mithilfe der Daten eines Fluchtkantengraphen Fluchtkantenreduzierungen feststellt und eine Einschätzung der Situation am Standort, sowie eine Fluchtwegempfehlung ausgibt. Der Sukey-Algorithmus, der in diesem Kapitel schrittweise entwickelt wird, muss die in [Abschnitt 2.4.1] erarbeiteten Parameter berücksichtigen und einen qualitativen Lösungsansatz verfolgen.

Der Sukey-Algorithmus baut auf dem Fluchtkantengraphen auf, der eine Abstraktion des Umgebungs- oder Stadtplanes darstellt, die durch die dezentral gelieferten Informationen wie eigene Position und Positionen der Hindernisse erweitert werden. Da die zentrale Datenstruktur ein Graph ist, wird ein Graphalgorithmus, nämlich der A\*-Algorithmus als Ausgangspunkt gewählt.

Der Sukey-Algorithmus ist ein informierter Suchalgorithmus, da der zugrunde liegende Fluchtkantengraph bekannte Ziele hat, auf der Wegsuche die Güte des Fluchtweges monoton sinkt und eine monotone Heuristik verwendet werden kann.

Zusätzlich soll der optimale Pfad als erstes gefunden werden und eine Vielzielsuche mit Veränderungen der Kantenbewertung des Fluchtkantengraphen nach jedem gefundenen Ziel möglich sein.

### 3.1 Darstellung des A\*-Algorithmus und seiner Varianten

Der A\*-Algorithmus nach [Hart u. a. (1968)] ist ein informierter Suchalgorithmus. Er sucht den kürzesten Weg in einem Graphen  $G$  und findet immer den kostengünstigsten Weg, sofern ein solcher existiert. Während der Suche entsteht ein Subgraph  $G_n$  vom Startknoten bis zu einem Knoten  $n$  mit den Kosten  $\hat{g}(n)$ , die während der Suche nicht den optimalen Kosten  $g(n)$  entsprechen müssen. Dabei gilt immer  $\hat{g}(n) \geq g(n)$ . Die Kosten von  $n_i$  nach  $n_j$  einer Kante  $a_{ij}$  sind  $c_{ij}$ . Die Kosten  $\hat{g}(n)$  werden ermittelt, nachdem der Nachfolgeoperator  $\Gamma$  auf  $n_i$  seine Nachfolger  $n_j$  mit den Kosten  $c_{ij} : \{(n_j, c_{ij})\}$  expandiert hat. Die Kosten für  $n_j$  sind  $\hat{g}(n_j) = g(n_i) + c_{ij}$ . Der A\*-Algorithmus sucht den kostengünstigsten Weg. Die Kosten sind beliebig hoch, solange  $c_{ij} \geq \delta > 0$  gilt. Dabei beschreibt  $\delta$  die geringsten anzunehmenden Kosten, die es im Graphen gibt.

Der  $A^*$ -Algorithmus arbeitet mit einer Schätzfunktion  $\hat{h}(n)$ , die die echten Restkosten  $h(n)$  eines Knotens  $n_i$  zu einem Ziel  $t \in T \subset \{n_i\}$  nicht überschätzen darf. Ist  $\hat{h}(n) \leq h(n)$ , so ist die Heuristik zulässig. Außerdem gilt nach [Hart u. a. (1968)]<sup>1</sup>:  $h(n) = \min_{t \in T} h(n, t)$ . Jede monotone Heuristik (z.B. die Luftlinie) ist zulässig.

Die Evaluationsfunktion<sup>2</sup> für jeden Knoten  $n$  ist  $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$ . Nach ihr werden die als »offen« markierte Knoten während der Suche in einer Liste (Open-List) sortiert. Knoten, zu denen der optimale Pfad  $g(n)$  bereits bekannt ist, werden als »geschlossen« markiert, damit sie nicht doppelt untersucht werden (Closed-List). Nach [Hart u. a. (1968), II A.] läuft der Suchalgorithmus  $A^*$  in folgenden Schritten ab (Übersetzung durch den Autor):

1. Markiere den Startknoten  $s$  als »offen« und berechne  $\hat{f}(s)$ .
2. Wähle den offenen Knoten  $n$  mit dem niedrigsten Wert von  $\hat{f}$ . Knoten mit gleichen Werten dürfen willkürlich gewählt, Zielknoten  $n \in T$  müssen jedoch bevorzugt behandelt werden.
3. Ist  $n \in T$  ein Zielknoten, markiere  $n$  als geschlossen und terminiere den Algorithmus.
4. Markiere  $n$  andernfalls als »geschlossen« und wende den Nachfolgeoperator  $\Gamma$  auf  $n$  an. Berechne  $\hat{f}$  für jeden Nachfolgeknoten von  $n$  und markiere jeden als »offen«, der noch nicht als »geschlossen« markiert ist, es sei denn, das neue  $\hat{f}(n_i)$  des als »geschlossen« markierten Nachfolgers  $n_i$  ist geringer als sein ursprüngliches  $\hat{f}$ . Gehe zu Schritt 2.

#### 3.1.1 Algorithmen für den Sukey-Algorithmus

Es folgt eine Auswahl möglicher informierter Suchalgorithmen, die auf dem  $A^*$ -Algorithmus beruhen.

- IDA\* (Iterative Deepening A\*) [vgl. Mahanti u. a. (1992)]

Zur Verringerung des Speicherverbrauchs auf Kosten der Laufzeit kommt der IDA\*-Algorithmus infrage. IDA\* ist ein  $A^*$ <sup>3</sup> mit iterativer Tiefensuche. Er verfolgt einen Fluchtweg bis eine bestimmte Kosten-Tiefenschranke erreicht ist. Dann wendet er sich dem nächsten Fluchtweg zu.

- LPA\* (Lifelong Planning A\*) [vgl. Koenig u. a. (2004)]

---

<sup>1</sup> [Hart u. a. (1968), I B.] »For simplicity, we shall represent the unique cost of an optimal path from  $n$  to a preferred goal node of  $n$  by the symbol  $h(n)$ ; i.e.,  $h(n) = \min_{t \in T} h(n, t)$ .«

<sup>2</sup> [Hart u. a. (1968), II A.] »Suppose some evaluation function  $\hat{f}(n)$  could be calculated for any node  $n$ .«

<sup>3</sup>Die Erklärung des  $A^*$ -Algorithmus nach [Hart u. a. (1968)] befindet sich im [Abschnitt 3.1].

Normaler  $A^*$ , der sich an vorherige Ergebnisse erinnern kann. Die Resultate der Versuchen werden übernommen, wenn durch Änderungen im Graphen keine Unterschiede entstanden sind.

- $AD^*$  (Anytime Dynamic  $A^*$ ) [vgl. Maxim Likhachev und Thrun (2005)]

Der  $AD^*$  rechnet einen Graphen einmal durch und passt die Werte der Fluchtwege bei Änderungen durch z.B. Hindernisse oder Kantenabwertungen an.

- $D^*$  (Dynamic  $A^*$ ) [vgl. Stentz (1995)]

Um Veränderungen durch Hindernisse während der Wegsuche zuzulassen, käme der  $D^*$ -Algorithmus infrage. Eine Verbesserung ist der  $D^*$  Lite [vgl. Koenig und Likhachev (2002)], welcher leichter zu analysieren und noch etwas schneller als der  $D^*$  ist. Diese Art von Algorithmus wird in der vorliegenden Arbeit für den vorgestellten Fluchtkantengraphen nicht infrage kommen, da Veränderungen am Fluchtkantengraphen stets vor der Suche abgeschlossen sein werden.

Bei ausreichendem Speicher des Gerätes (z.B. eines Mobiltelefons) eines Teilnehmers am Sukey-Netz ist wegen der punktuellen Veränderungen durch Hindernisse und durch Kantenabwertungen bei der Durchführung der Vielzielsuche des Sukey-Algorithmus ein dynamischer  $A^*$ -Algorithmus ( $AD^*$ ) zu bevorzugen. Hat das Gerät keinen großen Speicher, oder kann das Programm das den Sukey-Algorithmus ausführt nicht genug Speicher verwenden, sollte ein speichersparender Algorithmus wie  $IDA^*$  gewählt werden. Dessen Laufzeit ist bei komplexeren Graphen dann aber durch die Neuberechnung von bereits verworfenen Wegen höher als beim normalen  $A^*$ -Algorithmus oder einem dynamischen  $A^*$ -Algorithmus [vgl. Mahanti u. a. (1992)].

Es ist vorauszusehen, dass das  $A^*$ -Komplexitätsproblem bei den zur Anwendung kommenden Fluchtkantengraphen keine große Rolle spielen wird, da die Art von Evakuierungsszenarien voraussichtlich nicht sehr groß sein wird. Da die vorgestellten Algorithmen, die entweder das Speicher- oder das Laufzeitproblem ins Auge fassen, den herkömmlichen  $A^*$ -Algorithmus zugrunde legen, verwendet die vorgelegte Arbeit den  $A^*$ -Algorithmus, um eine Grundlage für Weiterentwicklungen zu schaffen und um die Entscheidung für schnellere Laufzeit oder für weniger Speicherverbrauch in der vorgelegten Arbeit nicht treffen zu müssen.

Den Sukey-Algorithmus auf Basis des  $A^*$ -Algorithmus nach [Hart u. a. (1968)] zu entwickeln, ist Ziel dieses Kapitels.

## 3.2 Adaption des Sukey-Algorithmus

Der wesentliche Unterschied zu dem herkömmlichen A\*-Algorithmus ist, dass dieser quantitative Strategien umsetzt. Es wird der kürzeste Pfad gesucht. Der Sukey-Algorithmus jedoch ist qualitativ. Es wird der beste Pfad gesucht.

Der beste Pfad berücksichtigt neben der Länge des Fluchtweges und seiner Breite (der Sicherheit) auch die Anzahl abgehender (zusätzlicher) Fluchtwege, die die Qualität einer Fluchtkante zusätzlich steigern. Darum werden die Werte relativ bewertet und miteinander multipliziert. Bei gleicher Länge soll ein Fluchtweg mit vielen gleichmäßig verteilten Ausweichmöglichkeiten einem Fluchtweg mit weniger Ausweichmöglichkeiten vorgezogen werden. [Abbildung 3.7] stellt die Bevorzugung eines Fluchtweges mit einer zusätzlichen Ausweichmöglichkeit anhand eines Beispiels dar.

Die Sicherheit einer Fluchtkante und die Anzahl an Personen am Standort, sowie ihre Mobilität, werden von vorneherein als relative Werte eingestellt. Die Längen der Fluchtkanten werden ins Verhältnis zur Gesamtausdehnung des Fluchtkantengraphen gesetzt und liegen damit ebenfalls als relative Werte vor.

Der Sukey-Algorithmus ist informiert und gewinnorientiert, verwendet relative Fluchtweggüten zum Finden von ggf. mehreren möglichen Fluchtwegen und findet immer den möglichst *sichersten* Fluchtweg als erstes. In [Abschnitt 3.7] werden zusammenfassend die Variablen des A\*-Algorithmus und die des Sukey-Algorithmus einander kongruent zugeordnet.

Trotz dieser Unterschiede zum A\*-Algorithmus kann der A\* als Grundlage für den Sukey-Algorithmus genommen werden, weil sich relative Werte über eine Umkehrfunktion in absolute Zahlen umwandeln lassen. So ist der Zusammenhang zwischen A\*- und Sukey-Algorithmus hergestellt. Die Funktion  $k$  kehrt die Fluchtweggüte in absolute Kosten um. Die Funktion  $w$  kehrt absolute Kosten in eine Fluchtweggüte um.

$$k(x) = -\ln_x \text{ mit } x \in ]0, 1[$$

$$w(x) = e^{-x} \text{ mit } x \in ]0, \infty[$$

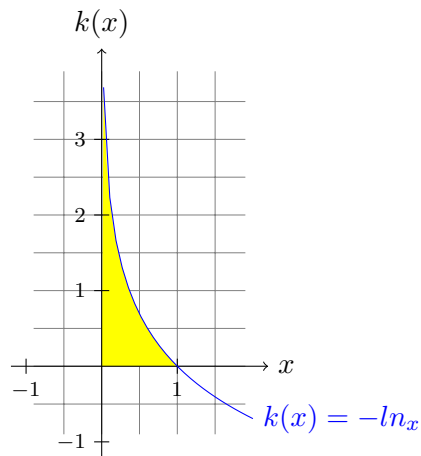


Abbildung 3.1: Die Umkehrfunktion  $k(x) = -\ln x$

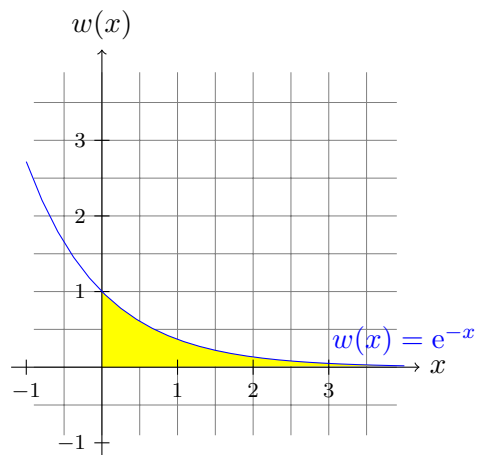


Abbildung 3.2: Die Umkehrfunktion  $w(x) = e^{-x}$

Für Addition und Multiplikation gelten durch die Umkehrung:

$$x^2 = w(2k(x)) \text{ mit } x \in ]0, 1[$$

$$2x = k(w(x)^2) \text{ mit } x \in ]0, \infty[$$

weil:

$$x = w(k(x)) \text{ mit } x \in ]0, 1[$$

$$x = k(w(x)) \text{ mit } x \in ]0, \infty[$$



Durch diese Umkehrung lassen sich Fluchtwege im Fluchtkantengraphen durch Algorithmen, die Längen addieren, berechnen. Der *natürliche Logarithmus*  $\ln_x$  ist durch  $-\ln_x$  an der x-Achse gespiegelt,  $e^x$  ist durch  $e^{-x}$  an der y-Achse gespiegelt. Daher werden die Relationen gedreht:

$$x^2 < x \text{ ist kongruent zu } 2k(x) > k(x) \text{ mit } x \in ]0, 1[$$

$$2x > x \text{ ist kongruent zu } w(x)^2 < w(x) \text{ mit } x \in ]0, \infty[$$

Der in der vorgelegten Arbeit vorgestellte Sukey-Algorithmus nutzt eine Heuristik. Er multipliziert dabei relative Werte zwischen 0 und 1 und dreht die Vorzeichen. Die Heuristik darf beim Sukey-Algorithmus im Gegensatz zum A\*-Algorithmus also niemals *unterschätzen*.

### 3.3 Der Fluchtkantengraph

Der Sukey-Algorithmus sucht einen Fluchtweg in einem Fluchtkantengraphen  $F$  und findet immer den möglichst sichersten Weg, sofern einer existiert. Der Fluchtkantengraph beinhaltet alle Daten, die der Sukey-Algorithmus zur gewinnorientierten Berechnung des sichersten Fluchtweges benötigt.

Der Fluchtkantengraph hat eine Dateneingabe. Diese wird in diesem Abschnitt erläutert.

Kürzel	Beschreibung
N	Alle Knoten
A	Alle Fluchtkanten
H	Alle Hindernisse
d	Der aktuelle Standort
pos	Eine xy-Position im Fluchtkantengraphen
sav	Die Sicherheit einer Fluchtkante
len	Die Länge einer Fluchtkante
cnt	Der Zähler einer Fluchtkante für die Vielzielsuche
v	Die Gefahr durch ein Hindernis
e	Die Performanz
p	Die Anzahl der Personen am Standort
m	Die Mobilität der Personen am Standort

Tabelle 3.1: Übersicht - Dateneingabe des Fluchtkantengraphen

$$F = (N, A, H, d, pos, sav, len, cnt, v, e, p, m)$$

- Die Position  $pos$  ist die Position von Objekten  $K$  im Fluchtkantengraphen. Eine Position besteht aus einer  $x$ - und einer  $y$ -Koordinate  $(pos_x, pos_y)$ . Die Position der Objekte  $K$  ist injektiv, weil sie im Fluchtkantengraphen nicht aufeinander stehen sollen. Die Objekte  $K$  des Fluchtkantengraphen sind paarweise disjunkt.

$$K = N \cup H \cup \{d\} : \text{ paarweise disjunkt.}$$

$$pos : K \rightarrow \mathbb{R} \times \mathbb{R} \text{ ist injektiv.}$$

- Knoten  $n$  (*nodes*) im Fluchtkantengraphen werden modelliert als Objekte mit Abzweigungen oder Kreuzungen mit einer exakten Position. Ihr Knotengrad<sup>4</sup> ist ungleich 2. Es gibt Knoten, die Ziele  $t$  sind. Jeder Knoten hat ein Ziel  $z$ , dessen Abstand zu ihm am geringsten ist.

$$N = \{n_i \mid 1 \leq i \leq |N|\}$$

$$T \subseteq N \text{ die möglichen Ziele.}$$

$$z : N \rightarrow T \text{ mit } z(n_i) = \min(\overline{n_i t_j})$$

- Fluchtkanten  $a$  (*arcs*) im Fluchtkantengraphen werden modelliert als Objekte mit Straßen mit einer relativen Länge  $len$  und einer relativen Sicherheit  $sav$  zwischen zwei Knoten. Die Sicherheit einer Fluchtkante wird aus den Geodaten und dem Wissen der Teilnehmer ermittelt. Die Fluchtkanten dürfen sich kreuzungsfrei überschneiden (z.B. als Brücken oder Unterführungen). Sie sind schleifenfrei und nicht parallel (d.h. zwei Knoten dürfen nur durch eine Fluchtkante verbunden sein). Eine maximale Länge ( $maxlen$ ) wird verwendet, um aus absoluten Längen relative Längen zu machen. Diese ist die Diagonale des Fluchtkantengraphen, wie in [Abbildung 3.3] dargestellt.

---

<sup>4</sup>Knotengrad ist die Anzahl der abgehenden Kanten eines Knotens.

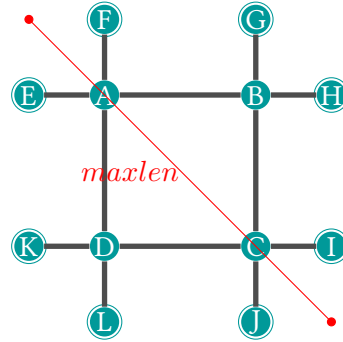


Abbildung 3.3: Diagonale des Fluchtkantengraphen *maxlen*

Ein Zähler *cnt* (*counter*) ist notwendig bei der Einschätzung der Situation am Standort bei der Suche nach mehr als einem Fluchtweg. Sein maximaler Wert entspricht der eingestellten Performanz *e*.

$a_{ij}$  ist die Kante von  $n_i$  nach  $n_j$

$A \subseteq \{a_{ij} \mid \text{mit } (n_i, n_j) \in N \times N \text{ und } i \neq j\}$

$sav : A \rightarrow ]0, 1]$

$len : A \rightarrow ]0, 1]$

$$maxlen = \sqrt{(\max(pos_x(N)) - \min(pos_x(N)))^2 + (\max(pos_y(N)) - \min(pos_y(N)))^2}$$

$$len(a_{ij}) = \frac{\sqrt{(pos_x(n_i) - pos_x(n_j))^2 + (pos_y(n_i) - pos_y(n_j))^2}}{maxlen}$$

$cnt(a_{ij}) : A \rightarrow \{1, \dots, e\} \subseteq \mathbb{N}^+$

- Hindernisse  $h$  im Fluchtkantengraphen werden modelliert als Objekte mit einer Intensität  $v$  (*value*). Die Position und die Intensität von Hindernissen werden von den Teilnehmern am Sukey-Netz ermittelt und können hinzugefügt und wieder entfernt werden.

$$H = \{h_i \mid 1 \leq i \leq |H|\}$$

$$v(H) : H \rightarrow ]0, 1]$$

- Der Standort  $d$  ist das technische Gerät (*device*), welches jeder Teilnehmer im Sukey-Netz mit sich führt. Der Standort wird modelliert als Objekt mit einer Performanz  $e$ , einem Personenaufkommen  $p$  und deren Mobilität  $m$ . Der Standort liegt immer auf einer Kante.

$$\begin{aligned} \text{pos}(d) \text{ auf } a_{ij}, \text{ sodass stets gilt: } & \frac{|\text{pos}_x(n_i) - \text{pos}_x(n_j)|}{|\text{pos}_x(n_i) - \text{pos}_x(d)|} = \frac{|\text{pos}_y(n_i) - \text{pos}_y(n_j)|}{|\text{pos}_y(n_i) - \text{pos}_y(d)|} \\ e : \{d\} & \rightarrow [1, \dots, \infty[ \in \mathbb{N}^+ \\ p : \{d\} & \rightarrow ]0, 1] \\ m : \{d\} & \rightarrow ]0, 1] \end{aligned}$$

Für einen Weg vom Standort  $d$  zu einem Teilziel  $n_j$  im Fluchtkantengraphen wird die Fluchtkante  $a_{ij} \in A : \text{pos}(d)$  auf  $a_{ij}$  unterbrochen. In dem so erweiterten Fluchtkantengraphen  $F'$  liegt der Standort auf keiner Kante mehr, sondern stellt selbst einen Knoten mit einem Knotengrad = 2 dar.

$$\begin{aligned} N' &= \{n_i \mid 1 \leq i \leq |N|\} \in N \cup \{d\} \\ \text{sei } \text{pos}(d) & \text{ auf } a_{ij} \\ A' &\subseteq \{A \setminus \{a_{ij}\} \cup \{a_{id}, a_{dj}\} \mid \text{mit } (n_i, n_j) \in N \times N \text{ und } i \neq j\} \\ F' &= (N', A', H, d, \text{pos}, \text{sav}, \text{len}, \text{cnt}, v, e, p, m) \end{aligned}$$

### 3.4 Hindernisregeln

In einem Fluchtkantengraphen treten Hindernisse  $H$  auf. Ein Hindernis hat eine Intensität  $v(h)$ , welche von den Teilnehmern am Sukey-Netz eingeschätzt und den beobachteten Hindernissen gegeben werden, und kann sich unabhängig von den Fluchtkanten bewegen. Zusätzlich kann ein Hindernis jederzeit unvorhergesehen auftauchen. Hindernisse haben eine Position und keinen eindeutigen Bezeichner. Da Hindernisse paarweise disjunkt sind, ist die Position eines Hindernisses als sein eindeutiger Bezeichner ausreichend.

In dieser Arbeit wird zwischen *bekannten Hindernissen* und *unbekannten Hindernissen* unterschieden:

- bekannte Hindernisse

Der Einfluss auf Fluchtkanten ist vorhersehbar. Bekannte Hindernisse beeinflussen eine Fluchtkante mithilfe der *direkten Hindernisregel (dhr)*, wie in [Abschnitt 3.4.1] beschrieben, direkt oder mithilfe der *indirekten Hindernisregel (ihr)*, wie in [Abschnitt 3.4.2] beschrieben, indirekt.

- unbekannte Hindernisse

Der Einfluss unbekannter Hindernisse auf Fluchtkanten ist nicht vorhersehbar, weil ihre Position nicht bekannt ist. Ihr Einfluss wird implizit über die stärkere Abwertung langer Fluchtkanten, sowie durch die Abwertung mehrfach benutzter Fluchtkanten bei der Suche nach mehr als einem Fluchtweg in [Abschnitt 3.5] berücksichtigt.

Die in diesem Abschnitt vorgestellten Regeln zur Behandlung bekannter Hindernisse,  $dhr$  und  $ihr$ , ermitteln für jede Fluchtkante  $a_{ij}$  die Einflüsse  $dhr(a_{ij})$  und  $ihr(a_{ij})$  auf die Güte einer Fluchtkante. Tabelle 3.2 stellt Beispiele für Hindernisregel-Werte und ihre Bedeutung vor.

$dhr(a_{ij})$	$ihr(a_{ij})$	Bedeutung
0.0	-	$a_{ij}$ ist unpassierbar
0.5	0.5	$a_{ij}$ ist noch passierbar
1.0	1.0	$a_{ij}$ ist passierbar

Tabelle 3.2: Beispiele für Einflüsse von  $dhr(a_{ij})$  und  $ihr(a_{ij})$  auf die Güte einer Fluchtkante und ihre Bedeutung

### 3.4.1 Die direkte Hindernisregel

Die direkte Hindernisregel  $dhr$  ermittelt die Wirkung aller Hindernisse  $H_{arc}$  auf eine Fluchtkante  $a_{ij}$ , die genau auf dieser Fluchtkante liegen. Zur Ermittlung der Wirkung werden die Intensitäten  $v(h)$  der Hindernisse hinzugezogen. Die Wirkung ist:

$$\begin{aligned}
 H_{arc} &= \{h \mid \exists a_{ij} \in A : pos(h) \text{ auf } a_{ij}\} \subseteq H \\
 dhr &: A \rightarrow [0, 1] \\
 dhr(a_{ij}) &= \prod_{h \in H_{arc}} (1 - v(h)) \\
 \text{weiterhin gilt } v &: H \rightarrow ]0, 1]
 \end{aligned}$$

### 3.4.2 Die indirekte Hindernisregel

Die indirekte Hindernisregel  $ihr$  ermittelt die Wirkung aller Hindernisse  $H_{sph}$  (*sphere*) auf eine Fluchtkante  $a_{ij}$ , wo auch immer sie sich befinden. Sie bezieht sich auf die Position des Teilzielknotens  $n_j$  einer Fluchtkante.

Unter der Voraussetzung, dass sich Hindernisse nicht schneller als die Personen am Standort bewegen<sup>5</sup>, gilt:

$\overline{hn}_j$  ist der Abstand des Hindernisses  $h$  zum Teilzielknoten  $n_j$

$\overline{dn}_j$  ist der Abstand des Standortes  $d$  zum Teilzielknoten  $n_j$

$$H_{sph} = \{h \mid \frac{\overline{hn}_j}{\overline{dn}_j} \leq 1\}$$

$$ihr : A \rightarrow ]0, 1]$$

$$ihr(a_{ij}) = \prod_{h \in H_{sph}} \left( 1 - v(h) \left( 1 - \frac{\overline{hn}_j}{\overline{dn}_j} \right) \right)$$

weiterhin gilt  $v : H \rightarrow ]0, 1]$

Die indirekte Hindernisregel (ihr) ermittelt die Wirkung von Hindernissen durch Inbezugsetzen der Entfernungen vom Hindernis und dem Standort zum Teilzielknoten, wie in [Abbildung 3.4] an einem Beispiel verdeutlicht.

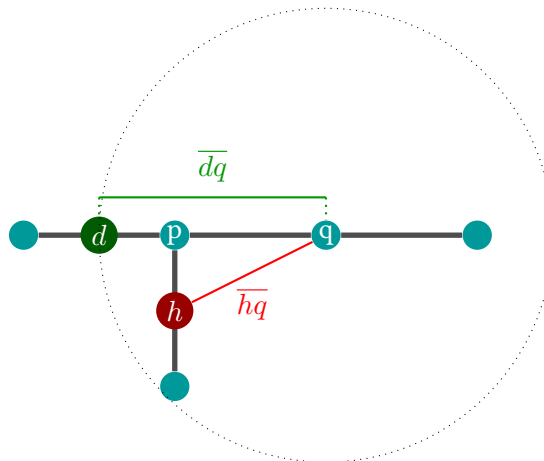


Abbildung 3.4: Wirkungsweise der indirekten Hindernisregel (ihr)

Die Formel für die indirekte Hindernisregel (ihr) berücksichtigt die Richtung, aus der eine Fluchtkante durchlaufen wird. Am Beispiel in [Abbildung 3.5] wird gezeigt, dass für den Standort  $d_1$  die Fluchtkante  $a_{pq}$  vorteilhafter ist, da er sich beim Durchlaufen der Fluchtkante

<sup>5</sup> Durch Einbeziehung der Mobilität  $m$  in die Formel kann der Einfluss noch genauer ermittelt werden. Für die vorliegende Arbeit wird die Voraussetzung als ausreichend betrachtet, dass sich ein Hindernis nicht schneller als die Personen am Standort bewegt.

von dem Hindernis  $h$  wegbewegt, als für den Standort  $d_2$ , der sich beim Durchlaufen von  $a_{qp}$  aus der Gegenrichtung zunächst auf das Hindernis  $h$  zubewegt.

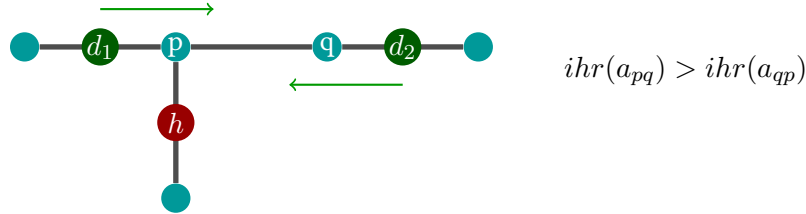


Abbildung 3.5: Berücksichtigung der Bewegungsrichtung durch die indirekte Hindernisregel (ihr)

### 3.5 Die Fluchtweggüte

Ein Flucht- oder Teilfluchtweg lässt sich als Teilgraph des aktuellen Fluchtkantengraphen auffassen. Ein solcher Fluchtweg vom Standort  $d$  bis zum Teilziel- oder Zielknoten  $n_j \in N$  ist  $FW(d, n_j) \subseteq F'$  (im weiteren Verlauf kurz:  $FW_n$ ).

Ein Fluchtweg  $FW_n$  eines Fluchtkantengraphen hat die Fluchtweggüte  $fwg(FW_n)$  und ist einer von allen möglichen Wegen  $W$ . Die Fluchtweggüte  $fwg(FW_n)$  ist für den Sukey-Algorithmus das, was die absolute Länge zu jedem Knoten auf seiner Suche für den A\*-Algorithmus ist.

$$W = \{(a_{ij}, a_{jk}, a_{kl}, \dots) \mid n_i n_j n_k n_l \in N\} \text{ alle Wege in } F'$$

$$FW(d, n_j) = \{a_{ij} \mid (a_{di}, \dots, a_{kj}) \in W\} \text{ (kurz: } FW_n)$$

$$fwg : N \rightarrow [-1, 1[$$

Die Fluchtweggüte wird aus der Sicherheit  $sav$  des Fluchtweges und seiner Qualität  $qual$  ermittelt. Sie ist der Wert, der zur Einsortierung der expandierten Knoten in die »Open-List« durch den Sukey-Algorithmus verwendet wird (eben das, was die absolute Länge an jedem expandierten Knoten für den A\*-Algorithmus ist).

$$sav : FW \rightarrow [-1, 1[$$

$$qual : FW \rightarrow [0, 1[$$

$$fwg(FW_n) = sav(FW_n) \cdot qual(FW_n) \in [-1, 1[$$

Ein Fluchtweg, dessen Fluchtweggüte  $fwg(FW_n) \leq 0$  ist, ist unpassierbar und wird vom Suchalgorithmus nicht weiter verfolgt. Durch diese Einschränkung gilt zusätzlich:

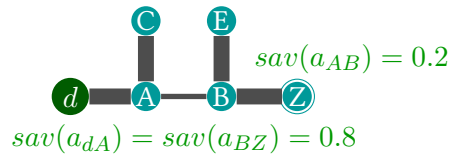
$$fwg(FW_n)^+ : N \rightarrow ]0, 1[$$

**Die Sicherheit** einer Flucht-kante ist definiert durch die Stelle, an der der geringste Durchsatz (*throughput*) auf ihr entsteht (Engpass). Wie in [Goldratt (2004) S. 238 »(...) the bottlenecks dictate (...) throughput.«] festgestellt, richtet sich der Durchsatz nach dem Flaschenhals, hier also nach der *unsichersten* Flucht-kante  $\min(sav(a_{ij}))$  im Fluchtweg  $FW_n$ . Der Durchsatz sind die Anzahl an Personen  $p$ , die eine Flucht-kante zeitgleich durchlaufen. Die Hindernisse verringern die Sicherheit einer Flucht-kante zusätzlich durch die Hindernisregeln  $dhr(a_{ij})$  und  $ihr(a_{ij})$ . Es gilt:

$$savh(a_{ij}) = sav(a_{ij}) \cdot dhr(a_{ij}) \cdot ihr(a_{ij}) - p$$

$$sav(FW_n) = \min(savh(a_{ij}), \dots, savh(a_{kl})) \text{ mit } a_{ij}, a_{kl} \in FW_n \text{ alle Kanten in } FW_n$$

Das Beispiel in [Abbildung 3.6] zeigt einen Flucht-kantengraphen mit einem Engpass durch die Flucht-kante  $a_{AB}$  für einen Standort  $d$  auf dem Weg zum Ziel  $Z$ . Dieser Engpass »diktiert« (vgl. [Goldratt (2004) S. 238]) unabhängig von der Beschaffenheit vom Rest des Weges den Durchsatz an Personen die den Fluchtweg gefahrlos durchlaufen können.



sei  $dhr(a_{ij}) = 1.0$  und  $ihr(a_{ij}) = 1.0$  (keine Hindernisse)

sei  $p = 0.1$  die Anzahl an Personen

$$\begin{aligned} sav(FW_Z) &= \min\left((sav(a_{dA}) - p), (sav(a_{AB}) - p), (sav(a_{BZ}) - p)\right) \\ &= \min\left((0.8 - 0.1), (0.2 - 0.1), (0.8 - 0.1)\right) \\ &= \min\left(0.7, 0.1, 0.7\right) \\ &= \mathbf{0.1} \end{aligned}$$

Abbildung 3.6: Auswirkung eines Engpasses auf die Sicherheit des Fluchtweges



Das Beispiel in [Abbildung 3.6] zeigt, dass der Engpass für die Sicherheit eines Fluchtweges entscheidend und dass die Sicherheit aller anderen Fluchtkanten für das Ergebnis unwichtig ist.

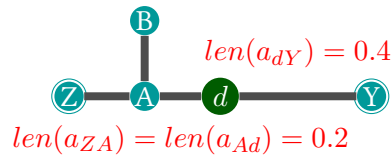
**Die Bewertung der Länge** stellt die Qualität  $qual$  der Kante dar. Sie wird in der hier vorgelegten Arbeit logarithmisch (durch Multiplikation relativer Längen) gewertet, damit Fluchtwege mit vielen kurzen Fluchtkanten, also mit vielen Ausweichmöglichkeiten, Fluchtwege mit wenigen langen Fluchtkanten, bei gleicher Gesamtlänge, vorgezogen werden.

In Abhängigkeit von der Mobilität  $m$  [Abschnitt 2.3.4] der Personen am Standort hat die Qualität eines Fluchtweges  $FW_n$  Einfluss auf seine Güte  $fwg(FW_n)$ .

Unter Berücksichtigung des Zählers jeder Fluchtkante  $cnt(a_{ij})$ , über den die Anzahl der Verwendungen dieser Fluchtkante während der Vielzielsuche durch den Sukey-Algorithmus festgestellt wird, gilt für die Qualität einer Fluchtkante:

$$qual(FW_n) = \prod_{a_{ij} \in FW_n} \frac{1 - len(a_{ij}) \cdot (1 - m)}{cnt(a_{ij})} \in [0, 1[$$

Das Beispiel in [Abbildung 3.7] verdeutlicht die Bevorzugung von Fluchtwegen mit mehr Ausweichmöglichkeiten als ansonsten gleichwertige Alternativen. Vereinfacht wird ein Zählerstand von  $cnt(a_{ij}) = 1$  für jede Fluchtkante angenommen und daher als Nenner des Bruchs in den Beispielformeln weggelassen. Für die Mobilität der Personen am Standort wird der Wert  $m = 0.5$  angenommen. Das Beispiel zeigt, dass der Fluchtweg mit besseren Ausweichmöglichkeiten eine höhere Qualität  $qual$  hat und dem schlechteren Fluchtweg vorgezogen werden wird.



sei  $m = 0.5$

$$qual(FW_Y) = (1 - 0.4 \cdot (1 - 0.5)) = \mathbf{0.8}$$

$$qual(FW_Z) = (1 - 0.2 \cdot (1 - 0.5))^2 = \mathbf{0.81}$$

Abbildung 3.7: Bevorzugung eines Fluchtweges mit einer zusätzlichen Ausweichmöglichkeit

Eine hohe Mobilität  $m$  der Personen am Standort beeinflusst die Fluchtweggüte weniger als eine niedrige Mobilität  $m$ . Je länger ein Weg ist, desto stärker ist die Abwertung der *Bewertung der Länge* durch die Mobilität. Es wird angenommen, dass die unmobilsten Personen mit der Mobilität  $\gamma$ , wobei gilt  $m \geq \gamma > 0$  mit  $m : \{d\} \rightarrow ]0, 1]$ , in der Lage sind den längst möglichen Fluchtweg in einer der Gefahr angemessenen Zeit zu durchlaufen. Ferner wird angenommen, dass für die theoretisch mobilsten Personen mit  $m = 1.0$  die Länge eines Fluchtweges keine Rolle spielt.<sup>6</sup>

Das folgende Beispiel veranschaulichen den Einfluss einer hohen Mobilität von  $m = 0.9$  auf die Qualität der Länge. Vereinfacht wird für jedes  $cnt(a_{ij}) = 1$  angenommen und hier daher weggelassen. In [Abbildung 3.8] und [Abbildung 3.9] wird im Koordinatensystem veranschaulicht, dass, bei sich vergrößernder Entfernung eines Ziel- oder Teilzielknotens, die Auswirkung der Mobilität auf die Bewertung der Länge linear ansteigt.<sup>7</sup>

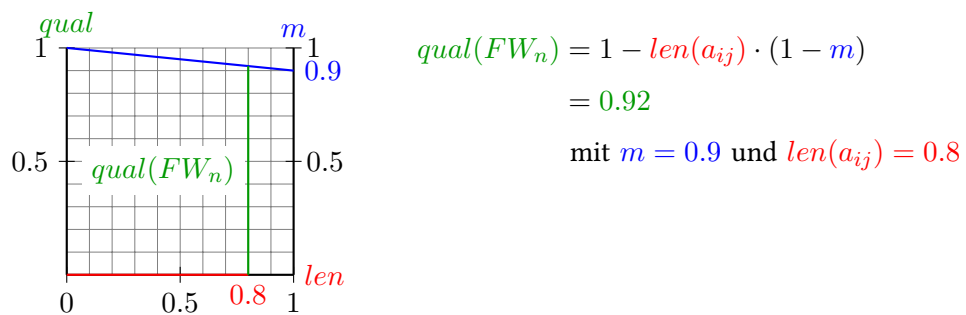
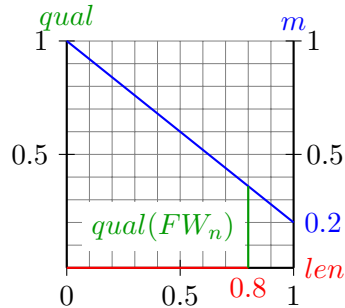


Abbildung 3.8: Einfluss der Länge auf  $FW_n$  bei hoher Mobilität von Personen

Eine niedrige Mobilität  $m = 0.2$  von Personen beeinflusst die Fluchtweggüte bei größer werdender Entfernung stärker als eine hohe Mobilität von Personen. [Abbildung 3.9] veranschaulicht den Einfluss einer niedrigen Mobilität von Personen auf die Bewertung der Länge.

<sup>6</sup> $m = 1.0$  könnte etwa Lichtgeschwindigkeit sein.

<sup>7</sup>Für die vorliegende Arbeit reicht ein linearer Anstieg der Auswirkung der Mobilität als einfachste Kurve. Nach eingehenden empirischen Tests, die nicht Bestandteil dieser Arbeit sind, kann diese Kurve durch eine komplexere Formel verbessert werden.



$$\begin{aligned} \text{qual}(FW_n) &= 1 - \text{len}(a_{ij}) \cdot (1 - m) \\ &= 0.36 \\ &\text{mit } m = 0.2 \text{ und } \text{len}(a_{ij}) = 0.8 \end{aligned}$$

Abbildung 3.9: Einfluss der Länge auf  $FW_n$  bei geringer Mobilität von Personen

Für die Einschätzung der Situation an einem Standort ist unter anderem die Anzahl der Fluchtwege, die der Sukey-Algorithmus findet, von Bedeutung. Hat der Sukey-Algorithmus einen Fluchtweg zu einem Ziel  $FW_t$  gefunden, werden die Zähler  $\text{cnt}(a_{ij})$  aller Fluchtkanten dieses Fluchtweges um 1 inkrementiert. Der Sukey-Algorithmus startet seine Suche nach einem Fluchtweg solange neu, bis er eine von der Performanz  $e$  abhängige Anzahl an Durchläufen vorgenommen hat oder kein Ziel gefunden wurde. Die Performanz  $e$  wird am Standort  $d$  als Parameter des Fluchtkantengraphen eingestellt. Der Sukey-Algorithmus terminiert also erst ab dem  $e$ -ten Durchlauf oder wenn er keinen Fluchtweg findet.

Das Beispiel in [Abbildung 3.10] zeigt den Sinn eines Zählers, wenn Fluchtwege durch gleiche Fluchtkanten verlaufen.

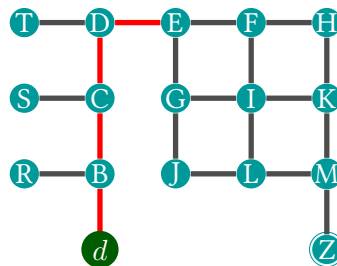


Abbildung 3.10: Fluchtkantengraph mit 12 unterschiedlichen Fluchtweisen, die gleiche Fluchtkanten nutzen

Werden Fluchtkanten durch verschiedene Fluchtwege mehrfach genutzt, ist die Situation am Standort schlechter als es die ausschließliche Betrachtung ihrer Anzahl suggeriert. Der Abwertungsvorgang bewirkt, dass lediglich der erste gefundene Fluchtweg der möglichst

sicherste ist. Die Werte aller weiteren Fluchtwege sind Ausdruck ihrer *Andersartigkeit* zu ihren Vorgängern. Eine Einordnung in eine Qualitäts-Rangfolge der Fluchtwege untereinander ist dadurch nicht mehr möglich - und auch nicht Ziel der Vielzielsuche durch den Sukey-Algorithmus.

Das Beispiel in [Abbildung 3.10] wurde mit dem Simulator in [Kapitel 4] berechnet. Die Fluchtkanten  $a_{dB}, a_{BC}, a_{CD}, a_{DE}$  werden von allen denkbaren Fluchtwegen benutzt und ihr Zähler  $cnt(a_{ij})$  bei jedem erneuten Durchlauf des Sukey-Algorithmus um 1 inkrementiert (bis der Durchlauf  $e$  erreicht ist). Dadurch werden bis auf den ersten gefundenen Fluchtweg alle Fluchtwege abgewertet. Je später sie gefunden werden, desto stärker ist ihre Abwertung. Weiterhin gilt: Die Reihenfolge der gefundenen Fluchtwege beschreibt lediglich ihre *Andersartigkeit* zueinander.

1. d - B - C - D - E - G - I - L - M - (Z) = 0.33171
  2. d - B - C - D - E - F - I - K - M - (Z) = 0.00985
  3. d - B - C - D - E - G - J - L - M - (Z) = 0.0013
  4. d - B - C - D - E - F - H - K - M - (Z) = 0.0003
- die restlichen Fluchtwege 5.-12. (...)  $\approx 0.0$

**Die Fluchtweggüte**  $fwg(FW_n)$  einer Fluchtkante  $FW_n$  wird aus der Sicherheit  $sav(FW_n)$  und der Bewertung ihrer Länge  $qual(FW_n)$  ermittelt. Unpassierbare Fluchtwege mit einer Fluchtweggüte  $fwg(FW_n) \leq 0$  werden vom Suchalgorithmus nicht weiter verfolgt. Im weiteren Verlauf wird daher die Bezeichnung  $fwg(FW_n)^+$  für einen Fluchtweg verwendet.

$$fwg(FW_n) = sav(FW_n) \cdot qual(FW_n) \in [-1, 1[$$

$$fwg(FW_n)^+ : N \rightarrow ]0, 1[$$

### 3.6 Die heuristische Fluchtweggüte

Der Sukey-Algorithmus ist ein informierter Suchalgorithmus. Er nutzt die *relative heuristische Fluchtweggüte* ( $hwg$ ) als Heuristik.

Ein Fluchtweg  $FW_n$  vom Standort  $d$  bis zum Teilzielknoten  $n \in N$  eines Fluchtkantengraphen hat die *relative heuristische Fluchtweggüte*  $\widehat{hwg}(FW_n)$ . Das Ergebnis von  $\widehat{hwg}(FW_n)$  darf  $hwg(FW_n)$ , die echte Restgüte, nicht *unterschätzen*. Das nächstgelegene Ziel des Knotens  $n_j$  in  $FW_n$  ist  $z(n_j) \in T$ . Die Luftlinie von  $n_j$  nach  $z(n_j)$  wird durch Teilen durch die bereits bekannte *maxlen* zu einer relativen Länge:  $\frac{n_j z(n_j)}{maxlen}$ . Unter [Abschnitt 3.5] wird erläutert, dass

Fluchtwege mit vielen Abzweigungen und Kreuzungen bei gleicher Gesamtlänge eine bessere Güte, als die mit wenigen Alternativen haben. Für den Knoten  $n_j$  wird angenommen, dass alle anderen Knoten in gleichen Abständen auf der Luftlinie zwischen  $n_j$  und  $z(n_j)$  liegen. Dies entspricht, aus Sicht des Knotens  $n_j$ , bezüglich der Fluchtweggüte dem best-vorstellbaren Fluchtkantengraphen.

$$\widehat{hwg} : N \rightarrow ]0 \dots 1[$$

$$\widehat{hwg}(FW_n) = \left(1 - \frac{\overline{n_j z(n_j)}}{\overline{maxlen} \cdot (|A'| - 1)} \cdot (1 - m)\right)^{(|A'| - 1)}$$

stets gilt  $\widehat{hwg}(FW_n) \geq hwg(FW_n)$

$\widehat{hwg}(FW_n) = 1$  ist zulässig

Die Einschätzung einer *heuristischen Sicherheit* (und damit auch der Hindernisregeln) ist nicht möglich. Der Flaschenhals  $\min(sav(a_{ij}))$  des Fluchtweges  $fwg(FW_n)^+$  ist im besten Fall  $(1 - p) < 1$  und kommt bereits durch  $fwg(FW_n)^+$  in der Formel vor. Jede Schätzung der Sicherheit mit einem Wert  $< 1$  könnte  $hwg(FW_n)$  unterschätzen und ist daher nicht zulässig.

### 3.7 Herleitung des Sukey-Algorithmus

Die Fluchtweggüte eines Fluchtkantengraphen ist  $fwg(FW_n)^+ \in ]0 \dots 1[$  wie in [Abschnitt 3.5] gezeigt, die im besten Fall gegen 1.0 geht.

Der Sukey-Algorithmus nutzt die Formulierung des  $A^*$ -Algorithmus nach [Hart u. a. (1968)], mit der Modifizierung, dass statt der *Addition von absoluten Kosten* eine *Multiplikation von relativen Güten* angewendet wird und alle Relationen gedreht werden, damit mit relativen Güten gearbeitet werden kann ([vgl. Abschnitt 3.2]).

$A^*$ nach [Hart u. a. (1968)]	Beschreibung
$g(n)$	die optimalen Kosten nach der Suche
$\hat{g}(n)$	die optimalen Kosten während der Suche
$h(n)$	die echten Restkosten
$\hat{h}(n)$	die geschätzten Restkosten
$\hat{f}(n')$	Pfad » $P = s$ to $n$ « mit $n'$ ein Knoten in $P$
$f(s)$	der optimale Weg zum Ziel

Tabelle 3.3: Die Funktionen des  $A^*$ -Algorithmus nach [Hart u. a. (1968)]

Sukey-Algorithmus	Beschreibung
$fwg(FW_n)^+$	die beste Güte nach der Suche
$\widehat{fwg}(FW_n)^+$	die beste Güte während der Suche
$hwg(FW_n)$	die echte Restgüte
$\widehat{hwg}(FW_n)$	die geschätzte Restgüte
$\hat{f}(n')$	Pfad $P = d$ nach $t$ mit $n'$ ein Knoten in $P$
$f(t)$	der Fluchtweg mit der besten Güte zum Ziel

Tabelle 3.4: Die Funktionen des Sukey-Algorithmus

A*	Sukey-Algorithmus
$\hat{g}(n) \geq g(n)$	$\widehat{fwg}(FW_n)^+ \leq fwg(FW_n)^+$
$\hat{h}(n) \leq h(n)$	$\widehat{hwg}(FW_n) \geq hwg(FW_n)$
$\hat{f}(n') \leq f(s)$	$\hat{f}(n') \geq f(t)$
$\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$	$\hat{f}(n) = \widehat{fwg}(FW_n)^+ \cdot \widehat{hwg}(FW_n)$

Tabelle 3.5: Vergleich von A\*- und Sukey-Algorithmus. Alle weiteren Regeln des A\* bleiben für den Sukey-Algorithmus unberührt

Aus der Drehung der Relationen folgt, dass die Heuristik eines Sukey-Algorithmus niemals unterschätzen darf.

### 3.8 Status

Der *STATUS* ist eine Einschätzung der Situation an einem Standort. Ein hoher *STATUS* ist besser als ein niedriger *STATUS*. Der *STATUS* ist die Summe aller Güten der Fluchtwege  $\{FWG_t\}$ , die der Sukey-Algorithmus bis zum letzten Durchlauf, der durch die Performanz  $e$  vorgegeben wird, gefunden hat. Bis zum letzten Durchlauf startet die Fluchtwegsuche mit abgewerteten Fluchtkanten der bereits gefundenen Fluchtwege immer wieder von neuem. Da es vorkommen kann, dass gleiche Fluchtwege gefunden werden (die Abwertung von Fluchtkanten verhindert dies nicht), wird die Einsortierung von Fluchtwegen, die in  $\{FWG_t\}$  bereits vorkommen, nicht zugelassen. Doppelt gefundene Fluchtwege werden also ungeachtet

des Durchlaufzählers im Vorfeld aussortiert. Je höher die Performanz  $e$  eingestellt ist, desto genauer und positiver schätzt der *STATUS* die Situation ein. Für den *STATUS* gilt:

$$STATUS = \sum_{g \in \{FWG_t\}} g \in [0 \dots |T|]$$

## 4 Implementierung des Sukey-Algorithmus

In diesem Kapitel wird der Sukey-Algorithmus in einem Simulator implementiert<sup>1</sup>. Für die Beschreibung des Fluchtkantengraphen in einer Texteingabe wird eine Grammatik vorgestellt. Der Simulator umfasst zusätzlich einen *Compiler* zur Übersetzung der Grammatik in eine Kommandoliste und eine *Benutzeroberfläche* zur Eingabe und Darstellung der Ergebnisse.

### 4.1 Aufbau des Simulators

Um *Compiler* und *Fluchtkantengraphen* klar voneinander zu trennen, übernimmt eine *Datenaufbereitung* die Steuerung beider Komponenten. Die Datenaufbereitung aktiviert den Fluchtkantengraphen zur Ausführung des Sukey-Algorithmus und reicht die Ergebnisse an die *Benutzeroberfläche* zur Darstellung weiter. Der Simulator besteht aus vier Komponenten:

- Fluchtkantengraph  
Der Fluchtkantengraph berechnet mithilfe des *Sukey-Algorithmus* den Status und den besten Fluchtweg.
- Compiler  
Der Compiler erstellt aus Texteingaben Kommandos für den Fluchtkantengraphen.
- Benutzeroberfläche  
Die Benutzeroberfläche stellt in einer Grafik den Fluchtkantengraphen dar. In der Benutzeroberfläche können die Parameter *Mobilität der Personen*, *Anzahl der Personen* und die *Performanz* für den Fluchtkantengraphen eingestellt werden. In der Benutzeroberfläche wird eine Texteingabe für den Compiler bereitgestellt und die Ergebnisse und eventuelle Fehler beim Kompilieren werden dargestellt.
- Datenaufbereitung

---

<sup>1</sup>Die für die Implementierung verwendete Programmiersprache ist *REALbasic 2008* (REAL Software, Inc. - <http://www.realbasic.com>).



Die Datenaufbereitung schickt an den Compiler Texteingaben zur Übersetzung in Kommandos für den Fluchtkantengraphen. Sie nimmt die Kommandos vom Compiler entgegen und erstellt mit ihrer Hilfe den Fluchtkantengraphen in der Komponente *Fluchtkantengraph*. Sie leitet die Ergebnisse an die graphische Oberfläche zur Darstellung weiter.

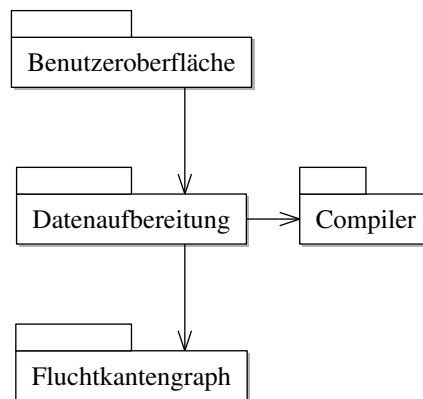


Abbildung 4.1: Zusammenspiel der Komponenten des Simulators

Die vorliegende Arbeit beschreibt im weiteren Verlauf die Grammatik und vertieft die Komponente *Fluchtkantengraph*.

## 4.2 Die Grammatik des Fluchtkantengraphen

Zur Ausführung des Sukey-Algorithmus wird der Fluchtkantengraph für den Simulator mithilfe einer einfachen Grammatik beschrieben. Die in der vorliegenden Arbeit verwendete LL(k)-Grammatik wurde vom Autor für eine vereinfachte Beschreibung von Fluchtkantengraphen entwickelt. Ihre Ableitungsschritte lauten:

Identifikator	Beschreibung
letter	= [a-zA-Z]
digit	= [0-9]
eol	= [ \n\r ]
word	= letter {word}.
integer	= digit {digit}.
single	= { integer ('   ') integer }.
goal	= 'z'.
position	= '(' single ';' single ')'
node	= letter position [goal].
arc	= ( position '-' position single   letter '-' letter '(' single ')') .
obstacle	= '+' position single ['!'].
device	= '! position.
nodelist	= [node {eol nodelist} eol].
arclist	= [arc {eol arc} eol].
obstaclelist	= [obstacle {eol obstacle} eol].
protocol	= nodelist arclist obstaclelist device.

Tabelle 4.1: Ableitungsschritte der Grammatik für die Beschreibung des Fluchtkantengraphen

Mithilfe dieser Grammatik lässt sich ein Fluchtkantengraph beschreiben. Ein *Compiler* erzeugt aus dieser Eingabe (einer Zeichenfolge die sich an die beschriebene Grammatik hält) Kommandos für die *Datenaufbereitung* zum Aufbau des Fluchtkantengraphen. Zum besseren Verständnis werden in [Tabelle 4.2] Beispiele für die sich aus der Grammatik ergebenden Kommandos gezeigt.

Beispielzeile	Beschreibung
A(10;10)	Knoten A an Position (10,10)
B(40;20)	Knoten B
C(40;40)z	Zielknoten C
A-B(0.8)	Kante A nach B mit der Sicherheit 0.8
(40;20)-(40;40)0.1	Kante der Knoten an (40,20) (B) und (40,40) (C) mit der Sicherheit 0.1
+(60;60)0.4	Hinzufügen eines Hindernisses an Position (60,60) mit dem Wert 0.4
+(29;34)0.4!	Hinzufügen eines Hindernisses auf die nächste Kante bei (29,34)
!(20;20)	Der Standort befindet sich an Position (20,20)

Tabelle 4.2: Beispiele für die Beschreibung eines Fluchtkantengraphen mithilfe der vorgestellten Grammatik

### 4.3 Die Klassen des Fluchtkantengraphen

Die Komponente *Fluchtkantengraph* hält alle Referenzen auf Knoten, Kanten, Hindernisse und Standort und organisiert ihr Zusammenwirken. Nur über die Methoden der Komponente *Fluchtkantengraph* kann von außen zugegriffen werden.

<b>module:Fluchtkantengraph</b>
<pre>private actionIndex as Arc private arcs() as Arc private nodes() as Node private obstacles() as Obstacle private device as Device</pre>
<pre>addArc(x1 as Single, x2 as Single, y2 as Single, sav as Single) addNode(letter as String, x as Single, y as Single, goal as Boolean) addObstacle(x as Single, y as Single, value as Single, onedge as Boolean) moveDevice(x as Single, y as Single) removeArc(x as Single, y as Single) removeNode(x as Single, y as Single) removeObstacle(x as Single, y as Single) clearGraph() calculate() getBestFW() as String getStatus() as Single</pre>

Tabelle 4.3: Das Modul Fluchtkantengraph

Die Methoden der Komponente *Fluchtkantengraph* werden von der Komponente *Datenaufbereitung* aufgerufen. Nach dem Aufruf der Methode *calculate()*, welche den Sukey-Algorithmus auf dem Fluchtkantengraphen ausführt, wird das Ergebnis über *getBestFW()* und *getStatus()* abgerufen. Für einen rekursiven Zugriff auf den Fluchtkantengraphen sind Knoten und Kanten doppelt verknüpft. Ein Aktionszähler (*actionIndex*) verhindert dabei, dass Schleifen auftreten. Für einen iterativen Zugriff werden die Referenzen von Knoten, Kanten und Hindernissen in der Komponente zusätzlich in einfach indizierten Arrays gehalten.

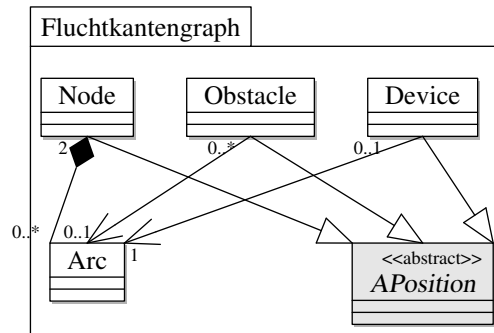


Abbildung 4.2: Die Komponente Fluchtkantengraph.

Die Klassen der Komponente *Fluchtkantengraph* werden aus der formalen Beschreibung des Fluchtkantengraphen aus [Abschnitt 3.3] hergeleitet.

#### 4.3.1 Die Position

Die Position von Objekten im Fluchtkantengraphen.

<b>class:APosition</b>
x as Single y as Single
constructor(x as Single, y as Single) getX() as Single getY() as Single posEquals(pos as APosition) as Boolean

Tabelle 4.4: Die abstrakte Klasse APosition

Knoten, Hindernisse und das Gerät am Standort erben von der abstrakten Klasse *APosition* die Position  $K \rightarrow \mathbb{R} \times \mathbb{R}$  wie verlangt. Für  $\mathbb{R}$  wird der Datentyp *Single* verwendet. Um sicherzustellen, dass zwei Objekte, die von *APosition* erben, paarweise disjunkt sind, gibt die Methode *posEquals(pos as APosition)* ( $self \cap pos = \emptyset$ ) mit  $self, pos \in K$  einen *Boolean*-Wert zurück. Die Injektivität (Linkseindeutigkeit) ist gegeben.

#### 4.3.2 Die Knoten

Knoten im Fluchtkantengraphen sind Abzweigungen oder Kreuzungen mit einer exakten Position. Für die Implementierung im Simulator erhalten die Knoten eine Bezeichnung (*letter*),

welche keine Auswirkung auf den Sukey-Algorithmus hat, sondern der besseren graphischen Darstellung des Fluchtkantengraphen und vereinfachten Beschreibung einer Kante in der Grammatik dient.

<b>class:Node extends APosition</b>
letter as String goal as Boolean nextGoalsDistance as Single arcs() as Arc pre as Node
constructor(letter as String, x as Single, y as Single, goal as Boolean = False) equals(other as Node) as Boolean expand(openList as NodeQueue, closedList as NodeQueue)

Tabelle 4.5: Die Klasse Node

Die Berechnung der Distanz *nextGoalsDistance* jedes Knotens zum nächst gelegenen Ziel wird nur einmal am Anfang, wenn der Fluchtkantengraph vollständig aufgebaut ist, oder wenn Änderungen an Knoten ausgeführt werden, vorgenommen. Jeder Knoten führt die Referenzen auf seine Kanten in einem Array mit sich. Jeder Knoten kann auf einen Vorgänger verweisen, damit der Fluchtweg vom Ziel bis zum Start zurückverfolgt werden kann. Die Methode *expand(openList as NodeQueue, closedList as NodeQueue)* berechnet ihre Nachfolgeknoten, fügt sie in die *openList* und sich selbst in die *closedList* ein. Die entwickelte Klasse *NodeQueue* ist eine sortierte Liste für die Knoten.

### 4.3.3 Die Fluchtkanten

Fluchtkanten im Fluchtkantengraphen sind Straßen mit einer Länge *len* und einer Sicherheit *sav* zwischen zwei Knoten.

<b>class:Arc</b>
nodeA as Node nodeB as Node sav as Single len as Single cnt as UInt16 = 1
constructor(nodeA as Node, nodeB as Node, sav as Single) calcLen(maxlen as Single) incCnt() resetCnt()

Tabelle 4.6: Die Klasse Arc

Jede Kante hat exakt zwei Knoten (*nodeA* und *nodeB*). Das heißt, kein übergebener Knoten darf `nil`<sup>2</sup> sein. Außerdem dürfen die beiden übergebenen Knoten nicht identisch sein (*not nodeA.equals(nodeB)*). Die Länge *len* kann erst nach Ermittlung von *maxlen* erfolgen. Dies ist erst nach Erstellung des kompletten Fluchtkantengraphen möglich und wird dann einmal durch `calcLen(maxlen as Single)` durchgeführt. *cnt* ist ein `UInt16`-Wert und wird mit 1 vorinitialisiert. `resetCnt()` setzt *cnt* auf 1 zurück, `incCnt()` inkrementiert *cnt* um 1.

#### 4.3.4 Die Hindernisse

Hindernisse im Fluchtkantengraphen werden als Objekte mit einer Intensität modelliert.

<b>class:Obstacle extends APosition</b>
value as Single arc as Arc
constructor(x as Single, y as Single, value as Single, arc as Arc = nil)

Tabelle 4.7: Die Klasse Obstacle

Hindernisse können überall auftreten. Die Hindernisregeln aus [Abschnitt 3.4] *dhr* und *ihr* unterscheiden zwischen Hindernissen auf Kanten und Hindernissen neben Kanten. Soll ein Hindernis auf einer Kante liegen, wird sie im Konstruktor mit übergeben. Hindernisse werden im Fluchtkantengraphen in einer separaten Hindernis-Liste geführt.

---

<sup>2</sup>Aus der REALbasic-Sprachreferenz *Nil Datatype*: »Nil means no value. When an object is first instantiated, it is automatically initialized to Nil.«

### 4.3.5 Der Standort

Der Standort ist ein technisches Gerät mit einer Performanz und einem Personenaufkommen mit deren Mobilität. Der Standort liegt immer auf einer Kante.

<b>class:Device extends APosition</b>
arc as Arc
performance as UInt16
crowd as Single
mobility as Single
abstract Create() as Device
setPosition(x as Single, y as Single, arc as Arc)

Tabelle 4.8: Die Klasse Device

Die Performanz  $e$  wird für die Implementierung bei 16-Bit (65.535) begrenzt. Vom Standort gibt es nur eine Instanz. Da der Standort von *APosition* ableiten soll und zur Einhaltung einer geringen Kopplung ist der Standort kein *Singleton*, sondern eine Instanzvariable im Fluchtkantengraphen. Seine Einmaligkeit im System wird durch Überprüfung bereits instantiiert Standorte in der Fabrikmethode *Create* geprüft und bei Fehlbedienung das *nil*-Objekt zurückgeliefert.

Für einen Weg vom Standort  $d$  zu einem Teilziel  $n_j$  im Fluchtkantengraphen wird die Fluchtkante  $a_{ij} \in A : pos(d)$  auf  $a_{ij}$  unterbrochen. In der praktischen Umsetzung wird kein zweiter Fluchtkantengraph  $F'$  angelegt, sondern der erste zeitlich begrenzt verändert.  $N' = \{n_i \mid 1 \leq i \leq k\} \in N \cup d$  verlangt das Hinzufügen eines Knotens mit den Koordinaten des Standortes (einen *Standort-Knoten*). Dies ist ein zeitlich begrenzter Bruch mit der Regel  $K = N \cup H \cup \{d\} : paarweise\ disjunkt$ , der aber zur Optimierung in Kauf genommen wird. Wie durch  $A' = A \setminus \{a_{ij}\} \cup \{a_{id}, a_{dj}\}$  (gekürzt) beschrieben, wird die ursprüngliche Fluchtkante entfernt. In der Praxis werden die Bezüge ihrer Knoten zu ihr entfernt. Gleichzeitig werden ihre Knoten durch zwei neue Kanten mit der Sicherheit  $sav(a_{ij})$  über den neuen *Standort-Knoten* miteinander verbunden. Der neugeschaffene *Standort-Knoten* ist zugleich der Startknoten für die Durchführung des Sukey-Algorithmus. Terminiert der Sukey-Algorithmus, wird der *Standort-Knoten* samt der Kanten  $a_{id}, a_{dj}$  entfernt und der ursprünglichen Kante  $a_{ij}$  ihre alten Knoten zurückgegeben. Der Fluchtkantengraph  $F$  ist dann wieder in seinem ursprünglichen Zustand.



## 4.4 Ausblick - Kurvenführung

Dieser Abschnitt ist ein Ausblick auf das Problem der Kurvenführung. Der in der vorliegenden Arbeit entwickelte Simulator eines Sukey-Algorithmus berücksichtigt dieses Problem nur teilweise, da es erst bei der Entwicklung eines Fluchtkantengraphen direkt aus einem Stadtplan heraus auftritt. Sollte ein *Editor für Fluchtkantengraphen*, auf dem die Kanten an den Straßen entlanggeführt und um Kurven herumgeführt werden, entwickelt werden, ist dieses Problem von Bedeutung.

Die in der Evaluation in [Kapitel 5] gewählten Beispiele sind so gewählt, dass sie ohne Berücksichtigung des Problems der Kurvenführungen auskommen und dort also keine groben Verfälschungen im Ergebnis auftreten werden. Die einzige Ausnahme ist das Interview 8 [Tabelle 5.9], das einen Knoten mit einem Knotengrad = 2 beinhaltet. Dies hat jedoch keine gravierende Auswirkung auf das Ergebnis.

### 4.4.1 Beschreibung von Kurven

Wie im [Abschnitt 3.3] ausgeführt, haben Knoten im Fluchtkantengraphen einen Knotengrad ungleich 2. Hintergrund ist die im [Abschnitt 3.5] gezeigte - in der [Abbildung 3.7] dargestellte - Aufwertung von Fluchtwegen mit *vielen Ausweichmöglichkeiten*. Ein Fluchtweg mit einem zusätzlichen Knoten mit einem Knotengrad gleich 2 hat dadurch eine höhere Güte als ein Fluchtweg gleicher Länge mit keinem zusätzlichen Knoten, wie in [Abbildung 4.3] dargestellt. Dies führt zu einer nicht gewünschten Aufwertung und wurde daher von vornherein ausgeschlossen.

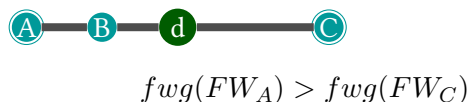
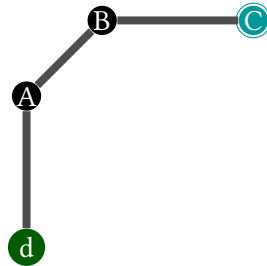


Abbildung 4.3: Ungewünschte Aufwertung des Fluchtweges mit einem zusätzlichen Knoten mit einem Knotengrad gleich 2

Für einen Fluchtkantengraphen, der auf einem Stadtplan basiert, sind Knotengrade gleich 2 möglich, wenn ein Weg um eine langgezogene Kurve führt. Die Kurvenführung soll nicht durch den direkten Weg abgekürzt werden, weil die Simulation auf einem Stadtplan basieren und die Bewegung des Standortes auf den Straßen entlangführen - und somit auch um Kurven herumführen - soll. Eine Kurve im Fluchtkantengraphen würde dann, wie in [Abbildung 4.4] dargestellt, durch die Aufeinanderfolge von Knoten mit je einem Knotengrad gleich 2 gebildet.

Der Einfluss der Sicherheit  $sav(FW_C)$ , welche der Flaschenhals ist, bleibt von Knoten mit Knotengraden gleich 2 unbeeindruckt, weil sie unabhängig von der Länge ist. Die Qualität der Länge jedoch wird verfälscht. Die Lösung liegt für den Sukey-Algorithmus darin, Knoten mit Knotengraden = 2 solange zu expandieren, bis ein Knoten mit einem Knotengrad  $\neq 2$  gefunden wird oder man sich im Kreis gedreht hat.



$$\text{sei } a_{curve} = \{a_{dA}, a_{AB}, a_{BC}\}$$

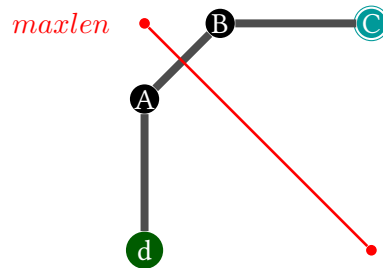
$$qual(FW_C) = \frac{1 - (\text{len}(a_{dA}) + \text{len}(a_{AB}) + \text{len}(a_{BC})) \cdot (1 - m)}{\text{cnt}(a_{dA}) + \text{cnt}(a_{AB}) + \text{cnt}(a_{BC})} \cdot |a_{curve}|$$

Abbildung 4.4: Berechnung von  $qual(FW_C)$  um eine Kurve

In der Praxis bedeutet das, dass entweder der Sukey-Algorithmus den Nachfolgeoperator  $\Gamma$  solange *ungeachtet der Open-List* auf die Knoten  $n_q$  eines Knotens  $n_p$  anwendet, bis er terminiert oder der Knotengrad von  $n_p$  ungleich 2 ist, oder dass vor dem Ausführen des Sukey-Algorithmus alle Kurven im Fluchtkantengraphen durch Kanten mit der Kurve entsprechendem  $len$  und  $sav$  ersetzt und nach der Berechnung wieder eingefügt werden. In beiden Fällen werden parallele Kanten - also Kanten, die dieselben Knoten verbinden - möglich.

#### 4.4.2 Auswirkung von Kurven auf die maximale Länge

Im [Abschnitt 3.3] ist die *maximale Länge*  $maxlen$ , die der Bezug für die relativen Längen des Fluchtkantengraphen ist,  $maxlen = \text{Diagonale des Graphen}$ . Durch die Kurvenführung kann eine (Kurven-) Fluchtkante länger sein als die Diagonale des Graphen wie in [Abbildung 4.5] veranschaulicht.



$$\text{len}(a_{\text{curve}}) = \text{len}(a_{dA}) + \text{len}(a_{AB}) + \text{len}(a_{BC}) > 1$$

Abbildung 4.5: Darstellung des Falls  $\text{len}(a_{\text{curve}}) > 1$

Um diesen Fall auszuschließen, wird die Diagonale mit der längsten Kurve (deren absolutem Wert) vor Durchführung der Simulation verglichen. Für *maxlen* wird der größere der beiden Werte angenommen.  $\text{len}(a_{\text{curve}}) > 1$  ist auf diese Weise unmöglich.

Um die Länge jeder Kurve zu berechnen, werden in der praktischen Umsetzung alle Knoten mit nur 2 Kanten betrachtet und auf jedem dieser Kurvenknoten eine rekursive Funktion aufgerufen, die die Länge seiner gemeinsam gebildeten Kurve wiedergibt. Dabei wird der Aktions-Zähler (*Fluchtkantengraph.actionIndex*) durchgereicht, um doppelte Abfragen und somit Schleifen zu verhindern. Ist die längste der ermittelten Kurven länger als das zuvor errechnete *maxlen*, wird dieses durch den neuen Wert ersetzt.

Da das Interview 8 [Tabelle 5.9] genau dieses Problem hat, ist die Suche nach möglichen Kurven, die länger als die Diagonale des Fluchtkantengraphen sind, im Simulator implementiert.

## 5 Evaluation

In diesem Kapitel werden die Ausgaben des Sukey-Algorithmus den Einschätzungen aus einem Online-Interview gegenübergestellt. Anschließend werden die Einstellungen der Parameter *Anzahl der Personen* und *Mobilität der Personen* am Standort diskutiert.

### 5.1 Die Interviews

In einem Online-Interview wurden 23 Interview-Partnern Fluchtkantengraphen gezeigt und zu jedem je 2 Fragen gestellt, die auch der Sukey-Algorithmus beantwortet. Die gestellten Fragen lauteten:

1. »Welchen Weg würdest du nehmen, wenn du fliehen müsstest?«

Die Antworten bestanden aus den Buchstaben der Knoten des bevorzugten Fluchtweges.

2. »Wie schätzt du deine Lage ein?«

Geantwortet wurde mit vorgegebenen Einschätzungen von »sehr gut«, »gut«, »mittel«, »schlecht« bis »sehr schlecht«. Die Einschätzung »sehr schlecht« entspricht dabei einer Einkesselung.

Den Interview-Partnern wurde es freigestellt, Fluchtkantengraphen zu bewerten oder sie zu überspringen. Zusätzlich wurde die Anordnung der Interviews bei jedem Interviewten zufällig gewählt. Daher gibt es pro Fluchtkantengraphen eine unterschiedliche Anzahl von Interview-Partnern.

Zusätzlich wurde den Interview-Partnern mitgeteilt, dass sich außer ihnen auch noch ca. 1000 andere Personen am Standort befinden, die sich an ihren Weisungen orientieren werden. Der Parameter *Anzahl an Personen* des Fluchtkantengraphen wird für den Sukey-Algorithmus auf  $p = 0.2$  eingestellt. Für die *Mobilität der Personen* wird für den Sukey-Algorithmus  $m = 0.5$  und die *Performanz*  $e = 10$  eingestellt. Diesen Werten liegen Erfahrungen bei der Entwicklung zugrunde.

Ziel dieser Interviews ist es herauszufinden, ob der in der vorgelegten Arbeit entwickelte Sukey-Algorithmus nach menschlicher Einschätzung nachvollziehbare Ergebnisse liefert. Dabei

wurden die Fluchtkantengraphen so entwickelt, dass sie auf die korrekte Behandlung folgender Themen durch den Sukey-Algorithmus eine Antwort geben.

1. Wahl des sichersten Fluchtweges
2. Direkte Hindernisregel (dhr) [vgl. Abschnitt 3.4.1]
3. Indirekte Hindernisregel (ihr) [vgl. Abschnitt 3.4.2]
4. Bevorzugung von Fluchtwegen mit zusätzlichen Ausweichmöglichkeiten [vgl. Abbildung 3.7]
5. Mehrfaches Benutzen gleicher Fluchtkanten bei unterschiedlichen Fluchtwegen [vgl. Abbildung 3.10]

Den Interview-Partnern wurde folgende Legende für die Fluchtkantengraphen gegeben.

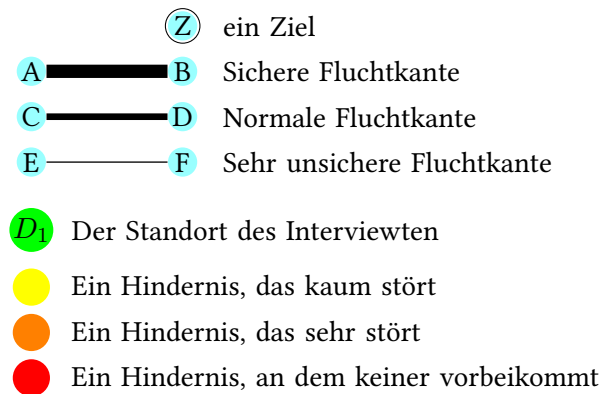


Abbildung 5.1: Interview - Legende des Fluchtkantengraphen für die Interview-Partner

*Die Interviews sind anonym und die Ergebnisse sind weder repräsentativ, noch werden sie in der vorliegenden Arbeit stochastisch aufbereitet. Sie können lediglich zeigen, dass der Sukey-Algorithmus die Situation am Standort ungefähr wie ein Mensch einschätzt.*

### 5.1.1 Vorstellung der Ergebnisse

In diesem Abschnitt werden die Ergebnisse der Interviews vorgestellt. Zur Gegenüberstellung wird das Resultat, zu dem der in der vorliegenden Arbeit implementierte Simulator des Sukey-Algorithmus kommt, mit angeführt und der Vergleich diskutiert. Die Interview-Partner stuften ihre Einschätzungen für die Situation am Standort von »sehr gut« bis »sehr schlecht« ein. Der Sukey-Algorithmus gibt einen Status wieder, wie in [Abschnitt 3.8] beschrieben.

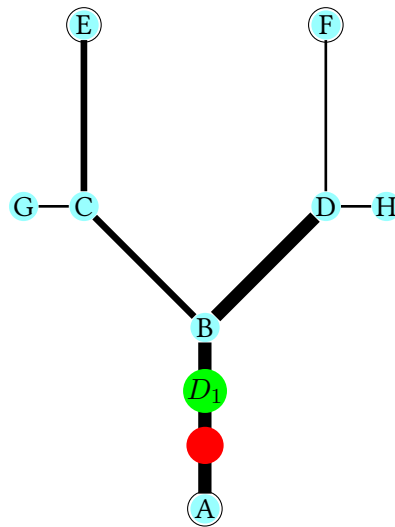
Für die Interviews können die Einschätzungen für die Werte des Status [Tabelle 5.1] entnommen werden. Der Bezug zwischen Einschätzung und zugehörigem Wert des Status ist eine Schätzung des Autors auf Basis seiner Erfahrung mit der Entwicklung des Sukey-Algorithmus und den durchgeführten Interviews.

Einschätzung	<i>STATUS</i>
sehr schlecht	= 0
schlecht	]0, 0.1[
mittel	[0.1, 0.3[
gut	[0.3, 0.7[
sehr gut	≥ 0.7

Tabelle 5.1: Interviews - Einschätzung der Variable *STATUS*

### Interview - Fluchtkantengraph 1

Wahl des sichersten Fluchtweges



#### Ergebnis des Interviews

Anzahl Interviewte: 18

#### Bewertung

sehr gut:	3x	(17%)
gut:	11x	(61%)
mittel:	3x	(17%)
schlecht:	1x	(6%)
sehr schlecht:	0x	(0%)

#### Bester Fluchtweg

BCE:	17x	(94%)
BDF:	1x	(6%)

#### Ergebnis des Sukey-Algorithmus

Status: 0.1978706 (mittel)

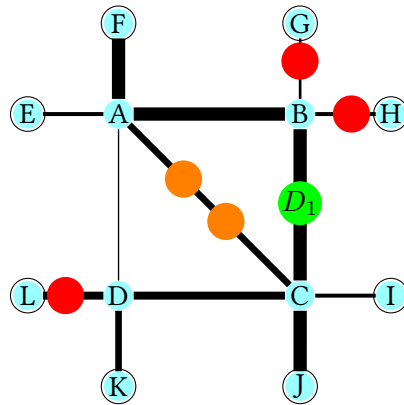
Bester Fluchtweg: BCE

Der Sukey-Algorithmus empfiehlt wie der große Teil der Interviewten den Fluchtweg über BCE und schätzt die Situation am Standort etwas schlechter als die Interviewten auf *mittel* ein. Der Engpass der Kante D-F wird wie gewünscht erkannt und gemieden.

Tabelle 5.2: Interview - Fluchtkantengraph 1

**Interview - Fluchtkantengraph 2**

Wahl des sichersten Fluchtweges



*Ergebnis des Interviews*

Anzahl Interviewte: 19

*Bewertung*

sehr gut:	14x	(74%)
gut:	2x	(11%)
mittel:	0x	(0%)
schlecht:	0x	(0%)
sehr schlecht:	3x	(16%)

*Bester Fluchtweg*

CJ:	14x	(74%)
CI:	2x	(11%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.6586208 (gut)

Bester Fluchtweg: CJ

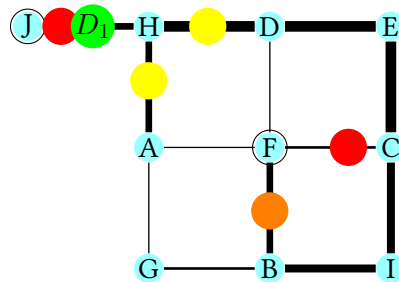
Wie die meisten Interview-Partner schätzt der Sukey-Algorithmus die Situation am Standort als *gut* bis *sehr gut* ein. Ebenso empfiehlt er den Fluchtweg CJ.

Tabelle 5.3: Interview - Fluchtkantengraph 2



**Interview - Fluchtkantengraph 3**

Direkte Hindernisregel

*Ergebnis des Interviews*

Anzahl Interviewte: 19

*Bewertung*

sehr gut:	0x	(0%)
gut:	1x	(5%)
mittel:	7x	(37%)
schlecht:	8x	(42%)
sehr schlecht:	3x	(16%)

*Bester Fluchtweg*

HDF:	11x	(58%)
HDECIBF:	2x	(11%)
HDECF:	2x	(11%)
HAF:	1x	(5%)

*Ergebnis des Sukey-Algorithmus*

Status: 0 (sehr schlecht)

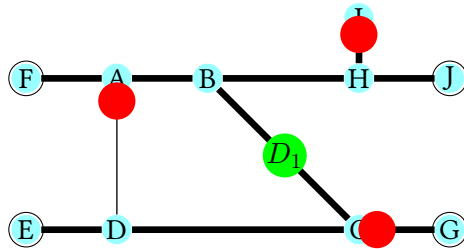
Bester Fluchtweg: -

Der Sukey-Algorithmus empfiehlt keinen Fluchtweg, da die Fluchtwege alle zu unsicher sind. Würde die *Anzahl an Personen*  $p = 0.001$  (also sehr gering) eingestellt, so empfiehlt der Sukey-Algorithmus die Flucht über den Fluchtweg HAGBF. Trotz Hindernis auf der Fluchtkante B-F ist dieser Weg sicherer, als die anderen Fluchtkanten nach Knoten F. Wie gewünscht, weil sich zwischen dem Standort und dem Knoten J ein unüberwindbares Hindernis befindet, wird der Zielknoten J nicht in Betracht gezogen.

Tabelle 5.4: Interview - Fluchtkantengraph 3

**Interview - Fluchtkantengraph 4**

Indirekte Hindernisregel



*Ergebnis des Interviews*

Anzahl Interviewte: 18

*Bewertung*

sehr gut:	0x	(0%)
gut:	2x	(11%)
mittel:	7x	(39%)
schlecht:	6x	(33%)
sehr schlecht:	3x	(17%)

*Bester Fluchtweg*

BHJ:	11x	(61%)
BAF:	3x	(17%)
CG:	1x	(6%)
BHI:	1x	(6%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.0033743 (schlecht)

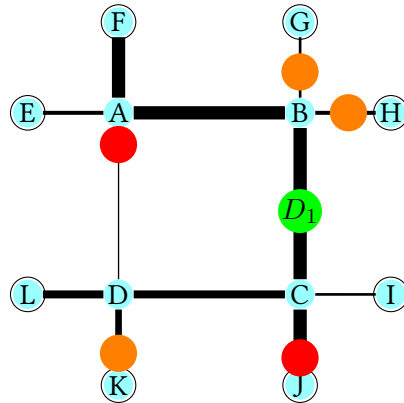
Bester Fluchtweg: BHJ

Der Sukey-Algorithmus schätzt die Situation am Standort wie viele Interviewte als *schlecht* ein. Da das Hindernis am Knoten H eine größere Distanz zum Fluchtweg hat als die Hindernisse an den Knoten A und C, empfiehlt der Sukey-Algorithmus, wie die meisten Interviewten, eine Flucht über den Fluchtweg BHJ.

Tabelle 5.5: Interview - Fluchtkantengraph 4

**Interview - Fluchtkantengraph 5**

Indirekte Hindernisregel



*Ergebnis des Interviews*

Anzahl Interviewte: 17

*Bewertung*

sehr gut:	1x	(6%)
gut:	4x	(24%)
mittel:	10x	(59%)
schlecht:	1x	(6%)
sehr schlecht:	1x	(6%)

*Bester Fluchtweg*

CDL:	4x	(24%)
BH:	4x	(24%)
CI:	4x	(24%)
BAF:	3x	(18%)
BAE:	1x	(6%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.03455 (schlecht)

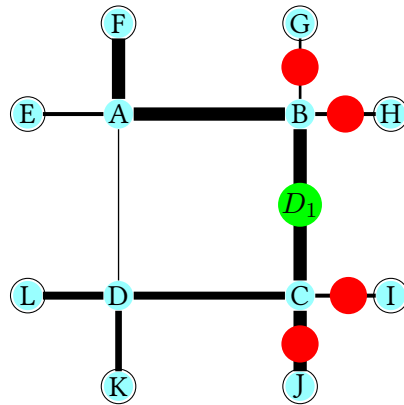
Bester Fluchtweg: CDL

Der Sukey-Algorithmus schätzt die Situation am Standort als *schlecht* ein. Er empfiehlt den Fluchtweg, dessen Knoten die größten Abstände zu den Hindernissen haben.

Tabelle 5.6: Interview - Fluchtkantengraph 5

### Interview - Fluchtkantengraph 6

Indirekte Hindernisregel



#### Ergebnis des Interviews

Anzahl Interviewte: 17

#### Bewertung

sehr gut:	0x	(0%)
gut:	3x	(18%)
mittel:	3x	(18%)
schlecht:	8x	(47%)
sehr schlecht:	3x	(18%)

#### Bester Fluchtweg

BAF:	12x	(71%)
CDL:	2x	(12%)

#### Ergebnis des Sukey-Algorithmus

Status: 0.1220978 (mittel)

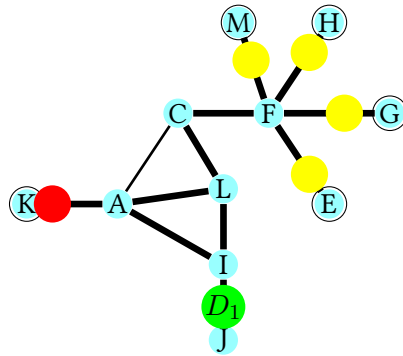
Bester Fluchtweg: CDL

Sukey-Algorithmus und Interviewte schätzen die Situation am Standort als *mittel* bis *schlecht* ein. Der Sukey-Algorithmus wählt den Fluchtweg - anders als die meisten Interviewten - über den Knoten C, weil die Hindernisse auf den Kanten C-I und C-J ein kleines Stück weiter von C entfernt sind, als die auf der gegenüberliegenden Seite. Das war für die Interviewten schwer zu erkennen.

Tabelle 5.7: Interview - Fluchtkantengraph 6

## Interview - Fluchtkantengraph 7

Indirekte Hindernisregel

*Ergebnis des Interviews*

Anzahl Interviewte: 19

*Bewertung*

sehr gut:	2x	(11%)
gut:	7x	(37%)
mittel:	6x	(32%)
schlecht:	2x	(11%)
sehr schlecht:	2x	(11%)

*Bester Fluchtweg*

ILCFM:	10x	(53%)
ILCFG:	4x	(21%)
ILCFE:	2x	(11%)
ILCAK:	1x	(5%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.0975252 (schlecht)

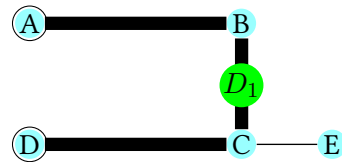
Bester Fluchtweg: **ILCFE**

Der Sukey-Algorithmus schätzt die Situation am Standort schlechter ein als die meisten Interviewten. Außerdem entscheidet er sich für den Fluchtweg zum Zielknoten E, obwohl der Zielknoten M offensichtlich die bessere Alternative ist, weil F-M leicht kürzer ist als F-E. Hier verschätzt sich der Sukey-Algorithmus. Die Hindernisse in dem von der indirekten Hindernisregel [Abschnitt 3.4.2] um den Knoten E gelegten Zirkel mit dem Radius  $\overline{D_1E}$  haben einen weniger großen Einfluss auf die Sicherheit als die Hindernisse in dem um den Knoten M gelegten Zirkel mit dem Radius  $\overline{D_1M}$ . Wegen der indirekten Hindernisregel (ihr) empfiehlt der Sukey-Algorithmus daher den Fluchtweg über ILCFE. Befände sich der Standort auf der Kante C-F, so ist in der Simulation der Vorschlag des Sukey-Algorithmus in der Tat der Fluchtweg FM.

Tabelle 5.8: Interview - Fluchtkantengraph 7

**Interview - Fluchtkantengraph 8**

Bevorzugung von Fluchtwegen mit zusätzlichen Ausweichmöglichkeiten

*Ergebnis des Interviews*

Anzahl Interviewte: 17

*Bewertung*

sehr gut:	16x	(94%)
gut:	0x	(0%)
mittel:	0x	(0%)
schlecht:	0x	(0%)
sehr schlecht:	1x	(6%)

*Bester Fluchtweg*

CD:	8x	(47%)
BA:	8x	(47%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.9917356 (sehr gut)

Bester Fluchtweg: CD

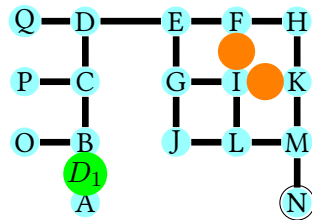
Der Sukey-Algorithmus schätzt die Lage am Standort als *sehr gut* ein wie die große Zahl der Interviewten. Der Fluchtweg über Knoten C wird gewählt, weil hier eine *zusätzliche Ausweichmöglichkeit* besteht.

In diesem Interview hat der Knoten B einen Knotengrad = 2. Zusätzlich ist die Diagonale des Fluchtkantengraphen *maxlen* kürzer als die Kurve  $D_1BA$ . Da der implementierte Simulator Kurven bei der Berechnung von *maxlen* schon berücksichtigt, jedoch keine Kurven bei der Wegsuche berücksichtigt, ist der Status lediglich leicht niedriger als hier angegeben (bleibt aber bei »sehr gut«).

Tabelle 5.9: Interview - Fluchtkantengraph 8

**Interview - Fluchtkantengraph 9**

Mehrfaches Benutzen gleicher Fluchtkanten bei unterschiedlichen Fluchtwegen

*Ergebnis des Interviews*

Anzahl Interviewte: 17

*Bewertung*

sehr gut:	5x	(29%)
gut:	7x	(41%)
mittel:	4x	(24%)
schlecht:	0x	(0%)
sehr schlecht:	1x	(6%)

*Bester Fluchtweg*

BCDEGJLMN:	13x	(76%)
BCDEFHKMN:	2x	(12%)
ABCDEFHKMN:	1x	(6%)

*Ergebnis des Sukey-Algorithmus*

Status: 0.0203146 (schlecht)

Bester Fluchtweg: BCDEGJLMN

Wie der größte Teil der Interviewten empfiehlt auch der Sukey-Algorithmus den Fluchtweg mit dem größten Abstand zu den beiden Hindernissen. Die Situation am Standort hingegen wird vom Sukey-Algorithmus sehr viel schlechter eingeschätzt. Der Grund ist der, dass alle Fluchtwegen die Hälfte der Strecke über dieselben Fluchtkanten verlaufen [vgl. Abbildung 3.10].

Tabelle 5.10: Interview - Fluchtkantengraph 9

**5.1.2 Zusammenfassung - Interviews**

Der oben angeführte Vergleich der Ergebnisse der Interviews mit den Ergebnissen des Sukey-Algorithmus zeigt, dass der in der vorliegenden Arbeit entwickelte Sukey-Algorithmus nicht gravierend von menschlichen Einschätzungen abweicht. Die Fluchtwegempfehlungen entsprechen zum großen Teil denen der Interview-Partner.

Der Sukey-Algorithmus schätzt die Situationen an den Standorten viel kritischer ein als die Interviewten es tun. Dies kann daran liegen, dass die Interviewten über die Länge des

Interviews wahrscheinlich vergaßen, dass ihnen die in der Situationsbeschreibung vorausgesetzte 1000 weitere Personen folgten, die Lage also zusätzlich erschwert war. Zusätzlich sind die Fluchtkantengraphen mit oft nur einem Zielknoten so gewählt, dass die Einschätzung der Situation am Standort schlecht ausfallen musste.

## 5.2 Parametereinstellungen

Die Parameter des Fluchtkantengraphen, die den Sukey-Algorithmus steuern, sind *Anzahl der Personen*, *Mobilität der Personen* und die *Performanz*. Die Anzahl der Personen und die Mobilität der Personen sind relative Parameter, deren Einstellung durch die Personen vor Ort vorgenommen werden und die eine Einschätzung der Situation am Standort sind.

Die korrekten Werte für die Parameter müssen in praktischen Versuchen herausgefunden werden, die nicht Bestandteil der vorliegenden Arbeit sind. Die Erfahrung aus den Interviews und dem Vergleich mit den Ergebnissen des Sukey-Algorithmus lassen jedoch eine Einschätzung für die ersten praktischen Versuche zu, welche Einstellungen welche Bedeutung haben könnten.

### 5.2.1 Die Anzahl der Personen am Standort

In den Interviews wurde als Situationsbeschreibung vorausgesetzt, dass den Interviewten weitere 1000 Personen folgten. Für die Anzahl der Personen am Standort wurde  $p = 0.2$  angenommen. Wie sich zeigte, war die Einschätzung durch den Sukey-Algorithmus immer schlechter als die Einschätzung der Interviewten. Die Interviewten könnten vergessen haben, dass Ihnen 1000 Personen folgten, was der Sukey-Algorithmus natürlich immer berücksichtigte. Im Fluchtkantengraphen 3 [Tabelle 5.4] wird das Szenario auch mit  $p = 0.001$  getestet und ein sehr unsicherer Fluchtweg gewählt. Daher kann angenommen werden, dass  $p = 0.2$  bedeutet, dass *viele* Personen folgen, und  $p = 0.001$ , dass *wenige* Personen folgen (man eventuell sogar alleine unterwegs ist). Eine genauere Einschätzung ist auf Grundlage der vorgelegten Arbeit nicht möglich.

### 5.2.2 Die Mobilität der Personen am Standort

In den Interviews sollten die Interviewten annehmen, dass sie selbst den Fluchtkantengraphen durchlaufen würden. Für die Mobilität der Personen am Standort wurde 0.5 angenommen. Die Mobilität beeinflusst die Einschätzung der Situation am Standort durch den Sukey-Algorithmus. Je geringer der Wert, desto schlechter der Status. Da der Sukey-Algorithmus die Situation



am Standort oft schlechter einschätzte als die Interviewten, kann angenommen werden, dass  $m = 0.5$  zu niedrig eingestellt war. Für eine Gruppe sehr mobiler Personen mag ein Wert über  $m = 0.5$  eher zutreffen. Eine genauere Einschätzung ist auf Grundlage der vorliegenden Arbeit nicht möglich.

### 5.2.3 Die Performanz

Durch eine höhere Einstellung der Performanz schätzt der Sukey-Algorithmus die Situation am Standort, wie in [Abschnitt 3.8] beschrieben, genauer und positiver ein. Für die vorgestellten Fluchtkantengraphen wurde eine Performanz von  $e = 10$  gewählt. Betrachtet man die Fluchtkantengraphen, wäre auch eine Performanz von  $e = 3$  oder  $e = 4$  zu denselben Einschätzungen gekommen. Die Auswirkung der Performanz hängt von der Komplexität des Fluchtkantengraphen ab. Sie sollte jedoch in jedem Fall größer als  $e = 1$  sein.

## 6 Zusammenfassung

In der vorliegenden Arbeit wurde der Fokus auf den *Sukey-Algorithmus* als kombinierte Lösungsstrategie für die Berechnung geeigneter Fluchtwege in einer Kesselsituation im zivilen Umfeld gelegt.

Es wurde gezeigt, dass die Lösungsstrategie eines *dezentralen Sukey-Netzes* zum Erkennen von Fluchtkantenreduzierungen, die der intuitiven Wahl des besten Fluchtweges durch Menschen bei bekannten Randbedingungen entspricht, möglich ist. Dafür wurde ein *Sukey-Algorithmus* entwickelt, der Fluchtkantenreduzierungen automatisch erkennt, sie bewertet und vor Einkesselungen warnt.

Die technischen Voraussetzungen für ein dezentrales Sukey-Netz sind durch die vorhandenen Mobiltelefone gegeben, so dass ein Einsatz im zivilen Umfeld möglich ist. Damit kann der Schwachpunkt eines zentralen Aufbaus (die Zentrale) durch den Einsatz eines dezentralen Sukey-Netzes umgangen werden.

### 6.1 Aufbau der Arbeit

Von den Strategien zur Lösung von Fluchtkantenreduzierungen wurde der Schwerpunkt in der vorliegenden Arbeit auf die Entwicklung des Algorithmus eines dezentralen Sukey-Netzes gelegt.

Zunächst wurden in [Kapitel 2] die Voraussetzungen des dezentralen Sukey-Netzes und die technischen Randbedingungen für seinen Betrieb vorgestellt. Anschließend wurden die Faktoren, die die Bewertung einer Fluchtkante ermöglichen herausgearbeitet.

In [Kapitel 3] wurde der Sukey-Algorithmus beschrieben, der auf einer Adaption des A\*-Algorithmus beruht.

In [Kapitel 4] wurde der Sukey-Algorithmus als Simulationswerkzeug implementiert.

Schließlich wurden in [Kapitel 5] die Ergebnisse mittels Online-Interviews an den Gegebenheiten der Realität verglichen.

## 6.2 Ergebnisse

Die Erkenntnisgewinne durch die Online-Interviews führen zu folgenden Resultaten:

Der in der vorliegenden Arbeit entwickelte Sukey-Algorithmus weicht nicht gravierend von menschlichen Einschätzungen ab. Dort, wo sich das Ergebnis des Sukey-Algorithmus von der menschlichen Intuition unterschied, konnte dies entweder auf eine vernachlässigbare Schwäche des Sukey-Algorithmus oder menschliche Fehler zurückgeführt werden.

Allerdings kam der Sukey-Algorithmus insgesamt zu viel kritischeren Einschätzungen, als die Interview-Partner. Dies konnte einerseits auf zu restriktiv eingestellte Parameter des Sukey-Algorithmus, andererseits auf die unbedarfte Herangehensweise der Interview-Partner, die möglicherweise die angenommenen ihnen folgenden Mitmenschen vergaßen, zurückgeführt werden.

Die Simulationen konnten nur mithilfe vom Autor geschätzten Voreinstellungen vorgenommen werden. Erkenntnisse über die Feineinstellungen werden in praktischen Tests erfolgen, die nicht Bestandteil dieser Arbeit waren.

## 6.3 Ausblicke und Visionen

Die vorgelegte Arbeit konzentrierte sich auf die Entwicklung des Sukey-Algorithmus als Voraussetzung für den Einsatz eines dezentralen Sukey-Netzes.

Über diese Entwicklung hinaus werden weitere Schritte bis zum praktischen Einsatz folgen.

### 1. Entwicklung einer Protokollsprache

Zur Kommunikation in einem dezentralen Sukey-Netz soll je nach Entwicklungssoftware entweder auf vorhandene Frameworks oder Bibliotheken zugegriffen, oder eine eigene Protokollsprache entwickelt werden. Da der exakt adressierte Versand von Daten in einem dezentralen Sukey-Netz nicht nötig sein sollte, weil jeder Teilnehmer zu seiner eigenen Sicherheit möglichst anonym und alle Daten für alle Teilnehmer bestimmt sind, sollen - aus Sicht des Autors - nur Broadcast-Protokolle verwendet werden.

Ein verschlüsseltes Sukey-Netz, welches Daten ausschließlich an alle Teilnehmer versendet, ist die Vision der Entwicklung einer Protokollsprache.

### 2. Entwicklung einer Benutzeroberfläche

Für den Betrieb eines Sukey-Netzes benötigen die Teilnehmer Mobiltelefone mit entsprechender Software. Um die Eingabe zu erleichtern muss ein Umgebungs- oder Stadtplan

auf den Mobiltelefonen dargestellt werden, auf dem die Hindernisse, sowie der aktuelle Standort zu sehen sind. Die zu entwickelnde Oberfläche soll es ermöglichen neue Hindernisse hinzuzufügen oder vorhandene Hindernisse zu entfernen. Darüber hinaus soll der aktuelle *STATUS* als Balken, Ampel o.ä., sowie die empfohlene Fluchtrichtung angezeigt werden.

### 3. Entwicklung eines Stadtplanparsers

Da das dezentrale Sukey-Netz ohne Anbindung an das Internet auskommen muss, soll der Umgebungs- oder Stadtplan während des Einsatzes auf jedem Mobiltelefon gespeichert werden. Aus dem Plan leitet sich der Fluchtkantengraph ab, auf dem der Sukey-Algorithmus laufen soll. Die Umwandlung eines Stadtplans in den Fluchtkantengraphen kann entweder manuell - durch Nachzeichnen der Straßenverläufe - oder automatisch - durch Parsen der Straßenverläufe - erfolgen.

Die automatische Ableitung eines Fluchtkantengraphen aus jedem beliebigen digitalen Umgebungs- oder Stadtplan ist die Vision der Entwicklung eines Stadtplanparsers.

## Literaturverzeichnis

- [Balasubramanian u. a. 2010] BALASUBRAMANIAN, Aruna ; MAHAJAN, Ratul ; VENKATARAMANI, Arun: Augmenting mobile 3G using WiFi. In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. New York, NY, USA : ACM, 2010 (MobiSys '10), S. 209–222. – URL <http://doi.acm.org/10.1145/1814433.1814456>. – ISBN 978-1-60558-985-5
- [Dirk Helbing ] DIRK HELBING, Péter Molnár und Tamás V.: Simulation von Fußgängergruppen in normalen Situationen und im Evakuierungsfall
- [ETSI 2011a] ETSI: *Mobile technologies GSM*. August 2011. – URL <http://www.etsi.org/WebSite/Technologies/gsm.aspx>
- [ETSI 2011b] ETSI: *UMTS*. August 2011. – URL <http://www.etsi.org/WebSite/Technologies/UMTS.aspx>
- [F. G. Pajonk 2006] F. G. PAJONK, W. R. D.: Panik bei Großschadensereignissen. (2006)
- [Goldratt 2004] GOLDRATT, Eliyahu M.: THE GOAL - A Process of Ongoing Improvement THIRD REVISED EDITION. (2004)
- [Hart u. a. 1968] HART, P.E. ; NILSSON, N.J. ; RAPHAEL, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: *Systems Science and Cybernetics, IEEE Transactions on* 4 (1968), july, Nr. 2, S. 100 –107. – ISSN 0536-1567
- [IEEE Standard for Information Technology 2011] IEEE STANDARD FOR INFORMATION TECHNOLOGY: *802.11n-2009*. 2011. – URL <http://standards.ieee.org/findstds/standard/802.11n-2009.html>
- [Jin-Shyan Lee und Shen 2007] JIN-SHYAN LEE, Yu-Wei S. ; SHEN, Chung-Chou: A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. (2007), Nov. – URL [http://eee.guc.edu.eg/Announcements/Comparative\\_Wireless\\_Standards.pdf](http://eee.guc.edu.eg/Announcements/Comparative_Wireless_Standards.pdf)

- [Koenig und Likhachev 2002] KOENIG, Sven ; LIKHACHEV, Maxim: D\*lite. In: *Eighteenth national conference on Artificial intelligence*. Menlo Park, CA, USA : American Association for Artificial Intelligence, 2002, S. 476–483. – URL <http://dl.acm.org/citation.cfm?id=777092.777167>. – ISBN 0-262-51129-0
- [Koenig u. a. 2004] KOENIG, Sven ; LIKHACHEV, Maxim ; FURCY, David: Lifelong planning A\*. In: *Artif. Intell.* 155 (2004), May, S. 93–146. – URL <http://dl.acm.org/citation.cfm?id=992861.992865>. – ISSN 0004-3702
- [Mahanti u. a. 1992] MAHANTI, Ambuj ; GHOSH, Subrata ; NAU, Dana S. ; PAL, Asim K. ; KANALL, Laveen: Performance of IDA on trees and graphs. In: *Proceedings of the tenth national conference on Artificial intelligence*, AAAI Press, 1992 (AAAI'92), S. 539–544. – URL <http://dl.acm.org/citation.cfm?id=1867135.1867218>. – ISBN 0-262-51063-4
- [Maxim Likhachev und Thrun 2005] MAXIM LIKHACHEV, Geoff Gordon Anthony S. ; THRUN, Sebastian: Anytime Dynamic A\*: An Anytime, Replanning Algorithm. In: *Artif. Intell.* (2005). – URL <https://www.aaai.org/Papers/ICAPS/2005/ICAPS05-027.pdf>
- [Stentz 1995] STENTZ, Anthony: The focussed D\* algorithm for real-time replanning. In: *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1995, S. 1652–1659. – URL <http://dl.acm.org/citation.cfm?id=1643031.1643113>. – ISBN 1-55860-363-8
- [Sukey.org 2011a] SUKEY.ORG: *Sukey*. Juli 2011. – URL <http://sukey.org/>
- [Sukey.org 2011b] SUKEY.ORG: *Sukey - Tutorial*. Juli 2011. – URL <http://sukey.org/tutorial>
- [Wi-Fi Alliance, 2011 2011] WI-FI ALLIANCE, 2011: *Wi-Fi Alliance*. 2011. – URL <http://www.wi-fi.org/>
- [Wikipedia 2011a] WIKIPEDIA: *IEEE 802.11*. July 2011. – URL [http://de.wikipedia.org/wiki/IEEE\\_802.11](http://de.wikipedia.org/wiki/IEEE_802.11)
- [Wikipedia 2011b] WIKIPEDIA: *Sukey*. April 2011. – URL <http://en.wikipedia.org/wiki/Sukey>
- [Wikipedia 2011c] WIKIPEDIA: *Wireless access point*. August 2011. – URL [http://en.wikipedia.org/wiki/Access\\_point](http://en.wikipedia.org/wiki/Access_point)

[Wikipedia 2011d] WIKIPEDIA: *Wireless ad hoc network*. July 2011. – URL [http://en.wikipedia.org/wiki/Wireless\\_ad\\_hoc\\_network](http://en.wikipedia.org/wiki/Wireless_ad_hoc_network)

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 16. Dezember 2011

---

Micha Pikora