

Kvalitativne metode za obvladovanje zaupanja

David Jelenc

DOKTORSKA DISERTACIJA

PREDANA

FAKULTETI ZA RAČUNALNIŠTVO IN INFORMATIKO

KOT DEL IZPOLNJEVANJA POGOJEV ZA PRIDOBITEV NAZIVA

DOKTOR ZNANOSTI

S PODROČJA

RAČUNALNIŠTVA IN INFORMATIKE



Ljubljana, 2014

IZJAVA

Izjavljam, da sem avtor dela in da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali na drugem visokošolskem zavodu, razen v primerih, kjer so navedeni viri.

— David Jelenc —
april 2014

ODDAJO SO ODOBRILI

dr. Denis Trček

redni profesor za računalništvo in informatiko

MENTOR IN ČLAN OCENJEVALNE KOMISIJE

dr. Franc Solina

redni profesor za računalništvo in informatiko

PRESEDNIK OCENJEVALNE KOMISIJE

dr. Branko Kavšek

docent za računalništvo in informatiko

ZUNANJI ČLAN OCENJEVALNE KOMISIJE

Univerza na Primorskem

dr. Borut Žalik

redni profesor za računalništvo in informatiko

ZUNANJI ČLAN OCENJEVALNE KOMISIJE

Univerza v Mariboru

PREDHODNA OBJAVA

Izjavljam, da so bili rezultati obravnavane raziskave predhodno objavljeni oz. sprejeti za objavo v recenzirani reviji ali javno predstavljeni v naslednjih primerih:

- [1] JELENC, David, HERMOSO, Ramón, OSSOWSKI, Sascha, TRČEK, Denis. Alpha test-bed: a new approach for evaluating trust models. V: RICCI, Alessandro (ur.). *ITMAS 2012: proceedings of the Third International Workshop on Infrastructures and Tools for Multiagent Systems*, 5. junij, 2012, Valencia, Španija.
- [2] JELENC, David, HERMOSO, Ramón, SABATER-MIR, Jordi, TRČEK, Denis. Decision making matters: a better way to evaluate trust models. *Knowledge-based Systems*, 2013. doi: [10.1016/j.knosys.2013.07.016](https://doi.org/10.1016/j.knosys.2013.07.016)
- [3] JELENC, David, TRČEK, Denis. Qualitative trust model with a configurable method to aggregate ordinal data. *Autonomous Agents and Multi-Agent Systems*, 2013. doi: [10.1007/s10458-013-9239-8](https://doi.org/10.1007/s10458-013-9239-8)

Potrjujem, da sem pridobil pisna dovoljenja vseh lastnikov avtorskih pravic, ki mi dovoljujejo vključitev zgoraj navedenega materiala v pričujočo disertacijo. Potrjujem, da zgoraj navedeni material opisuje rezultate raziskav, izvedenih v času mojega podiplomskega študija na Univerzi v Ljubljani.

“If I have seen a little further it is by standing on the shoulders of Giants.”

— Isaac Newton, 1676.

POVZETEK

Pri mnogih opravilih je naš uspeh odvisen od drugih. Ko se peljemo z avtom, zaupamo izdelovalcu, da bo vozilo brezhibno, gradbenikom, da bo cesta varna, ter soudeležencem v prometu, da bodo vozili po predpisih; če jim ne bi, se ne bi peljali. Zaupanje nam omogoča, da lahko delujemo v negotovem okolju. Iz podobnih razlogov postaja zaupanje vse pomembnejši dejavnik v računalništvu, zlasti na področju porazdeljenih sistemov, v katerih uporabniki dosegajo svoje cilje s skupnim delovanjem in zanašanjem drug na drugega. Sistemi za obvladovanje zaupanja so računalniški programi, ki uporabnikom svetujejo, kako zaupanja vredne so ostale v sistemu prisotne entitete. V disertaciji se lotimo dveh problemov. Najprej se ukvarjamo z modeli zaupanja, tj. matematičnimi postopki za izračun zaupanja, ki vhodne podatke kot tudi končne izračune predstavijo z opisnimi oznakami imenovanimi kvalitativne vrednosti. Razvijemo model zaupanja, ki do izbranega uporabnika in ponujen tip storitve oceni zaupanje. Izračun zaupanja vršimo na podlagi izkušenj iz preteklih interakcij ter mnenj ostalih uporabnikov. Posebno pozornost namenimo obdelavi kvalitativnih podatkov, zato razvijemo novo metodo, ki iz podanega seznama kvalitativnih oznak izračuna najbolj reprezentativno. Model zaupanja eksperimentalno ovrednotimo in ga primerjamo z najvplivnejšimi modeli v literaturi. V drugem delu disertacije se lotimo problema merjenja učinkovitosti modelov zaupanja. Predlagamo nov način, ki lahko neposredno vrednoti in primerja tako kvalitativne kot kvantitativne modele. Testno okolje implementiramo in v njem zasnujemo eksperiment, v katerem pokažemo, da posredni načini vrednotenja, ki v literaturi prevladujejo, dajejo dvomljive rezultate. S tem utemeljimo neposreden način vrednotenja ter splošnost predlagane rešitve.

Ključne besede: zaupanje, agent, kvalitativni podatki, agregacija, testno okolje

ABSTRACT

In performing various tasks we often rely on others. While driving, we trust the manufacturer to produce a working car, civil engineers to build safe roads, and fellow drivers to comply to driving rules; if we did not, we would not be driving. Trust thus allows us to function in an unpredictable environment. For similar reasons, it is also becoming prevalent in computing, particularly in distributed systems, in which different users fulfill their goals by interacting with and thus relying on each other. Trust management systems are computer programs that advise users about trustworthiness of other entities in the system. This dissertation tackles two problems. First, we deal with trust models—mathematical procedures that compute trust—that use data presented with linguistic labels, which we term qualitative values. The model uses qualitative data to express assessments in past interactions, judgments in received opinions, and degrees of trust in trust estimations. We manipulate qualitative values appropriately by devising a method for estimating the most representative qualitative value in a collection of such values. We experimentally evaluate the model and compare it against the most influential works in the literature. In the second part, we deal with the evaluation of trust models. We propose a method that can directly evaluate and compare the performances of trust models regardless of their type: be it qualitative or quantitative. We implement the method and devise an experiment, which shows that the assumption underlying most of existing approaches, does not always hold. This shows that our approach, free of this assumption, is more general.

Key words: trust, agent, qualitative data, aggregation, testbed

ZAHVALA

Pri vsakem delu, ki nastaja tako dolgo kot je pričujoče, postane seznam ljudi, ki si zaslužijo zahvalo, hitro neobvladljiv. V zadnjih štirih letih in pol sem se pogovarjal z mnogimi, ki so mi bili pripravljene pomagati. Že vnaprej se opravičujem tistim, ki sem jih v spodnjem besedilu izpustil: hvala vam za vse.

Hvala mentorju prof. Denisu Trčku za vse nasvete, spodbude, usmeritve ter pomoč, ki sem jo bil deležen tekom doktorskega izobraževanja.

Hvala Javni agenciji za raziskovalno dejavnost Republike Slovenije za finančno podporo v obliki programa Mladi raziskovalec.

Hvala prof. Saschi Ossowskemu z Univerze Rey Juan Carlos v Madridu in programu COST Agreement Technologies IC0801, ki sta mi omogočila raziskovalno gostovanje v Madridu. Hvala dr. Ramónu Hermosu iz Univerze Rey Juan Carlos ter dr. Jordiju Sabater-Miru z Inštituta za umetno inteligenco v Barceloni (IIIA-CSIC) za prijetno in več kot uspešno sodelovanje.

Hvala zdajšnjim in nekdanjim članom Laboratorija za e-medije, ki so mi z vprašanji, odgovori, komentarji ter pripombami bili vedno na voljo. V naključnem vrstnem redu hvala mag. Iztoku Starcu, Jerneju Kosu, Damjanu Tratniku, Evi Zupančič ter dr. Damjanu Kovaču.

Hvala staršema za moralno podporo ter da sta me naučila, da se vztrajnost obrestuje.

Nenazadnje hvala Tanji. Brez tvoje ljubezni, razumevanja, potrpežljivosti in zaupanja mi ne bi uspelo.

— David Jelenc, Ljubljana, april 2014.

KAZALO

<i>Povzetek</i>	<i>i</i>
<i>Abstract</i>	<i>iii</i>
<i>Zahvala</i>	<i>v</i>
<i>1 Uvod</i>	<i>1</i>
1.1 O zaupanju	2
1.2 O računalniško podprtem obvladovanju zaupanja	2
1.3 O kvalitativnih metodah in motivaciji za disertacijo	3
1.4 Prispevki k znanosti	4
1.5 Pregled vsebine	5
<i>2 Osnovne definicije ter pregled področja</i>	<i>7</i>
2.1 Uvod	8
2.2 Zaupanje	8
2.2.1 Definicije v literaturi	8
2.2.2 Formalizacija zaupanja in ostalih gradnikov	9
2.3 Model zaupanja	10
2.3.1 Formalizacija vhodnih podatkov	11
2.3.2 Formalizacija modela zaupanja	12
2.3.3 Ugled in modeli ugleda	12
2.3.4 Sistemi za obvladovanje zaupanja	13
2.4 Pregled nekaterih obstoječih modelov zaupanja	14
2.4.1 Abdul-Rahman in Hailes	14

2.4.2	Beta reputation system	15
2.4.3	EigenTrust	16
2.4.4	Travos	17
2.4.5	Yu, Singh, Sycara	17
2.4.6	Modeli na osnovi mehke logike	18
2.4.7	Modeli, ki uporabljajo ostale informacije	19
2.4.8	Kvalitativno dinamično ocenjevanje	20
2.4.9	Sklep in povzetek odprtih problemov	20
2.5	Pregled načinov merjenja učinkovitosti modelov zaupanja	21
2.5.1	Vrednotenje zaupanja preko odločitev	22
2.5.2	Vrednotenje zaupanja neposredno	26
2.5.3	Sklep in povzetek odprtih problemov	27
3	<i>Kvalitativni podatki</i>	29
3.1	Uvod	30
3.2	Množica kvalitativnih vrednosti	30
3.3	Izračun reprezentativne kvalitativne vrednosti	30
3.4	Dve lastnosti predlagane metode	34
3.5	Zaključek	34
4	<i>Kvalitativni model zaupanja</i>	37
4.1	Uvod	38
4.2	Viri informacij	38
4.3	Izračun zaupanja	39
4.3.1	Vrednotenje izkušenj	39
4.3.2	Vrednotenje mnenj	40
4.3.3	Končni izračun zaupanja	43
4.3.4	Izračun pretekle natančnosti ter kredibilnosti	44
4.4	Zaključek	47
5	<i>Namensko testno okolje Alpha</i>	49
5.1	Uvod	50
5.2	Alpha in ostali agenti	50
5.3	Visokonivojska arhitektura	52
5.4	Protokol evaluacije	52

5.4.1	Brez odločitev	53
5.4.2	Izbiranje ponudnikov storitev	54
5.4.3	Izbiranje ponudnikov storitev in izbiranje poročevalcev	55
5.5	Scenarij	56
5.5.1	Generiranje izkušenj	56
5.5.2	Generiranje mnenj	59
5.6	Metrike	62
5.6.1	Natančnost	62
5.6.2	Koristnost	64
5.6.3	Stroški mnenj	65
5.6.4	Pomen različnih metrik	66
5.7	Primer poteka testiranja	66
5.8	Implementacija	68
5.9	Zaključek	69
6	<i>Evaluacija</i>	71
6.1	Uvod	72
6.2	Vpliv odločitvenega modela na model zaupanja	72
6.2.1	Hipoteza in postopek preverjanja	72
6.2.2	Uporabljeni modeli zaupanja	73
6.2.3	Odločitvena modela	75
6.2.4	Scenarij	76
6.2.5	Rezultati	77
6.2.6	Diskusija	78
6.3	Evaluacija kvalitativnega modela zaupanja	84
6.3.1	Uporabljeni scenariji	84
6.3.2	Rezultati	87
6.3.3	Diskusija	88
6.3.4	Povzetek rezultatov	94
6.4	Zaključek	94
7	<i>Zaključek</i>	95
7.1	Povzetek vsebine	96
7.2	Odperta vprašanja	97

<i>A</i>	<i>Dokaza</i>	<i>99</i>
A.1	Uvodni gradniki	100
A.2	Dokaz enakosti z mediano	101
A.3	Dokaz enakosti s povprečjem rangov	105
	<i>Literatura</i>	<i>111</i>

Uvod

1.1 O zaupanju

Vsak dan se večkrat zanašamo na druge. Ko v restavraciji naročimo hrano, pričakujemo, da bo ta neoporečna, ker ocenjujemo, da je bila pravilno pridelana, ustrezno pripravljena in da bo hitro postrežena. Vseeno pa se zavedamo, da lahko katerikoli udeleženec postopka priprave hrane svojo nalogo opravi slabo in naš obrok pokvari. V splošnem lahko rečemo, da je vsaka naša odločitev vsaj malo tvegana, saj v praksi nikoli nimamo vseh informacij: pravi nameni ljudi niso vedno znani kot tudi ne njihove dejanske zmožnosti. Nekomu bomo naložili nalogo le, če bomo ocenili, da jo lahko dobro opravi: v mislih si bomo torej najprej izdelali oceno in na njeni podlagi sprejeli odločitev. V takih primerih govorimo o zaupanju. Nemški sociolog Niklas Luhmann je skušal orisati njegovo pomembnost, ko je dejal, da v svetu brez zaupanja zjutraj še vstali ne bi [1]. Zaupanje je nujno za naše delovanje, saj brez njega v negotovem okolju ne bi znali sprejemati odločitev.

1.2 O računalniško podprtem obvladovanju zaupanja

Omrežje vseh omrežij ali Internet je v zadnjih 20 letih v velikem porastu. Sprva so se nanj povezovali zgolj akademski in vojaški računalniki, kasneje osebni, zatem mobilne naprave, dandanes pa tudi sami predmeti; govorimo o t. i. Internetu stvari [2]. Takšen porast je navdahnil razvoj aplikacij, v katerih uporabniki skupaj dosegajo svoje cilje. Tako danes ni neobičajno, da preko interneta opravljamo nakupe, beremo recenzije knjig, ocene avtov, naprav in drugih predmetov, ki jih podajajo anonimni in neznani uporabniki, ter celo najemamo delovno silo za opravljanje manjših nalog (*angl.* crowdsourcing) [3]. Situacije, v katerih se moramo zanašati na druge, so tako enako kot v vsakdanjem svetu prisotne tudi v virtualnem. Vendar je tu obvladovanje tveganja veliko težje, saj akterjev, na katere se zanašamo, ne samo, da ne poznamo, temveč lahko ti svoje zmožnosti in dejanske namene veliko lažje prikrijejo. Z namenom premostitve tega problema so nastali računalniški sistemi za obvladovanje zaupanja, katerih naloga je oceniti zaupanje do uporabnikov v sistemu.

Problem obvladovanja zaupanja je še posebej aktualen na področju več-agentnih sistemov (*angl.* multiagent systems, MAS) [4]. V njih delno ali polno avtonomni računalniški programi, ki jih imenujemo agenti, opravljajo različne naloge. Model MAS bi naj postal osnova za velike, odprte in porazdeljene sisteme prihodnosti, za katere se pričakuje, da bodo sčasoma postali prevladujoči [5]. Ker v splošnem ni zagotovil,

da se bodo agenti obnašali le dobronamerno, moramo zaupanje obvladovati tudi v več-agentnih sistemih. To zavedanje je postalo tako izrazito, da računalniško podprto obvladovanje zaupanja dandanes najpogosteje uvrščamo kar na področje več-agentnih sistemov. To se najbolj vidi v terminologiji, ko denimo namesto besed uporabnik ali akter uporabljamo kar besedo agent.

Obvladovanje zaupanja se pogosto uvršča tudi na področje informacijske varnosti, pod katero običajno razumemo storitve zagotavljanja zaupnosti (*angl.* confidentiality), celovitosti (*angl.* integrity) in razpoložljivosti (*angl.* availability). Vendar se je zaradi storitve "varovanja pred slabimi ponudniki" na tem področju znašlo tudi obvladovanje zaupanja. Nekateri so šli celo tako daleč, da so zaupanje enostavno enačili z računalniško varnostjo [6], čeprav prevladujoči in celovitejši pogledi priznavajo, da same storitve informacijske varnosti še ne zagotavljajo zaupanja [7]. Da gre za soroden, a drugačen koncept lepo ponazorita Rasmusson in Jansson [8], ko ločita med *trdo varnostjo*, v katero spadajo klasične storitve informacijske varnosti, ter *mehko varnostjo*, v katero uvrščata sisteme za obvladovanja zaupanja. Medtem ko prva praviloma varuje komunikacijski kanal, druga varuje vsebino: zagotavlja, da so storitve izvedene v skladu z dogovori.

1.3 O kvalitativnih metodah in motivaciji za disertacijo

Študije kažejo, da se ljudem zdijo numerični podatki preveč kompleksni [9], predvsem pa imajo težave z njihovo interpretacijo [10]. Zato je pri sistemih za obvladovanje zaupanja, katerim ljudje bodisi podajo vhodne podatke bodisi interpretirajo njihove rezultate, smiselno uporabiti kvalitativne vrednosti [11–13]. Kvalitativne vrednosti so opisne in ljudem razumljive oznake, med katerimi je zgolj določen vrstni red. Primeri takih vrednosti so npr. oznake *slabo*, *srednje* in *dobro*. Ker jih ne moremo obravnavati kot števila, je obdelava takih vrednosti otežena, zato je sistemov, ki bi tovrstne podatke uporabljali, v literaturi malo in s kvalitativnimi vrednostmi pogosto ravnajo neprimereno. Zato je bila naša prva naloga narediti matematični nastavek za izračun zaupanja tj. model zaupanja, ki bo osnovan na kvalitativnih podatkih, katere bo pravilno obdeloval, ter bo po učinkovitosti primerljiv z najboljšimi kvantitativnimi modeli.

Naslednji problem, ki se ga v nalogi lotimo, je merjenje učinkovitosti modelov zaupanja. Iz literature poznamo dva pristopa: neposrednega in posrednega. Pri prvem merimo natančnost izračunanega zaupanja, pri drugem pa kakovost odločitev, ki iz izračunanega zaupanja izhajajo. Težava neposrednega merjenja je, da od modelov zaupa-

nja zahteva izračun zaupanja v vnaprej določenemu formatu. Zato z njim ne moremo primerjati denimo kvalitativnih modelov s kvantitativnimi, ker uporabljajo različne formate. Težava posrednega merjenja učinkovitosti pa je, da vrednoti le tisto zaupanje, ki je pripeljalo do odločitev, ne pa tudi tistega, ki ni. Poleg tega lahko na odločitve vplivajo tudi drugi dejavniki. Pomembno je še izpostaviti, da avtorji svoje modele običajno predstavijo brez postopkov, ki bi opisali, kako agent iz izračunanih stopenj zaupanja sprejema odločitve [14], zato moramo pri posrednem vrednotenju takšne postopke definirati sami. Tako posredno vrednotenje temelji na predpostavki, da sprejemanje odločitev ne vpliva na delovanje modela zaupanja, kar je drzna trditev, glede na to da večina modelov izračunava zaupanje iz podatkov, ki so nastali kot posledica agentovih preteklih odločitev. Zato je bila naša druga naloga poiskati neposreden način merjenja učinkovitosti, s katerim bi lahko primerjali tako kvalitativne kot kvantitativne modele.

1.4 Prispevki k znanosti

V disertaciji podamo naslednje izvirne prispevke k znanosti.

Razvijemo metodo za izračun reprezentativne vrednosti. Za potrebe obdelave kvalitativnih podatkov predstavimo metodo, s katero lahko v zbirki kvalitativnih vrednosti izračunamo vrednost, ki je med njimi najbolj reprezentativna. Metoda je uporabna širše in predstavlja alternativo izračunu mediane ali metodi povprečenja rangov.

Razvijemo kvalitativen model zaupanja. Predstavimo matematični nastavek za izračun zaupanja, katerega vhodni podatki kot tudi končni izračuni vsebujejo kvalitativno izražene vrednosti. Posebno pozornost namenimo mehanizmu za zaznavo lažnih mnenj. Model eksperimentalno ovrednotimo in rezultate primerjamo z najodmevnejšimi kvalitativnimi in kvantitativnimi modeli v literaturi.

Razvijemo neposreden način merjenja učinkovitosti modelov zaupanja. S predlaganim se lahko vrednoti tako kvalitativne kot kvantitativne modele ter njihove rezultate primerja. Idejo implementiramo v obliki namenskega testnega okolja, implementacijo pa objavimo pod odprtokodno licenco. S testnim okoljem ovrednotimo kvalitativen model zaupanja ter pokažemo, da predpostavka, na kateri temeljijo obstoječi načini za posredno vrednotenje modelov zaupanja, vedno ne drži, s čimer dodatno podkrepimo veljavnost naše rešitve.

1.5 Pregled vsebine

Celotno delo obsega sedem poglavij in dodatek.

- Poglavlje 2 podaja osnovne definicije konceptov kot so zaupanje, model zaupanja, sistem za obvladovanje zaupanja ipd. Ko vzpostavimo potrebno terminologijo, nadaljujemo s pregledom področja modelov zaupanja in načinov, kako jih vrednotiti. Poglavlje sklenemo s povzetkom odprtih problemov.
- V poglavju 3 formaliziramo kvalitativne podatke ter predstavimo metodo izračuna reprezentativne vrednosti. Prav tako podamo nekatere njene lastnosti.
- Poglavlje 4 predstavi kvalitativni model zaupanja. V njem podamo tipe vhodnih podatkov, postopke, ki iz njih izračunajo zaupanje, ter posebno pozornost namenimo mehanizmom za zaznavanje zlaganih mnenj.
- V poglavju 5 se posvetimo problemu vrednotenja učinkovitosti modelov zaupanja. Opišemo težave obstoječih pristopov ter podamo konceptualno zasnovo in arhitekturo namenskega testnega okolja Alpha. Podrobno opišemo delovanje vsake od njegovih komponent.
- V poglavju 6 zasnujemo eksperiment, v katerem pokažemo, da predpostavka, ki jo uporabljajo posredni načini vrednotenja modelov zaupanja, ne drži vedno. Zatem ovrednotimo kvalitativni model zaupanja in njegove rezultate primerjamo z najodmevnejšimi modeli v literaturi.
- Poglavlje 7 povzema predstavljeno vsebino, glavne rezultate ter ugotovitve in podaja seznam odprtih problemov.
- V dodatku A dokažemo pravilnost Izrekov 1 in 2.



*Osnovne definicije ter pregled
področja*

2.1 Uvod

Namen tega poglavja je podati pregled področja. A pred tem je potrebno vzpostaviti besedišče, ki ga bomo pri pregledu uporabili. Tako najprej podamo definicije osnovnih konceptov kot so zaupanje, model zaupanja in sistemi za obvladovanje zaupanja. Pri pregledu najprej obravnavamo modele zaupanja, zatem pa še načine, kako meriti njihovo učinkovitost. Vsak pregled se zaključi povzetkom odprtih problemov.

2.2 Zaupanje

V tem razdelku najprej podamo pregled najbolj pogostih definicij zaupanja, nato pa zaupanje tudi matematično formaliziramo.

2.2.1 Definicije v literaturi

Kljub temu, da besedo zaupanje v vsakdanjem pogovoru pogosto uporabljamo, je formalna definicija zaupanja precej težavna. Besedo namreč uporabljamo precej splošno in v različnih kontekstih, kar se odraža tudi v literaturi, ki ponuja več različnih definicij. Med temi pa Jøsang *et al.* [9] izpostavljajo najpogostejši, ki jih bomo imenovali *ožja* ter *širša* definicija zaupanja. Prvo, ožjo, podaja sociolog Gambetta [15].

Definicija 1 (Ožje zaupanje): Zaupanje je subjektivna verjetnost, ki ponazarja stopnjo prepričanja nekoga, osebe A, da bo nekdo drug, oseba B, izpolnila neko (osebi A pomembno) nalogo.

Definicija 1 je za modeliranje zelo priročna, saj vsebuje tako element odvisnosti (*angl.* dependence; osebe A od osebe B) kot tudi element zanesljivosti (*angl.* reliability; osebe B kot jo doživlja oseba A). Ponekod je ta formulacija poimenovana kot *zaupanje za zanesljivost* (*angl.* reliability trust). Spet drugi trdijo, da je zaupanje bolj kompleksno. Tako denimo Falcone in Castelfranchi [16] ugotavljata, da četudi je oseba A prepričana, da je oseba B zanesljiva, to še ne pomeni, da bo oseba A *zaupala* nalogo osebi B. Čeprav je možnost neuspeha (izredno) majhna, lahko za osebo A to še vedno predstavlja nesprejemljivo stopnjo tveganja. Takšne situacije so navdahnile *širšo* definicijo zaupanja, ki jo podajata McKnight in Chervany [17].

Definicija 2 (Širše zaupanje): Zaupanje pomeni pripravljenost, da se v dani situaciji in z občutkom relativne varnosti zanesemo na nekoga (ali nekaj), kljub temu da so mogoče negativne posledice.

Jøsang *et al.* [9] ugotavljajo, da je ta definicija manj eksplicitna in bolj dvoumna kot prejšnja, a ravno zaradi tega bolj splošna, saj vsebuje elemente odvisnosti (tistega, ki zaupa, od tistega, ki mu je zaupano), zanesljivosti (tistega, ki mu je zaupano), koristi (*angl.* utility; če je naloga uspešno opravljena, tisti, ki zaupa pridobi korist, če ni, nastane škoda) in tveganja (*angl.* risk; tisti, ki zaupa, mora sprejeti določeno tveganje). Pomembna razlika med širšo in ožjo definicijo je tudi v tem, da je zaupanje v ožji definiciji predstavljeno kot spekter vrednosti, v širši pa zgolj binarno, tj. v dani situaciji nekomu dano nalogo bodisi zaupamo bodisi ne. A kljub temu da je širša definicija bolj splošna in zajema več dejavnikov kot ožja – posledično bi lahko celo rekli, da je bolj pravilna – pa se v praksi izkaže, da je za modeliranje preobsežna. Večina računsko obvladljivih modelov zaupanja zato izhaja iz ožje definicije: v preglednem članku Jøsang *et al.* [9] navajajo le dva modela [18, 19], ki poleg zanesljivosti modelirata še druge dejavnike. Ostali modeli izhajajo iz ožje definicije.

2.2.2 Formalizacija zaupanja in ostalih gradnikov

Za potrebe modela zaupanja, ki bo predstavljen v tej disertaciji, kot tudi za lažje primerjanje med obstoječimi modeli zaupanja, v nadaljevanju podajamo formalno definicijo zaupanja. Najprej formaliziramo osnovne gradnike modeliranega sistema.

Definicija 3 (Čas): Čas v sistemu modeliramo diskretno in ga predstavimo z množico T , posamezno časovno enoto pa označimo s $t \in T$. Med časovnimi enotami velja linearna urejenost: $t_i < t_{i+1}$ za vsak $i \in \mathbb{N}$. Slednje pomeni, da enota t_i nastopi pred enoto t_{i+1} .

Ko je govora o zaupanju, omenjamo različne posameznike, osebe, entitete, ki jih bomo v nadaljevanju imenovali *agenti*. Izraz prihaja s področja več-agentnih sistemov [4], ki je v zadnjem času postalo eno izmed najbolj aktivnih področij aplikacij modelov zaupanja.

Definicija 4 (Agenti in tipi storitev): Agenti so entitete, med katerimi se vzpostavlja-jo relacije zaupanje. Množico vseh agentov predstavimo z množico A . Ta se lahko s časom spreminja, saj lahko obstoječi agenti sistem zapustijo, prav tako pa lahko v sistem pridejo novi. Tako množico agentov ob času $t \in T$ označimo z $A(t)$. Agenti opravljajo različne storitve. S simbolom S označimo množico vseh tipov storitev, ki jih agenti nudijo drug drugemu.

V Definiciji 1 je govora o *subjektivni verjetnosti* kot meri zaupanja. To mero bomo v nadaljevanju imenovali *stopnja zaupanja*.

Definicija 5 (Stopnja zaupanja): Stopnja zaupanja je mera, s katero izrazimo količino zaupanja do drugih agentov. Množica vseh stopenj zaupanja je linearno urejena množica, ki jo označimo z Ω .

Z zgornjimi gradniki lahko zaupanje tudi matematično formaliziramo. Izhajamo iz ožje definicije, kjer zaupanje predstavlja mero zanesljivosti za opravljanje določenega tipa storitve.

Definicija 6 (Formalizacija zaupanja): Zaupanje je mera zanesljivosti, s katero agente opravljajo različne storitve. Zaupanje formaliziramo s preslikavo $\tau : A \times S \mapsto \Omega$, kjer $\omega = \tau(a, s) \in \Omega$ predstavlja našo stopnjo zaupanja v agenta $a \in A$, da opravi storitev tipa $s \in S$. Z $\Omega^{A \times S}$ označimo univerzalno množico zaupanja.

Definicija 6 je za razliko od Definicij 1 in 2 podana z vidika agenta, ki zaupa, in ne z vidika zunanjega opazovalca. Tako je zaupanje τ indeksirano le s ciljnim agentom (ter tipom storitve) ne pa tudi z izvornim agentom. Ta formulacija se razlikuje od Kovačeve [20], kjer je zaupanje indeksirano tako s ciljnim kot s izvornim agentom (ter tipom storitve, ki je tam imenovana kontekst), a bo za nas bolj priročna, saj bomo večino stvari definirali z vidika agenta, ki uporablja model zaupanja.

2.3 Model zaupanja

Model zaupanja je računski postopek, ki izračuna zaupanje. Kot vsak računski postopek ima model zaupanja določene vhode in izhode. Medtem ko smo izhode, tj.

zaupanje, definirali v prejšnjem razdelku, bomo v tem definirali vhode ter na koncu formalizirali model zaupanja.

2.3.1 Formalizacija vhodnih podatkov

V literaturi [9, 14, 21] najdemo precej modelov zaupanja. Ti modeli se med seboj razlikujejo v mnogih vidikih, vendar je eden najpomembnejših vrsta informacij, iz katerih modeli izračunavajo zaupanje. Tako Sabater in Sierra [21] ter tudi Pinyol in Sabater [14] ugotavljajo, da prevladujejo modeli, ki za izračun zaupanja uporabljajo izkušnje iz preteklih interakcij ter mnenja ostalih agentov v sistemu. Nekateri modeli uporabljajo tudi druge vrste informacij, vendar so takšni primeri redkejši. Najprej formaliziramo izkušnje.

Definicija 7 (Izkušnje): Ko agent za nas opravi določeno storitev, pravimo, da smo bili s tem agentom v interakciji. Izkušnja je tako zapis interakcije, pri katerem dodatno shranimo oceno, s katero izrazimo zadovoljstvo z opravljeno storitvijo. Naše izkušnje predstavimo s preslikavo $\varepsilon : A \times S \times T \mapsto \Lambda$, kjer $\lambda = \varepsilon(a, s, t) \in \Lambda$ predstavlja oceno, s katero smo ovrednotili, kako dobro je agent $a \in A$ ob času $t \in T$ za nas opravil storitev tipa $s \in S$. Množica ocen je linearno urejena in jo označimo z Λ , oznaka $\Lambda^{A \times S \times T}$ pa predstavlja univerzalno množico izkušenj.

V odprtih sistemih so izkušnje z agenti redke. Zaradi tega modeli zaupanja pogosto uporabijo mnenja pridobljena od drugih agentov. Naslednja definicija formalizira mnenje.

Definicija 8 (Mnenja): Mnenje je vrednostna sodba, ki nam jo nek agent poda o drugemu agentu. Predstavimo s preslikavo $\phi : A \times A \times S \times T \mapsto \Theta$, kjer $\theta = \phi(a_p, a_c, s, t) \in \Theta$ predstavlja vrednost mnenja, ki nam ga je ob času $t \in T$ sporočil agent $a_p \in A$ o agentu $a_c \in A$ za opravljanje storitve tipa $s \in S$. Množica vrednosti mnenj Θ je linearno urejena, s simbolom $\Theta^{A \times A \times S \times T}$ pa predstavimo univerzalno množico mnenj.

Poleg izkušenj in mnenj pa nekateri modeli uporabljajo tudi podatke iz drugih virov. Tako denimo model Regret [22] analizira družbena omrežja in iz njih ugotavlja, kateri agenti so bolj kredibilni za podajanje mnenj. Podobno so Hermoso *et al.* [23] pred-

stavili model, ki analizira strukture virtualnih organizacij (*angl.* virtual organizations), tj. navideznih okolji, ki delovanje agentov delno omejujejo. Takšne vire informacij formalizira naslednja definicija.

Definicija 9 (Ostale informacije): Ostale informacije so informacije, iz katerih modeli zaupanja izračunajo zaupanje in niso izkušnje ali mnenja. Univerzalno množico ostalih informacij predstavimo z oznako Γ .

Zgornja definicija je namenoma ohlapna; njen namen je zaobjeti vse ostale možne tipe informacij, iz katerih modeli zaupanja izračunavajo zaupanje.

2.3.2 Formalizacija modela zaupanja

Sedaj ko smo formalizirali izkušnje, mnenja ter ostale vire informacij, lahko model zaupanja tudi matematično opišemo.

Definicija 10 (Model zaupanja): Model zaupanja je računski postopek, ki v danem trenutku na vhodu sprejme izkušnje, mnenja in ostale informacije ter na izhodu vrne zaupanje. Formalno ga predstavimo kot funkcijo višjega reda.

$$\text{ModelZaupanja} : T \times \Lambda^{A \times S \times T} \times \Theta^{A \times A \times S \times T} \times \Gamma \mapsto \Omega^{A \times S}$$

Na model zaupanja lahko torej gledamo kot na funkcijo višjega reda, ki za podan čas $t \in T$, nabor izkušenj $\varepsilon \in \Lambda^{A \times S \times T}$, mnenj $\phi \in \Theta^{A \times A \times S \times T}$ in ostalih informacij $\gamma \in \Gamma$, izračuna zaupanje $\tau \in \Omega^{A \times S}$, torej $\tau = \text{ModelZaupanja}(t, \varepsilon, \phi, \gamma)$. Skozi prizmo zgornje formulacije bomo v razdelku 2.4 pregledali najbolj znane modele zaupanja; za vsak model bomo navedli, kakšne množice uporablja, ter (kratko) opisali postopek izračuna.

2.3.3 Ugled in modeli ugleda

Zaupanju soroden je pojem ugled. Za potrebe te disertacije si bomo izposodili definicijo iz slovarja Oxford Concise Dictionary, ki ugled definira kot "tisto, kar se običajno govori ali verjame o naravi osebe ali reči". Ta opredelitev vsebinsko precej spominja na definicijo zaupanja, vendar pojma ne smemo enostavno enačiti. Da sta si različna,

lepo ponazorijo Jøsang *et al.* [9], ko pravijo, da lahko nekomu zaupamo zaradi dobre ugleda, enako verjetno pa se lahko zgodi, da nekomu drugemu ne zaupamo, kljub temu da je njegov ugled enako dober.

Tako kot modeli zaupanja obstajajo tudi modeli ugleda. Slednji imajo podobno nalogo kot modeli zaupanja: svetovati o zanesljivosti potencialnih partnerjev za interakcije. Zaradi podobnega namena je meja med prvimi in drugimi precej zabrisana. Tako nekateri avtorji govorijo o modelih ugleda, drugi pa o modelih zaupanja. Nekateri sicer skušajo podati ločnico med njimi, a trenutno o tem ni širšega konsenza. Tako na primer Kovač [20] kot modele zaupanja označi tiste, ki pri izračunu upoštevajo subjektivne dejavnike agenta, ki vrši izračun (denimo dajo večji poudarek lastnim izkušnjam), kot modele ugleda pa tiste, ki pri izračunu upoštevajo vsa mnenja enakomerno. Po Kovačevi klasifikaciji je izračunana stopnja zaupanja odvisna od agenta, ki jo izračuna, stopnja ugleda pa ne, saj jo izračunajo vsi agenti enako. Pinyol in Sabater [14] pa pravita, da model ugleda zgolj izračuna stopnjo zanesljivosti, model zaupanja pa poda končno odločitev ali nekomu zaupati ali ne. Ta delitev je analogna delitvi o ožji in širši definiciji zaupanja; po njej ožja definicija zaupanja sovpada z ugledom, širša pa z zaupanjem. Da je zmeda še večja, mnogo obstoječih modelov elemente zaupanja in ugleda združuje. Ker naš namen ni klasificirati dosedanje delo, bomo v tej disertaciji izraza model zaupanja in model ugleda obravnavali kot sopomenki. A kljub temu se je treba zavedati, da sta ugled in zaupanje ločena pojma.

2.3.4 Sistemi za obvladovanje zaupanja

Do sedaj smo govorili o modelih zaupanja (in ugleda) zgolj abstraktno. V praksi so modeli implementirani v računalniških rešitvah, ki se imenujejo sistemi za obvladovanje zaupanja (ugleda). Sistem za obvladovanje zaupanja je širši koncept kot model zaupanja, saj mora poleg implementacije sistem skrbeti tudi za komunikacijo med agenti ter shranjevanje bistvenih informacij.

Sistem za obvladovanje zaupanja je lahko realiziran na dva načina, in sicer centralizirano ali porazdeljeno [20]. Pri centralizirani implementaciji nek osrednji sistem shranjuje vse informacije in preko njega tudi poteka vsa komunikacija. Primer centraliziranih sistemov najdemo na spletiščih eBay¹ in Amazon². Nasprotno pa se v porazdeljenem sistemu informacije shranjujejo porazdeljeno, navadno pri agentih samih.

¹<http://www.ebay.com>

²<http://www.amazon.com>

Vsak agent tako sam pridobiva relevantne informacije in vrši izračune. Takšni modeli so pogosti v omrežjih P2P [24].

Tako so po Kovačevi klasifikaciji [20] centralizirani sistemi naravni način implementacije modelov ugleda, porazdeljeni sistemi pa modelov zaupanja. Centralizirani sistemi lahko namreč enostavno izračunajo (konsistentno) oceno o posameznem agentu iz vseh informacij, medtem ko je takšna naloga v porazdeljenih sistemih precej otežena in je navadno odvisna od tistega, ki vrši izračun. Vendar tudi ta ločnica ni stroga; prav mogoče bi bilo realizirati porazdeljen sistem ugleda ali centraliziran sistem zaupanja.

2.4 Pregled nekaterih obstoječih modelov zaupanja

V literaturi [9, 14, 21, 25] je mogoče najti precej modelov zaupanja ter tudi načinov, kako jih klasificirati. Poleg že omenjene (in pogosto dvoumne) delitve na modele zaupanja in ugleda, je precej pogosta delitev po tipu informacij, ki jih modeli uporabljajo. V kontekstu definicij iz razdelka 2.3 tip informacij pomeni način, na katerega so realizirane naslednje množice: množica ocen Λ , množica vrednosti mnenj Θ ter množica stopenj zaupanja Ω . Tako denimo Kovač [20], Jøsang *et al.* [9] in Sabater in Sierra [21] ločijo modele na kvantitativne in kvalitativne (ponekod imenovane tudi diskretne). Na tem mestu omenimo, da obstajajo tudi hibridni modeli, ki na vhodu sprejmejo kvalitativne na izhodu pa vrnejo kvantitativne informacije. V nadaljevanju podajamo pregled najodmevnejših modelov.

2.4.1 Abdul-Rahman in Hailes

Med kvalitativne modele lahko štejemo model zaupanja, ki sta ga predstavila Abdul-Rahman in Hailes (ARH) [26]. V modelu ARH so podporne množice (množica ocen, vrednosti mnenj ter zaupanja) definirane kvalitativno, in sicer $\Lambda = \Theta = \Omega = \{vb < b < g < vg\}$, kjer elementi teh množic zaporedoma pomenijo vrednosti *zelo slabo* (*angl.* very bad), *slabo* (*angl.* bad), *dobro* (*angl.* good) in *zelo dobro* (*angl.* very good). Z njimi agenti ocenjujejo zadovoljstvo opravljenih storitev, izražajo mnenja o drugih agentih ter izračunajo stopnje zaupanja.

Pri izračunu stopnje zaupanja (do izbranega agenta in za izbrano storitev) ARH najprej pregleda lastne izkušnje in, v kolikor te obstajajo, kot stopnjo zaupanja vrne vrednost, ki predstavlja najpogostejšo oceno. Če neposrednih izkušenj ni, model izračuna stopnjo zaupanja iz mnenj. Pri tem na podlagi preteklih opažanj mnenja ustrezno preoblikuje; če denimo opazi da nekdo konsistentno sporoča drugačna mnenja kot se

kasneje izkaže v interakcijah (mnenje je npr. vedno slabše od dejanske izvedbe storitve), bo ARH mnenja takega agenta ustrezno popravil: prejeto mnenje bo vnaprej “izboljšal”.

2.4.2 Beta reputation system

Beta reputation system (BRS) [27] je kvantitativen model za obvladovanje ugleda, ki temelji na beta porazdelitvah. Stopnja ugleda je v BRS predstavljena kot realno število $\Omega \subseteq [0, 1]$, kjer vrednosti 1 in 0 pomenita najvišjo in najnižjo možno stopnjo ugleda. Ocene iz izkušenj ter vrednosti mnenj pa so v BRS predstavljene kot dvojice $\langle r, s \rangle$, kjer $r \in [0, 1]$ predstavlja količino pozitivne $s \in [0, 1]$ pa negativne ocene oz. vrednosti mnenja, pri čemer za vsako dvojico velja $r + s = 1$; $\Lambda = \Theta \subseteq [0, 1] \times [0, 1]$. Denimo ocena $\langle 0.7, 0.3 \rangle$ pomeni, da je bila storitev v 70% dobra, v preostalih 30% pa slaba.

Pri izračunu stopnje ugleda BRS najprej izračuna *funkcijo ugleda* (*angl.* reputation function), nato pa predstavi stopnjo ugleda kot njeno pričakovano vrednost. Funkcija ugleda je namreč običajna beta porazdelitvena funkcija, katere parametri se neposredno izračunajo iz izkušenj in pridobljenih mnenj. Tako BRS po komponentah sešteje vse dvojice ocen iz izkušenj ter k temu, prav tako po komponentah, prišteje še diskontirane dvojice vrednosti mnenj. Mnenja se diskontirajo s faktorjem, ki se izračuna iz količine pozitivnih ocen. Tako BRS predvideva, da je nekdo, ki je zaupanja vreden za opravljanje neke storitve, hkrati tudi kredibilen za podajanje mnenj. Tej predpostavki se v literaturi pogosto pravi predpostavka *tranzitivnosti* [28]. Zanimivo je, da avtorji BRS opredelijo kot model ugleda, hkrati pa je izračun ugleda – ki bi naj bil enak za vse agente – odvisen od tega, kdo ga vrši. To je nazoren primer, kako nejasna je ločnica med modeli ugleda in zaupanja.

Kasneje sta Jøsang in Haller [29] posplošila BRS v Dirichlet Reputation System (DRS). DRS tako uporablja petstopenjsko kvalitativno lestvico za predstavitev ocen in vrednosti mnenj $\Lambda = \Theta = \{m < b < a < g < e\}$, kjer oznake zaporedoma nosijo naslednje pomene: drugorazreden (*angl.* mediocre), slab (*angl.* bad), povprečen (*angl.* average), dober (*angl.* good) in odličen (*angl.* excellent). Za predstavitev stopenj zaupanja pa DRS uporablja več formatov: od preprostega 5-dimenzionalnega vektorja, ki predstavlja agregat vseh ocen in mnenj, do preproste numerične vrednosti na intervalu $[0, 4]$, ki se izračuna tako, da 5-dimenzionalni vektor normiramo in ga obravnavamo kot funkcijo verjetnosti, nato pa izračunamo njegovo matematično upanje. Pri tem kvalitativne vrednosti obravnavamo numerično in ekvidistančno: najslabša ocena (m)

je predstavljena s številom 0, najboljša (e) pa s 4.

2.4.3 *EigenTrust*

EigenTrust (ET) [28] je model za obvladovanje ugleda v omrežjih P2P in je v literaturi najbolj odmeven. V tem primeru je opredelitev modela ugleda skladna s Kovačev o opredelitvijo, saj EigenTrust na podlagi vseh mnenj izračuna globalno zaupanje, tj. enotno oceno zaupanja za vse agente, ki ji lahko rečemo ugled. V EigenTrust agenti ocenjujejo interakcije z binarnimi ocenami $\Lambda = \{0, 1\}$, kjer število 0 označuje slabo, 1 pa dobro oceno. Pri podajanju vrednosti mnenj pa agenti tvorijo dvojice $\langle m, n \rangle$, kjer $m \in \mathbb{N}_0$ in $n \in \mathbb{N}_0$ zaporedoma označujeta število pozitivnih in negativnih ocen iz preteklih interakcij; $\Theta \subseteq \mathbb{N}_0 \times \mathbb{N}_0$. Tako mnenje ni neko izoblikovano stališče (kot npr. stopnja zaupanja v primeru modela ARH ali BRS), temveč predstavlja zgolj zbirko izkušenj. Kljub temu da so vhodi v model binarni, pa je izračunana stopnja zaupanja kvantitativna: $\Omega \subseteq [0, 1]$, kjer 0 in 1 predstavljata najnižjo in najvišjo stopnjo; to pomeni, da je EigenTrust hibriden model. Vendar avtorji opozarjajo, da se morajo končne stopnje zaupanja, četudi so kvantitativne, interpretirati kvalitativno tj. relativno – stopnje zaupanja služijo zgolj za to, da lahko agente med seboj razvrstimo po zaupanju, v absolutnem smislu (numerično) pa stopnje nimajo pomena.

Zgoraj opisana relativna interpretacija sledi iz delovanja modela. V EigenTrust algoritmu vse dvojice mnenj $\langle m, n \rangle$ 1) uredimo v kvadratno matriko. V tej matriki stolpec i ponazarja izkušnje agenta i z vsemi ostalimi agenti. Nato v tej matriki 2) pri vsaki komponenti od pozitivnih ocen odštejemo negativne (v primeru negativne razlike vzamemo 0) in zatem 3) matriko po stolpcih normiramo. Na koncu z uporabo vektorja *vnajprej zaupanja vrednih vrstnikov* (angl. *pretrusted peers*) izračunamo stacionarne vrednosti omenjene matrike. EigenTrust se namreč tako imenuje zato, ker lahko nastalo matriko interpretiramo kot markovsko verigo, katere končno stanje – globalne stopnje zaupanja – izračunamo kot njen levi lastni vektor. V kolikor imamo opravka s sistemom, kjer vnajprej zaupanja vrednih vrstnikov ne moremo določiti, lahko kot njihov nadomestek uporabimo vektor lastnih izkušenj [30]. Glavni vzrok za relativno interpretacijo je tako drugi korak, v katerem EigenTrust ne loči med agentom, o katerem ni mnenj, in agentom, o katerem obstajajo le slaba mnenja; v obeh primerih pridemo do števila 0, potem ko od dobrih interakcij odštejemo slabe. Slednje pomeni, da EigenTrust ne meri t. i. *negativnega zaupanja* in – tako kot BRS – med samim izračunom daje večjo težo mnenjem agentov, ki so dobri pri ponujanju storitev: model uporablj

predpostavko tranzitivnosti.

2.4.4 *Travos*

Travos (TRA) je model zaupanja za agentno-usmerjene navidezne organizacije (*angl.* virtual organization) [31]. Uporablja binarne vrednosti za podajanje ocen interakcij $\Lambda = \{0, 1\}$ ter predstavlja množico vrednosti mnenj na enak način kot EigenTrust: agenti tvorijo dvojice $\langle m, n \rangle$, kjer $m \in \mathbb{N}_0$ in $n \in \mathbb{N}_0$ zaporedoma označujeta število pozitivnih in negativnih ocen iz preteklih interakcij; $\Theta \subseteq \mathbb{N}_0 \times \mathbb{N}_0$. Pri predstavitvi ter po načinu izračuna stopenj zaupanja pa Travos spominja na BRS, saj iz izkušenj in mnenj izračuna beta porazdelitveno funkcijo ter stopnjo zaupanja predstavi kot njeno pričakovano vrednost. Tako je ocena zaupanja podana kot realno število z zveznega intervala $\Omega \subseteq [0, 1]$ in predstavlja verjetnost, da bo interakcija z izbranim agentom uspešna.

Travos in BRS pa se tudi precej razlikujeta. Medtem ko BRS uporablja predpostavko tranzitivnosti, je Travosov mehanizem diskontiranja mnenj naprednejši. Travos namreč vsako prejeto mnenje shrani in ga, podobno kot ARH, primerja s kasnejšo interakcijo. Vendar mnenj za razliko od ARH ne preoblikuje, temveč jih diskontira. Diskontni faktor se izračuna po zapletenem postopku, a osnovni princip je, da se mnenja agentov, ki so v preteklosti podali napačna mnenja, diskontirajo močneje kot mnenja tistih, ki so bili v preteklosti pri podajanju mnenj natančni. Druga razlika med BRS in Travos je ta, da prvi vedno uporabi vsa prejeta mnenja, Travos pa iz količine lastnih izkušenj izračuna prepričanje (*angl.* confidence). V kolikor slednje doseže vnaprej podan prag, se mnenja zavržejo, v nasprotnem primeru pa se z izkušnjami združijo.

2.4.5 *Yu, Singh, Sycara*

Model zaupanja, ki so ga predlagali Yu, Singh in Sycara (YSS) [32], je namenjen za velikim omrežjem P2P. Model je v celoti kvantitativen, saj so tako ocene v izkušnjah, vrednosti mnenj kot tudi stopnje zaupanja predstavljene kot realna števila, tj. $\Lambda = \Theta = \Omega \subseteq [0, 1]$.

Pri izračunu zaupanja YSS ločeno ovrednoti izkušnje ter prejeta mnenja. Pri ovrednotenju izkušenj izračuna povprečje vseh podanih ocen ter, podobno kot v modelu Travos, prepričanje (*angl.* confidence) v lastne izkušnje. Prepričanje, predstavljeno kot število iz intervala $[0, 1]$, se izračuna kot razmerje med številom *dejanskih* izkušenj in

številom *zahtevanih* izkušenj (slednje se poda kot parameter modela). Če je npr. število dejanskih izkušenj 7 potrebnih pa 10, znaša prepričanje $\frac{7}{10}$. Pri obdelavi mnenj YSS izračuna uteženo povprečje vrednosti mnenj, pri čemer se za uteži uporabi *faktor kredibilnosti*, ki se izračuna po algoritmu utežene večine (*angl.* weighted majority algorithm) [33]. Slednji na začetku vsakemu agentu priredi kredibilnost 1.0, nato pa jo pri vsakem nenatančno podanem mnenju zniža za faktor, ki je sorazmeren razliki med podanim mnenjem in dejansko izkušnjo. V zadnjem koraku YSS izračuna uteženo povprečje med povprečjem lastnih ocen izkušenj in povprečno vrednostjo mnenj, pri čemer se ocene utežijo s parametrom prepričanja.

2.4.6 Modeli na osnovi mehke logike

Modele zaupanja, ki temeljijo na mehki logiki, lahko uvrstimo med hibridne, saj imajo lastnosti kvantitativnih in kvalitativnih modelov. Tako denimo model Afras avtorjev Carbo, Molina in Davilla [34] predstavlja vhodne podatke (ocene v izkušnjah, vrednosti mnenj) kot mehke vrednosti (*angl.* fuzzy values) z intervala [0, 100]. Vsaka mehka vrednost je predstavljena s štirimi pari števil, ki predstavljajo oglišča trapezne pripadnostne funkcije (*angl.* trapezium-based membership function). Mehke vrednosti se združuje z uporabo utežene aritmetične sredine, kjer se uteži izračunajo dinamično in so odvisne od različnih dejavnikov, kot je denimo vnaprej določena spominska kapaciteta ter natančnost preteklih napovedi. Posledično so tudi stopnje zaupanje predstavljene kot mehke vrednosti.

Na osnovi mehke logike tudi deluje model avtorjev Ramchurn, Jennings, Sierra in Godo [35]. Pri obdelavi izkušenj – v modelu poimenovanih lokalno zaupanje (*angl.* local confidence) – model izračunava *variacije koristnosti*, ki so definirane kot razlika med dogovorjeno kakovostjo storitve in tisto, ki je bila dejansko opravljena. Variacije, predstavljene kot vrednosti z intervala [-1, 1], so preslikane v nivoje zaupanja (*angl.* confidence levels) treh mehkih vrednosti: *slabo*, *srednje* in *dobro*. Te nato agenti med seboj izmenjujejo v obliki mnenj. Za nekega agenta lahko npr. povemo, da so njegovi nivoji zaupanja 0.60 dobro, 0.25 srednje in 0.00 slabo; slednje se navadno predstavi kot trojica (0.60, 0.25, 0.00). Model nato združi lastne izkušnje s prejetimi mnenji in iz tega izlušči pričakovano izgubo za interakcijo z vsakim agentom. Komplemente izgube, normiran na interval [0, 1], pa predstavlja končno stopnjo zaupanja.

Model avtorjev Griffiths, Chao in Younas [36] pri izračunu zaupanja uporablja vnaprej določen nabor pravil za sklepanje (*angl.* inference rules). Za vsako interakcijo

ovrednoti več vidikov in pri vsakem oceni interakcijo bodisi kot pozitivno bodisi negativno, nakar iz teh ocen izlušči splošno oceno interakcije, ki se predstavi z realnim številom z intervala $[-1, 1]$. Ta ocena se zatem zmehta (*angl.* fuzzify) in obdela z naborem pravil za sklepanje. Rezultat obdelave je nova mehka vrednost, ki se na koncu izostrí (*angl.* defuzzify). Zaradi tega so končne stopnje zaupanja predstavljene kot realna števila z intervala $[-1, 1]$.

Nekaj ostalih primerov modelov na osnovi mehke logike lahko najdemo v [37–39]. V splošnem za njih velja, da (1) sprejmejo kvantitativne vhodne podatke, (2) jih spremenijo v mehke vrednosti, ki (3) se nato obdelajo z ustaljenimi postopki mehke logike in na koncu (4) pretvorijo rezultat nazaj v kvantitativno vrednost.

2.4.7 *Modeli, ki uporabljajo ostale informacije*

Kot primer modela, ki pri izračunu zaupanja uporablja tudi ostale informacije, si najprej oglejmo model Regret avtorjev Sabater in Sierra [22]. Regret izračunava zaupanje iz pridobljenih izkušenj, zbranih mnenj ter analiziranih družbenih relacij (*angl.* social relations) med agenti. Medtem ko so ocene v izkušnjah, vrednosti mnenj ter stopnje zaupanja predstavljene kot realna števila, $\Lambda = \Theta = \Omega \subseteq [0, 1]$, se za predstavitev družbenih relacij uporabljajo usmerjeni in uteženi grafi imenovani sociogrami (*angl.* sociograms). Ti vsebujejo informacije o naravi razmerij med agenti – sociogram denimo pove ali sta izbrana agenta v tekmovalnem ali kooperativnem razmerju. Na podlagi podatkov zapisanih v sociogramih se model odloči, od katerih agentov bo pridobil mnenja in kako jih bo vrednotil. V primerih, ko mnenj ni mogoče dobiti, model na podlagi družbenih razmerij celo sklepa na stopnje zaupanja po načelu *povej mi, s kom se družiš, in povem ti, koliko si zaupanja vreden*.

Model zaupanja, ki so ga predstavili Hermoso *et al.* [23, 40] pa poleg izkušenj in mnenj pri izračunu zaupanja uporablja informacije iz virtualnih organizacij (*angl.* virtual organizations). Virtualne organizacije so organizacijske strukture, v katerih morajo agenti, v kolikor želijo stopiti v želeno interakcijo, sprejeti določeno vlogo (*angl.* role), ki njihovo avtonomijo do neke mere omejuje [41]. Omenjen model zaupanja zato izkorišča informacije, ki so zapisane v definicijah vlog in relacijah med njimi. Denimo, če iz lastnih izkušenj in prejetih mnenj vemo, da je izbrani agent zaupanja vreden kadar nastopa v vlogi *A*, potem v tem modelu sklepamo, da bo isti agent v vlogi *B*, ki je vlogi *A* zelo podobna, prav tako zaupanja vreden.

2.4.8 Kvalitativno dinamično ocenjevanje

Kvalitativno dinamično ocenjevanje (*angl.* Qualitative Assessment Dynamics, QAD) [11, 20, 42–44] ni klasični model zaupanja, temveč simulacijsko ogrodje za modeliranje relacij zaupanja med agenti s človeškimi lastnostmi. Tako QAD modelira različne dejavnike kot so iracionalnost, subjektivnost, družbeno vpetost ipd. Za potrebe takšnega modeliranja so avtorji izvedli raziskavo [13], ki je pokazala, da ljudje za izražanje ocen, vrednosti mnenj in stopenj zaupanja raje kot kvantitativne uporabljajo kvalitativne urejenostne vrednosti. Tako QAD uporablja petstopenjsko lestvico $\Lambda = \Theta = \Omega = \{D < PD < U < PT < D\}$, kjer elementi zaporedoma pomenijo nezaupanje (*angl.* distrust), delno nezaupanje (*angl.* partial distrust), neodločenost (*angl.* undecidedness), delno zaupanje (*angl.* partial trust) in zaupanje (*angl.* trust). A ker gre za simulacijsko ogrodje, QAD nima mehanizmov za odpravo lažnih mnenj in zato v okoljih, kjer so lažna mnenja pogosta, ne dosega dobrih rezultatov. Kljub temu, da QAD na vhodu sprejema kvalitativne vhodne podatke in prav tako podaja zaupanje kvalitativno, model ni v celoti kvalitativen, saj se pri obdelavi podatkov pogosto zateče k kvantitativnim metodam. Pri tem vhodne kvalitativne podatke pretvori v kvantitativne, jih obdela in na koncu rezultat pretvori nazaj v kvalitativno domeno; tako delujeta npr. Kovačev [42] in Trčkov [44] model.

2.4.9 Sklep in povzetek odprtih problemov

Predstavljen seznam modelov zaupanja ni izčrpen – obširne preglede je moč najti v [9, 14, 21, 35] – a lahko kljub temu podamo nekaj sklepnih ugotovitev. Iz pregleda izhaja, da so najpogostejši kvantitativni ter hibridni modeli, medtem ko je kvalitativnih precej manj. Glavna prednost kvantitativnih modelov je v tem, da uporabljajo ustaljene in učinkovite računske postopke, medtem ko za njihovo glavno pomanjkljivost velja, da so za navadne uporabnike prekompleksni in težko razumljivi [9]. Ljudje namreč raje podajamo ocene in mnenja ter izražamo zaupanje kvalitativno [13], saj nam je interpretacija kvantitativnih podatkov (denimo verjetnosti) tuja in nerazumljiva [10]. Tako lahko zaključimo, da je uporaba modelov zaupanja v sistemih, kjer podatke podajajo in interpretirajo ljudje, neprimerna. Hibridni modeli v takih okoljih predstavljajo izboljšavo, saj sprejemajo kvalitativne vhodne podatke, a so ljudem še vedno težko razumljivi, saj so njihovi izračuni kvantitativni. Kvalitativnih modelov je občutno manj in tudi ti pogosto obravnavajo kvalitativne podatke kot kvantitativne. Takšna praksa ni

neobičajna in je v nekaterih disciplinah celo prevladujoča. Kljub temu obstajajo precej burne razprave o tem, ali so takšne operacije sploh dopustne [45, 46]. Denimo Kahler *et al.* [47] so pokazali, kako lahko zgolj izračun povprečne vrednosti nad urejenostnimi kvalitativnimi podatki privede do napačnih ugotovitev. Tako lahko ugotovimo, da so obstoječi kvalitativni modeli sicer primerni za uporabo v sistemih, kjer podatke podajajo in interpretirajo ljudje, a hkrati ti modeli s podatki ne ravnajo primerno. Njihova dodatna težava je tudi v tem, da so v primerjavi s kvantitativnimi manj natančni, saj jih omejuje grobost (*angl.* coarseness) kvalitativnih podatkov.

2.5 Pregled načinov merjenja učinkovitosti modelov zaupanja

Ker je cilj modelov zaupanja znan – preprečevati interakcije s slabonamernimi ter zanesljivimi agenti in promovirati tiste z dobronamernimi ter zanesljivimi – lahko govorimo o tem, kako dobri so modeli zaupanja pri doseganju tega cilja. Merjenje učinkovitosti delovanja modelov zaupanja je tako postopek, v katerem preverjamo, kako dobri so modeli pri izračunu zaupanja. Merjenju učinkovitosti pogosto pravimo tudi testiranje ali evaluacija. Težava pri evaluaciji je, da so modeli zaupanja – kot smo prikazali v prejšnjem razdelku – precej različni: medtem ko se njihovi izhodi razlikujejo zgolj po vrsti (kvalitativni, kvantitativni), se njihovi vhodi razlikujejo še po vsebini (izkušnje, mnenja, strukture družbenih omrežij, strukture navideznih organizacij in drugo). V tem pogledu je področje obvladovanja zaupanja precej manj ustaljeno kot denimo področje prepoznavne vzorcev (*angl.* pattern recognition), kjer so v klasifikacijskih problemih vhodi in izhodi algoritmov dobro znani.

Ključna komponenta izračunanega zaupanja so stopnje zaupanja, ki predstavljajo količino zaupanja do agentov (glej Definicijo 5). Te se uporabijo v različnih odločitvenih procesih (*angl.* decision making processes). Ker naj bi dobri modeli zaupanja vodili k sprejemanju dobrih odločitev, se pri vrednotenju modelov zaupanja pogosto vrednoti kakovost odločitev agentov, ki te modele uporabljajo: odločitve bi naj bile največji skupni imenovalce vseh modelov zaupanja. A vendar obstajajo tudi drugačni pristopi, zato bomo v nadaljevanju pri pregledu obstoječih načinov evaluacije modelov zaupanja – poimenovanih modeli evaluacije – ločeno obravnavali tiste, ki testirajo modele zaupanja posredno, tako da vrednotijo kakovost odločitev, ki iz zaupanja izhajajo, in tiste, ki testirajo modele zaupanja neposredno, tako da vrednotijo zaupanje samo.

Pred tem si pa oglejmo še načine, kako so lahko modeli evaluacije realizirani. Tu imamo dve možnosti, in sicer *ad-hoc* testiranja (*angl.* ad-hoc evaluation) in namenska

testna okolja (*angl.* dedicated testbed). *Ad-hoc* testiranje navadno nastane kot “stranski produkt” pri predstavitvi novega modela zaupanja, zato je primerno za evalvacijo spremnega modela zaupanja, a hkrati premalo splošno, da bi lahko bilo širše uporabno. Bolj kot ozka uporabnost pa je problematičen njegov namen: *ad-hoc* testiranje je namenjeno promociji spremnega modela zaupanja, zato običajno prikaže zgolj njegove prednosti ne pa pomanjkljivosti. Kljub temu so *ad-hoc* testiranja postala nekakšen standard pri predstavitvi novih modelov zaupanja in predstavljajo dober vir modelov evalvacije. Drugi način realizacije pa predstavljajo namenska testna okolja, ki so samostojni računalniški programi, katerih snovalci niso nujno avtorji modelov zaupanja. Takšna okolja morajo biti dovolj splošna, da so uporabna za čim večji krog modelov zaupanja in zato predstavljajo korak naprej od *ad-hoc* testiranj. Posledično so tudi manj pristranska, saj njihovi avtorji niso “obremenjeni” z rezultati določenega modela zaupanja.

2.5.1 Vrednotenje zaupanja preko odločitev

V tem razdelku predstavljeni modeli evalvacije vrednotijo modele zaupanja tako, da vrednotijo odločitve agentov, ki modele zaupanja uporabljajo. Ker so lahko odločitve zelo različne, bomo v nadaljevanju ločeno obravnavali tiste, ki vrednotijo odločitve v igri iterativne zapornikove dileme (*angl.* iterated prisoner dilemma), in tiste, ki vrednotijo domensko pogojene odločitve. Del agenta, ki izračunano zaupanje pretvori v odločitev, bomo v nadaljevanju imenovali *odločitveni model*.

Modeli evalvacije za odločitve v igri iterativna zapornikova dilema

Ti modeli evalvacije vrednotijo odločitve agentov v igri iterativne zapornikove dileme. Zapornikova dilema je igra z neničelno vsoto, v kateri dva igralca sočasno – ne da bi vedela za dejanje nasprotnika – izbirata med dvema potezama: *sodeluj* (*angl.* cooperate) ali *goljufaj* (*angl.* defect). Glavna ideja igre je v tem, da medtem ko obema igralcema obojestransko sodelovanje koristi, saj vsak prejme nagrado S , oba igralca tudi mika goljufanje: če eden od njiju sodeluje, medtem ko drugi goljufa, goljuf priigra korist G , kjer je $G > S$, medtem ko ogoljufani igralec prejme miloščino M , pri čemer $G > S > M$. A v primeru, ko oba igralca goljufata, vsak prejme kazen K , kjer $G > S > K > M$. Slednje pomeni, da tudi v primeru obojestranskega goljufanja vsak igralec prejme več, kot bi prejel, če bi bil ogoljufan (tj. če bi sam sodeloval, nasprotnik pa bi goljufal). Takšna shema narekuje, da bo vsak racionalen igralec – takšen, ki skuša maksimirati

skupno vsoto dobitkov brez obzira do nasprotnika – vedno goljufal.

V iterativni različici zapornikove dileme isti par igralcev igra večkrat. Tako ima vsak igralec priložnost, da nasprotnikovo goljufanje kaznuje; v preteklem krogu ogoľufani igralec bo lahko v prihodnjem sam goljufal. Če pa med igralci obstaja dovolj zaupanja, se ti lahko odločijo za obojestransko sodelovanje, saj bodo, če se to razvije, na dolgi rok pridobili več [48].

Mnogi so uporabili iterativno različico zapornikove dileme za eksperimentiranje z modeli zaupanja ter za njihovo vrednotenje. V nadaljevanju sledi opis nekaterih pristopov. Hkrati z verjetno prvim računskim modelu zaupanja je Marsh [49] razvil simulacijo *the PlayGround*. Z njo je demonstriral, kako različne konfiguracije predlaganega modela zaupanja vodijo do vzpostavitve relacij zaupanja med agenti tj. pripravljenosti za sodelovanje v igri iterativne zapornikove dileme. Omenjena simulacija dejansko ni bila *pravi* model evaluacije, temveč prvi eksperiment, ki je povezal računski model zaupanja z eksperimentom zapornikove dileme. Podobno je Mui [50] preverjal uspešnost strategije za igranje zapornikove dileme, ki vključuje elemente modela ugleda, imenovane *milo-za-drago-z-ugledom* (*angl.* reputation-tit-for-tat), v primerjavi s klasičnimi strategijami kot so vedno goljufaj (*angl.* always defect), vedno sodeluj (*angl.* always cooperate) ter *milo-za-drago* (*angl.* tit-for-tat). Podobno kot *the PlayGround*, omenjen eksperiment ni pravi model evaluacije, temveč simulacija o vzajemnih učinkih med zaupanjem (ugledom) ter sodelovanjem.

Schillo [51] je predstavil model evaluacije, ki temelji na *razkriti* iterativni zapornikovi dilemi z izbiro partnerjev. V njem vsak agent izbere soigralca po vnaprej določenem protokolu: agenti najprej plačajo stave (*angl.* stakes) in javno napovejo svoje poteze. Protokol nato določi pare igralcev glede na napovedi in stave. A v dejanski igri lahko agenti svoje napovedi tudi kršijo: prav mogoče je, da agent napove sodelovanje, v dejanski igri pa goljufa. Po koncu vsake igre se zasluzki izplačajo in rezultati javno – a le v delnem obsegu – objavijo, tako da lahko drugi agenti vidijo, kdo se je obnašal pošteno. Opisan postopek se nato večkrat ponovi in končni zmagovalec je agent z največjim zasluzkom. Avtorji so z omenjenim modelom evaluacije pokazali, da je predlagan model zaupanja, *TrustNet*, učinkovit tako pri identifikaciji slabonamernih agentov kot tudi pri izbiri strategije. Omenjeno delo je tako prvi pravi model evaluacije, ki temelji na zapornikovi dilemi. Testno okolje je bilo realizirano *ad-hoc*.

Nedavno sta Salehi-Abari in White [52] predlagala namensko testno okolje DART (*angl.* framework for a distributed analysis of reputation and trust models). V okolju

DART se agenti povezujejo v družbeno mrežo (*angl.* social network), v kateri lahko medsebojno komunicirajo in izvajajo interakcije le neposredno povezani agenti: množica agentov, s katerimi je nek agent povezan, se imenuje njegova soseska (*angl.* neighborhood). Vsak agent nato nenehno igra igro zapornikova dilema z vsemi agenti v njegovi soseski. Poleg tega lahko vsak agent zaprosi svoje sosede za mnenja o agentih iz njegove soseske, vsak agent lahko prejme priporočila o tem, koga vključiti v svojo sosesko in analogno, vsak agent lahko zaprosi sosede, da ga priporočijo svojim sosedom za vključitev v njihovo sosesko. In nenazadnje, vsak agent lahko kadarkoli iz svoje soseske agente tudi odstrani. Za sprejemanje vseh teh odločitev agenti uporabljajo modele zaupanja in različne politike. Glavni metriki v DART sta torej koristnost, ki jo agent pridobi pri igranju zapornikove dileme, ter število povezav, ki jih vsak agent iz svoje soseske odstrani.

Chandrasekaran in Esfandiari [53] predlagata model evaluacije, ki sicer temelji na teoriji grafov, a vsebuje nekatere podobnosti z igro zapornikove dileme, zato ga obravnavamo v tem razdelku. Kot modeli evaluacije na osnovi zapornikove dileme, je namreč tudi ta pristop splošen in zahteva od modelov zaupanja odločitve: zaupati ali ne zaupati. Model evaluacije zgradi zaporedoma tri usmerjene grafe: graf informacij (*angl.* feedback graph), graf ugleda (*angl.* reputation graph) in graf zaupanja (*angl.* trust graph). V vsakem grafu vozlišča predstavljajo agente, medtem ko je pomen povezav pri vsakem drugačen: v grafu informacij povezave predstavljajo vhode v model zaupanja (izkušnje, mnenja), v grafu ugleda izračunane stopnje zaupanja, v grafu zaupanja pa binarno zaupanje – povezava od agenta a do agenta b pomeni, da agent a v celoti zaupa agentu b , medtem ko bi odsotnost take povezave pomenila, da a v celoti ne zaupa b . Evaluacija poteka tako, da program generira informacije (izkušnje, mnenja), katere agenti z uporabo modelov zaupanja pretvorijo v graf ugleda. Nato agenti z uporabo nekega odločitvenega postopka, npr. upragovljenje (*angl.* thresholding), graf ugleda pretvorijo v graf zaupanja. Vrednotenje poteka tako, da se izračunani graf zaupanja primerja z grafom, ki bi nastal, če bi model zaupanja deloval perfektno.

Modeli evaluacije za domensko pogojene odločitve

Modeli evaluacije iz tega razdelka prav tako vrednotijo modele zaupanja po kakovosti odločitev, a se od modelov evaluacije iz prejšnjega razdelka razlikujejo v tem, da so odločitve domensko pogojene. Tako so rezultati teh vrednotenj v okviru podanih domen bolj razumljivi, jih je pa težje posplošiti.

V to skupino lahko uvrstimo sploh prvo namensko testno okolje ART (*angl.* Agent Reputation and Trust) [54]. Postopek vrednotenja se v ART izvaja kot tekmovanje, v katerem agenti nastopajo v vlogi ocenjevalcev vrednosti slik. Pri tem lahko vsak agent uporabi lastno presojo, ki mu jo na začetku evaluacije določi testno okolje, ali pa za pomoč povpraša druge agente. Agent lahko uporabi model zaupanja pri dveh nalogah: da ugotovi, katere agente lahko povpraša za pomoč pri ocenjevanju slik, in da ugotovi, koliko so tako pridobljene informacije sploh veljavne. Na koncu vsak agent odda cenitev, za katero mu testno okolje izplača *zaslužek*, ki je sorazmeren natančnosti cenitve. V nadaljnjih krogih lahko agent pridobljen zaslužek porabi za nakup dodatnih cenitev od drugih agentov ali za nakup mnenj o drugih agentih. Kot glavna metrika se uporablja končni zaslužek agenta.

Da bi odpravila nekatere pomanjkljivosti okolja ART, sta Kerr in Cohen predlagala okolje TREET (*angl.* Trust and Reputation Experimentation and Evaluation Testbed) [55]. Model evaluacije v TREET obsega tržnico (*angl.* marketplace), na kateri se srečujejo kupci in prodajalci. Agenti nastopajo v vlogi kupcev, medtem ko so prodajalci v celoti pod nadzorom testnega okolja. Testno okolje vsakemu agentu dodeli seznam predmetov, ki jih mora v določenem časovnem oknu kupiti, pri čemer lahko svobodno izbira prodajalce. Slednji pa lahko pri trgovanju goljufajo, saj lahko oglašujejo predmete, ki jih dejansko nimajo, ali pa predmeta, potem ko je kupec zanj že plačal, ne pošljejo. Testno okolje namreč modelira časovni zamik informacij (*angl.* information lag), tako da kupci nek (vnaprej določen) čas ne izvejo ali je kupčija uspela (ali pa so bili ogoljufani). Za preprečevanje goljufij agenti uporabljajo modele zaupanja, da ugotovijo, kateri prodajalci so zaupanja vredni. Glavna metrika primerja zaslužke poštenih in nepoštenih trgovcev. Če model zaupanja deluje dobro, potem bodo pošteni trgovci zaslužili več kot nepošteni.

S področja računalniških omrežij prihajata dve namenski testni okolji, in sicer program TRMSim-WSN (*angl.* Trust & Reputation Models Simulator for Wireless Sensor Networks) [56, 57] ter program P2P-Sim (*angl.* P2P Simulator) [58]. Okolje TRMSim-WSN vrednoti modele zaupanja in ugleda namenjene za uporabo v brezžičnih senzorskih omrežjih (*angl.* wireless-sensor network, WSN) [59]. Program simulira delovanje teh omrežij tako, da ustvari topologijo vozlišč, jih opremi z modeli zaupanja, določi njihovo obnašanje – katera bodo dobro- in katera slabo-namerna – in simulira omrežni promet. Cilj vsakega vozlišča je pošiljati pakete dobrim vozliščem ter se izogibati slabim. Pri usmerjanju prometa si vsako vozlišče zato pomaga z modelom

zaupanja. Tako je glavna metrika natančnost (*angl.* accuracy), ki predstavlja odstotek vozlišč, ki pošiljajo pakete dobronamernim vozliščem. Drugo testno okolje, P2P-Sim, simulira delovanje omrežja vsak-z-vsakim (*angl.* peer-to-peer, P2P), kjer vozlišča izmenjujejo datoteke. Podobno kot TRMSim-WSN okolje P2P-Sim zgradi topologijo omrežja, določi obnašanje vozlišč (slabo- ali dobronamerno), definira nabor datotek, ki jih vsako vozlišče ima, ter nabor datotek, ki jih vsako vozlišče še mora pridobiti, nakar prične simulacijo. Med izvajanjem si vozlišča izmenjujejo datoteke, pri čemer je cilj dobronamernih pridobiti le prave datoteke, cilj slabonamernih vozlišč pa razširiti čim več okuženih. Za doseg tega cilja si vsako vozlišče pomaga z modelom zaupanja. Glavna metrika v P2P-Sim je učinkovitost (*angl.* effectiveness), ki predstavlja delež prenosov veljavnih datotek med dobronamernimi vozlišči.

2.5.2 Vrednotenje zaupanja neposredno

V nasprotju z modeli evaluacije iz prejšnjega razdelka, tukaj opisani modeli evaluacije ocenijo izračunano zaupanje neposredno. Pri tem domnevajo, da vsi modeli zaupanja uporabljajo enake (standardne) stopnje zaupanja, denimo vrednosti z intervala $[0, 1]$. Izračunano zaupanje se tipično vrednoti tako, da se primerja s pravimi vrednostmi (*angl.* ground truth), tj. z vrednostmi, ki bi jih izračunali nezmotljivi modeli zaupanja. Glavna prednost takega modela evaluacije je v tem, da je neposreden, saj ne zahteva, da agenti sprejemajo odločitve. Njegova glavna pomanjkljivost pa je, da je zaradi domneve o standardnih stopnjah zaupanja manj splošen, saj lahko medsebojno primerja zgolj modele, ki uporabljajo iste stopnje zaupanja. Zaradi slednjega so takšni pristopi precej redkejši.

Verjetno prvi tovrstni model evaluacije sta uporabila Zacharia in Maes pri predstavitvi modelov zaupanja Histos in Sporas [60]. Zasnovala sta eksperiment, v katerem sta merila število interakcij, ki jih model zaupanja potrebuje, da izračunano zaupanje konvergira k pravim vrednostim. Prave vrednosti so tiste, ki jih določi testno okolje. Podoben pristop je bil uporabljen pri vrednotenju modelov Travos [31] ter Habit [61], kjer avtorji izračunavajo povprečno absolutno napako (*angl.* mean absolute error) med izračunanim in dejanskim zaupanjem. Omenjena pristopa tako zahtevata, da so stopnje zaupanja izražene v vnaprej določenem formatu: prvi na intervalu celih števil $[0, 3000]$ drugi pa na zveznem intervalu realnih števil $[0, 1]$. Sicer sta bili obe testiranji izvedeni *ad-hoc*, a je bil prvi eksperiment [60] ponovno uporabljen za oceno modelov Regret [22] in Afras [34].

2.5.3 Sklep in povzetek odprtih problemov

Iz pregleda modelov evaluacije lahko zaključimo, da je teh v primerjavi z modeli zaupanja precej manj. Predstavili smo neposreden in posreden način evaluacije, od katerih ima vsak svoje prednosti in slabosti. Neposredna evaluacija tako zahteva, da modeli zaupanja uporabljajo vnaprej dogovorjene stopnje zaupanja. Njena prednost je v preprostih in nazornih metrikah, zaradi zahteve o enakih stopnjah zaupanja pa je njena uporabnost zelo omejena. Vrednotenje modelov preko vrednotenja odločitev pa zahteva, da modeli (oz. agenti, ki jih uporabljajo) sprejemajo odločitve. Ker je slednje mogoče pri vsakem modelu zaupanja – ne glede na to ali je kvanti- ali kvalitativen – je taka evaluacija bolj splošna. Slabost tovrstnega vrednotenja pa je v tem, da je delno in posredno. Delno zaradi tega, ker ne ocenjujemo vsega izračunanega zaupanja, temveč le tisto, ki privede do odločitev: lahko izračunamo zaupanje do 100 agentov, a če gremo v interakcijo le z enim, smo posledično ocenili le zaupanje do tistega, do ostalih 99 pa ne. Posredno pa zaradi tega, ker vrednotimo odločitve – ki sicer izhajajo iz zaupanja – in ne zaupanje samo. Slednje sicer ne bi bilo problematično, če bi bili modeli zaupanja predstavljeni skupaj z odločitvenimi modeli. Običajno pa je situacija obratna: modeli zaupanja so predstavljeni kot črne škatle, ki na izhodih vračajo zaupanje, o uporabi slednjega pa navadno ni govora. To je razumljivo, saj so take odločitve domensko pogojene, modeli zaupanja pa bi naj bili širše uporabni. Ta problem se pri evaluacijah premosti tako, da se modelu zaupanja priredi arbitraren odločitveni model, kar pa je lahko problematično, saj odločitveni model določa t. i. razmerje med raziskovanjem (*angl.* exploration) in izrabljanjem (*angl.* exploitation). To razmerje opisuje tendence agenta pri pridobivanju novega znanja: v nekem trenutku se lahko agent zanaša na že izračunano zaupanje in gre v interakcijo s tistim, za katerega misli, da je najbolj zaupanja vreden, ali pa pridobi dodatne izkušnje (z novimi, nepoznanimi agenti) in tako obstoječe izračune zaupanja izboljša. V prvem primeru je pridobljeno zaupanje *izkoristil*, da je izbral čim boljšega partnerja, v drugem pa *raziskal* okolje, da bi izboljšal prihodnje izračune zaupanja. Ker pa se lahko modeli zaupanja obnašajo različno glede na količino in obseg izkušenj, postane takšno vmešavanje odločitvenega modela, ki je prisoten zgolj za potrebe evaluacije, problematično. Privede lahko namreč do situacije, kjer modele zaupanja vrednotimo z različnimi vhodnimi podatki; ko agent (oz. odločitveni model) sprejema odločitve (denimo s kom iti v interakcijo), dejansko nadzoruje, katere izkušnje (oz. s kom) bo agent pridobil. Ker izkušnje pred-

stavlja prihodnje vhodne podatke, odločitveni model tako dejansko izbira podatke, iz katerih bo model zaupanja vrednoten. V zaključku zapišimo, da navkljub mnogim obstoječim načinom evaluacij še nimamo takega, ki bi bil dovolj splošen, da bi zmožni primerjati tako kvalitativne kot kvantitativne modele, ali takega, ki bi lahko modele zaupanja vrednotil brez odločitvenih modelov.

Kvalitativni podatki

3.1 Uvod

Ljudje se rajši kot natančno in s števkami izražamo približno in z besedami. Tako nam je bližje reči, da je včeraj padlo veliko dežja, kot pa natančno podati volumen padle vode na ploščinsko enoto. Podobno bi lahko trdili tudi za domeno obvladovanja zaupanja, saj smo že v pregledu področja navedli primere, ki kažejo, da so ljudem številke neintuitivne [10], in da pri obvladovanju rajši uporabljamo kvalitativne vrednosti [13]. V tem poglavju predstavimo množico kvalitativnih vrednosti in metodo, ki iz zbirke več kvalitativni vrednosti izbere najbolj reprezentativno.

3.2 Množica kvalitativnih vrednosti

Kot kvalitativne vrednosti bomo v tem delu obravnavali vrednosti, ki so medsebojno primerljive z relacijami *večji*, *manjši* ali *enak*, a hkrati med njimi ni definirane *razdalje*. Neformalno to pomeni, da lahko za dve poljubni kvalitativni vrednosti ugotovimo, katera je večja, ne moremo pa reči, kolikšna je ta razlika. Ko je Stevens [62] definiral merse lestvice, je lestvico takih vrednosti poimenoval urejenostna lestvica. Množico kvalitativnih vrednosti in njihov vrstni red podaja Definicija 11.

Definicija 11 (Kvalitativne vrednosti): Naj množica $\mathbb{W} = \{ZS, S, N, D, ZD\}$ predstavlja linearno urejeno množico, kjer elementi zaporedoma predstavljajo vrednosti *zelo slabo*, *slabo*, *nevtravno*, *dobro* in *zelo dobro*. Za podane elemente naj velja naslednji vrstni red $ZS < S < N < D < ZD$. Za ugotavljanje vrstnega reda elementov znotraj množice definiramo funkcijo $RANG : \mathbb{W} \mapsto \mathbb{N}$, ki elementom ZS, S, N, D, ZD zaporedoma priredi vrednosti 1, 2, 3, 4, 5.

Množico kvalitativnih vrednosti bomo v nadaljevanju uporabili pri konstrukciji modela zaupanja. A pred tem moramo definirati še eno izmed najpogostejših operacij nad elementi kvalitativne množice.

3.3 Izračun reprezentativne kvalitativne vrednosti

Pri obdelavi zbirke kvalitativnih vrednosti pogosto želimo izračunati njihovo *reprezentativno* vrednost, tj. vrednost, ki najbolje predstavlja vse vrednosti v zbirki. Tu imamo na izbiro kar nekaj možnosti. Ker imajo elementi določen vrstni red lahko – po pravilu

Stevensovih merskih lestvic – izračunamo *mediano* ali *modus*. Mnogi se v takih situacijah zatečejo k *povprečenju kvalitativnih vrednosti*; tako npr. Kovač [20] izračuna povprečje rangov kvalitativnih vrednosti in rezultat zaokroži k najbližjemu celemu številu. Reprezentativna kvalitativna vrednost je tako tista, katere rang sovпада z izračunanim številom. Takšen postopek določevanja reprezentativne vrednosti ni neobičajen in je na področju družboslovja in humanistike pri obdelavi Likertovih vprašalnikov precej pogost. A v strogem smislu so takšne operacije na urejenostnih lestvicah nedopustne, *saj moramo za njihovo izvedbo o podatkih vedeti nekaj več kot le njihov relativni vrstni red*.¹ [62]. Opisani pristopi tako uporabljajo predpostavko *ekvidistančnosti* med kvalitativnimi vrednostmi: po njej je denimo razlika med ZS in N enaka kot denimo razlika med S in D . Alternativni pristopi za izračun reprezentativne vrednosti prihajajo iz teorije mehkih množic. Takšna sta primera uteženih različic maksimalnih in minimalnih operatorjev, ki sta jih predstavila Dubois in Prade [63]. Herrera, Viedma in Verdegay [64] pa so pripravili izračunu povprečja podobno operacijo LOWA (*angl.* Linguistic Ordered Weighted Averaging), ki implicitno pretvarja med kvalitativnimi in kvantitativnimi vrednostmi. Prav tako je svojo različico kvalitativnih operatorjev OWA (*angl.* Ordered Weighted Averaging) pripravil Yager [65]. Na koncu omenimo še delo raziskovalcev Godo in Torra [66], v katerem predlagata operator, ki je v celoti kvalitativen in spominja na izračun povprečne vrednosti. Njun pristop temelji na definiciji dodatnih asociativnih operatorjev, ki igrajo vlogo seštevanja in normiranja.

Medtem ko vsi opisani pristopi izračunajo reprezentativno vrednost pa nobeden od njih ne omogoča, da je postopek izračuna prilagodljiv. Pomen kvalitativnih vrednosti je namreč lahko za vsakega posameznika drugačen in takšen je lahko tudi izračun reprezentativne vrednosti; ni si težko predstavljati dveh uporabnikov, ki bi iz istega seznama kvalitativnih vrednosti izračunala drugačno reprezentativno vrednost. Nekateri pristopi naslavljajo omenjeno neposredno: denimo v Marshovem modelu zaupanja [49] lahko agent izračuna oceno zaupanja (reprezentativno vrednost iz seznama zbranih podatkov) kot optimist, pesimist ali realist, v ogrodju QAD [11, 42] agenti za ta namen uporabljajo različne operatorje, v modelu avtorjev Griffiths, Chao in Younas [36] pa se vsak agent odloča, kako bo zaupal ob odsotnosti podatkov. Da bi realizirali takšen subjektivni izračun reprezentativne vrednosti nad zbirko kvalitativnih podatkov, predlagamo novo metodo, ki temelji na primerjanju distribucij. Na začetku za

¹V izvorniku: "For such statistics imply a knowledge of something more than the relative rank order of data."

neko zbirko kvalitativnih vrednosti definiramo njen *porazdelitveni vektor*.

Definicija 12 (Porazdelitveni vektor): Naj vektor $\vec{p} = [f_{ZS}, f_S, f_N, f_D, f_{ZD}]$ predstavlja porazdelitev (kvalitativnih) vrednosti v neki zbirki, kjer komponente $f_{ZS}, f_S, f_N, f_D, f_{ZD}$ zaporedoma predstavljajo relativne frekvence vrednosti ZS, S, N, D, ZD .

Posamezno komponento vektorja \vec{p} bomo označevali z indeksi, denimo $\vec{p}(D) = f_D$. Za ilustracijo vzemimo zbirko vrednosti $ZS, ZS, S, S, N, ZD, ZD, ZD$. Njen porazdelitveni vektor je tako $\vec{p} = [\frac{2}{8}, \frac{2}{8}, \frac{1}{8}, \frac{0}{8}, \frac{3}{8}]$, medtem ko $\vec{p}(ZD) = \frac{3}{8}$ označuje relativno frekvenco vrednosti ZD . V naslednjem koraku bomo za vsako kvalitativno vrednost iz množice \mathbb{V} ustvarili pripadajoč porazdelitveni vektor. Slednji bo predstavljal porazdelitev kvalitativnih vrednosti, katero *bi pričakovali*, če bi nam nekdo dejal, da je kvalitativna vrednost, kateri ta vektor pripada, reprezentativna vrednost omenjene porazdelitve. Takšne porazdelitvene vektorje bomo imenovali *reprezentativni porazdelitveni vektorji*.

Definicija 13 (Reprezentativni porazdelitveni vektor): Podani porazdelitveni vektorji zaporedoma predstavljajo vrednosti ZS, S, N, D, ZD : $\vec{r}_{ZS} = [1, 0, 0, 0, 0]$, $\vec{r}_S = [0, 1, 0, 0, 0]$, $\vec{r}_N = [0, 0, 1, 0, 0]$, $\vec{r}_D = [0, 0, 0, 1, 0]$, $\vec{r}_{ZD} = [0, 0, 0, 0, 1]$. Omenjene vektorje imenujemo reprezentativni porazdelitveni vektorji, množico vseh reprezentativnih porazdelitvenih vektorjev pa označimo z $\mathbf{R} = \{\vec{r}_{ZS}, \vec{r}_S, \vec{r}_N, \vec{r}_D, \vec{r}_{ZD}\}$.

Reprezentativni porazdelitveni vektorji so tako ključni dejavnik pri izračunu reprezentativne vrednosti; pri drugačnih reprezentativnih vektorjih bo tudi izračunana reprezentativna vrednost drugačna. Tako vektorji iz Definicije 13 služijo le kot primer, ki ga bomo uporabili v nadaljevanju.

Delovanje predlagane metode lahko opišemo sledeče. Reprezentativno vrednost neke zbirke dobimo tako, da najprej izračunamo njen porazdelitveni vektor in ga nato primerjamo z vsemi reprezentativnimi. Reprezentativna vrednost je tako tista, ki pripada reprezentativnemu porazdelitvenemu vektorju, ki je vektorju zbirke najbolj *podoben*. Slednje je formalizirano v Definiciji 14.

Definicija 14: S \vec{p} označimo porazdelitveni vektor neke distribucije in z $d(\vec{x}, \vec{y})$ razdaljo med porazdelitvenima vektorjema \vec{x} in \vec{y} kot je podano v enačbi (3.1). Reprezentativna vrednost zbirke je tista vrednost $v \in \mathbb{V}$, ki pripada reprezentativnemu

porazdelitvenemu vektorju \vec{r}_v , ki ima najkrajšo razdaljo do porazdelitvenega vektorja zbirke \vec{p} .

$$\text{Reprezentativna}(\vec{p}) = \max_{v \in \mathbb{V}} \operatorname{argmin} d(\vec{p}, \vec{r}_v)$$

Lahko se zgodi, da je porazdelitveni vektor zbirke enako oddaljen od več reprezentativnih porazdelitvenih vektorjev in da so te razdalje hkrati tudi najkrajše. Takrat bo operator argmin vrnil več vrednosti, operator \max pa bo izbral največjo in tako zagotovil enolično rešitev. Operator \max uporabimo zato, da je metoda skladna z zaokroževanjem, kjer se denimo število 1.5 zaokroži k številu 2 in ne k 1, kljub temu, da je enako oddaljeno od obeh.

Ostane nam le še definicija razdalje med porazdelitvenimi vektorji. Podajamo ga v enačbi (3.1). Razdalja se izračuna kot preprosta evklidska razdalja med kumulativno različico porazdelitvenih vektorjev \vec{x} in \vec{y} . Kumulativna različica je uporabljena zato, ker upošteva, da imajo kvalitativne vrednosti definiran vrstni red.

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{v_1 \in \mathbb{V}} \left(\sum_{\substack{v_2 \leq v_1 \\ v_2 \in \mathbb{V}}} (\vec{x}(v_2) - \vec{y}(v_2)) \right)^2} \quad (3.1)$$

Za prikaz delovanja metode bomo z uporabo reprezentativnih porazdelitvenih vektorjev iz Definicije 13 izračunali reprezentativno vrednost v naslednji zbirki: $5 \times ZS$, $10 \times S$, $25 \times N$, $6 \times D$ in $4 \times ZD$. Ko preštejemo vse pojavitve in normiramo frekvence, dobimo porazdelitveni vektor $\vec{p} = [0.10, 0.20, 0.50, 0.12, 0.08]$. Da bo izračun lažji, bomo porazdelitvene vektorje preoblikovali v kumulativne različice in izračunali evklidsko razdaljo med njimi. Tako je kumulativna različica vektorja \vec{p} vektor $[0.10, 0.30, 0.80, 0.92, 1.00]$. Na podoben način preoblikujemo še reprezentativne porazdelitvene vektorje. Nato izračunamo razdaljo med \vec{p} in vsakim reprezentativnim porazdelitvenim vektorjem.

$$d(\vec{p}, \vec{r}_{ZS}) = \sqrt{(1 - 0.10)^2 + (1 - 0.30)^2 + (1 - 0.80)^2 + (1 - 0.92)^2} = 1.16$$

$$d(\vec{p}, \vec{r}_S) = \sqrt{(0 - 0.10)^2 + (1 - 0.30)^2 + (1 - 0.80)^2 + (1 - 0.92)^2} = 0.74$$

$$d(\vec{p}, \vec{r}_N) = \sqrt{(0 - 0.10)^2 + (0 - 0.30)^2 + (1 - 0.80)^2 + (1 - 0.92)^2} = 0.38$$

$$d(\vec{p}, \vec{r}_D) = \sqrt{(0 - 0.10)^2 + (0 - 0.30)^2 + (0 - 0.80)^2 + (1 - 0.92)^2} = 0.86$$

$$d(\vec{p}, \vec{r}_{ZD}) = \sqrt{(0 - 0.10)^2 + (0 - 0.30)^2 + (0 - 0.80)^2 + (0 - 0.92)^2} = 1.26$$

Ker ima reprezentativni porazdelitveni vektor \vec{r}_N najkrajšo razdaljo do porazdelitvenega vektorja zbirke \vec{p} ($d(\vec{p}, \vec{r}_{ki}) = 0.38$ je najkrajša od vseh razdalj) lahko zaključimo, da je reprezentativna vrednost podane zbirke vrednost N .

3.4 Dve lastnosti predlagane metode

Na koncu tega poglavja zgolj omenimo dva izreka, ki opisujeta obnašanje predlagane metode. Dokaz je podan v dodatku A.

Izrek 1 (Enakost z mediano): Če množica reprezentativnih porazdelitvenih vektorjev vsebuje vektorje $\vec{r}_{ZS} = [1, 0, 0, 0, 0]$, $\vec{r}_S = [0, 1, 0, 0, 0]$, $\vec{r}_N = [0, 0, 1, 0, 0]$, $\vec{r}_D = [0, 0, 0, 1, 0]$, $\vec{r}_{ZD} = [0, 0, 0, 0, 1]$, metoda izračuna reprezentativne vrednosti vrne enak rezultat kot mediana.

Izrek 2 (Enakost s povprečjem rangov): Če množica reprezentativnih porazdelitvenih vektorjev vsebuje naslednje vektorje $\vec{r}_{ZS} = [1, 0, 0, 0, 0]$, $\vec{r}_S = [\frac{3}{4}, 0, 0, 0, \frac{1}{4}]$, $\vec{r}_N = [\frac{1}{2}, 0, 0, 0, \frac{1}{2}]$, $\vec{r}_D = [\frac{1}{4}, 0, 0, 0, \frac{3}{4}]$, $\vec{r}_{ZD} = [0, 0, 0, 0, 1]$, metoda izračuna reprezentativne vrednosti vrne enak rezultat kot izračun povprečja rangov.

3.5 Zaključek

V tem poglavju smo predstavili množico kvalitativnih vrednosti in metodo, ki iz zbirke takih vrednosti izbere najbolj reprezentativno. Predlagana metoda je še posebej primerna za področje računalniško podprtega obvladovanja zaupanja. Tu običajno vsak uporabnik pričakuje, kakšna bi lahko bila porazdelitev prejetih ocen in mnenj agenta,

katerega stopnja zaupanja je denimo *zelo dobro* – reprezentativni porazdelitveni vektorji predstavljajo takšna pričakovanja. Predlagana metoda tako le primerja agente s *stereotipi*, torej primerja porazdelitvene vektorje agentov z reprezentativnimi porazdelitvenimi vektorji. In ker imajo lahko različni uporabniki različne stereotipe, pravimo, da je predlagana metoda prilagodljiva. V primerih, kjer slednje ni potrebno, pa uporabimo reprezentativne porazdelitvene vektorje iz Definicije 13. Slednje bomo – predvsem zavoljo enostavnejše razlage – uporabljali tudi v nadaljevanju.



Kvalitativni model zaupanja

4.1 Uvod

V tem poglavju predstavimo model zaupanja, ki uporablja v prejšnjem poglavju predstavljene kvalitativne vrednosti in metodo za izračun reprezentativne vrednosti. V modelu se kvalitativne vrednosti uporabijo za ocenjevanje preteklih interakcij, za izmenjevanje mnenj med agenti ter za podajanje izračunanih stopenj zaupanja. Model zato poimenujemo kvalitativni model zaupanja (*angl.* qualitative trust model, QTM).

4.2 Viri informacij

Model zaupanja (oz. agent, ki ga uporablja) računa zaupanje iz dveh virov informacij: iz izkušenj, ki jih agent pridobi v interakcijah z drugimi agenti, ter iz mnenj, ki jih agent prejme od drugih agentov. Model tako predpostavlja, da agent vodi evidenco lastnih izkušenj ter da lahko komunicira z drugimi agenti in od njih pridobiva mnenja.

Algoritem 1

Podatkovni strukturi, ki predstavljata izkušnje in mnenja.

```

1: Struct Experience {
2:   Agent agent
3:   Service service
4:   Time time
5:   Omega assessment
6: }
7:
8: Struct Opinion {
9:   Agent reporter
10:  Agent agent
11:  Service service
12:  Time time
13:  Omega assessment
14: }
```

Ker bo delovanje modela predstavljeno z različnimi algoritmi, bomo v tem razdelku opredelili ustrezne podatkovne strukture. Najprej si oglejmo tiste za predstavitev izkušenj in mnenj. Vsaka izkušnja vsebuje štiri podatke: identiteto *agenta*, s katerim je interakcija potekala, tip *storitve*, ki je bila v interakciji opravljena, *čas*, ob katerem se je interakcija zgodila, in *oceno*, s katero je bil ponudnik storitve ocenjen. Analogno

posamezno mnenje vsebuje pet podatkov: identiteto agenta, ki je mnenje podal in ga bomo imenovali *poročevalec*, identiteto *agenta*, katerega mnenje zadeva, tip *storitve*, na katero se mnenje nanaša, *čas*, ob katerem je bilo mnenje podano, ter *vrednost mnenja*, ki jo mnenje podaja. Pripadajoči podatkovni strukturi za izkušnjo ter mnenje sta podani v Algoritmu 1.

4.3 Izračun zaupanja

Na kratko lahko izračun stopnje zaupanja – do določenega agenta in za določen tip storitve – opišemo sledeče. Najprej iz preteklih izkušenj – s tem agentom in za podan tip storitve – izračunamo *porazdelitveni vektor ocen*. Podobno iz prejetih mnenj – o izbranem agentu in za podan tip storitve – izračunamo *porazdelitveni vektor vrednosti mnenj*. Nato porazdelitvena vektorja združimo v *porazdelitveni vektor zaupanja* in ga z metodo izračuna reprezentativne vrednosti pretvorimo v kvalitativno stopnjo zaupanja. Naslednji razdelki podrobneje opisujejo posamezne korake.

4.3.1 Vrednotenje izkušenj

Funkcija, ki izračuna porazdelitveni vektor iz izkušenj, je podana v Algoritmu 2. Na vходу sprejme dva argumenta, in sicer identiteto agenta (*agent*), do katerega želimo zaupanje izračunati, ter tip storitve (*service*), na katero se izračun nanaša. Funkcija vrne par (\vec{p}, c) , kjer \vec{p} predstavlja porazdelitveni vektor izkušenj, c pa mero prepričanja v izkušnje (*angl. confidence*). Slednja pove, koliko informacij izkušnje vsebujejo: če imamo malo izkušenj ali pa so te stare, je prepričanje nizko, v obratnem primeru – če jih je veliko ali so nedavne – pa visoko.

Na začetku funkcija inicializira prazen porazdelitveni vektor, nastavi skupno utež izkušenj (*total_weight*) na vrednost 0 in prebere sistemski čas (*time*). Nato v zanki obdelava vse relevantne izkušnje; klic funkcije `LOADEXPERIENCES(agent, service)` v vrstici 5 sproži poizvedbo v agentovi lokalni bazi in vrne izkušnje s podanim agentom za podan tip storitve. Vsaki izkušnji dodelimo utež, ki je odvisna od starosti izkušnje: v ta namen uporabimo funkcijo eksponentnega razpada, kjer s parametrom β določimo stopnjo pojemanja pomembnosti izkušnje v odvisnosti od časa. Izračunano utež nato prištejemo na ustrezno mesto v porazdelitvenemu vektorju (mesto določa rang ocene, vrstica 8) in h skupni uteži (vrstica 9). Zatem porazdelitveni vektor normiramo (vrstice 12-14) in na koncu z logističnim izrazom izračunamo parameter prepričanja (vrstica

16). Logistični izraz zagotavlja, da bo v primeru, ko ima agent veliko nedavnih izkušenj, spremenljivka *total_weight* blizu vrednosti 1. V obratnem primeru – ko imamo malo izkušenj ali pa so te stare – bo *total_weight* blizu vrednosti 0. Pri tem pomembno vlogo igra parameter κ , ki nadzira hitrost rasti. Pri evaluaciji modela v razdelku 6.3 bomo za β uporabili 0.1 za κ pa 1.

Algoritem 2

Funkcija COMPUTEEXPERIENCES(*agent*, *service*) izračuna porazdelitveni vektor izkušenj in mero prepričanja vanje.

Vhod: *agent* identiteta agenta, katerega izkušnje vrednotimo; *service* tip storitve, za katero vrednotimo izkušnje.

Izhod: *vector* porazdelitveni vektor izkušenj; *confidence* mera prepričanja v izkušnje

```

1: vector  $\leftarrow$  [0, 0, 0, 0, 0]
2: total_weight  $\leftarrow$  0
3: time  $\leftarrow$  CURRENTTIME()
4:
5: for all ex  $\in$  LOADEXPERIENCES(agent, service) do
6:   weight  $\leftarrow e^{-\beta(\text{time}-\text{ex.time})}$ 
7:   rank  $\leftarrow$  RANK(ex.assessment)
8:   vector[rank]  $\leftarrow$  vector[rank] + weight
9:   total_weight  $\leftarrow$  total_weight + weight
10: end for
11:
12: for i  $\leftarrow$  1 ... 5 do
13:   vector[i]  $\leftarrow$   $\frac{\text{vector}[i]}{\text{total\_weight}}$ 
14: end for
15:
16: confidence  $\leftarrow$   $\frac{\text{total\_weight}}{\kappa+\text{total\_weight}}$ 
17:
18: return (vector, confidence)

```

4.3.2 Vrednotenje mnenj

Lastne izkušnje so za izračun zaupanja običajno najzanesljivejši vir informacij, a so redke in posledično pogosto zastarele. Zato QTM pri izračunu zaupanja uporablja tudi mnenja, ki jih je sicer lažje pridobiti, a so v primerjavi z izkušnjami manj zanesljiva. Razlogi za to so različni, eden izmed najpogostejših pa je goljufivo obnašanje agentov, saj lahko agenti s ciljem lastne promocije lažno širijo slab glas o svojih konkurentih ali,

analogno, promovirajo svoje pajdaše. V tem razdelku predstavimo računski postopek, ki je na lažniva mnenja do neke mere odporen. Pri izračunu porazdelitvenega vektorja mnenj tako upoštevamo sledeče dejavnike:

- Nedavnost (*angl. recency*) mnenj. Kot pri izkušnjah nosijo nedavno podana mnenja večjo težo kot starejša.
- Zanesljivost poročevalca (*angl. reporter's trustworthiness*). Ta sestoji iz pretekle natančnosti (*angl. past accuracy*) in kredibilnosti (*angl. credibility*). Pretekla natančnost odraža resničnost mnenj, ki jih je nek poročevalec podal v preteklosti, kredibilnost pa meri vpliv, ki so ga njegova mnenja imela. QTM uporablja oba dejavnika, da lahko učinkovito zajame situacije, ko je resnična mnenja posebej težko dobiti.
- Družbena povezanost (*angl. social link*) med poročevalcem in agentom. QTM domneva, da imajo agenti, ki pripadajo istemu družbenemu krogu, drug o drugem bolj natančna mnenja kot agenti izven takega kroga.

V Algoritm 3 podamo postopek izračuna porazdelitvenega vektorja mnenj, ki temelji na zgornjih predpostavkah. Funkcija na vhodu sprejme dva argumenta: identiteto agenta (*agent*), o katerem mnenja vrednotimo, ter tip storitve (*service*), na katero se mnenja nanašajo. Najprej nastavimo začetne vrednosti spremenljivk: ustvarimo prazen porazdelitveni vektor (*vector*), nastavimo skupno utež (*total_weight*) na 0 ter ugotovimo sistemski čas (*time*). Nato v zanki obdelamo vsa mnenja, ki govorijo o izbranem agentu za podani tip storitve; klic `LOADOPINIONSABOUT(agent, service)` v vrstici 5 vrne množico takih mnenj iz agentove lokalne baze. Pri vsakem mnenju izračunamo njegovo nedavnost (*recency*, vrstica 6) zanesljivost poročevalca (*reporter_trustworthiness*, vrstice 7-17) ter stopnjo družbene povezanosti med obravnavanim agentom in poročevalcem (*social_link*, vrstice 18-20). Medtem ko je izračun nedavnosti povsem enak kot pri vrednotenju izkušenj, si izračuna zanesljivosti poročevalca ter stopnje družbene povezanosti med agentom in poročevalcem zaslužita dodatno razlago.

Zanesljivost poročevalca sestoji iz pretekle natančnosti (*past_accuracy*), ki opisuje, kako natančen je bil poročevalec pri podajanju mnenj v preteklosti, ter iz kredibilnosti (*credibility*), ki pove, koliko vpliva so njegova mnenja imela. Preteklo natančnost izračunamo tako, da iz lokalne baze naložimo seznam časovnih oznak (*angl. timestamps*), ob katerih je poročevalec podal pravilna oz. napačna mnenja (vrstici 8 in 12); kako

Algoritem 3

Funkcija $\text{COMPUTEOPINIONS}(\text{agent}, \text{service})$ izračuna porazdelitveni vektor mnenj.

Vhod: agent identiteta agenta, o katerem vrednotimo mnenja; service tip storitve, za katero vrednotimo mnenja

Izhod: *vector* porazdelitveni vektor mnenj

```

1:  $\text{vector} \leftarrow [0, 0, 0, 0, 0]$ 
2:  $\text{total\_weight} \leftarrow 0$ 
3:  $\text{time} \leftarrow \text{CURRENTTIME}()$ 
4:
5: for all  $op \in \text{LOADOPINIONSABOUT}(\text{agent}, \text{service})$  do
6:    $\text{recency} \leftarrow e^{-\lambda(\text{time}-op.\text{time})}$ 
7:    $\text{number\_correct} \leftarrow 0$ 
8:   for all  $t \in \text{LOADTIMESTEPSOFCORRECTOPINIONS}(op.\text{reporter}, \text{service})$  do
9:      $\text{number\_correct} \leftarrow \text{number\_correct} + e^{-\lambda(\text{time}-t)}$ 
10:  end for
11:   $\text{number\_incorrect} \leftarrow 0$ 
12:  for all  $t \in \text{LOADTIMESTEPSOFINCORRECTOPINIONS}(op.\text{reporter}, \text{service})$  do
13:     $\text{number\_incorrect} \leftarrow \text{number\_incorrect} + e^{-\lambda(\text{time}-t)}$ 
14:  end for
15:   $\text{past\_accuracy} \leftarrow (1 + e^{\text{number\_incorrect}-\text{number\_correct}})^{-1}$ 
16:   $\text{credibility} \leftarrow \text{LOADCREDIBILITY}(op.\text{reporter}, \text{service})$ 
17:   $\text{reporter\_trustworthiness} \leftarrow \sqrt{\text{past\_accuracy} \cdot \text{credibility}}$ 
18:   $\text{agent\_links} \leftarrow \text{EXTRACTAGENTS}(\text{LOADOPINIONSGIVENBY}(\text{agent}, \text{service}))$ 
19:   $\text{reporter\_links} \leftarrow \text{EXTRACTAGENTS}(\text{LOADOPINIONSGIVENBY}(op.\text{reporter}, \text{service}))$ 
20:   $\text{social\_link} \leftarrow \frac{|\text{agent\_acquaintances} \cap \text{reporter\_acquaintances}|}{|\text{agent\_acquaintances} \cup \text{reporter\_acquaintances}|}$ 
21:   $\text{weight} \leftarrow \text{MIN}(\text{recency}, \text{reporter\_trustworthiness}, \text{social\_link})$ 
22:   $\text{rank} \leftarrow \text{RANK}(op.\text{assessment})$ 
23:   $\text{vector}[\text{rank}] \leftarrow \text{vector}[\text{rank}] + \text{weight}$ 
24:   $\text{total\_weight} \leftarrow \text{total\_weight} + \text{weight}$ 
25: end for
26: for  $i \leftarrow 1 \dots 5$  do
27:    $\text{vector}[i] \leftarrow \frac{\text{vector}[i]}{\text{total\_weight}}$ 
28: end for
29:
30: return  $\text{vector}$ 

```

pridobiti take časovne oznake in kako ugotoviti, ali je mnenje pravilno ali napačno, je razloženo v razdelku 4.3.4. Časovne oznake se nato uporabijo za izračun časovne

diskontiranega števila pravih ($num_correct$) in napačnih ($num_incorrect$) preteklih mnenj. Ti številki nato združimo v preteklo natančnost z logističnim izrazom v vrstici 15. Slednji zagotavlja, da bo pretekla natančnost za večinsko pravilna mnenja blizu 1, za večinsko napačna blizu 0, v primerih enakega števila pravih in napačnih mnenj pa bo blizu $\frac{1}{2}$. V naslednjem koraku iz lokalne baze preberemo kredibilnost poročevalca (vrstica 16); izračun kredibilnosti je razložen v razdelku 4.3.4. Preteklo natančnost in kredibilnost nato združimo v zanesljivost poročevalca ($reporter_trustworthiness$) z geometrijsko sredino (vrstica 17).

Namen izračuna družbene povezanosti je ugotoviti, ali poročevalec in agent pripadata isti družbeni skupini. Nekateri modeli, denimo Regret [22], v ta namen uporabljajo dodatne vire informacij kot so družbena omrežja. Ker pa QTM ne predvideva, da leti obstajajo, se zatečemo enostavnejši rešitvi. Najprej zgradimo dve množici agentov: prvo, ki jo označimo z $agent_links$, vsebuje agente, o katerih je izbrani agent v preteklosti podal mnenja, in drugo, označena z $reporter_links$, ki vsebuje agente, o katerih je izbrani poročevalec v preteklosti podal mnenja (vrstici 18 in 19). Ti dve množici povesta, katere agente podani agent in poročevalec poznata. Če sta si množici zelo podobni, potem agent in poročevalec pripadata istemu družbenemu krogu, če ne, je njuna družbena povezanost nižja. Podobnost med množicama in posledično stopnjo družbene povezanosti ($social_link$) zato izračunamo z uporabo Jaccardovega koeficienta podobnosti (vrstica 20).

Za končno utež posameznega mnenja vzamemo najmanjšo izmed izračunanih mer (vrstica 21) ter na koncu zanke izračunano utež dodamo na pripadajoče mesto v porazdelitvenem vektorju in k skupni vsoti (vrstice 22-24). Porazdelitveni vektor na koncu še normaliziramo (vrstice 26-28).

4.3.3 Končni izračun zaupanja

Ko izračunamo porazdelitvena vektorja (prvega iz izkušenj in drugega iz mnenj), ju združimo v porazdelitveni vektor zaupanja. Slednji nam podaja celostno sliko agenta.

Definicija 15: Naj bosta par (\vec{e}, c) porazdelitveni vektor izkušenj in mera prepričanja kot ju vrne funkcija podana v Algoritmu 2, in naj bo \vec{d} porazdelitveni vektor mnenj kot ga podaja funkcija v Algoritmu 3. Porazdelitveni vektor zaupanja \vec{f} izračunamo

kot linearno kombinacijo porazdelitvenega vektorja izkušenj in mnenj.

$$\vec{t} = c \cdot \vec{e} + (1 - c) \cdot \vec{d}$$

V zadnjem koraku iz porazdelitvenega vektorja zaupanja \vec{t} izračunamo reprezentativno vrednost, ki predstavlja kvalitativno stopnjo zaupanja. Za to uporabimo metodo Reprezentativna : $\mathbb{R}^5 \mapsto \mathbb{V}$ (glej Definicijo 14).

$$v = \text{Reprezentativna}(\vec{t})$$

4.3.4 Izračun pretekle natančnosti ter kredibilnosti

Pri opisu postopka vrednotenja mnenj v razdelku 4.3.2 smo kot pomemben dejavnik izpostavili zanesljivost poročevalca, ki sestoji iz pretekle natančnosti in kredibilnosti. Prva se izračuna iz natančnosti mnenj, ki jih je poročevalec podal v preteklosti, medtem ko druga izhaja iz vpliva, ki so ga njegova mnenja imela. Ta razdelek opisuje izračun omenjenih mer.

QTM shranjuje podatke v lokalno podatkovno bazo: sem shranjuje izkušnje in mnenja. Pri nastanku nove izkušnje, se ta primerja z mnenji v bazi. Če sta izkušnja in izbrano mnenje *skladna*, slednjega označimo kot pravilnega (sicer pa kot napačnega). Pri tem za vsako tako ugotovitev v lokalno bazo zabeležimo čas, ob katerem je bilo mnenje podano. Iz teh podatkov izračunamo preteklo natančnost.

Pri ugotavljanju vpliva mnenj upoštevamo, kako težko je bilo pravilna mnenja pridobiti. Denimo, če je nek poročevalec podal pravilno mnenje takrat, ko so vsi ostali poročevalci prav tako podali pravilnega, je vpliv takšnega mnenja precej manjši, kot če se izkaže, da je izbrani poročevalec edini govoril resnico. (Analogno velja za podajanje napačnih mnenj.) Iz teh podatkov računamo kredibilnost. Konkretnije, QTM vsake-mu agentu (potencialnemu poročevalcu) pripiše začetno kredibilnost 1.0. Če se neko mnenje izkaže kot napačno, se kredibilnost poročevalcu, ki ga je podal, razpolovi in odstranjen del porazdeli med vse ostale poročevalce, ki so takrat podali mnenje – tudi poročevalcu, čigar mnenje je bilo napačno. Če se ta želi odkupiti in si ponovno zvišati kredibilnost, mora ne samo začeti podajati pravilna mnenja, ampak se morajo hkrati ostali poročevalci zmotiti. Slednje pomeni, da lahko agent kredibilnost izgubi hitro, dvigne pa precej počasneje. Omenjen postopek posnema delovanje algoritma utežene večine [33] in je podoben mehanizmu, ki je uporabljen v modelu Yu, Singh, Sycara

(YSS) [32]. Razlike med postopkoma podamo v nadaljevanju.

Algoritem 4

Procedura `VERIFYOPINIONS(experience)` preveri obstoječa mnenja, kadar agent pridobi novo izkušnjo. Procedura ne vrne rezultata, temveč kot stranski učinek (*angl. side effect*) posodobi kredibilnosti in seznam pravih in nepravilnih mnenj.

Vhod: `experience` novo nastala izkušnja

```

1: credibilities_old ← LOADCREDIBILITIES(experience.service)
2: credibilities_new ← COPY(credibilities_old)
3: opinions ← LOADOPINIONSABOUT(experience.agent, experience.service)
4:
5: for all op ∈ opinions do
6:   if  $|\text{RANG}(op.assessment) - \text{RANG}(experience.assessment)| > \delta$  then
7:     STOREINCORRECTPREDICTION(op.reporter, op.service, op.time)
8:     credibilities_new[op.reporter] ←  $\frac{1}{2} \cdot \text{credibilities\_new}[op.reporter]$ 
9:
10:    if credibilities_new[op.reporter] <  $\epsilon$  then
11:      credibilities_new[op.reporter] ←  $\epsilon$ 
12:    end if
13:  else
14:    STORECORRECTPREDICTION(op.reporter, op.service, op.time)
15:  end if
16: end for
17:
18: old_sum ← 0
19: new_sum ← 0
20:
21: for all reporter ∈ EXTRACTREPORTERS(opinions) do
22:   old_sum ← old_sum + credibilities_old[reporter]
23:   new_sum ← new_sum + credibilities_new[reporter]
24: end for
25:
26: for all reporter ∈ reporters do
27:   credibilities_new[reporter] ← credibilities_new[reporter] ·  $\frac{\text{old\_sum}}{\text{new\_sum}}$ 
28: end for
29:
30: STORECREDIBILITIES(credibilities_new, experience.service)

```

Procedura za preverjanje mnenj je podana v Algoritmu 4. Ker ji kot argument podamo novonastalo izkušnjo (`experience`), jo vedno izvedemo takoj po končani interakciji

– ko izkušnja nastane. Na začetku (vrstici 1-2) iz lokalne baze preberemo kredibilnosti poročevalcev za podajanje mnenj (o istem tipu storitve kot novonastala izkušnja) ter jih skupiramo. Nato v spremenljivko *opinions* naložimo mnenja, ki zadevajo istega agenta in tip storitve kot novonastala izkušnja (vrstica 3). Za vsako od teh v zanki preverimo pravilnost: mnenje je nepravilno, če se rang ocene v novonastali izkušnji ter rang vrednosti mnenja razlikujeta za več kot parameter δ (vrstica 6). V takem primeru (i) dodamo časovno oznako na seznam nepravilnih mnenj (vrstica 7) in (ii) poročevalcu razpolovimo kredibilnost. Ukazi na vrsticah 10-12 zgolj poskrbijo, da kredibilnost ostane nad mejo, ki jo določa parameter ϵ . Slednji zagotavlja, da se lahko poročevalci za podajanje napačnih mnenj odkupijo hitreje, in hkrati tudi preprečuje, da bi prišlo do zaokrožitvene napake, če bi nek poročevalec nenehno podajal lažna mnenja. Pri evaluaciji modela v razdelku 6.3 bomo za parameter δ uporabili vrednost 1 za ϵ pa 0.001. Če pa je mnenje pravilno, le shranimo časovno oznako (vrstica 14). V nadaljevanju izračunamo vsoti starih ter posodobljenih kredibilnosti (vrstice 18-24) in popravimo kredibilnosti za faktor, ki se izračuna kot razmerje med vsotama (vrstice 26-28). Ta popravek zagotavlja, da bo skupna vsota kredibilnosti ostala konstantna (vedno enaka številu poročevalcev) ter da bo nagrada (analogno kazni) za podajanje pravilnih (analogno napačnih) mnenj odvisna od števila poročevalcev, ki so jih podali. Na koncu posodobljene kredibilnosti zapišemo v lokalno podatkovno bazo (vrstica 30).

Za primer delovanja posodabljanja kredibilnosti vzemimo tri poročevalce a_1, a_2, a_3 , katerih kredibilnosti na začetku so zaporedoma 1.00, 1.00, 1.00. Potem ko poročevalci podajo svoja mnenja o nekem tretjem agentu, se izkaže, da je le agent a_1 govoril resnico, medtem ko sta se a_2 in a_3 zlagala. Ko izvedemo postopek posodobitve kredibilnosti iz Algoritma 4, se kredibilnosti spremenijo na vrednosti 1.50, 0.75, 0.75. V naslednjem koraku poročevalci znova podajo mnenja, a tokrat se izkaže da sta a_1 in a_2 govorila resnico, medtem ko se je a_3 zopet zlagal. Po ponovni posodobitvi kredibilnosti te zaporedoma znašajo 1.71, 0.86, 0.43. Vidimo lahko, da kljub temu da se je a_2 enkrat zlagal in enkrat povedal resnico, njegova končna kredibilnost ni enaka začetni: ko se je zlagal, je bila laž bolj škodljiva (dva agenta sta lagala, eden je govoril resnico), medtem ko je resnica bila manj informativna (dva agenta sta govorila resnico, eden pa lagal). Prav tako je v drugem primeru a_2 pridobil le del kredibilnosti, ki je bila odvzeta agentu a_3 , ta pa je bila nizka.

Model zaupanja avtorjev Yu, Singh in Sycara (YSS) [32] za izračun kredibilnosti prav tako uporablja algoritem utežene večine [33], vendar se v primerjavi s QTM razlikuje

v treh točkah. Prvič, model YSS zgolj znižuje kredibilnosti in jih nikoli ne poveča, zaradi česar je zelo občutljiv na poročevalce, ki se v sistemu pojavijo na novo, saj ti vedno dobijo največjo možno kredibilnost (1.0). Drugič, QTM za izračun zanesljivosti poročevalca uporablja še preteklo natančnost, medtem ko YSS uporablja le kredibilnost. In tretjič, YSS znižuje kredibilnost za faktor, ki se izračuna kot razlika med podanim mnenjem in lastno izkušnjo in leži na intervalu $[0.5, 1.0]$, medtem ko QTM razpolovi kredibilnost le, če je razlika med mnenjem in izkušnjo večja od vnaprej podanega praga. Ker YSS podaja ocene izkušenj in vrednosti mnenj kot realna števila, je popolno ujemanje med njima praktično nemogoče. Posledično je vsako mnenje vsaj malenkost napačno, kar pomeni, da se še pri tako natančnih mnenjih kredibilnost rahlo zniža. A ker se kredibilnost spreminja z množenjem, tudi majhne spremembe – ko jih večkrat ponovimo – hitro pripeljejo do večjih. Zaradi tega je model YSS občutljiv na šum, kar bo pokazano v nadaljevanju.

4.4 *Zaključek*

V tem poglavju smo predstavili model zaupanja QTM, ki uporablja kvalitativne podatke. Model izračuna zaupanje iz dveh virov informacij, in sicer iz izkušenj, kjer so kot kvalitativne predstavljene ocene interakcij, ter iz pridobljenih mnenj, kjer so kvalitativno podane vrednosti mnenj. Prav tako so kvalitativne stopnje izračunanega zaupanja. Logični naslednji korak pri predstavitvi modela je njegova evaluacija, tj. ugotoviti kako *dobro* deluje. Zato bomo v nadaljevanju obravnavali problem evaluacije modelov zaupanja.



Namensko testno okolje Alpha

5.1 Uvod

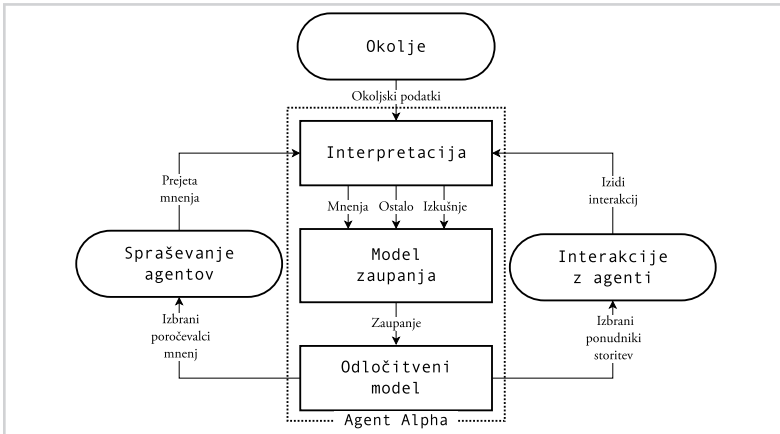
Že pri pregledu področja v razdelku 2.5 smo zapisali, da lahko merimo učinkovitost modelov zaupanja (jih evaluiramo, vrednotimo, testiramo), ter podali obstoječe načine merjenja, ki smo jih poimenovali modeli evaluacije. Omenjeni razdelek se zaključí s sklepom, da obstoječe rešitve bodisi (i) vrednotijo izračunano zaupanje neposredno, tako da od modelov zaupanja zahtevajo izračun stopenj zaupanja v vnaprej določenem formatu, bodisi (ii) vrednotijo izračunano zaupanje posredno, tako da vrednotijo iz izračunanega zaupanja izpeljane odločitve. Slabost prvega pristopa je v tem, da že vnaprej omejuje nabor modelov zaupanja, ki jih lahko vrednoti, slabost drugega pa, da od modelov zaupanja (oz. agentov, ki jih uporabljajo) zahteva sprejemanje odločitev. A ker so modeli zaupanja pogosto podani kot črne škatle – imajo natančno definirane vhode in izhode, a ne podajo navodil, kako izračune uporabiti – morajo takšni modeli evaluacije modelom zaupanja dodati odločitvene postopke, ki iz izračunanega zaupanja privedejo do odločitev. Takim postopkom pravimo odločitveni modeli. Ker rezultati odločitev vodijo do nastanka podatkov, ki bodo v prihodnosti postali vhod v model zaupanja (izbor partnerja za interakcijo vodi do nastanka izkušnje, izbor poročevalca pa do nastanka mnenja), lahko takšen način evaluacije privede do dvomljivih rezultatov, saj se model zaupanja in odločitveni model vmešavata v nastajanje vhodnih podatkov. Poleg tega odločitveni model določa razmerje med raziskovanjem in izrabljanjem (*angl.* exploration vs. exploitation ratio), zato ni gotovo, da bosta poljuben model zaupanja in poljuben odločitveni model združljiva.

V tem poglavju predstavimo namensko testno okolje Alpha (*angl.* Alpha Testbed, ATB), ki zgoraj opisanih omejitev nima, saj lahko testira modele zaupanja (i) z ali brez odločitvenih modelov in hkrati (ii) od njih ne zahteva uporabe vnaprej določenega formata zaupanja. Na začetku predstavimo osnovno idejo modela evaluacije, nato pa opišemo testno okolje z implementacijskega vidika: podamo arhitekturni vidik in razložimo delovanje vsake od sodelujočih komponent. Poglavje zaključimo s primerom.

5.2 Alpha in ostali agenti

Testno okolje je osnovano na posebnem agentu z imenom Alpha, ki med evaluacijo edini uporablja model zaupanja, ostali agenti pa so le simulirani. Slika 5.1 prikazuje njegovo zgradbo ter vire informacij za izračun zaupanja.

Alpha pridobiva informacije iz izkušenj iz *interakcij*, iz *mnenj*, ki jih agenti podajo



Slika 5.1

Agent Alpha, njegovi sestavni deli in viri informacij za izračun zaupanja. Agent sestoji iz komponent interpretacija, model zaupanja in odločitveni model. Prva pretvarja zbrane podatke in jih posreduje testiranemu modelu, ki iz njih izračuna zaupanje. To se posreduje odločitvenemu modelu, ki na podlagi izračunanega zaupanja sprejema odločitve, kot so denimo s kom iti v interakcijo ali koga in o kom povprašati za mnenje. V okolju ATB pri testiranju uporabljamo zgolj mnenja in izkušnje, okoljske podatke pa izpuščamo.

o drug drugem, ter iz *okoljskih* podatkov. Med slednje spadajo npr. družbena omrežja (*angl.* social networks) ali strukture virtualnih organizacij (*angl.* virtual organization). Ker je uporaba izkušenj in mnenj najpogostejša, se bomo omejili le nanje in pustili obravnavo okoljskih podatkov za kdaj drugič. Agent Alpha sestoji iz treh komponent, in sicer iz *interpretacije*, *modela zaupanja* in *odločitvenega modela*. Pogosto se bomo namesto na posamezno komponento sklicevali kar na Alfo, vendar bo iz konteksta razvidno, o kateri komponenti govorimo.

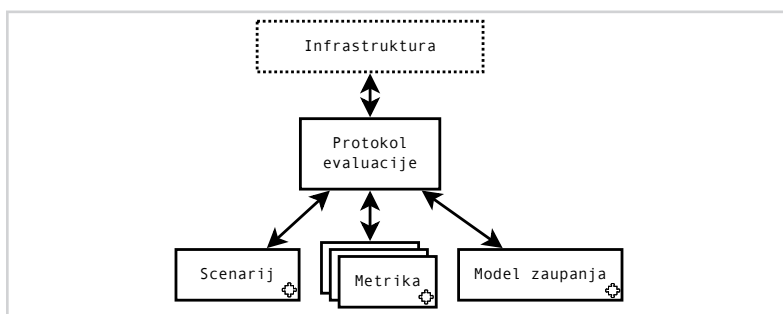
Vsaka komponenta je zadolžena za opravljanje določene naloge. Tako je naloga interpretacije, da pretvarja prejete podatke v obliko, ki je modelu zaupanja razumljiva, naloga modela pa da izračunava zaupanje. Odločitveni model je zadolžen za sprejemanje različnih odločitev, kot je npr. izbiranje ponudnikov storitev ali izbiranje poročevalcev mnenj. Ta komponenta je v praksi lahko zelo kompleksna in pri sprejemanju odločitev upošteva poleg zaupanja tudi druge dejavnike. A ker so ti domensko specifični, jih avtorji modelov zaupanja običajno ne podajo [14]. Odločitvene modele v ATB uporabimo le izjemoma in še takrat zgolj za izbiranje partnerjev za interakcije ali za izbiranje poročevalcev mnenj.

5.3 Visokonivojska arhitektura

Testno okolje ATB sestavlja pet osnovnih komponent: scenarij, model zaupanja, metrike, protokol evaluacije ter podporna infrastruktura. Scenarij je zadolžen za generacijo podatkov, iz katerih testirani model zaupanja izračunava zaupanje, nato pa njegovo natančnost ocenijo različne metrike. Komponente povezuje protokol evaluacije, ki skrbi za komunikacijo in izmenjavanje sporočil ter tako posledično določa veljavne načine kombiniranja scenarijev, modelov zaupanja in metrik. Protokol se izvaja na izbrani infrastrukturi; v ta namen je uporabljeno simulacijsko orodje Repast Symphony [67, 68]. Visokonivojska arhitektura je podana na Sliki 5.2.

Slika 5.2

Komponente testnega okolja ATB: protokol evaluacije, scenarij, model zaupanja in nabor metrik. Med testiranjem scenarij generira podatke, iz katerih model zaupanja izračunava zaupanje, zatem njegovo natančnost ocenijo različne metrike. Protokol evaluacije nadzoruje ostale komponente in se izvaja na izbrani infrastrukturi.



Scenarij je zadolžen za generiranje podatkov in predstavlja celotno okolico agenta Alpha. Tako nadzoruje obnašanje vseh ostalih agentov v simulaciji: nadzira njihovo populacijo, sposobnost izvajanja storitev, poštenost in voljnost za podajanje mnenj ter določa s kom bo Alpha imel interakcije (generacija izkušenj) ter od koga in o kom bo prejel mnenja (generacija mnenj). Skratka, scenarij določa vse parametre, ki so pomembni za generiranje vhodnih podatkov za model zaupanja. Generirani podatki se nato posredujejo agentu Alpha oz. modelu zaupanja, ki iz njih izračuna zaupanje. Slednje kot tudi ostala dejanja modela zaupanja ovrednotijo izbrane metrike. Delovanje posameznih komponent je podrobno opisano v sledečih razdelkih.

5.4 Protokol evaluacije

Protokol evaluacije predstavlja osrednji del testnega okolja. Zadolžen je za potek evaluacije, saj skrbi za izmenjavo informacij med komponentami. Od protokola evaluacije je

tako odvisno, katere podatke mora scenarij generirati, katere podatke mora model zaupanja izračunati in katere podatke morajo metrike ovrednotiti. Posledično protokol evaluacije določa, katere modele zaupanja, scenarije in metrike smemo kombinirati. Trenutno ATB podpira tri protokole, in sicer evaluacijo *Brez odločitev*, evaluacijo *Izbiranje ponudnikov storitev* ter evaluacijo *Izbiranje ponudnikov storitev in izbiranje poročevalcev*. Vsak protokol je namenjen evaluaciji posebnega tipa modela zaupanja: prvi testira modele zaupanja, ki zgolj izračunavajo zaupanje, drugi testira modele, ki poleg izračunavanja zaupanja še izbirajo partnerje za interakcije, medtem ko tretji testira modele zaupanja, ki poleg izračunavanja zaupanja in izbiranja partnerjev za interakcije, izbirajo še poročevalce mnenj.

5.4.1 *Brez odločitev*

Protokol *Brez odločitev* je namenjen testiranju modelov zaupanja, katerih avtorji zgolj podajo pravila, formule ali algoritme, ki opisujejo, kako zaupanje izračunati, ne podajo pa navodil, kako izračunano zaupanje v neki situaciji uporabiti. Takšni modeli so v literaturi najpogostejši. Težava pri testiranju teh modelov je v tem, da jim moramo tekom testiranja podati izkušnje, pri ustvarjanju le-teh pa mora *nekdo* izbrati partnerje za interakcije. Zato obstoječa testna okolja takim modelom zaupanja dodajo *ad-hoc* odločitvene modele, ki izbirajo partnerje; običajno agenta, ki ima najvišjo stopnjo zaupanja. Kot smo že omenili, se želimo takšnemu početju izogniti, saj ni gotovo, da je poljuben model zaupanja kompatibilen s poljubnim odločitvenim modelom.

Protokol *Brez odločitev* postopa drugače. V njem partnerjev za interakcije ter poročevalcev za podajanje mnenj ne izbira agent Alpha (oz. testirani model zaupanja), temveč jih določa scenarij. Tako generacija izkušenj in mnenj ni več odvisna od modela zaupanja, temveč le od parametrov evaluacije, kar prinese dve ključni prednosti: prvič, modeli zaupanja so testirani neodvisno in brez odločitvenih modelov, in drugič, vsak model zaupanja je testiran z enakimi vhodnimi podatki. Slednje je nujno, če želimo, da bo evaluacija nepristranska. Če bi sklopili model zaupanja s poljubnim odločitvenim modelom, bi omenjeno načelo kršili. Denimo, da imamo dva modela zaupanja, ki iz danih vhodnih podatkov izračunata različno zaupanje. Nato vsak od njiju na podlagi izračunanega zaupanja izbere partnerja za interakcijo, tako da uporabi vnaprej podan odločitveni model; denimo, da vsak izbere agenta, za katerega misli, da je najbolj zaupanja vreden. A ker sta modela izračunala različno zaupanje, bo izbira partnerja za interakcijo – kljub temu, da sta uporabila enak odločitveni model – različna. To pa

pomeni, da bosta modela zaupanja v prihodnosti, ko bosta na vходу sprejela izkušnjo s trenutno izbranim agentom, uporabljala različne vhodne podatke. Posledično bodo rezultati takšne evaluacije dvomljivi. A če so partnerji za interakcije podani neodvisno od modela zaupanja, takšnih težav ne bomo imeli, ker bo vsak model zaupanja dobil enake vhodne podatke.

V tem protokolu ATB testira modele tako, da vrednoti natančnost izračunanega zaupanja. Podrobno delovanje metrike natančnosti razložimo v razdelku 5.6.1. Konkretni koraki omenjenega protokola so naslednji.

1. Protokol sporoči vsem komponentam, da je nastopila nova časovna enota.
2. Protokol naroči scenariju naj generira mnenja in jih izroči modelu zaupanja.
3. Protokol naroči scenariju naj izbere partnerja, simulira interakcijo med njim in agentom Alpha ter generirano izkušnjo izroči modelu zaupanja.
4. Protokol naroči modelu zaupanja, naj izračuna zaupanje.
5. Protokol preda zaupanje metriki natančnosti, ki ga ovrednoti.

Omenjeno zaporedje korakov se ponovi v vsaki časovni enoti. Ta protokol zagotavlja, da vsak model zaupanja prejme enake vhodne podatke.

5.4.2 Izbiranje ponudnikov storitev

V nasprotju s protokolom *Brez odločitev* pa protokol *Izbiranje ponudnikov storitev* od modela zaupanja oz. agenta, ki ga uporablja, zahteva, da ta izbira partnerje za interakcije. Posledično ta protokol uporablja dodatni korak, v katerem model zaupanja izbere partnerja, ter dodatno metriko, ki oceni kakovost izbire; metrika koristnosti je podrobneje razložena v razdelku 5.6.2.

1. Protokol sporoči vsem komponentam, da je nastopila nova časovna enota.
2. Protokol naroči scenariju naj generira mnenja in jih izroči modelu zaupanja.
3. Protokol naroči modelu zaupanja, naj izbere partnerja za interakcijo in ga naznani scenariju.
4. Protokol naroči scenariju naj simulira interakcijo med izbranim partnerjem in agentom Alpha ter generirano izkušnjo izroči modelu zaupanja.

5. Protokol naroči modelu zaupanja, naj izračuna zaupanje.
6. Protokol z metriko natančnosti ovrednoti izračunano zaupanje.
7. Protokol z metriko koristnosti ovrednoti izbiro partnerja.

Opisano zaporedje se ponovi v vsaki časovni enoti.

5.4.3 Izbiranje ponudnikov storitev in izbiranje poročevalcev

Ko testiramo model zaupanja, ki poleg postopka za izračun zaupanja in izbor partnerja podaja še postopek za izbor poročevalcev mnenj, uporabimo protokol evaluacije *Izbiranje ponudnikov storitev in izbiranje poročevalcev*. Protokol je zelo podoben zgornjemu, a vključuje dva dodatna koraka.

1. Protokol sporoči vsem komponentam, da je nastopila nova časovna enota.
2. Protokol naroči modelu zaupanja, naj izbere poročevalce mnenj in izbor preda scenariju.
3. Protokol naroči scenariju naj za izbrane poročevalce generira mnenja in jih izroči modelu zaupanja.
4. Protokol naroči modelu zaupanja, naj izbere partnerja za interakcijo in ga naznani scenariju.
5. Protokol naroči scenariju naj simulira interakcijo med izbranim partnerjem in agentom Alpha ter generirano izkušnjo izroči modelu zaupanja.
6. Protokol naroči modelu zaupanja, naj izračuna zaupanje.
7. Protokol z metriko natančnosti ovrednoti izračunano zaupanje.
8. Protokol z metriko koristnosti ovrednoti izbiro partnerja.
9. Protokol z metriko stroškov mnenj ovrednoti stroške zahtevanih mnenj.

Opisano zaporedje se ponovi v vsaki časovni enoti. Ker tukaj model zaupanja (natančnejše agent Alpha in njegov odločitveni model) izbira poročevalce, protokol dodatno oceni kakovost teh izbir. Pri tem se uporabi metrika stroškov mnenj, ki je podrobneje opisana v razdelku 5.6.3.

Glavna motivacija za različne protokole evaluacije izhaja iz dejstva, da raziskovalci pri predstavitvi modelov zaupanja le-te podajo na različne načine: medtem ko nekateri zgolj podajo navodila, kako zaupanje izračunati, drugi dodatno podajo postopke, ki povedo, kako na podlagi izračunanega zaupanja sprejeti odločitve; bodisi le o tem, kako izbrati partnerja za interakcijo, bodisi tudi o tem, koga in o kom povprašati za mnenje. Z uporabo različnih protokolov lahko primerjamo modele, ne da bi jim morali dodati elemente, ki bi lahko vplivali na rezultate evaluacije.

5.5 Scenarij

Scenarij predstavlja okolico agenta Alpha, saj nadzoruje obnašanje ostalih agentov, ki sodelujejo pri evaluaciji. V scenariju določimo naslednje vidike testiranja.

Populacija agentov. Scenarij določa množico agentov, ki so prisotni v sistemu, ter njeno spreminjanje tekom evaluacije; množica agentov se namreč lahko spremeni, saj lahko v sistem pridejo novi agenti, obstoječi pa odidejo.

Generiranje izkušenj. Scenarij določa postopke za generiranje izkušenj. Tako scenarij določa zmožnosti agentov za opravljanje različnih storitev ter njihove spremembe tekom evaluacije, kot tudi pravila, kako se te zmožnosti izrazijo v interakcijah. V protokolih, kjer Alpha ne izbira partnerjev, mora te izbrati scenarij.

Generiranje mnenj. Scenarij določa postopke za generiranje mnenj. Določiti mora tendence in zmožnosti agentov za podajanje mnenj: definirati, kateri agenti se poznajo in kakšna mnenja podajajo drug o drugem. V protokolih, kjer Alpha ne izbira poročevalcev mnenj, mora te izbrati scenarij.

Scenariji ponujajo prilagodljiv način za ustvarjanje testnih primerov, v katerih lahko simuliramo različne napade na modele zaupanja. Nekatere od njih bomo opisali v nadaljevanju, ko bo govora o evaluaciji (razdelka 6.2.4 in 6.3.1). V naslednjih podrazdelkih podrobneje opišemo generiranje izkušenj ter mnenj.

5.5.1 Generiranje izkušenj

Agent Alpha dobi izkušnje v interakcijah z drugimi agenti, ko zanj opravijo neko storitev (glej Definicijo 7). S kakovostjo izvedbe slednjih se izražajo zmožnosti (*angl.* capability) agentov. Zato moramo pred opisom postopka za generiranje izkušenj še opredeliti koncept zmožnosti agentov.

Zmožnosti agentov za opravljanje storitev

Zmožnosti agentov za opravljanje storitev modeliramo s konceptom zmožnosti (*angl.* capability).

Definicija 16 (Zmožnosti): Zmožnost agenta $a \in A$ za opravljanje storitve tipa $s \in S$ v času $t \in T$ predstavlja njegovo sposobnost in voljnost, da opravi kakovostno storitev za agenta Alpha. Zmožnost predstavimo z realnim številom z intervala $[0, 1]$, kjer 0 in 1 zaporedoma predstavljata najnižjo in najvišjo stopnjo zmožnosti. Matematično jo predstavimo kot preslikavo $Zmoznost : A \times S \times T \mapsto [0, 1]$.

Zmožnosti predstavljajo dejanske zanesljivosti agentov za opravljanje storitev v interakcijah; večja kot je zmožnost agenta, več koristi agent Alpha pridobi v interakciji z njim. Zaradi tega jih uporabimo kot osnovo za (i) modeliranje interakcij (generiranje izkušenj) ter (ii) modeliranje mnenj (generiranje mnenj). Na tem mestu še zapišimo, da zmožnosti modelu zaupanja (oz. agentu Alpha) niso nikoli razkrite in da mora slednji o njih sklepati le na podlagi pridobljenih izkušenj in mnenj. Zaradi tega jih bomo uporabili kot osnovo za vrednotenje izračunanega zaupanja.

Modeliranje interakcij

Interakcije modeliramo tako, da generiramo ustrezne pare, ki ponazarjajo izkušnje. Kot smo podali v Definiciji 7, je posamezna izkušnja urejena četverka $\langle a, s, t, \lambda \rangle$, kjer $a \in A$ predstavlja agenta, s katerim je Alpha imel interakcijo, $s \in S$ tip opravljene storitve, $t \in T$ čas, ob katerem se je interakcija zgodila, ter $\lambda \in \Lambda$ oceno, s katero je Alpha ocenil ponudnika storitve. V pregledu področja smo izpostavili, da različni modeli zaupanja različno predstavijo ocene iz interakcij: poznamo vse od binarnih do zveznih lestvic. V želji, da bi bili čim bolj splošni, v ATB predstavimo ocene interakcij kot realna števila z $[0, 1]$, kjer 0 in 1 zaporedoma predstavljata najslabšo in najboljšo oceno. Takšnim (notranjim) ocenam v ATB pravimo izidi interakcij. Če testirani model uporablja drugačno predstavitev ocene, mora izide interakcij pretvoriti v svojo domeno [69].

Ko generiramo posamezno izkušnjo, moramo zgolj izračunati (simulirati) izid interakcije. Pri tem izhajamo iz zmožnosti agenta, ki opravlja storitev. A da bo simulirani izid bolj realističen, moramo upoštevati, da je lahko izid posamezne interakcije drugačen od same zmožnosti agenta, ki opravlja storitev. Vendar razlika ne sme biti znatna

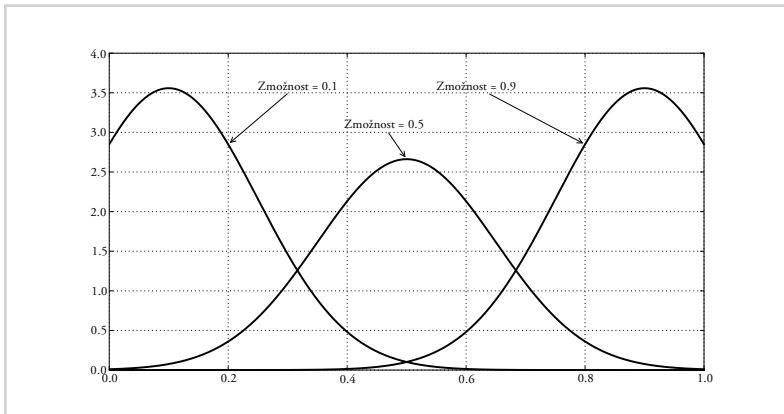
in mora – tekom več interakcij – sovpadati z dejansko zmožnostjo agenta; razlika mora biti zgolj posledica šuma. Zato ATB generira posamezne izide interakcij z uporabo psevdonaključnega generatorja. Generator uporabi funkcijo gostote verjetnosti (*angl.* probability density function, PDF), ki jo uglasimo z dvema parametra: (i) z zmožnostjo agenta, ki opravlja storitev, in (ii) s količino šuma. V ta namen uporabimo prisekano normalno porazdelitev (*angl.* truncated normal distribution) na intervalu $[0, 1]$.

$$p(x; \mu, \sigma) = \begin{cases} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\int_0^1 e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt} & 0 \leq x \leq 1 \\ 0 & \text{sicer} \end{cases} \quad (5.1)$$

Oblika funkcije gostote verjetnosti je odvisna od povprečne vrednosti μ in standardnega odklona σ . Ko simuliramo izid interakcije med Alpo in agentom a ob času t za storitev s , le pokličemo psevdonaključni generator in nastavimo μ na zmožnost agenta, ki opravlja storitev (tj. $\mu = \text{Zmožnost}(a, s, t)$), medtem ko lahko za σ podamo poljubno vrednost, ki določa količino šuma. Na primer, Slika 5.5.1 prikazuje funkcije gostote verjetnosti za tri agente, katerih zmožnosti so zaporedoma 0.1, 0.5 in 0.9. Parameter σ je v vseh treh primerih nastavljen na 0.15.

Slika 5.3

Testno okolje generira izkušnje z uporabo psevdonaključnega generatorja, ki kot funkcijo gostote verjetnosti uporablja prisekano normalno porazdelitev. Pri simuliranju izidov interakcij se za vrednost parametra μ uporabi zmožnost agenta, ki ponuja storitev, medtem ko s parametrom σ arbitrarno določamo stopnjo šuma. Slika prikazuje funkcije gostote verjetnosti za agente, katerih zmožnosti so zaporedoma 0.1, 0.5 in 0.9, medtem ko za σ v vseh treh primerih uporabimo 0.15.



5.5.2 Generiranje mnenj

Ostali agenti lahko agentu Alpha podajo mnenja o drug drugem. V dejanskih sistemih bi bila takšna mnenja odvisna od dveh dejavnikov: od modelov zaupanja, ki jih ostali agenti uporabljajo, in od njihove poštenosti do agenta Alpha. Ker nas prvenstveno zanima le delovanje modela zaupanja, ki ga uporablja Alpha, in ker želimo pri tem zmanjšati vpliv ostalih dejavnikov, bomo modele zaupanja, ki jih uporabljajo ostali agenti, le simulirali, medtem ko bomo različne stopnje poštenosti modelirali z modeli prevar (*angl.* deception models).

Modeliranje zaupanja med ostalimi agenti

Z mnenji modeliramo zaupanje med ostalimi agenti v sistemu. Kot smo podali v Definiciji 8, posamezno mnenje predstavlja urejena peterka $\langle a_p, a_c, s, t, \theta \rangle$, kjer a_p predstavlja poročevalca oz. agenta, ki je mnenje podal, a_c agenta, o komur je govora, s tip storitve, ki jo mnenje zadeva, t čas, ob katerem je bilo mnenje podano, ter θ vrednost mnenja. Podobno kot izkušnje lahko različni modeli zaupanja različno predstavijo tudi mnenja. Zavoljo večje splošnosti bomo v testnem okolju ATB predstavili množico vrednosti mnenj kot množico parov realnih števil $\Theta \subseteq [0, 1] \times [0, 1]$. V vsakem paru $\langle w, u \rangle \in \Theta$ prva komponenta $w \in [0, 1]$ označuje vrednost mnenja, s katero poročevalec izrazi svoje mnenje o ciljnem agentu, medtem ko druga komponenta $u \in [0, 1]$ sporoča stopnjo negotovosti (*angl.* uncertainty), ki jo ima poročevalec o podanem mnenju; števili 0 in 1 predstavljata najnižji in najvišji vrednosti. Tako denimo vrednost $\langle 0.95, 0.00 \rangle$ pomeni zelo dobro mnenje ter da je poročevalec o povedanem povsem gotov. Če model uporablja drugačno predstavitev vrednosti mnenj, mora podane podatke ustrezno preoblikovati.

Okolje ATB ustvari posamezno mnenje s psevdonaključnim generatorjem. Podobno kot pri izkušnjah uporabimo psevdonaključni generator in funkcijo gostote verjetnosti priskekane normalne porazdelitve, ki je odvisna od dveh parametrov: z zmožnostjo agenta, ki ga mnenje zadeva, ter s parametrom šuma. Tako pri ustvarjanju mnenja agenta a_p o agentu a_c za storitev tipa s in ob času t pokličemo naključni generator in nastavimo $\mu = \text{Zmoznost}(a_c, s, t)$, medtem ko parameter σ nastavimo na vrednost negotovosti mnenja, ki tako ponazarja stopnjo poznanstva med poročevalcem in agentom; če se agenta dobro poznata, je negotovost nizka, sicer visoka.

Predstavljen način generiranja mnenj nam omogoča troje. Prvič, z njim lahko si-

muliramo različne stopnje poznanstev med agenti – nekateri agenti se poznajo bolje kot drugi in tako generirana mnenja bodo to dejstvo upoštevala. Drugič, ker naključni generator vsebuje šum, se bodo tudi mnenja poročevalcev, ki izbranega agenta poznajo enako dobro, vsaj malo razlikovala. In tretjič, model zaupanja, ki ga testiramo, v ničemer ne vpliva na postopek generiranja mnenj; mnenja so namreč generirana povsem neodvisno od testiranega modela, kar pomeni, da bo v izbrani konfiguraciji testiranja vsak model zaupanja prejel enaka mnenja.

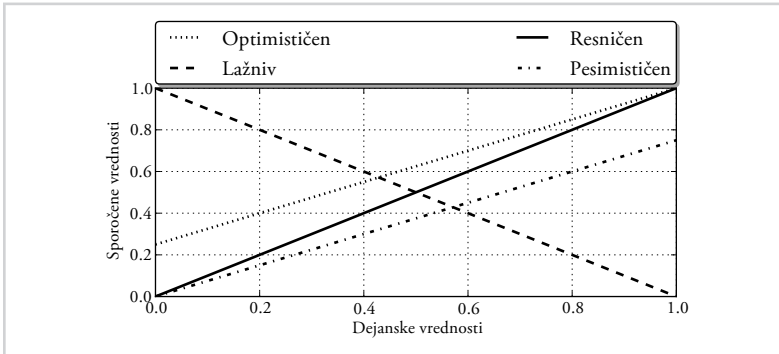
Modeliranje prevar

Nerealno je pričakovati, da bodo vsa sporočena mnenja vedno poštena, še posebej če med agenti obstajajo tekmovalna razmerja in govorimo o sistemih, kjer lahko agenti prosto prihajajo in odhajajo. Za modeliranje takšnih situacij v ATB uporabimo modele prevar (*angl.* deception models) [70], s katerimi določamo, kakšna mnenja bodo sporočena agentu Alpha.

V splošnem je model prevare funkcija $d : [0, 1] \mapsto [0, 1]$, ki spreminja komponente vrednosti mnenj. V prvotnem predlogu sta Yu in Sing [70] predstavila naslednje štiri modele prevare: *resnični* (*angl.* truthful), *lažnivi* (*angl.* complementary), *optimistični* in *pesimistični* (*angl.* exaggerated positive, negative). Resnični model pravzaprav ni pravi model prevare, saj vrednosti mnenj ne spreminja; namenjen je agentom, ki sporočajo resnična mnenja. V lažnivem modelu poročevalci podajajo mnenja, ki so čisto nasprotje dejanskih vrednosti mnenj, medtem ko optimisti in pesimisti zgolj olepšajo oz. očrtnijo dejanske vrednosti mnenj. Grafični prikaz delovanja modelov prevare je podan na Sliki 5.5.2.

K omenjeni množici modelov prevare dodajmo še *naključni* model, v katerem agenti podajajo naključne vrednosti mnenj; te izračunamo s psevdonaključnim generatorjem, ki uporablja enakomerno porazdelitev na intervalu $[0, 1]$. Množica modelov prevare tako vsebuje naslednje elemente $D = \{d_R, d_L, d_O, d_P, d_N\}$, kjer posamezne modele definiramo sledeče (parameter $0 \leq \kappa \leq 1$ označuje koeficient pretiravanja; na Sliki 5.5.2 je uporabljena vrednost $\kappa = 0.25$):

- Resničen model $d_R(x) = x$;
- Lažniv model $d_L(x) = 1 - x$;
- Optimističen model $d_O(x) = x \cdot (1 - \kappa) + \kappa$;



Slika 5.4

Simuliranje podajanja mnenj z uporabo modelov prevar. Če je dejanska vrednost mnenja 0,20, bi poročevalci z resničnim, lažnivim, optimističnim in pesimističnim modelom prevare zaporedoma sporočili vrednosti 0,20, 0,80, 0,40 in 0,15.

- Pesimističen model $d_P(x) = x \cdot (1 - \kappa)$;
- Naključen model $d_N = U(0, 1)$.

Modeli prevar v množici D predstavljajo najpogostejše načine podajanja mnenj v odprtih večagentnih sistemih: nekateri govorijo resnico, drugi lažejo [70] in tretji poročajo pristransko – bodisi v pozitivno bodisi v negativno smer [71]. Naš dodatek, naključni model, predstavlja agente, katerih mnenja so najmanj zanesljiva, medtem ko se situacije, ko se agenti med seboj bodisi ne poznajo bodisi ne želijo podati mnenj, modelirajo z odsotnostjo modela prevare.

Dodeljevanje modelov prevare je naloga scenarija. Ta mora modele prevare dodeliti v odvisnosti od poročevalca, ciljnega agenta, tipa storitve in časa. Zato dodeljevanje modelov prevare opišemo kot funkcijo Prevvara : $A \times A \times S \times T \mapsto D$. Ta funkcija ni nujno totalna: obstajajo vhodne kombinacije za katere je nedefinirana ($D(a_p, a_c, s, t) = \perp$). Slednje zgolj pomeni, da v času t poročevalec a_p za agenta a_c o storitvi tipa s ne bo podal mnenja.

Z modeli prevare lahko tudi simuliramo diskriminatorno obnašanje agentov pri ponujanju storitev. Če se nek agent kot ponudnik storitve do ostalih agentov obnaša drugače kot do agenta Alpha – denimo, da storitve za Alpha vedno opravi dobro, za preostale agente pa slabo – potem se Alphi sicer resnična mnenja, ki jih ostali podajo o omenjenem agentu, zdijo lažna, saj se razlikujejo od tistega, kar Alpha dejansko izkusi v interakciji. Pri modeliranju takih situacij preprosto uporabimo ustrezen model prevare.

5.6 Metrike

Testno okolje uporablja več metrik, vendar je njihova uporaba odvisna od modela zaupanja, ki ga testiramo. Pričujoči razdelek obravnava metrike in njihovo uporabo.

5.6.1 Natančnost

Pri najosnovnejšem protokolu evaluacije, *Brez odločitev*, testno okolje oceni zgolj pravilnost izračunanega zaupanja. Pri tem izhajamo iz zmožnosti agentov, saj te določajo njihovo kakovost za opravljanje storitev. V takih primerih alternativni pristopi preprosto izračunajo podobnost med izračunanim zaupanjem ter samimi zmožnostmi agentov. Omenjen postopek je problematičen, saj zahteva, da je izračunano zaupanje podano z enakimi enotami kot zmožnosti agentov; v našem primeru bi to pomenilo, da morajo vsi modeli zaupanja uporabljati stopnje zaupanja z intervala $[0, 1]$. A kot smo že večkrat omenili, temu ni tako. V skladu z Definicijo 5 lahko pričakujemo, da bodo stopnje zaupanja predstavljene z mero, ki bo imela definirano relacijo linearne urejenosti. Slednje pomeni, da bi naj vsak model znal razvrstiti agente glede na izračunane stopnje zaupanja. Zato bomo izračunano zaupanje vrednotili po podobnosti med razvrstitvama agentov po zaupanju ter po zmožnostih. Kot mero podobnosti razvrstitve lahko uporabimo katerokoli metriko, ki izračuna razdaljo med razvrstitvami.

Definicija 17 (Natančnost): Naj $\tau(a, s) \in \Omega$ označuje stopnjo zaupanja, ki jo ima agent Alpha do agenta $a \in A$ za ponujanje storitve tipa $s \in S$ ob času $t \in T$. Definirajmo $\#\text{Skladnih} : S \times T \mapsto \mathbb{N}_0$ kot funkcijo, ki vrne število parnih primerjav med agenti po izračunanih stopnjah zaupanja, ki so skladne s primerjavami po njihovih zmožnostih.

$$\#\text{Skladnih}(s, t) = \sum_{\substack{a_i, a_j \in A(t) \\ \text{Zmoznost}(a_i, s, t) > \text{Zmoznost}(a_j, s, t)}} [\tau(a_i, s) > \tau(a_j, s)]$$

Analogno definirajmo funkcijo $\#\text{Neskladnih} : S \times T \mapsto \mathbb{N}_0$, ki vrne vrne število parnih primerjav med agenti po izračunanih stopnjah zaupanja, ki so neskladne s primerjavami po njihovih zmožnostih.

$$\#Neskladnih(s, t) = \sum_{\substack{a_i, a_j \in A(t) \\ Zmoznost(a_i, s, t) > Zmoznost(a_j, s, t)}} [\tau(a_i, s) < \tau(a_j, s)]$$

Nato definirajmo metriko *Natančnost* : $S \times T \mapsto [0, 1]$, ki vrne stopnjo podobnosti med izračunanim zaupanjem in zmožnostmi agentov. Natančnost ovrednotimo v vsaki časovni enoti $t \in T$ in za vsak tip storitve $s \in S$ posebej.

$$\text{Natančnost}(s, t) = \frac{1}{2} + \frac{\#Skladnih(s, t) - \#Neskladnih(s, t)}{|A(t)| \cdot (|A(t)| - 1)}$$

Zgornja definicija uporablja metriko Kendall's Tau-A, pri čemer smo zgolj premaknili njeno domeno iz $[-1, 1]$ na $[0, 1]$. Tako rezultat 1.0 pomeni, da se izračunano zaupanje popolnoma sklada z zmožnostmi agentov, medtem ko rezultat 0.0 pomeni popolno neskladje. V slednjem primeru je model zaupanja dejansko škodljiv, saj uporabniku svetuje, da so agenti s slabšimi zmožnostmi bolj zaupanja vredni. Če je rezultat 0.5 pa izračunano zaupanje ne prinaša nobene informacije. Takšen rezultat dobimo, če npr. vsakemu agentu priredimo enako stopnjo zaupanja ali če stopnje zaupanja prirejamo naključno.

Primer Imejmo množico agentov $A = \{a_1, a_2, a_3\}$, ki imajo pripadajoče zmožnosti: $Zmoznost(a_1) = 0.1$, $Zmoznost(a_2) = 0.5$ in $Zmoznost(a_3) = 0.9$. (Da bo primer enostaven, predpostavimo konstantne zmožnosti in le en tip storitve, zato bomo oznaki za čas in tip storitve iz notacije izpustili.) Testiramo model zaupanja, ki uporablja kvalitativne stopnje zaupanja $\Omega = \{\text{slabo} < \text{srednje} < \text{dobro}\}$. Model izračuna naslednje: $\tau(a_1) = \text{srednje}$, $\tau(a_2) = \text{slabo}$ in $\tau(a_3) = \text{dobro}$. Izračunano zaupanje tako podaja drugačno urejenost agentov kot jo podajajo zmožnosti:

- Urejenost po zmožnostih: $a_1 < a_2 < a_3$.
- Urejenost po zaupanju: $a_2 < a_1 < a_3$.

Natančnost zaupanja izračunamo v treh korakih. Najprej izračunamo število skladnih primerjav.

$$\#Skladnih = 1_{[\tau(a_1) < \tau(a_3)]} + 1_{[\tau(a_2) < \tau(a_3)]} = 2$$

Nato izračunamo število neskladnih primerjav.

$$\#\text{Neskladnih} = 1_{[\tau(a_1) > \tau(a_2)]} = 1$$

Na koncu izračunamo natančnost kot jo podaja Definicija 17.

$$\text{Natančnost} = \frac{1}{2} + \frac{2-1}{3 \cdot (3-1)} = \frac{2}{3}$$

Metrika Kendall's Tau-A iz Definicije 17 predstavlja le eno izmed možnih mer natančnosti. Lahko bi denimo uporabili metriko Spearman's footrule ali katerokoli ustrezno zamenjavo [72]. Testno okolje dopušča uporabo različnih implementacij in dodajanje novih v obliki vtičnikov (*angl.* plug-in). A v tej disertaciji bomo uporabljali zgolj Kendall's Tau-A.

5.6.2 Koristnost

Kadar testiramo model zaupanja, ki poleg postopka za izračun zaupanja poda tudi postopek izbire partnerjev za interakcije, lahko testno okolje oceni kakovost takšnih izbir. Tipično se v ta namen meri koristnost, ki jo agent Alpha pridobi v interakciji. Ker je koristnost neposredno odvisna od zmožnosti ponudnika storitve, jo bomo merili kar z zmožnostmi agentov, ki jih Alpha izbere za partnerje.

Definicija 18 (Koristnost): Naj funkcija Izbira : $S \times T \mapsto A$ predstavlja izbiro agenta, s katerim bo agent Alpha ob podanem času in za izbran tip storitve imel interakcijo. Potem lahko definiramo metriko Koristnost : $S \times T \mapsto [0, 1]$, ki za podan tip storitve $s \in S$ in do vključno časa $t \in T$ vrne uspešnost agenta Alpha pri izbiranju partnerjev za interakcije.

$$\text{Koristnost}(s, t) = \frac{\sum_{t_i \leq t} \text{Zmožnost}(\text{Izbira}(s, t_i), s, t_i)}{\sum_{t_i \leq t} \max_{a \in A(t_i)} \text{Zmožnost}(a, s, t_i)}$$

Koristnost tako meri relativno uspešnost izbiranja partnerjev za interakcije, kjer vrednosti blizu 0.0 predstavljajo slabe izbire, vrednosti blizu 1.0 pa dobre. V števcu seštejemo zmožnosti agentov, s katerimi je Alpha imel interakcije od začetka evaluacije do vključno trenutka izračuna, v imenovalcu pa zmožnosti tistih agentov, ki so v posameznih trenutkih bili najzmogljivejši. Omenjena metrika bo tako vrnila rezultat

1.0 le, če bo Alpha vsakič izbral najzmogljivejšega agenta. Na tem mestu omenimo, da ima metrika koristnosti "spomin", saj novejše vrednosti vedno vključujejo tudi vse starejše.

Primer Uporabimo množico agentov iz prejšnjega primera. (Kot prej imamo konstantne zmožnosti in le en tip storitve, zato bomo iz notacije $Zmoznost(a, s, t)$ izpustili oznaki za čas in tip storitve, iz notacije $Izbira(s, t)$ pa oznako za tip storitve.) Če Alpha izbira agente po naslednjem zaporedju: $Izbira(1) = a_1$, $Izbira(2) = a_2$ in $Izbira(3) = a_3$, lahko izračunamo koristnost v času 3.

$$\begin{aligned} \text{Koristnost}(3) &= \frac{Zmoznost(a_1) + Zmoznost(a_2) + Zmoznost(a_3)}{Zmoznost(a_3) + Zmoznost(a_3) + Zmoznost(a_3)} \\ &= \frac{0.10 + 0.50 + 0.90}{0.90 + 0.90 + 0.90} = 0.55 \end{aligned}$$

Podobno kot natančnost lahko koristnost izračunamo na več načinov; testno okolje dopušča različne implementacije preko vtičnikov. A v tej disertaciji bomo za merjenje koristnosti vedno uporabili metriko iz Definicije 18.

5.6.3 Stroški mnenj

Kadar testiramo model zaupanja, ki poleg postopkov za izračun zaupanja in izbire partnerjev za interakcije poda tudi postopke, ki povedo, koga in o kom povprašati za mnenje, lahko testno okolje ovrednoti tudi kakovost le-teh.

Definicija 19 (Stroški mnenj): Z množico $ZM \subseteq A \times A \times S \times T$ označimo zahteve mnenj, ki jih podaja agent Alpha. Vsak zahtevek predstavimo s četverko $\langle a_p, a_c, t, s \rangle$, kjer $a_p \in A$ predstavlja agenta, ki ga Alpha sprašuje (poročevalca), $a_c \in A$ agenta in $s \in S$ tip storitve, ki ju mnenje zadeva, ter $t \in T$ čas, v katerem se zahtevek podaja. Z oznako $ZM(s, t)$ označimo podmnožico zahtevkov, ki so podani za podan tip storitve $s \in S$ v času $t \in T$. Sedaj lahko definiramo metriko $\text{StroskiMnenj} : S \times T \mapsto [0, 1]$, ki za podane zahteve mnenj vrne relativne stroške njihove pridobitve.

$$\text{StroskiMnenj}(s, t) = \frac{|ZM(s, t)|}{|A(t)| \cdot (|A(t)| - 1)}$$

V Definiciji 19 podana metrika izračuna razmerje med podanim številom zahtevkov ter številom zahtevkov, ki jih je sploh možno podati. Če agent Alpha zahteva

mnenje od vsakega o vsakemu agentu, bodo stroški mnenj znašali 1.0. Za razliko od natančnosti in koristnosti tukaj nižja vrednost pomeni boljši rezultat. Pomen metrike stroškov mnenj je domensko pogojen; z njo lahko denimo pokažemo stopnjo zgoštitve prometa (*angl.* congestion), ki jo model zaupanja s podajanjem zahtevkov povzroči v komunikacijskem kanalu.

5.6.4 Pomen različnih metrik

Predstavljene metrike nam omogočajo, da vrednotimo posamezne vidike modelov zaupanja posebej. Tako natančnost meri kakovost izračunanega zaupanja, koristnost kakovost izbire partnerjev za interakcije, metrika stroškov mnenj pa porabo virov pri pridobivanju mnenj. Tako ni gotovo, da bodo rezultati modelov zaupanja po posameznih metrikah vedno usklajeni. Denimo, če želi agent Alpha izboljšati natančnost izračunanega zaupanja, bo moral kdaj tudi vstopiti v interakcije z neznanimi in potencialno manj zmožnimi agenti, saj se bo le tako lahko prepričal, da so slednji res manj zmožni. Vendar se bo zaradi tega pridobljena koristnost zmanjšala. Alternativno bi lahko za mnenja povprašal več agentov, ampak bi s tem "zamašil" komunikacijski kanal ter hkrati tvegala, da bi prejel napačne informacije. Konkretnije, natančnost in koristnost kažeta tendence Alphe glede raziskovanja (*angl.* exploration) in izrabljanja (*angl.* exploitation). Za doseganje natančnejših izračunov mora Alpha kdaj tudi vstopiti v interakcije z neznanimi agenti, torej mora raziskati prostor potencialnih partnerjev za interakcije. Po drugi strani pa mora za doseganje večjih koristnosti izrabljati obstoječe znanje (tj. izračunano zaupanje) in za partnerje izbirati agente, katerih zmožnosti so visoke. Dober odločitveni model bi moral uskladiti tendence med raziskovanjem in izrabljanjem.

5.7 Primer poteka testiranja

V tem razdelku pokažemo, kako testiranje dejansko poteka. Imejmo množico agentov $A = \{a_1, a_2, a_3, a_4, a_5\}$ z zmožnostmi: $Zmoznost(a_1) = 0.10$, $Zmoznost(a_2) = 0.25$, $Zmoznost(a_3) = 0.50$, $Zmoznost(a_4) = 0.75$ in $Zmoznost(a_5) = 1.00$. Ker so zmožnosti konstantne in ker obstaja le en tip storitve, bomo iz notacije *Izbira*(s, t) izpustili oznako za tip storitve iz notacije $Zmoznost(a, s, t)$ pa še oznako za čas. Ker testiramo model zaupanja, ki poleg izračuna zaupanja podaja tudi postopek izbire partnerjev za interakcije, bomo uporabili protokol evaluacije *Izbiranje ponudnikov storitev* (glej razdelek 5.4.2); vrednotili bomo torej natančnost in koristnost. Testirani

model zaupanja uporablja kvantitativne stopnje zaupanja z intervala $[0, 1]$, pri izbiri partnerjev interakcij pa poda naslednje zaporedje: Izbira(1) = a_1 , Izbira(2) = a_2 , Izbira(3) = a_3 , Izbira(4) = a_4 , Izbira(5) = a_5 , Izbira(6) = a_5 , Izbira(7) = a_5 in Izbira(8) = a_5 . Tabela 5.1 prikazuje dinamiko izračunov zaupanja, ter pripadajoče vrednosti natančnosti in koristnosti.

Tabela 5.1

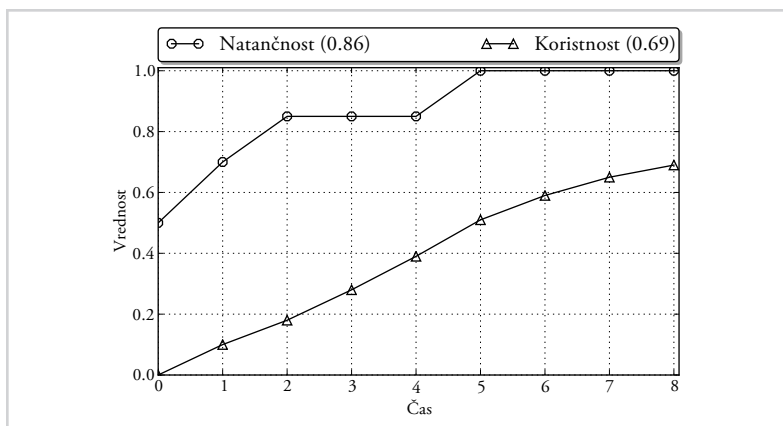
Dinamika izračuna stopenj zaupanja ter pripadajoče spremembe v natančnosti in koristnosti. Z osenčeno barvo so označeni agenti (ter pripadajoče stopnje zaupanja), s katerimi je v podani časovni enoti agent Alpha imel interakcijo.

t	$\tau(a_1, t)$	$\tau(a_2, t)$	$\tau(a_3, t)$	$\tau(a_4, t)$	$\tau(a_5, t)$	Nat(t)	Kor(t)
0	0.50	0.50	0.50	0.50	0.50	0.50	0.00
1	0.37	0.50	0.50	0.50	0.50	0.70	0.10
2	0.37	0.42	0.50	0.50	0.50	0.85	0.18
3	0.37	0.42	0.50	0.50	0.50	0.85	0.28
4	0.37	0.42	0.50	0.58	0.50	0.85	0.39
5	0.37	0.42	0.50	0.58	0.66	1.00	0.51
6	0.37	0.42	0.50	0.58	0.75	1.00	0.59
7	0.37	0.42	0.50	0.58	0.80	1.00	0.65
8	0.37	0.42	0.50	0.58	0.83	1.00	0.69

Na začetku priredi model vsem agentom privzeto stopnjo zaupanja (0.50), kar pomeni, da je natančnost 0.50, koristnost pa 0.00, saj Alpha še ni imel interakcij. Ob času $t = 1$ Alpha stopi v interakcijo z a_1 in ker je zmožnost slednjega nizka (0.10), mu model zmanjša stopnjo zaupanja iz 0.50 na 0.37. (Tukaj omenimo, da nas trenutno ne zanima, kako testirani model zaupanja dejansko vrši izračune, ampak le da se zaupanje *nekako* spremeni – spremembe v izračunih so v Tabeli 5.1 osenčene.) Omenjena sprememba poveča natančnost iz 0.50 na 0.70 ter koristnost iz 0.00 na 0.10. Analogno ima Alpha ob času $t = 2$ interakcijo z a_2 in zopet popravi izračunano stopnjo zaupanja iz 0.50 na 0.42. Zaradi tega se natančnost dvigne na 0.85, koristnost pa na 0.18. Podobno se zgodi v vsaki časovni enoti.

Iz opisanega lahko povzamemo dvoje. Prvič, natančnost zaupanja je odvisna od relativne razvrstitve agentov glede na stopnjo zaupanja in ne od njihovih absolutnih vrednosti. Slednje je še posebej očitno od $t = 5$ dalje, kjer se stopnja zaupanja do agenta a_5 čedalje bolj približuje njegovi dejanski zmožnosti, natančnost zaupanja pa ostaja

enaka. In drugič, takoj ko začne agent Alpha za partnerja izbirati agenta z največjo zmogljnostjo, bo začela koristnost konvergirati proti 1.00.



Slika 5.5

Natančnost in koristnost modela zaupanja v obravnavanem primeru. Vrednosti v legendi prikazujeta povprečno vrednost natančnosti ter najnovjšo vrednost koristnosti.

Ko bomo v naslednjem poglavju prikazovali rezultate, bomo vrednosti metrik prikazali na grafu; Slika 5.7 prikazuje gibanje vrednosti natančnosti in koristnosti in odvisnosti od časa. Za lažjo primerjavo med različnimi modeli bomo izračunali reprezentativno število za vsako metriko. Pri natančnosti bomo uporabili povprečno vrednost v celotnem časovnem intervalu – na testnem primeru znaša 0.86 – medtem ko bomo pri koristnosti zgolj prebrali njeno zadnjo vrednost – na testnem primeru ob času $t = 8$ znaša 0.69. To moramo storiti zaradi tega, ker metrika koristnosti, kot smo jo definirali v Definiciji 18, podaja tako oceno trenutne izbire kot tudi vseh prejšnjih, medtem ko natančnost podaja kakovost izračunanega zaupanja v posamezni časovni enoti. Reprezentativne vrednosti metrik so prikazane v legendi.

5.8 Implementacija

Namensko testno okolje Alpha smo implementirali v programskem jeziku Java za osnovo pa smo vzeli simulacijsko platformo Repast [67, 68]. Implementacija dosledno sledi arhitekturi iz razdelka 5.3 in ima modularno zasnovo, kar pomeni, da lahko večino elementov dodamo v obliki vtičnikov (*angl.* plug-in). Celotna implementacija kot tudi navodila za namestitev in uporabo so na voljo na sledeči spletni strani.

<http://atb.fri.uni-lj.si>

Trenutna implementacija vsebuje vse protokole evaluacije, scenarije, metrike, modele prevar ter modele zaupanja, ki jih omenjamo v poglavjih 4-6, medtem ko lahko nove elemente dodamo preko vtičnikov. Pri tem moramo zgolj implementirati želeni programski vmesnik, implementacijski razred zapakirati v datoteko JAR [73] ter slednjo podati javanskemu nalagalniku razredov (*angl.* class loader). Implementacija je ponujena pod odprtokodno licenco GNU GPL različica 3 [74].

5.9 Zaključek

V tem poglavju smo predstavili namensko testno okolje Alpha (ATB), s katerim lahko vrednotimo delovanje modelov zaupanja. V testnem okolju imamo enega *pravega* agenta, medtem ko ostale agente simuliramo. Naloga pravega agenta – agenta Alpha – je, da (i) izračunava zaupanje do drugih (simuliranih) agentov ter da (ii) sprejema za evaluacijo pomembne odločitve, denimo s katerim od simuliranih agentov iti v interakcijo ali koga in o kom povprašati za mnenje. V primerjavi z obstoječimi modeli evaluacije prinaša ATB dve bistveni prednosti. Prvič, od testiranega modela zaupanja ne zahteva, da ta uporablja vnaprej dogovorjeno množico stopenj zaupanja, temveč zgolj da so stopnje zaupanja medsebojno primerljive z relacijami večji, manjši ali enak. Zaradi tega je ATB v primerjavi z obstoječimi modeli evaluacije, ki vrednotijo zaupanje neposredno, bolj splošen. In drugič, od testiranega modela zaupanja (oz. agenta, ki ga uporablja) zahteva sprejemanje odločitev le v posebnih, prav za to namenjenih, primerih – če model zaupanja odločitvenih modelov ne podaja, mu testno okolje ne bo dodelilo *ad-hoc* odločitvenega modela, temveč ga bo ovrednotilo brez njega. Slednje nam omogoča, da testiramo modele zaupanja z enakimi vhodnimi podatki, kar pomeni, da bodo rezultati takšnih evaluacij objektivnejši.



Evaluacija

6.1 Uvod

V tem poglavju bomo koncepte iz prejšnjih dveh (kvalitativni model zaupanja iz poglavja 4 ter namensko testno okolje iz poglavja 5) eksperimentalno ovrednotili. Začeli bomo s preverjanjem hipoteze o vplivu odločitvenega modela na delovanje modela zaupanja ter v nadaljevanju ovrednotili delovanje kvalitativnega modela zaupanja.

6.2 Vpliv odločitvenega modela na model zaupanja

V poglavju 5 smo večkrat zapisali, da je tvegano sklopiti poljuben model zaupanja z poljubnim odločitvenim modelom, saj ni gotovo, da bo vsaka kombinacija delovala enako dobro. Razlog za naš dvom izhaja iz dejstva, da odločitveni model vpliva na razmerje med raziskovanjem in izrabljanjem, ter slutnje, da različni modeli zaupanja potrebujejo različno razmerje med omenjenima konceptoma. To je bila tudi ena izmed osrednjih kritik obstoječih modelov evaluacije, saj ti pri testiranju modele zaupanja navadno sklopijo z *ad-hoc* odločitvenimi modeli, ker predpostavljajo, da slednji na testiranje ne vplivajo.

6.2.1 Hipoteza in postopek preverjanja

Da bi lahko omenjeno hipotezo preverili, smo zasnovali dva eksperimenta. V prvem smo izbrane modele zaupanja sklopili z enim odločitvenim modelom, v drugem pa z drugim. Pričakovali smo, da v kolikor naša hipoteza – *izbira odločitvenega modela vpliva na delovanje modela zaupanja* – ne drži, bomo v obeh eksperimentih dobili enake rezultate. Če pa se zgodi, da bodo rezultati različni, potem mora naša hipoteza veljati. Pri tem dodajmo, da pri primerjanju rezultatov govorimo zgolj o razvrstitvi modelov zaupanja glede na rezultate in ne na absolutne vrednosti le-teh.

V eksperimentu smo odločitveni model uporabili zgolj za izbiranje partnerjev za interakcije ne pa tudi za izbiranje poročevalcev mnenj. Zaradi tega smo uporabili protokol evaluacije *Izbiranje ponudnikov storitev* (glej razdelek 5.4.2), kar je pomenilo, da smo uporabili metriki natančnosti in koristnosti. Pri preverjanju hipoteze smo torej primerjali modele glede na natančnost in koristnost v odvisnosti od odločitvenega modela.

6.2.2 Uporabljeni modeli zaupanja

Pri testiranju smo uporabili naslednje modele zaupanja: Beta Reputation System (BRS) [27], Travos (TRA) [31], EigenTrust (ET) [28], model zaupanja Abdul-Rahman in Hailes (ARH) [26] ter Yu, Singh in Sycara (YSS) [32]. Omenjen izbor vsebuje v literaturi najodmevnejše modele, hkrati pa zaradi velike raznolikosti modelov – nekateri so centralizirani, drugi porazdeljeni, nekateri kvalitativni, drugi kvantitativni – služi kot dokaz splošnosti namenskega testnega okolja. V sledečih podrazdelkih na kratko opišemo, kako smo omenjene modele realizirali v testnem okolju, tj. podamo postopke, ki generirane podatke pretvorijo v format, ki ga modeli zaupanja pričakujejo. Z drugimi besedami, opišemo, kako smo realizirali komponento interpretacija s Slike 5.1. Dodajmo še, da smo pri testiranju uporabili priporočene vrednosti parametrov, če pa slednje niso bile podane, smo skušali smiselne poiskati.

Abdul-Rahman in Hailes (ARH)

Model zaupanja avtorjev Abdul-Rahman in Hailes (ARH) [26], predstavljen v razdelku 2.4.1, uporablja kvalitativne vrednosti za predstavitev ocen interakcij in vrednosti mnenj. Pri generiranju izkušenj in mnenj smo zato morali izide interakcij in vrednosti mnenj preslikati iz numeričnih vrednosti v kvalitativne. Uporabili smo tri različne preslikave, ker nas je zanimalo, kako vplivajo na delovanje modela. Preslikave so podane v Tabeli 6.1.

Tabela 6.1

Tabela preslikav numeričnih vrednosti z intervala $[0, 1]$ v kvalitativne vrednosti modela ARH. Če npr. uporabimo preslikave ARH_S bomo število 0.50 pretvorili v vrednost 'g'. Črke N, S, V tako označujejo "višino" preslikave: nizka, srednja in visoka.

	'vb'	'b'	'g'	'vg'
ARH_N	[0.00, 0.10)	[0.10, 0.25)	[0.25, 0.50)	[0.50, 1.00]
ARH_S	[0.00, 0.25)	[0.25, 0.50)	[0.50, 0.75)	[0.75, 1.00]
ARH_V	[0.00, 0.50)	[0.50, 0.75)	[0.75, 0.90)	[0.90, 1.00]

Beta Reputation System (BRS)

Model BRS [27] smo podrobneje opisali v razdelku 2.4.2. Omenili smo, da so ocene izkušenj ter vrednosti mnenj predstavljene kot dvojice $\langle r, s \rangle$, kjer $r \in [0, 1]$ in $s \in [0, 1]$ predstavljata količino pozitivne int negativne ocene oz. vrednosti mnenja, pri čemer

za vsako dvojico velja $r + s = 1$. V ATB smo omenjene dvojice izračunali iz generiranih izkušenj (mnenj), tako da smo vrednosti r priredili izid interakcije (vrednosti mnenja), vrednosti s pa njen komplement tj. $s = 1 - r$. Tako je denimo generirana izkušnja $\langle a, s, t, \langle 0.8, 0.05 \rangle \rangle$ postala $\langle a, s, t, \langle 0.8, 0.2 \rangle \rangle$; ker BRS pri podajanju mnenj komponente negotovosti (0.05 v podanem primeru) ne uporablja, je nismo pretvorili.

EigenTrust (ET)

EigenTrust (ET) [28] smo obravnavali v razdelku 2.4.3, kjer smo zapisali, da model predstavlja ocene interakcij kot binarne dvojice. Tako dvojica $\langle 1, 0 \rangle$ predstavlja pozitivno, dvojica $\langle 0, 1 \rangle$ pa negativno oceno. Posamezno dvojico izračunamo tako, da izid interakcije primerjamo z vnaprej podanim pragom in če ga izid dosega, je ocena pozitivna sicer pa negativna. Kot pri modelu ARH, smo tudi tu testirali tri različne pragove, in sicer: ET_N s pragom 0.25, ET_S s pragom 0.50, in ET_V s pragom 0.75.

Ker EigenTrust predpostavlja, da bodo vsi agenti prav tako uporabljali enak model zaupanja, pričakuje, da bodo vrednosti mnenj podane kot zbirka ocen iz preteklih interakcij v obliki dvojic $\langle m, n \rangle$, kjer m in n predstavljata število pozitivnih in negativnih ocen. Ker je v okolju ATB edini agent, ki uporablja model zaupanja, agent Alpha in ker so ostali agenti simulirani, smo morali generirane vrednosti mnenj pretvoriti v omenjene dvojice. Pri tem smo uporabili naslednji postopek. Ko model prejme mnenje, denimo $\langle a_p, a_c, s, t, \langle 0.60, 0.05 \rangle \rangle$, simulira določeno število interakcij¹ med agentoma a_p in a_c , tako da uporabi naključni generator in prisekano normalno porazdelitev. Za μ vzame vrednost, ki je bila podana v izvornem mnenju (v zgornjem primeru 0.60), za σ pa vrednost, ki se modelu poda kot parameter in mora biti enaka parametru šuma, s katerim testno okolje generira izkušnje. Naključni generator nato poženemo tolikokrat, kolikor interakcij želimo simulirati, in vsako dobljeno vrednost primerjamo s pragom. Če ga generirana vrednost dosega, je simulirana interakcija pozitivna sicer pa negativna. Opisan postopek zagotavlja, da bo Alpha prejel povsem enake vrednosti mnenj, kot bi jih, če bi agent a_p imel z a_c deset interakcij in bi rezultate slednjih podal v mnenju. Denimo, če uporabimo prag 0.50, bi zgornje mnenje spremenili v $\langle a_p, a_c, s, t, \langle 8, 2 \rangle \rangle$.

¹V naših eksperimentih smo simulirali 10 interakcij. To število bi lahko izpeljali iz stopnje negotovosti izvirnega mnenja (0.05 v zgornjem primeru), kjer bi nizke vrednosti pomenile večje število interakcij, visoke negotovosti pa manjše. Ampak ker bomo v eksperimentih generirali vsa mnenja z enako stopnjo negotovosti, bi s tem postopek pretvarjanja le nepotrebno zakomplicirali.

Travos (TRA)

Model Travos (TRA) [31] smo opisali v razdelku 2.4.4. Pokazali smo, da predstavlja ocene interakcij ter vrednosti mnenj na enak način kot model EigenTrust: ocene so predstavljene kot binarne dvojice, vrednosti mnenj pa kot dvojice, ki vsebujejo število uspešnih in neuspešnih preteklih interakcij. Zaradi tega smo lahko za pretvorbo izkušenj in mnenj uporabili isti postopek, kot smo ga opisali v prejšnjem podrazdelku. Prav tako smo enako kot pri modelu EigenTrust tudi tu uporabili tri različne pragove: TRA_N s pragom 0.25, TRA_S s pragom 0.50, in TRA_V s pragom 0.75.

Yu, Singh in Sycara (YSS)

Model avtorjev Yu, Singh in Sycara (YSS) [32] smo predstavili v razdelku 2.4.5. Ker model predstavlja ocene interakcij in vrednosti mnenj na enak način kot jih generira ATB, nam podatkov ni bilo treba pretvoriti; $\Lambda = \Theta \subseteq [0, 1]$.

6.2.3 Odločitvena modela

Le pri enem od izbranih modelov zaupanja so avtorji v izvirnem predlogu podali odločitveni model, in sicer pri modelu EigenTrust [28]. Ta je vseboval kar dva odločitvena modela: *maksimalno* ter *verjetnostno izbiro*. V odločitvenem modelu maksimalna izbira Alpha za interakcijo izbere agenta, ki ima najvišjo stopnjo zaupanja, medtem ko pri verjetnostni izbiri Alpha izbira naključno, vendar je verjetnost, da bo nek agent izbran, sorazmerna stopnji zaupanja, ki ga agent uživa. Verjetnostna izbira lahko tako izbere kateregakoli agenta, a hkrati zagotavlja, da bodo zaupanja vredni agenti izbrani pogosteje. Pri maksimalni izbiri bo agent Alpha posledično zbral več koristnosti kot pri verjetnostni, vendar bo pri slednji več raziskoval.

Omenjeni izbiri smo uporabili pri preverjanju hipoteze tako, da smo v prvem eksperimentu modele kombinirali z maksimalno, v drugem pa z verjetnostno izbiro. Kombiniranje je bilo preprosto, izjemo smo morali narediti le pri verjetnostni izbiri in modelu ARH. Ker so v tem modelu stopnje zaupanja kvalitativne, verjetnostna izbira pa zahteva kvantitativne – da lahko vsakemu agentu pripiše verjetnost, da bo izbran – smo morali kvalitativnim stopnjam prirediti številke. V ta namen smo uporabili preslikave iz Tabele 6.1, pri čemer se posamezne kvalitativne vrednosti preslikajo v števila, ki predstavljajo spodnje meje posameznih intervalov. Tako se npr. vrednost 'vg' v modelu ARH_N preslika v število 0.50, v modelu ARH_S v 0.75, v modelu ARH_V pa v 0.90.

6.2.4 Scenarij

Pri testiranju smo uporabili le en scenarij, in sicer *Tranzitivni scenarij*.

- Scenarij je vseboval 100 agentov in en tip storitve ter je agentom dodeljeval zmožnosti naključno z uporabo enakomerne porazdelitve na intervalu $[0, 1]$. Nabor agentov kot tudi njihove zmožnosti so bile določene na začetku testiranja in se med potekom niso spreminjale.
- Pri dodeljevanju modelov prevare je scenarij sledil predpostavki *tranzitivnosti*. Tako je dodeljeval *resnične* modele agentom, katerih zmožnosti so bile visoke, ter *lažnive* tistim, katerih zmožnosti so bile nizke. Natančneje, verjetnost, da je nek poročevalec podal resnično mnenje, je bila linearno odvisna od njegove zmožnosti za opravljanje storitev. Enako kot zmožnosti so se modeli prevare dodelili na začetku in se tekom testiranja niso spreminjali.
- V scenariju je vsak agent poznal vsakega in o vsakem tudi podal mnenje.
- Posamezno testiranje je trajalo 500 časovnih enot. To je bilo dovolj, da so se izračuni ustalili. Dodatne simulacije kažejo, da podaljševanje izvajanja evaluacij ne vpliva na končne rezultate.
- Za generiranje izkušenj smo uporabili standardni odklon $\sigma = 0.10$ za generiranje mnenj pa $\sigma = 0.05$. Nižji šum za generiranje mnenj utemeljujemo s tem, da posamezno mnenje, v kolikor je resnično, nosi več informacije kot posamezna izkušnja. Ker smo želeli simulirati, da se vsi agenti poznajo enako dobro, in ker smo želeli imeti eksperiment čim bolj obvladljiv, smo vrednost $\sigma = 0.05$ uporabili za generiranje vseh mnenj.

Na tem mestu ponovimo, da je bil namen eksperimenta pokazati, kako lahko odločitveni model vpliva na delovanje modela zaupanja. Zaradi tega smo uporabili le tranzitivni scenarij. Če bi želeli omenjene modele temeljito testirati, bi morali uporabiti več scenarijev, predvsem pa bi jih morali – kot bo jasno v nadaljevanju – testirati brez odločitvenih modelov. Tako pa omenjen scenarij služi zgolj kot minimalni primer, s katerim bomo pokazali veljavnost naše trditve.

Tabela 6.2

Povzetek rezultatov testiranja. Iz tabele razberemo, da izbira odločitvenega modela lahko vpliva na delovanje modela zaupanja. Prav tako je pomembna izbira preslikave oz. praga za določitev ocen; ko slednje višamo, se večja tudi koristnost, natančnost pa je največja pri srednjih (S) vrednostih. Zato številke v oklepajih prikazujejo razvrstitev modelov po rezultatih z uporabo srednjih pragov in preslikav.

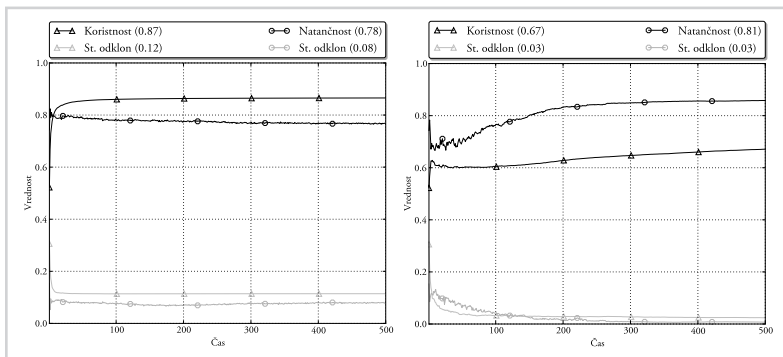
Model zaupanja	Maksimalna izbira		Verjetnostna izbira	
	Natančnost	Koristnost	Natančnost	Koristnost
ARH _N	0.62	0.79	0.75	0.62
ARH _S	0.77 (2)	0.87 (2)	0.81 (4)	0.67 (2)
ARH _V	0.73	0.94	0.78	0.74
BRS	0.68 (3)	0.79 (3)	0.92 (2)	0.56 (5)
ET _N	0.49	0.50	0.65	0.58
ET _S	0.62 (4)	0.58 (5)	0.72 (5)	0.71 (1)
ET _V	0.57	0.79	0.59	0.85
TRA _N	0.48	0.65	0.83	0.53
TRA _S	0.55 (5)	0.78 (4)	0.87 (3)	0.60 (4)
TRA _V	0.54	0.96	0.67	0.68
YSS	0.91 (1)	0.95 (1)	0.96 (1)	0.65 (3)

6.2.5 Rezultati

Vsako testiranje smo ponovili tridesetkrat, pri čemer smo spreminjali začetno naključno vrednost (*angl.* random seed). Nato smo dobljene rezultate združili v *povprečna testiranja*, ki so prikazana na Slikah 6.1-6.5; posamezna točka grafa predstavlja "povprečno" točko, ki smo jo izračunali kot povprečje tridesetih testiranj. Na vsaki sliki imamo dva grafa: levi prikazuje rezultate z uporabo maksimalne izbire, desni pa verjetnostne. Grafi prikazujejo evolucijo izračuna natančnosti in koristnosti, medtem ko so njihove reprezentativne vrednosti zapisane v oklepajih. Slednje dodatno povzemamo v Tabeli 6.2. Nadalje smo rezultate testirali za statistično značilnost. Pri preverjanju razlik v koristnosti smo uporabili Welchov t-test [75], pri preverjanju razlik v natančnosti pa Studentov parni t-test, kjer smo vrednosti paroma primerjalni po časovnih enotah. Vse trditve so statistično značilne s p-vrednostjo 0.05, razen v primerih, kjer izrecno zapišemo drugače.

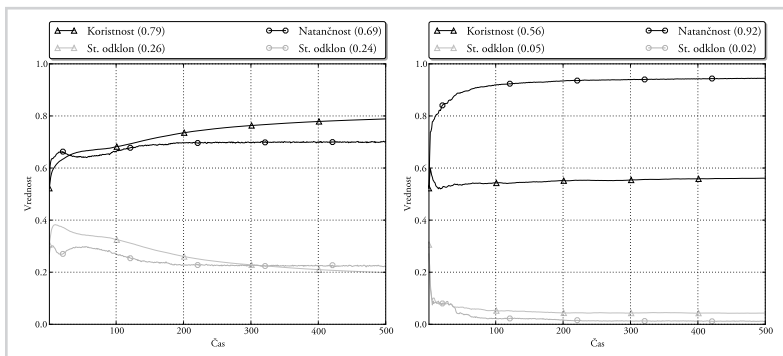
Slika 6.1

Rezultati modela Abdul-Rahman, Hailes (ARH) s srednjimi preslikavami (ARH_M) ter z maksimalno (levo) in z verjetnostno izbiro (desno). Model dosega dobre rezultate z obema odločitvenima modeloma. Zaradi kvalitativnih stopenj zaupanja je njegova natančnost omejena na 0.87.



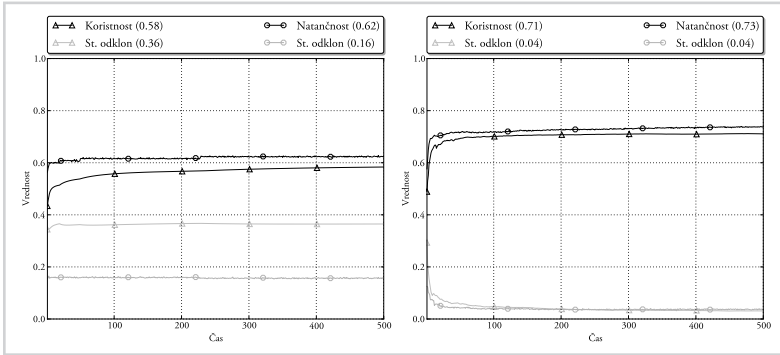
Slika 6.2

Rezultati BRS z maksimalno (levo) in z verjetnostno izbiro (desno). Ker model uporablja predpostavko tranzitivnosti, potrebuje več raziskovanja. Tako je njegova natančnost z maksimalno izbiro povprečna, z verjetnostno pa odlična. Zanimivo je, da pri verjetnostni izbiri dosega najslabšo koristnost navkljub odlični natančnosti: posledica majhnih razlik med izračunanimi stopnjami zaupanja.



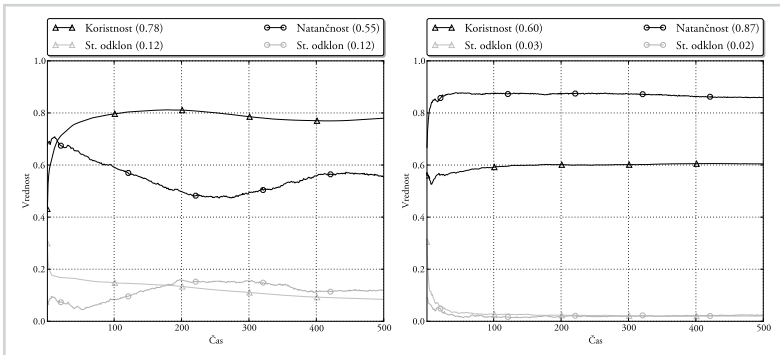
6.2.6 Diskusija

Dobljeni rezultati podpirajo trditev, da odločitveni model vpliva na delovanje modela zaupanja. Če razvrstimo testirane modele glede na koristnost, dobimo drugačen vrstni red pri uporabi maksimalne kot pri uporabi verjetnostne izbire. O tem se lahko prepričamo, če primerjamo tretji in peti stolpec Tabele 6.2. Tako je pri uporabi odločitvenega modela maksimalna izbira najboljši model YSS, medtem ko je pri verjetnostni izbiri najboljši EigenTrust. Nadalje lahko izračunamo korelacijo med dobljenimi rezultati; Kendallov koeficient rangov vrne -0.2, kar pomeni, da so rezultati celo rahlo negativno korelirani. Hipoteza drži tudi, če primerjamo delovanje modelov z ostalimi preslikavami in pragovi, saj koeficient korelacije rangov vrne 0.5 in -0.1 za nizke (N)



Slika 6.3

Rezultati EigenTrust (ET) z maksimalno (levo) in verjetnostno izbiro (desno). Ker potrebuje dovolj raziskovanja, je njegova koristnost pri maksimalni izbiri najnižja, pri verjetnostni pa najvišja. Prav tako je zanimivo, da pri verjetnostni izbiri dosega najboljšo koristnost z najslabšo natančnostjo. Razlog je v tem, da EigenTrust ne meri negativnega zaupanja, kar pokažemo s primerom na Sliki 6.6.



Slika 6.4

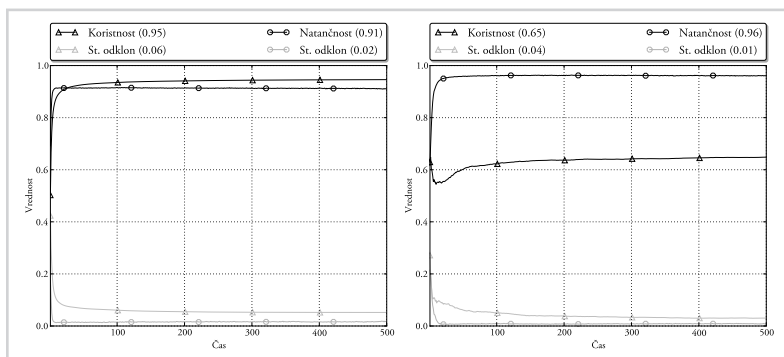
Rezultati modela TRAS z maksimalno (levo) ter verjetnostno izbiro (desno). Pri maksimalni izbiri rezultati precej variirajo, saj lahko model pade v *lokalno past*, ko zaradi grobih vhodnih podatkov slabo oceni najbolj zaupanja vrednega agenta. Pri verjetnostni izbiri je natančnost boljša, medtem ko je nizka koristnost posledica nizkih razlik med izračunanimi stopnjami zaupanja.

in visoke (V) pragove. Če primerjamo delovanje modelov glede na natančnost izračunanega zaupanja (drugi in četrti stolpec Tabele 6.2), so korelacije za nizke, srednje in visoke preslikave oz. pragove zaporedoma 0.6, 0.4 in 0.6, kar je malenkost boljše, vendar tudi ta metrika kaže, da je izbira odločitvenega modela vplivna. Dodajmo, da pri maksimalni izbiri model BRS precej variira, zato razlika v koristnosti med modeloma BRS (0.79) in TRAS (0.78) ter BRS (0.79) in ARHS (0.87) ni statistično značilna, razlika med ARHS (0.87) in TRAS (0.78) pa je. Vendar omenjeno na končni rezultat eksperimenta in našo ugotovitev ne vpliva, saj rezultati testiranj v odvisnosti od izbranega odločitvenega modela v vseh primerih slabo korelirani.

Pri modelih ARH, EigenTrust in Travos se z višanjem pragov oz. preslikav viša tudi njihova koristnost, medtem ko je natančnost najvišja pri srednjih vrednostih. Razlog

Slika 6.5

Rezultati modela Yu, Singh, Sycara (YSS) z maksimalno (levo) ter z verjetnostno izbiri (desno). YSS dosega dobre rezultate z obema odločitvenima modeloma. Pomanjkanje raziskovanja pri maksimalni izbiri nadomešča z dobrim delovanjem mehanizma za detekcijo lažnih mnenj.



za takšno obnašanje je v načinu dodeljevanja zmožnosti v testnem scenariju. Te so dodeljene naključno in enakomerno, zaradi česar srednje preslikave in pragovi priskrbijo modelom zaupanja najnatančnejše vhodne podatke, saj delijo interval $[0, 1]$ enakomerno. Denimo, če nastavimo prag na 0.75, potem bo kar 75% agentov podajalo storitve, ki jih bo agent Alpha ocenil z negativno oceno. Tako bo model zaupanja kar 75% agentov imel za enako zaupanja nevredne in bo med njimi le težka razlikoval. Po drugi strani pa visoki pragovi pomagajo pri razlikovanju med agenti z visokimi zmožnostmi, kar vodi do višjih koristnosti. Tako lahko vidimo, kako enostavno je modele pripraviti do tega, da dosežajo večje koristnosti za ceno nižje natančnosti. Če bi kot večina obstoječih namenskih testnih okolij merili le koristnost, bi ta ugotovitev ostala skrita.

Modeli zaupanja dosežejo višje koristnosti z maksimalno izbiri kot z verjetnostno. Slednje niti ni presenetljivo, saj verjetnostna izbira že po definiciji izbira med vsemi agenti, tudi med takimi z nižjimi zmožnostmi. Analogno pa modeli z verjetnostno izbiri dosežejo višje natančnosti. Tudi to ni presenetljivo, saj slednja vsebuje več raziskovanja.

Pri maksimalni izbiri rezultati variirajo bolj kot pri verjetnostni, kar je posebej izrazito pri modelih BRS ter EigenTrust (Sliki 6.2 in 6.3). Višja varianca je posledica pomanjkanja raziskovanja in posledično premajhnega števila izkušenj z različnimi agenti. Ker omenjena modela uporabljata predpostavko *transitivnosti* – prepričanje, da bo nekdo, ki dobro opravlja storitve, tudi podajal resnična mnenja – je pomanjkanje izkušenj še posebej problematično. Če takšen model napačno oceni izkušnje, kar je pri pomanjkanju raziskovanja zelo verjetno, bo napačno tudi vrednotil vsa prejeta mnenja

in končne izračune zaupanja še dodatno pokvaril.

Modela ARH in YSS delujeta dobro z obema odločitvenima modeloma. Oba dosega visoko koristnost z maksimalno ter visoko natančnost z verjetnostno izbiro. Natančnost modela YSS je visoka tudi pri maksimalni izbiri, saj nadomešča pomanjkanje izkušenj z dobrim delovanjem mehanizma za detekcijo lažnih mnenj. Po drugi strani pa je natančnost modela ARH zaradi kvalitativnih stopenj zaupanja omejena. V našem scenariju – 100 agentov z enakomerno dodeljenimi zmožnostmi – znaša teoretična zgornja meja njegove natančnosti 0.87. V to se lahko prepričamo s primerom, kjer mora ARH dva agenta s podobnima, a še vedno drugačnima, zmožnostma označiti z isto stopnjo zaupanja. Denimo, ARH_S označi vse agente, katerih zmožnosti padejo v interval $[0.50, 0.75)$, z isto stopnjo zaupanja.

Vpliv odločitvenega modela je zelo viden pri modelu BRS. Ko zamenjamo maksimalno izbiro z verjetnostno, njegova natančnost skoči iz 0.70 na kar 0.92. Takšna razlika je posledica delovanja mehanizma za diskontiranje mnenj, ki potrebuje določeno mero raziskovanja. A ker maksimalna izbira tega ne dopušča, je model v kombinaciji z njo dojemljiv za napake. Prav tako je zelo neintuitivno, da BRS z verjetnostno izbiro dosega nizko koristnost – celo najnižjo od vseh modelov. Natančni izračuni zaupanja bi vendar morali voditi k dobrim izbiram! Razlago moramo tokrat poiskati v načinu izračuna zaupanja. Če (zelo) poenostavimo, BRS vrši račune tako, da vsakemu agentu priredi začetno stopnjo zaupanja (0.5), ki jo ob prejemu novih podatkov posodablja v majhnih inkrementih: pri pozitivnih podatkih jo večja, pri negativnih pa manjša. Zaradi tega bodo izračunane stopnje zaupanja zgoščene okoli vrednosti 0.5, hkrati pa bo tudi razlika v stopnji zaupanja med najbolj in najmanj zmožnim agentom nizka. Posledično bodo pri verjetnostni izbiri vsi agenti imeli zelo podobne možnosti, da so izbrani; zaupanja vredni agenti ne bodo imeli kakšne občutne prednosti pred ostalimi. Če bi hoteli koristnost povečati, bi morali izračunane stopnje zaupanja bolj razpršiti.

Rezultati modela Travos precej variirajo. Medtem ko je njegova natančnost pri maksimalni izbiri slaba (0.55), je pri verjetnostni precej boljša (0.87). Pri tem omenimo, da govorimo o srednjih vrednostih pragov (TRA_S), saj natančnost pri nizkem in visokem pragu drastično pade. Zanimivo je tudi, da Travos dosega od vseh modelov najvišjo koristnost, ko uporabi visoki prag in maksimalno izbiro (0.96), medtem ko mu YSS (0.95) in ARH_V (0.94) tesno sledita. Pri tem je zanimivo, da lahko Travos izbira zmožljive ponudnike storitev, kljub temu, da je njegova natančnost porazna – le 0.54, kar je le za odtенок boljše kot golo ugibanje. To lahko pojasnimo s tem, da

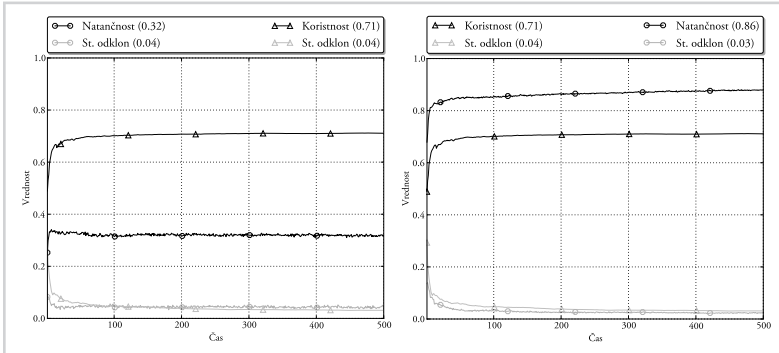
zna model (zaradi visokega praga) natančno izračunati zaupanje do najzmogljivejših agentov (tistih, katerih zmožnosti padejo v interval $[0.80, 1.00]$), do preostalih pa ne. Sicer ve, da so ostali agenti manj zmogljivi, a med njimi ne zna pravilno ločiti.

Travos uporablja binarne ocene interakcij, ki dajejo grobo sliko o kakovosti interakcij, zaradi česar lahko model ob pomanjkanju raziskovanja pade v t. i. *lokalno past*. S tem izrazom označujemo stanje, v katerem model zmotno meni, da je nek agent najbolj zaupanja vreden in ga zato vedno izbere za ponudnika storitve, saj se ne zaveda, da obstajajo drugi agenti z višjimi zmožnostmi. Denimo, da je zmožnost nekega agenta 0.75 in da Travos uporablja prag 0.50. Če na začetku, ko še nima nobene izkušnje, za partnerja izbere tega agenta, bo interakcija z njim zelo verjetno uspešna, zato bo stopnjo zaupanja do agenta povečal. Povečanje bo konstantno in povsem neodvisno konkretnega izida interakcije; pomembno je le, da je interakcija uspešna. V naslednji časovni enoti bo Alpha izbral istega agenta in Travos bo zaupanje vanj še okreplil. Podobno se bo zgodilo v vsaki časovni enoti. Sčasoma bo tak agent dobil največjo možno stopnjo zaupanja (v Travosu je to 1.00) pa čeprav so zmožnosti nekaterih drugih agentov večje. Takšni pasti se lahko Travos izogne, če bodisi nastavi visok prag – v tem primeru bodo le agenti z visokimi zmožnostmi dobili pozitivne ocene – bodisi raziskuje dovolj in ima interakcije z več različnimi agenti.

Zamenjava odločitvenega modela je pri merjenju koristnosti najbolj očitna pri modelu EigenTrust. Pri maksimalni izbiri slednji doseže najnižjo koristnost, pri verjetnosti pa najvišjo. Podobno kot BRS tudi EigenTrust uporablja predpostavko tranzitivnosti in zato potrebuje določeno mero raziskovanja, da se njegovi izračuni ustalijo. Pri maksimalni izbiri je slednje še posebej očitno, saj so rezultati bodisi zelo dobri bodisi zelo slabi, v povprečju pa slabši kot pri ostalih modelih. Omenjena dvojnost se kaže v obliki visokega standardnega odklona na levem grafu Slike 6.3. A ko ga sklopimo z verjetnostno izbiro, EigenTrust doseže najvišjo koristnost.

Visoka koristnost ter nizka natančnost modela EigenTrust sta pri verjetnosti izbiri neintuitivni. Kako lahko model izbira verjetnostno in dosega visoko koristnost, če je pa natančnost izračunov nizka? Odgovor se ponovno skriva v metodi izračuna zaupanja. V razdelku 2.4.3 smo omenili, da je EigenTrust znan po tem, da ne meri t. i. negativnega zaupanja. V okolju ATB to pomeni, da bo EigenTrust prisodil agentom, katerih zmožnosti so manj kot 0.5, stopnjo zaupanja 0.0.² Z drugimi besedami, za

²V praksi ni nobena vrednost povsem 0.0; t. i. lokalno zaupanje – izračunano zgolj iz izkušenj – je 0.0, končne stopnje zaupanja pa so neničelne, saj jih mnenja rahlo povečajo. A še vedno gre za števila, ki so nekaj



Slika 6.6

Model EigenTrust ne izračunava *negativnega zaupanja*. Posledično je izračunano zaupanje do agentov z nizkimi zmožnostmi (levo) zelo nenatančno, medtem ko so ostali izračuni (desno) dokaj natančni.

vsakega agenta, katerega zmožnost je manj kot 0,5, se smatra, da je povsem zaupanja nevreden. Posledici takega izračuna sta dve. Prvič, izračunano zaupanje ima nizko stopnjo natančnosti, saj se spodnji polovici agentov po zmožnostih pripiše enaka stopnja zaupanja. In drugič, omenjeni agenti imajo izredno majhne – skoraj nične – možnosti, da bodo pri verjetnostnem izboru izbrani; navadno bodo izbrani le agenti, katerih zmožnosti so nad 0,5, kar vodi do višjih koristnosti. Opisana dualnost je predstavljena na Sliki 6.6.

Opisan primer je še en dokaz več, da merjenje koristnosti daje delne rezultate, saj ta ocenjuje zgolj kakovost zaupanja, ki je pripeljalo do odločitev, ne pa tudi tistega, ki ni. Zato je metrika natančnosti veliko temeljitejša, saj oceni kakovost celotnega zaupanja. Slednje je pomembno, saj v praksi, ko izbiramo med potencialnimi ponudniki, ne izbiramo zgolj na podlagi izračunanega zaupanja. V takih primerih si želimo, da je izračunano zaupanje natančno tako do zmogljivih kot tudi manj zmogljivih agentov. Glavne ugotovitve lahko strnemo v naslednje točke.

1. Odločitveni modeli igrajo pomembno vlogo pri ocenjevanju modelov zaupanja, saj določajo razmerje med raziskovanjem in izrabljanjem, kar lahko vpliva na delovanje modelov.
2. Določeni odločitveni modeli so primernejši za nekatere modele zaupanja, z drugimi pa ne delujejo dobro. Če modele zaupanja in odločitvene modele kombiniramo poljubno, lahko dobimo dvomljive rezultate.

velikostnih redov manjša od stopenj zaupanja, ki jih uživajo agenti z zmožnostmi večjimi od 0,5.

3. Koristnost ocenjuje zgolj kakovost zaupanja, ki je pripeljalo do odločitev, ne pa tudi tistega, ki ni. Za temeljito testiranje je potrebno uporabiti metrike, ki ocenijo celotno zaupanje, denimo natančnost.

6.3 *Evaluacija kvalitativnega modela zaupanja*

V tem razdelku ovrednotimo delovanje kvalitativnega modela zaupanja (QTM), ki smo ga predstavili v poglavju 4. Pri testiranju smo njegove rezultate primerjali z rezultati modelov ARH, BRS, EigenTrust, Travos in YSS. Pri modelih ARH, EigenTrust in Travos smo uporabili srednje preslikave oz. pragove, zato tovrstne oznake v tem razdelku izpuščamo.

6.3.1 *Uporabljeni scenariji*

Pri evaluaciji smo uporabili štiri scenarije, in sicer: *preprosti*, *tranzitivni*, *dinamični* in *usmerjen napad*. Vsak scenarij je vseboval 100 agentov in vsi – z izjemo dinamičnega – so se izvajali 500 časovnih enot. V vseh scenarijih – z izjemo scenarija usmerjen napad – so se vsi agenti poznali in drug o drugem vedno podajali mnenja. V nasprotju s testiranjem predstavljenim v razdelku 6.2 modelov zaupanja tukaj nismo kombinirali z odločitvenimi mehanizmi, temveč so partnerje za interakcije dodeljevali scenariji. Podrobnosti posameznega scenarija so opisane v sledečih podrazdelkih.

Preprosti scenarij (P-10 in P-100)

Preprosti scenarij je agentom dodelil zmožnosti naključno in enakomerno. Pri dodeljevanju modelov prevar je 90 naključno izbranim agentom priredil lažnive modele, preostalim 10 pa resnične. Tako so bili agenti pri sporočanju mnenj konsistentni: 90 agentov je vedno sporočalo lažna, 10 pa vedno resnična mnenja. V primerjavi s tranzitivnim scenarijem iz razdelka 6.2.4 v tem scenariju zmožnosti agentov za opravljanje storitev niso vplivale na njihovo poštenost za podajanje mnenj; če je nek agent imel visoko zmožnost, to ni pomenilo, da je z večjo verjetnostno podajal resnična mnenja. Tu omenimo, da smo namenoma uporabili ekstremno razmerje med številom agentov, ki lažejo, in tistimi, ki govorijo resnico – kar 90% agentov je vedno lagalo, saj nas je zanimalo delovanje modelov v težavnih pogojih.

Preprosti scenarij smo poganjali z dvema različnima konfiguracijama. V prvi, označeni s Preprosti-10, je Alpha imel interakcije zgolj z 10 na začetku testiranja naključno

izbranimi agenti. V drugi, označeni s Preprosti-100, pa je Alpha imel interakcije z vsemi stotimi. Ker je testiranje trajalo 500 časovnih enot, je v konfiguraciji P-10 Alpha z vsakim od desetih agentov napravil 50 interakcij, v konfiguraciji P-100 pa z vsakim (od stotih) le 5. Z omenjenima konfiguracijama smo preverjali delovanje modelov pri različni količini raziskovanja.

Tranzitivni scenarij (T-10 in T-100)

Tranzitivni scenarij je praktično enak tistemu, ki smo ga opisali v razdelku 6.2.4; zmožnosti so dodeljene naključno in enakomerno, modeli prevar pa v linearni odvisnosti od dodeljenih zmožnosti – bolj zmožni agenti sporočajo resnična mnenja bolj verjetno kot manj zmožni. Pomembna razlika med preprostim in tranzitivnim scenarijem je tudi v tem, da pri slednjem agenti niso podajali mnenj konsistentno. Denimo, če je zmožnost nekega agenta znašala 0.75, je tak agent govoril resnico o 75% agentov, medtem ko je za ostalih 25% lagal; nasprotno bi v preprostem scenariju tak agent bodisi vedno govoril resnico bodisi vedno lagal. Zmožnosti in modeli prevar se tekom testiranja niso spreminjali. Edina razlika med tem scenarijem in tranzitivnim scenarijem iz razdelka 6.2.4 je ta, da je pri slednjem partnerje za interakcije izbral agent Alpha (oz. njegov odločitveni model), tukaj pa partnerje določa scenarij. Glede tega smo imeli dve konfiguraciji: v T-10 je imel Alpha interakcije z 10 naključnimi agenti, v T-100 pa z vsemi.

Dinamični scenarij (D-10 in D-100)

Z dinamičnim scenarijem smo preverjali, kako dobro se modeli zaupanja prilagajajo spremembam v okolju. Scenarij je na začetku naključno in enakomerno dodelil zmožnosti ter na podlagi tega še modele prevar – tudi tu je bila uporabljena predpostavka tranzitivnosti. Nato je zmožnosti in modele prevar tekom testiranja spreminjal: v vsaki časovni enoti je izbral naključnega agenta, mu naključno spremenil zmožnost in ustrezno popravil modele prevar; če je npr. zmožnost agenta padla iz 0.90 na 0.10, to ni pomenilo le, da je ta agent postal slab ponudnik storitev, temveč tudi, da je od takrat dalje podajal resnična mnenja o le 10% agentov. Da bi bili rezultati nazornejši, smo ta scenarij izvajali 3000 časovnih enot. Tudi tu smo uporabili dve konfiguraciji, ki podajata število agentov, s katerimi je agent Alpha imel interakcije: D-10 ter D-100.

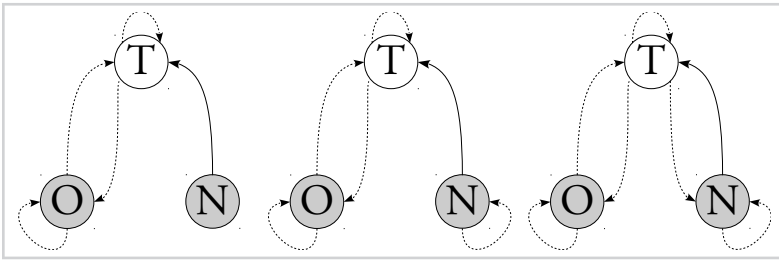
Scenarij usmerjen napad (UN-1, UN-2 in UN-3)

S scenarijem usmerjen napad smo preverjali delovanje modelov zaupanja ob različnih strategijah podajanja lažnih mnenj. Tako QTM kot tudi ostali testirani modeli zaupanja uporabljajo *eksogene* mehanizeme [9] za detekcijo lažnih mnenj. Ti resničnost mnenj preverijo tako, da prejeta mnenja primerjajo z lastnimi izkušnjami. Osnovna ideja tega scenarija pa je selektivno laganje, pri katerem Alpha lažnivcev ne more ujeti na laži, saj se mnenja, ki jih lahko preveri, izkažejo za resnična, mnenja, ki jih ne more, pa so dejansko zlagana. V ta namen so bili agenti v scenariju razdeljeni v tri disjunktne skupine, in sicer: (i) *napadalci*, ki širijo lažna mnenja, (ii) *tarče*, o katerih napadalci širijo laži, ter (iii) *ostali* agenti, ki z napadom niso povezani. V tem scenariju je bil cilj testiranega modela čim natančneje oceniti zaupanje do tarč, pri čemer Alpha z njimi ni imel interakcij; te je imel zgolj z ostalimi agenti ter napadalci. A medtem ko so ostali agenti podajali resnična mnenja, je bil cilj napadalcev čim bolj zмести model zaupanja.

Lastnosti agentov v posameznih skupinah so bile naslednje. Zmožnosti napadalcev ter ostalih agentov so bile navadne (dodeljene naključno z intervala $[0,0, 1,0]$), zmožnosti tarč pa izredno nizke (dodeljene naključno iz intervala $[0,0, 0,1]$). Agenti iz skupine ostali so drugo drugemu in o agentih iz skupine tarč podajali resnična mnenja. Podobno so tudi tarče dajale resnična mnenja o ostalih tarčah ter o agentih iz skupine ostali; tarče in ostali agenti so tako predstavljali dobronameren del sistema, napadalci pa zlonamernega. Glede na to kako so napadalci podajali mnenja, ločimo tri različice scenarija. V prvi (UN-1) so napadalci zgolj širili laži o tarčah: trdili so, da so bile tarče zelo zaupanja vredne, čeprav so bile njihove zmožnosti zelo nizke. V drugi (UN-2) so napadalci podajali lažna mnenja o tarčah, obenem pa še resnična o preostalih napadalcih. Njihov namen je bil pridobiti kredibilnost za podajanje mnenj, saj so se tista, katera je Alpha lahko preveril, izkazala za resnična. Tretja različica scenarija (UN-3) je razširila drugo (UN-2), in sicer tako da so tarče dodatno kolaborirale z napadalci, saj so o njih podajale resnična mnenja. Zaradi slednjega je bila ta različica izjemno težavna, saj z vidika modela zaupanja ni bilo nobene razlike med dobronamernim in zlonamernim delom sistema; model je prejemal o tarčah konfliktna mnenja, ni pa mogel ugotoviti, katera so resnična, saj so bili tako napadalci kot ostali agenti s tarčami enako dobro povezani. Vse tri različice scenarija so predstavljene na Sliki 6.7.

Ta scenarij predstavlja *napad orkestrirane samopromocije* (*angl.* orchestrated self-promoting attack), ki ga opisujejo Hoffman *et al.* [76]. Poleg tega vsebuje še dodaten

Slika 6.7



Prikaz podajanja mnenj med člani posameznih skupin (Tarče, Napadalci, Ostali). Leva shema podaja različico UN-1, srednja UN-2, desna pa UN-3. Črtkane puščice označujejo resnična, polne pa lažna mnenja. Izvor puščice ponazarja skupino poročevalca, ponor pa skupino agenta, o katerem mnenje govori. Alpha ima interakcije zgolj z agenti iz osenčenih skupin.

element: Alpha nima interakcij s tarčami in tako ne more ujeti napadalcev pri laganju. V praksi bi tak napad lahko izvedli s pomočjo napada Sybil (*angl.* Sybil attack) [77], pri katerem napadalec ustvari več identitet z namenom pokoritve napadenega sistema. Zato lahko trdimo, da je usmerjen napad mogoče izvesti na kateremkoli sistemu, ki je dojemljiv za napad Sybil. Pri tem dodajmo, da sta različici UN-1 in UN-2 realnejši, saj bi pri različici UN-3 morale tarče aktivno sodelovati z napadalci, kar pa se smatra za težko izvedljivo. Na tej predpostavki denimo temelji večina shem za detekcijo napada Sybil [78].

Scenarij usmerjen napad smo izvedli v vseh treh različicah, pri čemer smo pri vsaki spreminjali število agentov v skupinah napadalci in ostali; za vsako različico smo izvedli dve konfiguraciji. V lažji je bilo 40 napadalcev, 20 tarč in 40 ostalih agentov, v težji pa 60 napadalcev, 20 tarč in 20 ostalih agentov. Število agentov, s katerimi je Alpha imel interakcije, je bilo pri obeh konfiguracijah 20, izbrani pa so bili naključno izmed napadalcev in ostalih agentov. Pri posamezni oznaki podamo različico scenarija in konfiguracijo; denimo UN-1-40 ponazarja prvo različico usmerjenega napada v konfiguraciji s 40 napadalci.

6.3.2 Rezultati

Pri testiranju smo vsako konfiguracijo pognali tridesetkrat in rezultate združili v *povprečna testiranja*, ki jih prikazujemo na Slikah 6.8-6.13. Tudi tu smo zaradi lažjega primerjanja rezultatov izračunali reprezentativne vrednosti metrik; te so prikazane v legendah posameznih slik in povzete v Tabeli 6.3. Nadalje smo v vsaki konfiguraciji testirali ali je razlika med povprečnimi natančnostmi statistično značilna; uporabili smo

Studentov parni t-test, kjer smo rezultate paroma primerjalni po časovnih enotah. Vse trditve, ki jih podamo v nadaljevanju, so statistično značilne s p-vrednostjo 0.05.

Ponovimo, da kvalitativna modela (ARH, QTM) ne moreta doseči skrajnih natančnosti, kot sta 0 in 1. Razlog za to so njune stopnje zaupanja, ki jih je mnogo manj (ARH 4, QTM 5) kot agentov (100), ki jih je potrebno razvrstiti. Po načelu golobjnjaka (*angl.* pigeonhole principle) bodo nekateri agenti označeni z isto stopnjo zaupanja, čeprav bodo njihove zmožnosti različne. Ker je različnih zmožnosti teoretično neskončno, se ni težko prepričati, da sta zgornji in spodnji meji natančnosti za ARH in QTM zaporedoma 0.87 in 0.13 ter 0.90 in 0.10.

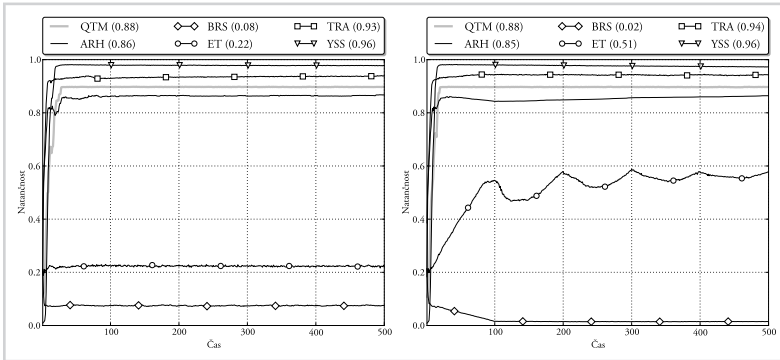
Tabela 6.3

Povzetek rezultatov testiranja. Z decimalnimi števili so prikazane povprečne vrednosti natančnosti, s števili v oklepajih pa mesta, ki jih modeli zaupanja dosegajo v posameznih konfiguracijah scenarijev.

	QTM	ARH	BRS	ET	TRA	YSS
P-10	0.88 (3)	0.86 (4)	0.08 (6)	0.22 (5)	0.93 (2)	0.96 (1)
P-100	0.88 (3)	0.85 (4)	0.02 (6)	0.51 (5)	0.94 (2)	0.96 (1)
T-10	0.89 (2)	0.72 (5)	0.75 (4)	0.71 (6)	0.85 (3)	0.93 (1)
T-100	0.89 (3)	0.84 (5)	0.92 (2)	0.75 (6)	0.89 (3)	0.96 (1)
D-10	0.88 (1)	0.54 (6)	0.77 (2)	0.57 (5)	0.61 (4)	0.68 (3)
D-100	0.89 (1)	0.56 (6)	0.87 (2)	0.57 (5)	0.61 (4)	0.63 (3)
UN-1-40	0.83 (1)	0.80 (2)	0.56 (4)	0.54 (5)	0.68 (3)	0.17 (6)
UN-1-60	0.78 (1)	0.74 (2)	0.18 (5)	0.53 (4)	0.58 (3)	0.16 (6)
UN-2-40	0.83 (1)	0.77 (2)	0.56 (4)	0.55 (5)	0.55 (5)	0.60 (3)
UN-2-60	0.78 (2)	0.30 (5)	0.19 (6)	0.53 (3)	0.49 (4)	0.89 (1)
UN-3-40	0.83 (1)	0.79 (2)	0.56 (4)	0.54 (6)	0.57 (3)	0.55 (5)
UN-3-60	0.25 (5)	0.29 (4)	0.18 (6)	0.47 (3)	0.51 (2)	0.85 (1)

6.3.3 Diskusija

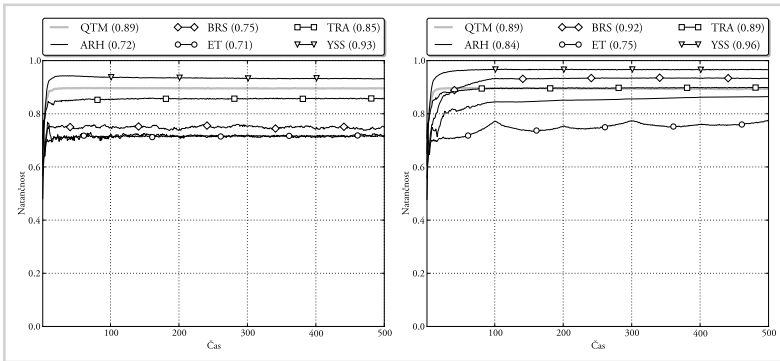
V preprostem scenariju se je QTM obnesel dobro, čeprav je dosegel le tretji rezultat. Razlog za to so kvalitativne stopnje zaupanja, katerih natančnost je omejena na 0.90, medtem ko YSS in Travos take omejitve nimata. EigenTrust in BRS sta se obnesla še posebej slabo, saj je ta scenarij kršil predpostavko tranzitivnosti. To je bilo sploh pogubno za BRS, saj so bili njegovi izračuni na trenutke skoraj povsem obratno-korelirani



Slika 6.8

Rezultati v preprostem scenariju. Leva slika prikazuje rezultate konfiguracije, v kateri dobi Alpha izkušnje z 10% agentov (P-10), desna pa rezultate, kjer dobi izkušnje z vsemi agenti (P-100).

z zmoglostmi. EigenTrust je v konfiguraciji P-100 svoje izračune sicer izboljšal, a ne dosti; ko je nabral več izkušenj, je izboljšal ocene do vnaprej zaupanih agentov (*angl.* pre-trusted peers), a končni izračuni – kombinacija zaupanja v vnaprej zaupane agente in globalnega zaupanja – so bili kljub temu slabi. Izračun globalnega zaupanja je namreč osnovan na predpostavki tranzitivnosti.

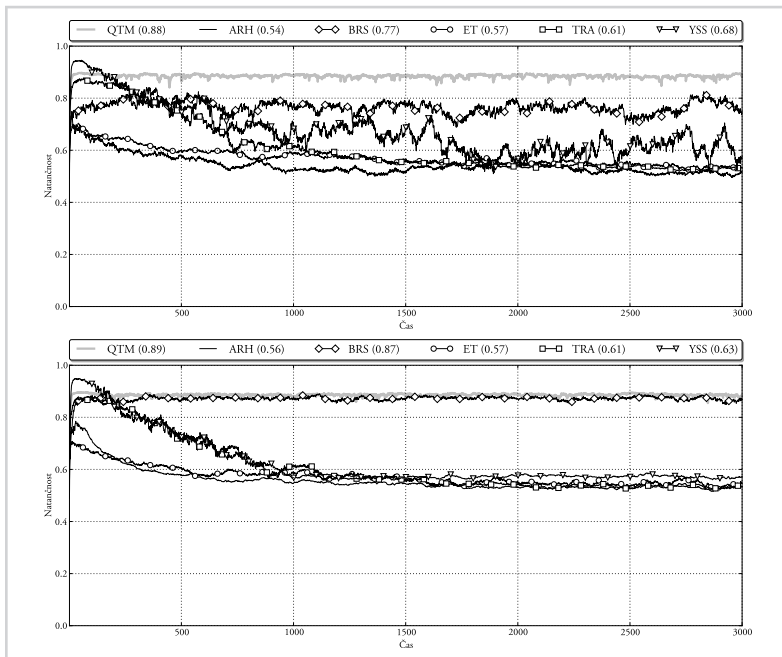


Slika 6.9

Rezultati v tranzitivnem scenariju. Leva slika prikazuje rezultate konfiguracije, v kateri dobi Alpha izkušnje z 10% agentov (T-10), desna pa rezultate, kjer dobi izkušnje z vsemi agenti (T-100).

V tranzitivnem scenariju s konfiguracijo T-10 je QTM dosegel drugi, v konfiguraciji T-100 pa tretji rezultat. Razlog, zakaj ga je v konfiguraciji T-100 BRS prehitel (in Travos dohitel), ni v tem, da QTM z dodatnimi izkušnjami svojih izračunov ne bi izboljšal, temveč v tem, da je že v konfiguraciji T-10 dosegel svojo zgornjo mejo. Vsi ostali modeli so z dodatnimi izkušnjami izboljšali natančnost zaupanja – najbolj BRS,

ARH in Travos – medtem ko je QTM svoj teoretični maksimum že dosegel, ko je imel izkušnje z zgolj 10% agentov. Iz tega sledi, da QTM deluje dobro tudi s malo izkušnjami, kar je posebej zaželena lastnost, saj so te navadno redke. V primerjavi s preprostim scenarijem sta se modela BRS in EigenTrust tukaj obnesla bolje; predpostavka tranzitivnosti je bila izpolnjena. A EigenTrust je kljub temu dosegel povprečen rezultat, ker ne izračunava negativnega zaupanja. O tem smo že govorili na koncu razdelka 6.2.6 (glej Sliko 6.6).

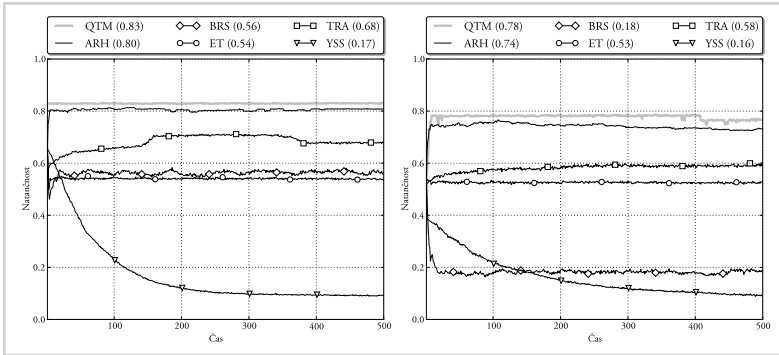


Slika 6.10

Rezultati v dinamičnem scenariju. Zgornja slika prikazuje rezultate konfiguracije, v kateri ima Alpha interakcije z 10 agenti (D-10), spodnja pa rezultate, kjer dobi izkušnje z vsemi (D-100).

V obeh konfiguracijah dinamičnega scenarija je QTM dosegel najboljši rezultat. V splošnem je bila natančnost vseh modelov na začetku dobra, a je sčasoma usahnila, saj so agenti začeli obnašanje spreminjati; le QTM in BRS sta obdržala začetno raven natančnosti. V konfiguraciji D-100 sta omenjena modela dosegla zelo podoben rezultat (QTM je sicer imel za 0.02 točki boljšega), razlika med njima pa je postala izrazitejša v konfiguraciji D-10, saj je BRS postal nestabilen. Ostali modeli so imeli s prilagajanjem

na spremembo precej težav. To niti ni presenetljivo, saj imata med testiranimi modeli le BRS in QTM ustrezne mehanizme za časovno diskontiranje podatkov.



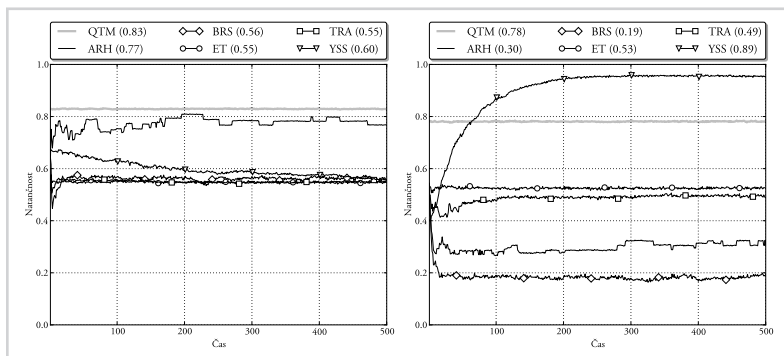
Slika 6.11

Rezultati v prvi različici scenarija usmerjen napad (UN-1). Leva slika prikazuje rezultate konfiguracije s 40 (UN-1-40), desna pa s 60 napadalci (UN-1-60).

Pri scenariju usmerjen napad smo uporabili rahlo spremenjeno različico natančnosti: ker je bil napad osredotočen na tarče, je natančnost v tem scenariju vrednotila zgolj zaupanje do njih. V prvi različici scenarija in v konfiguraciji s 40 napadalci (UN-1-40), sta se QTM (0.83) in ARH (0.80) obnesla najboljše, medtem ko je bila Travosova natančnost (0.68) že rahlo načeta. Izračuni BRS (0.56) in EigenTrust (0.54) so bili nekorelirani, YSS (0.17) pa je napad povsem dotolkel.

Ko smo povečali število napadalcev na 60 (UN-1-60) so QTM (0.78), ARH (0.74), EigenTrust (0.53) in YSS (0.16) obdržali podobne rezultate, medtem ko sta ga Travos (0.58) in BRS (0.18) vidno pokvarila. Pri modelu QTM je do izraza prišel izračun družbene povezanosti. Ker so bili tarče in napadalci družbeno nepovezani – napadalci so dajali mnenja o tarčah, ki pa napadalcev sploh niso poznale – so dobila mnenja napadalcev majhno težo. Dober rezultat je imel tudi ARH, saj je ignoriral mnenja napadalcev, ker jih ni mogel preveriti. Travos je diskontiral mnenja napadalcev, a ne dovolj močno. Ko se je njihovo število povečalo, so njihova mnenja prevladala. Od vseh modelov je najslabši rezultat dosegel YSS. Razlog za to tiči v delovanju njegovega mehanizma diskontiranja mnenj, saj diskontira kredibilnost le tistih poročevalcev, katerih mnenja lahko preveri. Ker je v tem scenariju lahko preverjal zgolj mnenja tarč in ostalih agentov, je njihove kredibilnosti (rahlo, a vztrajno) diskontiral, medtem ko so kredibilnosti napadalcev ostale na začetnih vrednostih (1.0), zato so sčasoma mnenja napadalcev prevladala. Rezultata modelov EigenTrust in BRS sta bila slaba, saj scenarij

usmerjen napad ne upošteva predpostavke tranzitivnosti.



Slika 6.12

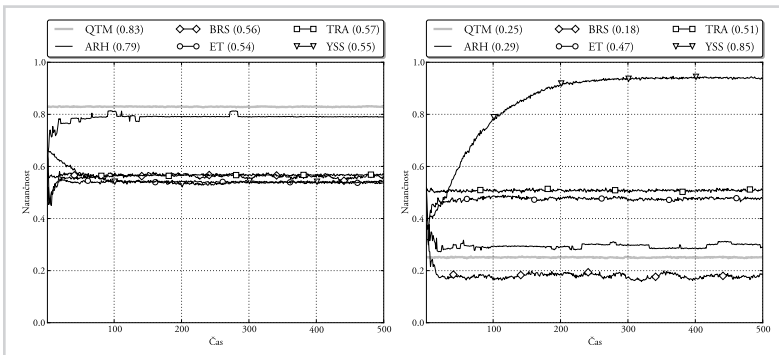
Rezultati v drugi različici scenarija usmerjen napad (UN-2). Leva slika prikazuje rezultate konfiguracije s 40 napadalci (UN-2-40), desna pa s 60 napadalci (UN-2-60).

V drugi različici usmerjenega napada s 40 napadalci (UN-2-40) so QTM (0.83), ARH (0.77), EigenTrust (0.55) in BRS (0.56) dosegli podobne rezultate kot v prvi (UN-1-40), medtem ko se je v isti primerjavi Travosov rezultat (0.55) poslabšal. Zanimivo je, da se je rezultat modela YSS (0.60) v primerjavi s konfiguracijo UN-1-40 izboljšal, a je bil v splošnem še vedno slab. V konfiguraciji s 60 napadalci (UN-2-60) so QTM (0.78), EigenTrust (0.53) in Travos (0.49) obdržali podobne rezultate kot v konfiguraciji s 40, medtem ko sta se rezultata ARH (0.30) in BRS (0.19) vidno poslabšala.

Tukaj je izstopal YSS (0.89), saj je dosegel najboljšo natančnost. To je še posebej presenetljivo, saj je konfiguracija UN-2-60 zahtevnejša kot v UN-2-40. Rezultat postane razumljiv, ko pobližje preučimo delovanje te različice scenarija ter si ogledamo, kako YSS diskontira mnenja. Na kratko, YSS je dosegel dober rezultat, ker je sčasoma bolj diskontiral mnenja napadalcev kot tarč in ostalih agentov. Slednje pa se ni zgodilo zaradi tega, ker bi model predvideval, da je z mnenji napadalcev kaj narobe, temveč zgolj slučajno, saj je imel z napadalci več interakcij (izkušenj). Faktor, s katerim YSS diskontira kredibilnost posameznega poročevalca, se izračuna kot absolutna razlika med podanim mnenjem ter dejansko izkušnjo. In ker YSS predstavlja informacije (vrednosti mnenj ter ocene izkušenj) z realnimi števili, ki jih je neskončno mnogo, se mnenje in izkušnja v praksi nikoli ne ujemata povsem – nekaj šuma bo vedno prisotnega. Zaradi tega bo YSS *vedno vsaj malo znižal kredibilnost poročevalca, katerega mnenje bo lahko preveril*; tudi če je mnenje resnično, se bo kredibilnost malenkostno znižala.

A ker YSS kredibilnosti spreminja z množenjem, tudi majhni popravki sčasoma pripeljejo do večjih sprememb. Ker je bilo v tej konfiguraciji več napadalcev kot tarč in ostalih agentov skupaj (60 napadalcev ter 20 tarč in 20 ostalih) in ker je scenarij izbiral partnerje za interakcije naključno med napadalci in ostalimi agenti, je YSS dobil več izkušenj z napadalci kot z ostalimi. Posledično je lahko preveril mnenja napadalcev večkrat kot mnenja ostalih agentov. Zaradi tega so kredibilnosti napadalcem padle hitreje kot kredibilnosti ostalih agentov. Ponovimo, da so dober rezultat modela YSS omogočile specifične okoliščine tega scenarija – če bi iz generiranih podatkov (mnenj in izkušenj) odstranili šum, ali če bi k napadalcem dodali že samo enega agenta, ki ne bi podajal mnenj o ostalih napadalcih, bi YSS dosegel podobno slabe rezultate kot v prvi različici napada.

QTM je v obeh konfiguracijah dosegel dober rezultat, saj je pravilno ocenil, da so napadalci in tarče slabo družbeno povezani. ARH je deloval dobro zgolj v konfiguraciji s 40 napadalci (0.77), v konfiguraciji s 60 pa je njegova natančnost strmoglavila (0.30). V tej različici napada so napadalci podajali resnična in preverljiva mnenja o drugih napadalcih in tako “prelisičili” ARH, da je njihovim mnenjem verjel. Pri 40 napadalcih se to ni opazilo, pri 60 pa je bilo za model pogubno.



Slika 6.13

Rezultati v tretji različici scenarija usmerjen napad (UN-3). Leva slika prikazuje rezultate konfiguracije s 40 (UN-3-40), desna pa s 60 napadalci (UN-3-60).

V tretji različici usmerjenega napada sta v konfiguraciji s 40 napadalci (UN-3-40) QTM (0.83) in ARH (0.79) delovala najbolje, medtem ko so bili rezultati ostalih modelov skoraj nekorelirani: BRS (0.56), EigenTrust (0.54), Travos (0.57) in YSS (0.55). Ko smo povečali število napadalcev (konfiguracija UN-3-60) so se rezultati v splošnem poslabšali: QTM (0.25), ARH (0.29) in BRS (0.18) so dejansko izračunali obratno-

korelirane rezultate, medtem ko sta Travos (0.51) in EigenTrust (0.47) ostala nekorelirana. Zopet se je delovanje modela YSS (0.85) drastično izboljšalo. Razlaga za to je tudi tokrat enaka kot zgoraj: ker je model imel več izkušenj z napadalci, je njihove kredibilnosti znižal hitreje. Ta konfiguracija je tudi edini primer, kjer je QTM povsem odpovedal. To se je zgodilo zaradi tega, ker so v tej različici napada tarče sodelovale z napadalci tako, da so o njih podajale (resnična) mnenja. Zaradi tega je QTM ocenil, da so tarče in napadalci dobro povezani in pripisal mnenjem slednjih večjo težo. Ko so postali napadalci številčnejši, je to postala težava, saj so njihova mnenja prevladala.

6.3.4 Povzetek rezultatov

Ko strnemo rezultate, lahko ugotovimo, da QTM podaja natančne izračune in da deluje dobro tako z malo kot tudi z več izkušnjami. Čeprav ni dosegel najboljšega rezultata v vseh scenarijih, je bila izračunana natančnost vedno visoka in blizu njegove teoretične zgornje meje. To omejitev smo zavedno sprejeli ob odločitvi za kvalitativne stopnje zaupanja tj. namenoma smo žrtvovali en del natančnosti za večjo prijaznost uporabniku.

6.4 Zaključek

V tem poglavju smo namensko testno okolje Alpha uporabili v praksi. Pokazali smo, da odločitveni model lahko vpliva na delovanje modela zaupanja, saj določa razmerje med raziskovanjem in izrabljanjem. Še več, pokazali smo, da sta lahko natančnost izračunanega zaupanja in koristnost, ki jo agent z modelom pridobi, celo nepovezana. Zato je pomembno, da pri evaluaciji uporabljamo metrike, ki vrednotijo zaupanje neposredno ne pa posredno preko odločitev, ki iz zaupanja izhajajo. Nadalje smo pokazali, da kvalitativen model zaupanja dosega dobre rezultate že ob malem številu izkušenj ter da je njegov mehanizem za ugotavljanje zlaganih mnenj učinkovit. Pri vsem tem smo uporabili namensko testno okolje, kar dokazuje, da je programska rešitev splošna in uporabna.

Zaključek

7

7.1 Povzetek vsebine

V disertaciji smo utemeljili uporabo kvalitativnih metod na področju računalniško podprtega obvladovanja zaupanja. V poglavju 2 smo podali osnovne definicije in nadaljevali s pregledom področja, kjer smo izpostavili, da je kvalitativnih modelov zaupanja občutno manj kot kvantitativnih, poleg tega pa smo tudi opozorili, da večina obstoječih kvalitativnih modelov obdeluje kvalitativne podatke neprimerno. Podali smo razloge in argumentirali, zakaj so v primerih, ko morajo podatke interpretirati ljudje, kvalitativni modeli primernejši kot kvantitativni. Zatem smo predstavili obstoječe načine testiranja modelov zaupanja in izpostavili njihove omejitve.

V poglavju 3 smo definirali kvalitativno množico podatkov, podali metodo, ki iz seznama kvalitativnih vrednosti izbere najbolj reprezentativno, ter predstavili nekatere njene lastnosti. V poglavju 4 smo zgradili kvalitativni model zaupanja, ki izračunava zaupanje na podlagi preteklih interakcij ter mnenj, ki jih agenti podajo drug o drugem. V modelu smo uporabili kvalitativne podatke na mestih, kjer lahko pride do interakcije z uporabnikom: pri podajanju ocen v interakcijah, pri izražanju mnenj ter pri podajanju izračunanih stopenj zaupanja. Razvili smo različne hevristične metode, da bi ublažili vpliv zlaganih mnenj: upoštevamo njihovo starost, zanesljivost agentov, ki jih podajo, ter stopnjo družbene povezanosti med agenti, ki jih podajo, ter agenti, ki jih mnenja zadevajo.

V poglavju 5 smo predstavili namensko testno okolje za vrednotenje modelov zaupanja. Naša motivacija je izhajala iz dejstva, da obstoječe rešitve bodisi od modelov zaupanja zahtevajo, da ti uporabljajo vnaprej dogovorjene stopnje zaupanja, bodisi kombinirajo modele zaupanja z odločitvenimi modeli in nato ocenjujejo učinkovitost prvih po tem, kako dobro delujejo v kombinaciji z drugimi. Ker smo želeli primerjati delovanje kvalitativnih modelov s kvantitativnimi in ker smo domnevali, da lahko odločitveni model vpliva na delovanje modela zaupanja, smo zasnovali namensko testno okolje, ki zgornjih omejitev nima. Naša rešitev omogoča testiranje modelov zaupanja brez odločitvenih modelov, pri čemer od njih ne zahteva, da uporabljajo vnaprej dogovorjene stopnje zaupanja. Poleg tega lahko okolje vrednoti modele zaupanja po ustaljenih načinih, kot je denimo merjenje koristnosti.

Zatem smo v poglavju 6 zasnovali eksperiment, da bi preverili, ali kombiniranje odločitvenih modelov z modeli zaupanja dejansko vpliva na delovanje slednjih. V eksperimentu smo kombinirali pet modelov zaupanja z dvema odločitvenima modeloma

in primerjali rezultate testiranj. Izkazalo se je, da je izbira odločitvenega modela pomembna in da poljubna modela zaupanja nista nujno enako združljiva s poljubnima odločitvenima modeloma. Na podlagi te ugotovitve, smo v drugem delu tega poglavja primerjali delovanje kvalitativnega modela zaupanja s petimi najbolj odmevnimi deli v literaturi. Pri tem smo zavestno uporabili protokol evaluacije, ki testira modele zaupanja brez odločitvenega modela. Kvalitativni model zaupanja je v vseh scenarijih dosegel dobre rezultate, večinoma celo najboljše. Razlog, zakaj so v nekaterih scenarijih bili drugi modeli boljši, je zgotovljen v naravi kvalitativnih podatkov. Slednji imajo namreč v primerjavi s kvantitativnimi nižjo ločljivost.

7.2 Odprta vprašanja

Tekom disertacije smo naleteli na več problemov. Nekatere smo rešili v celoti, nekatere pa le ublažili in jih pustili delno odprte. V tem razdelku podajamo tiste, za katere menimo, da si zaslužijo nadaljnjo obravnavo.

Posplošiti izračun reprezentativne kvalitativne vrednosti. Pri predstavitvi metode za izračun reprezentativne kvalitativne vrednosti smo vedno operirali z množico, ki vsebuje pet kvalitativnih vrednosti. Zanimivo bi bilo raziskati, kako splošiti Izreka 1 in 2, da bi veljala tudi v večjih množicah.

Najti mero razpršenosti kvalitativnih podatkov. Pri izračunu povprečja rangov bi lahko izračunali standardni odklon kot mero razpršenosti, pri predlagani metodi izračuna reprezentativne vrednosti pa take mere nimamo. Zato bi bilo smotno raziskati, kaj lahko uporabimo kot njen nadomestek.

Izboljšati izračun stopnje družbene povezanosti. Pri izračunu mnenj preverjamo družbeno povezanost med poročevalcem in agentom, ki ga mnenje zadeva. Pri tem preprosto izračunamo stopnjo podobnosti med množicama znancev poročevalca ter znancev agenta. Tukaj smo zaradi obvladovanja kompleksnosti zavedno izbrali preprosto rešitev. Za doseganje boljših rezultatov bi bilo smotno preskusiti pristope iz področja analize kompleksnih omrežij [79].

Izboljšati metodo ugotavljanja lažnih mnenj. Podobno bi lahko izboljšali način, kako v eno mero združiti nedavnost mnenja, kredibilnost poročevalca ter stopnjo družbene povezanosti med poročevalcem in agentom. V Algoritmu 3 preprosto uporabimo najmanjšo izmed vrednosti, smotno pa bi raziskati alternativne pristope.

Poiskati dober odločitveni model. Če želimo, da kvalitativni model zaupanja uporabi avtonomen agent, potrebujemo odločitveni model. Zato bi bilo zanimivo poiskati takega, ki bo zagotavljal, da bo QTM ob sprejemanju dobrih odločitev dosegal tudi dobro natančnost.

Razširiti namensko testno okolje Alpha. Namensko testno okolje lahko trenutno testira le tiste modele zaupanja, ki zaupanje izračunavajo iz izkušenj in mnenj. Programsko rešitev bi bilo dobro razširiti, da bi lahko z njo testirali modele, ki uporabljajo druge vire informacij, kot so denimo družbena omrežja. Prav tako bi bilo smotno pripraviti scenarije, ki implementirajo različne napade na modele zaupanja.

Dokaza

A

A.1 Uvodni gradniki

V tem dodatku dokažemo pravilnost Izreka 1 in 2. Najprej pripravimo gradnike, ki jih bomo potrebovali pri dokazovanju.

Lema 1: Naj bo $\vec{p} = [f_{ZS}, f_S, f_N, f_D, f_{ZD}]$ porazdelitveni vektor neke zbirke. Potem velja

$$(a) \quad 0 \leq f_v \leq 1 \text{ za vsak } v \in \mathbb{V},$$

$$(b) \quad \sum_{v \in \mathbb{V}} f_v = 1.$$

Dokaz. Obe lastnosti sledita neposredno iz Definicije 12. \square

Ker metrika iz enačbe (3.1) uporablja vgnezden izraz in ker je delo s slednjimi nerodno, uvedemo pojem kumulativnega porazdelitvenega vektorja.

Definicija 20 (Kumulativni porazdelitveni vektor): Kumulativni porazdelitveni vektor $\vec{P} = [F_{ZS}, F_S, F_N, F_D, F_{ZD}]$ izračunamo iz pripadajočega porazdelitvenega vektorja $\vec{p} = [f_{ZS}, f_S, f_N, f_D, f_{ZD}]$ kot podaja spodnji izraz.

$$F_v = \sum_{\substack{v_x \leq v \\ v_x \in \mathbb{V}}} f_{v_x}$$

Porazdelitvene vektorje in njihove komponente bomo označevali z malimi črkami (\vec{p} in f_D), njihove kumulativne različice pa z velikimi (\vec{P} in F_D).

Lema 2 (Lema o urejenosti): Naj bo $\vec{P} = [F_{ZS}, F_S, F_N, F_D, F_{ZD}]$ kumulativni porazdelitveni vektor neke zbirke. Potem velja naslednje.

$$(a) \quad F_{ZS} \leq F_S \leq F_N \leq F_D \leq F_{ZD}$$

$$(b) \quad 0 \leq F_{ZS}$$

$$(c) \quad F_{ZD} = 1$$

Dokaz. Vse lastnosti sledijo neposredno iz preteklih definicij in lem: trditev (a) sledi iz Definicije 20, trditev (b) iz Definicije 20 in Leme 1 ter trditev (c) iz Definicije 20 in Leme 1. \square

Podobno kot smo definirali razdaljo me med porazdelitvenimi vektorji, lahko definiramo razdaljo med kumulativnimi porazdelitvenimi vektorji.

$$D(\vec{X}, \vec{Y}) = \sqrt{\sum_{v \in \mathbb{V}} (\vec{X}(v) - \vec{Y}(v))^2} \quad (\text{A.1})$$

Lema 3: Naj bosta \vec{x} in \vec{y} porazdelitvena vektorja in naj bosta \vec{X} in \vec{Y} njeni kumulativni različici kot jo opredeljuje Definicija. 20. Potem velja

$$d(\vec{x}, \vec{y}) = D(\vec{X}, \vec{Y})$$

Dokaz. Enakost sledi neposredno iz definicij metrik v izrazih (3.1, A.1) in definicije kumulativnega porazdelitvenega vektorja v Definiciji 20. \square

A.2 Dokaz enakosti z mediano

V matematični enciklopediji MathWorld¹ [80] je mediana definirana sledeče: “Mediano v podani zbirki vrednosti izračunamo tako, da vse vrednosti uredimo po velikosti in kot mediano izberemo tisto, ki se v tej ureditvi nahaja na sredini. Če imamo v zbirki sodo število vrednosti, potem mediane ne moremo enolično določiti; v takih primerih se kot rezultat določi število, ki je aritmetična sredina srednjih dveh vrednosti.” A ker v našem primeru operiramo s kvalitativnimi vrednostmi, aritmetične sredine dveh kvalitativnih vrednosti ne moremo izračunati. Zato bomo v takih primerih kot vrednost mediane izbrali večjo izmed srednjih dveh. Izračun mediane kumulativnega porazdelitvenega vektorja \vec{P} je v kanonični obliki podan v enačbi (A.2).

¹<http://mathworld.wolfram.com>

$$\text{Mediana}(\vec{P}) = \begin{cases} ZS & F_{ZS} > \frac{1}{2} \\ S & F_{ZS} \leq \frac{1}{2} \wedge F_S > \frac{1}{2} \\ N & F_S \leq \frac{1}{2} \wedge F_N > \frac{1}{2} \\ D & F_N \leq \frac{1}{2} \wedge F_D > \frac{1}{2} \\ ZD & F_D \leq \frac{1}{2} \end{cases} \quad (\text{A.2})$$

Nazadnje še pretvorimo reprezentativne porazdelitvene vektorje iz Izreka 1 v njihove kumulativne različice kot narekuje Definicija 20: $\vec{R}_{ZS} = [1, 1, 1, 1, 1]$, $\vec{R}_S = [0, 1, 1, 1, 1]$, $\vec{R}_N = [0, 0, 1, 1, 1]$, $\vec{R}_D = [0, 0, 0, 1, 1]$, $\vec{R}_{ZD} = [0, 0, 0, 0, 1]$. Sedaj lahko z dokazom pričnemo. Samo dokazovanje poteka precej enostavno: pokazali bomo, da je poljubnemu kumulativnemu porazdelitvenemu vektorju, \vec{P} , katerega mediana je vrednost $v = \text{Mediana}(\vec{P})$, vedno najbližji kumulativni reprezentativni vektor \vec{R}_v .

Dokaz. Imejmo porazdelitveni vektor $\vec{p} = [f_{ZS}, f_S, f_N, f_D, f_{ZD}]$ in njegovo kumulativno različico $\vec{P} = [F_{ZS}, F_S, F_N, F_D, F_{ZD}]$. Najprej izračunajmo razdalje med \vec{P} in vsemi kumulativnimi reprezentativnimi vektorji.

$$D(\vec{P}, \vec{R}_{ZS}) = \sqrt{(1 - F_{ZS})^2 + (1 - F_S)^2 + (1 - F_N)^2 + (1 - F_D)^2} \quad (\text{A.3})$$

$$D(\vec{P}, \vec{R}_S) = \sqrt{F_{ZS}^2 + (1 - F_S)^2 + (1 - F_N)^2 + (1 - F_D)^2} \quad (\text{A.4})$$

$$D(\vec{P}, \vec{R}_N) = \sqrt{F_{ZS}^2 + F_S^2 + (1 - F_N)^2 + (1 - F_D)^2} \quad (\text{A.5})$$

$$D(\vec{P}, \vec{R}_D) = \sqrt{F_{ZS}^2 + F_S^2 + F_N^2 + (1 - F_D)^2} \quad (\text{A.6})$$

$$D(\vec{P}, \vec{R}_{ZD}) = \sqrt{F_{ZS}^2 + F_S^2 + F_N^2 + F_D^2} \quad (\text{A.7})$$

Najprej preverimo primer $ZS = \text{Mediana}(\vec{P})$. Tu moramo pokazati sledeče.

$$F_{ZS} > \frac{1}{2} \iff \begin{cases} D(\vec{P}, \vec{R}_{ZS}) < D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) < D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) < D(\vec{P}, \vec{R}_D) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) < D(\vec{P}, \vec{R}_{ZD}) \end{cases} \quad (\text{A.8})$$

V sistem (A.8) vstavimo izraze iz enačb (A.3-A.7) in dobimo naslednje.

$$F_{ZS} > \frac{1}{2} \iff \begin{aligned} & \frac{1}{2} < F_{ZS} \wedge \\ & 1 < F_{ZS} + F_S \wedge \\ & \frac{3}{2} < F_{ZS} + F_S + F_N \wedge \\ & 2 < F_{ZS} + F_S + F_N + F_D \end{aligned} \quad (\text{A.9})$$

Iz leme o urejenosti (Lema 2) izhaja, da je prva neenakost na desni strani sistema (A.9) najstrožja, zaradi česar lahko ostale tri izpustimo. Tako pridemo do prvega želenega rezultata.

$$F_{ZS} > \frac{1}{2} \iff \frac{1}{2} < F_{ZS} \quad (\text{A.10})$$

Kot drugo preverimo $S = \text{Mediana}(\vec{P})$. Sedaj moramo pokazati sledeče.

$$F_{ZS} \leq \frac{1}{2} \wedge F_S > \frac{1}{2} \iff \begin{aligned} & D(\vec{P}, \vec{R}_S) \leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ & D(\vec{P}, \vec{R}_S) < D(\vec{P}, \vec{R}_N) \wedge \\ & D(\vec{P}, \vec{R}_S) < D(\vec{P}, \vec{R}_D) \wedge \\ & D(\vec{P}, \vec{R}_S) < D(\vec{P}, \vec{R}_{ZD}) \end{aligned} \quad (\text{A.11})$$

Pri prvi komponenti na desni strani sistema (A.11) uporabimo relacijo večji-ali-enak. Razlog je v tem, da metoda v primeru $D(\vec{P}, \vec{R}_{ZS}) = D(\vec{P}, \vec{R}_S)$ kot rezultat vrne S (večjo izmed ZS in S ; glej Definicijo 14, kjer se uporabi operator \max). Vstavimo enačbe (A.3-A.7) in sistem (A.11) se poenostavi.

$$F_{ZS} \leq \frac{1}{2} \wedge F_S > \frac{1}{2} \iff \begin{aligned} & \frac{1}{2} \geq F_{ZS} \wedge \\ & \frac{1}{2} < F_S \wedge \\ & 1 < F_S + F_N \wedge \\ & \frac{3}{2} < F_S + F_N + F_D \end{aligned} \quad (\text{A.12})$$

Ker druga neenakost na desni strani sistema (A.12) omejuje strožje kot tretja in četrta (Lema 2), lahko tretjo in četrto neenakost izpustimo in tako pridemo do želenega rezultata.

$$F_{ZS} \leq \frac{1}{2} \wedge F_S > \frac{1}{2} \iff \frac{1}{2} \geq F_{ZS} \wedge \frac{1}{2} < F_S \quad (\text{A.13})$$

Kot tretjo preverimo možnost $N = \text{Mediana}(\vec{P})$. Tu pokažemo sledeče.

$$\begin{aligned}
 F_S \leq \frac{1}{2} \wedge F_N > \frac{1}{2} &\iff \begin{aligned} &D(\vec{P}, \vec{R}_N) \leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ &D(\vec{P}, \vec{R}_N) \leq D(\vec{P}, \vec{R}_S) \wedge \\ &D(\vec{P}, \vec{R}_N) < D(\vec{P}, \vec{R}_D) \wedge \\ &D(\vec{P}, \vec{R}_N) < D(\vec{P}, \vec{R}_{ZD}) \end{aligned}
 \end{aligned} \tag{A.14}$$

Vstavimo enačbe (A.3-A.7) in pridemo do sistema (A.15).

$$\begin{aligned}
 F_S \leq \frac{1}{2} \wedge F_N > \frac{1}{2} &\iff \begin{aligned} &1 \geq F_{ZS} + F_S \wedge \\ &\frac{1}{2} \geq F_S \wedge \\ &\frac{1}{2} < F_N \wedge \\ &1 < F_N + F_D \end{aligned}
 \end{aligned} \tag{A.15}$$

Lema 2 zagotavlja, da v desnem delu izraza (A.15) druga komponenta omejuje strožje kot prva, tretja pa strožje kot četrta; izpustimo prvo in četrto ter pridemo do želenega.

$$F_S \leq \frac{1}{2} \wedge F_N > \frac{1}{2} \iff \frac{1}{2} \geq F_S \wedge \frac{1}{2} < F_N \tag{A.16}$$

Sedaj preverimo možnost $D = \text{Mediana}(\vec{P})$. Tu moramo pokazati naslednje.

$$\begin{aligned}
 F_N \leq \frac{1}{2} \wedge F_D > \frac{1}{2} &\iff \begin{aligned} &D(\vec{P}, \vec{R}_D) \leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ &D(\vec{P}, \vec{R}_D) \leq D(\vec{P}, \vec{R}_S) \wedge \\ &D(\vec{P}, \vec{R}_D) \leq D(\vec{P}, \vec{R}_N) \wedge \\ &D(\vec{P}, \vec{R}_D) < D(\vec{P}, \vec{R}_{ZD}) \end{aligned}
 \end{aligned} \tag{A.17}$$

Vstavimo enačbe (A.3-A.7) in pridemo do sistema (A.18).

$$\begin{aligned}
 F_N \leq \frac{1}{2} \wedge F_D > \frac{1}{2} &\iff \begin{aligned} &\frac{3}{2} \geq F_{ZS} + F_S + F_N \wedge \\ &1 \geq F_S + F_N \wedge \\ &\frac{1}{2} \geq F_N \wedge \\ &\frac{1}{2} < F_D \end{aligned}
 \end{aligned} \tag{A.18}$$

Z uporabo Leme 2 se izraz (A.18) poenostavi in pridemo do želenega rezultata.

$$F_N \leq \frac{1}{2} \wedge F_D > \frac{1}{2} \iff \frac{1}{2} \geq F_N \wedge \frac{1}{2} < F_D \tag{A.19}$$

Na koncu preverimo še $ZD = \text{Mediana}(\vec{P})$. Sedaj pokažemo naslednje.

$$F_D \leq \frac{1}{2} \iff \begin{aligned} D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_D) \end{aligned} \quad (\text{A.20})$$

Vstavimo enačbe (A.3-A.7) in pridemo do sistema (A.21).

$$F_D \leq \frac{1}{2} \iff \begin{aligned} 2 &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ \frac{3}{2} &\geq F_S + F_N + F_D \wedge \\ 1 &\geq F_N + F_D \wedge \\ \frac{1}{2} &\geq F_D \end{aligned} \quad (\text{A.21})$$

Zaradi Leme 2 prve tri komponente izpustimo in pridemo do sistema (A.22).

$$F_D \leq \frac{1}{2} \iff \frac{1}{2} \geq F_D \quad (\text{A.22})$$

Pokazali smo, da ekvivalence v izrazih (A.8, A.11, A.14, A.17, A.20) držijo. Tako je jasno, da pri uporabi reprezentativnih vektorjev iz Izreka 1 predlagana metoda vrne enak rezultat kot mediana. \square

A.3 Dokaz enakosti s povprečjem rangov

Izraz povprečje rangov označuje postopek izračuna reprezentativne vrednosti, kjer vsaki kvalitativni vrednosti priredimo število – njen rang – nato pa iz njih izračunamo povprečje. Dobljeno povprečje nato zaokrožimo k najbližjemu celemu številu in kot reprezentativno izberemo tisto kvalitativno vrednost, katere rang sovpa z dobljenim celim številom. Natančen postopek izračuna je podan v enačbi (A.23); oglati oklepaji označujejo *funkcijo zaokroževanja*, medtem ko $\text{RANG}^{-1} : \{1, 2, 3, 4, 5\} \mapsto \mathbb{V}$ inverzno preslikavo iz rangov v kvalitativne vrednosti (osnovne preslikave so podane v Definiciji 11).

$$\text{pr}(\vec{p}) = \text{RANG}^{-1}([1 \cdot f_{ZS} + 2 \cdot f_S + 3 \cdot f_N + 4 \cdot f_D + 5 \cdot f_{ZD}]) \quad (\text{A.23})$$

Povprečje rangov lahko izrazimo tudi s kumulativnimi porazdelitvenimi vektorji.

$$\text{PR}(\vec{P}) = \text{RANG}^{-1}([5 - F_{ZS} - F_S - F_N - F_D]) \quad (\text{A.24})$$

Lema 4: Naj bo \vec{p} porazdelitveni vektor in naj bo \vec{P} njegova kumulativna različka. Potem velja naslednje.

$$\text{pr}(\vec{p}) = \text{PR}(\vec{P})$$

Dokaz. Enakost neposredno sledi iz postopka izračuna povprečja rangov podane-ga v enačbah (A.23, A.24) in definicije kumulativnega porazdelitvenega vektorja v Definiciji 20. \square

Da bo analiza lažja, zapišimo način izračuna povprečja rangov še v kanonični obliki.

$$\text{PR}(\vec{P}) = \begin{cases} ZD & 5 \geq 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{9}{2} \\ D & \frac{9}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{7}{2} \\ N & \frac{7}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{5}{2} \\ S & \frac{5}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{3}{2} \\ ZS & \frac{3}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq 1 \end{cases} \quad (\text{A.25})$$

Pred začetkom dokaza še pretvorimo reprezentativne porazdelitvene vektorje iz Izreka 2 v njihove kumulativne različice: $\vec{R}_{ZS} = [1, 1, 1, 1, 1]$, $\vec{R}_S = [\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, 1]$, $\vec{R}_N = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1]$, $\vec{R}_D = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 1]$, $\vec{R}_{ZD} = [0, 0, 0, 0, 1]$. Sam potek dokazovanja je analogen prejšnjemu: pokazali bomo, da je poljubnemu kumulativnemu porazdelitvenemu vektorju, \vec{P} , katerega povprečje rangov je vrednost $v = \text{PR}(\vec{P})$, vedno najbližji kumulativni reprezentativni vektor \vec{R}_v .

Dokaz. Imejmo porazdelitveni vektor $\vec{p} = [f_{ZS}, f_S, f_N, f_D, f_{ZD}]$ in njegovo kumulativno različico $\vec{P} = [F_{ZS}, F_S, F_N, F_D, F_{ZD}]$. Najprej izračunamo razdaljo med \vec{P} in vsemi kumulativnimi reprezentativnimi vektorji.

$$D(\vec{P}, \vec{R}_{ZS}) = \sqrt{(1 - F_{ZS})^2 + (1 - F_S)^2 + (1 - F_N)^2 + (1 - F_D)^2} \quad (\text{A.26})$$

$$D(\vec{P}, \vec{R}_S) = \sqrt{(\frac{3}{4} - F_{ZS})^2 + (\frac{3}{4} - F_S)^2 + (\frac{3}{4} - F_N)^2 + (\frac{3}{4} - F_D)^2} \quad (\text{A.27})$$

$$D(\vec{P}, \vec{R}_N) = \sqrt{(\frac{1}{2} - F_{ZS})^2 + (\frac{1}{2} - F_S)^2 + (\frac{1}{2} - F_N)^2 + (\frac{1}{2} - F_D)^2} \quad (\text{A.28})$$

$$D(\vec{P}, \vec{R}_D) = \sqrt{(\frac{1}{4} - F_{ZS})^2 + (\frac{1}{4} - F_S)^2 + (\frac{1}{4} - F_N)^2 + (\frac{1}{4} - F_D)^2} \quad (\text{A.29})$$

$$D(\vec{P}, \vec{R}_{ZD}) = \sqrt{F_{ZS}^2 + F_S^2 + F_N^2 + F_D^2} \quad (\text{A.30})$$

Najprej preverimo $ZD = PR(\vec{P})$. Tu pokažemo naslednje.

$$5 \geq 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{9}{2} \iff \begin{aligned} D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_{ZD}) &\leq D(\vec{P}, \vec{R}_D) \end{aligned} \quad (\text{A.31})$$

V sistem (A.31) vstavimo enačbe (A.26-A.30) in dobimo sistem (A.32).

$$\frac{1}{2} \geq F_{ZS} + F_S + F_N + F_D \geq 0 \iff \begin{aligned} 2 &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ \frac{3}{2} &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ 1 &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ \frac{1}{2} &\geq F_{ZS} + F_S + F_N + F_D \end{aligned} \quad (\text{A.32})$$

Ker je na desni strani sistema (A.32) zadnji člen najstrožji, prve tri izpustimo. Zaradi leme o urejenosti (Lema 2, primer (b)) pa lahko spodnjo mejo varno dodamo.

$$\frac{1}{2} \geq F_{ZS} + F_S + F_N + F_D \geq 0 \iff \frac{1}{2} \geq F_{ZS} + F_S + F_N + F_D \geq 0 \quad (\text{A.33})$$

Zdaj preverimo $D = PR(\vec{P})$. Tu pokažimo sledeče.

$$\frac{9}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{7}{2} \iff \begin{aligned} D(\vec{P}, \vec{R}_D) &\leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ D(\vec{P}, \vec{R}_D) &\leq D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_D) &\leq D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_D) &< D(\vec{P}, \vec{R}_{ZD}) \end{aligned} \quad (\text{A.34})$$

V sistem (A.34) vstavimo enačbe (A.26-A.30) in dobimo sistem (A.35). Slednjega z uporabo Leme 2 poenostavimo v sistem (A.36).

$$\frac{3}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{1}{2} \iff \begin{aligned} \frac{5}{2} &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ 2 &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ \frac{3}{2} &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ F_{ZS} + F_S + F_N + F_D &> \frac{1}{2} \end{aligned} \quad (\text{A.35})$$

$$\frac{3}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{1}{2} \iff \frac{3}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{1}{2} \quad (\text{A.36})$$

Naslednji je primer $N = \text{PR}(\vec{P})$, kjer moramo pokazati naslednje.

$$\frac{7}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{5}{2} \iff \begin{aligned} D(\vec{P}, \vec{R}_N) &\leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ D(\vec{P}, \vec{R}_N) &\leq D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_N) &< D(\vec{P}, \vec{R}_D) \wedge \\ D(\vec{P}, \vec{R}_N) &< D(\vec{P}, \vec{R}_{ZD}) \end{aligned} \quad (\text{A.37})$$

V sistem (A.37) vstavimo enačbe (A.26-A.30) in dobimo sistem (A.38). Slednjega z uporabo Leme 2 poenostavimo v sistem (A.39).

$$\frac{5}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{3}{2} \iff \begin{aligned} 3 &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ \frac{5}{2} &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ F_{ZS} + F_S + F_N + F_D &> \frac{3}{2} \wedge \\ F_{ZS} + F_S + F_N + F_D &> 1 \end{aligned} \quad (\text{A.38})$$

$$\frac{3}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{1}{2} \iff \frac{3}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{1}{2} \quad (\text{A.39})$$

Predzadnje preverimo $S = \text{PR}(\vec{P})$. Tu mora držati naslednje.

$$\frac{5}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq \frac{3}{2} \iff \begin{aligned} D(\vec{P}, \vec{R}_S) &\leq D(\vec{P}, \vec{R}_{ZS}) \wedge \\ D(\vec{P}, \vec{R}_S) &< D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_S) &< D(\vec{P}, \vec{R}_D) \wedge \\ D(\vec{P}, \vec{R}_S) &< D(\vec{P}, \vec{R}_{ZD}) \end{aligned} \quad (\text{A.40})$$

V sistem (A.40) vstavimo enačbe (A.26-A.30) in dobimo sistem (A.41). Slednjega z uporabo Leme 2 poenostavimo v sistem (A.42).

$$\frac{7}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{5}{2} \iff \begin{aligned} \frac{7}{2} &\geq F_{ZS} + F_S + F_N + F_D \wedge \\ F_{ZS} + F_S + F_N + F_D &> \frac{5}{2} \wedge \\ F_{ZS} + F_S + F_N + F_D &> 2 \wedge \\ F_{ZS} + F_S + F_N + F_D &> \frac{3}{2} \end{aligned} \quad (\text{A.41})$$

$$\frac{7}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{5}{2} \iff \frac{7}{2} \geq F_{ZS} + F_S + F_N + F_D > \frac{5}{2} \quad (\text{A.42})$$

Nazadnje preverimo še $ZS = \text{PR}(\vec{P})$. Tu pokažemo sledeče.

$$\frac{3}{2} > 5 - F_{ZS} - F_S - F_N - F_D \geq 1 \iff \begin{aligned} D(\vec{P}, \vec{R}_{ZS}) &< D(\vec{P}, \vec{R}_S) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) &< D(\vec{P}, \vec{R}_N) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) &< D(\vec{P}, \vec{R}_D) \wedge \\ D(\vec{P}, \vec{R}_{ZS}) &< D(\vec{P}, \vec{R}_{ZD}) \end{aligned} \quad (\text{A.43})$$

V sistem (A.43) vstavimo enačbe (A.26-A.30) in dobimo sistem (A.44).

$$4 \geq F_{ZS} + F_S + F_N + F_D > \frac{7}{2} \iff \begin{aligned} &F_{ZS} + F_S + F_N + F_D > \frac{7}{2} \wedge \\ &F_{ZS} + F_S + F_N + F_D > 3 \wedge \\ &F_{ZS} + F_S + F_N + F_D > \frac{5}{2} \wedge \\ &F_{ZS} + F_S + F_N + F_D > 2 \end{aligned} \quad (\text{A.44})$$

Ker je prvi člen desne strani sistema (A.44) najstrožji, preostale tri izpustimo. In ker Lema o urejenosti (Lema 2, primera (b) in (c)) pravi, da je vsaka komponenta kumulativnega porazdelitvenega vektorja največ 1, lahko rečemo, da je vsota katerihkoli štirih komponent kumulativnega porazdelitvenega vektorja največ 4. Torej lahko tudi zgornjo mejo varno dodamo.

$$4 \geq F_{ZS} + F_S + F_N + F_D > \frac{7}{2} \iff 4 \geq F_{ZS} + F_S + F_N + F_D > \frac{7}{2} \quad (\text{A.45})$$

Pokazali smo, da ekvivalence v izrazih (A.31, A.34, A.37, A.40, A.43) držijo. S tem je dokaz zaključen. \square

LITERATURA

- [1] Niklas Luhmann, Howard Davis, John Raffan, and Kathryn Rooney. *Trust, and, Power: two works by Niklas Luhmann*. Wiley Chichester, 1979.
- [2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 2010.
- [3] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 2006.
- [4] Nicholas R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 2001.
- [5] Sascha Ossowski, editor. *Agreement Technologies*. Springer Verlag, 2013.
- [6] Vishwas Patil and R. Shyamasundar. Trust management for e-transactions. *Sadhana*, 30:141–158, 2005.
- [7] Audun Josang and Nam Tran. Trust management for e-commerce. In *In proceedings Virtual Banking*, 2000.
- [8] Lars Rasmussen and Sverker Jansson. Simulated social control for secure internet commerce. In *Proceedings of the 1996 workshop on New security paradigms*, 1996.
- [9] Audun Josang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2007.
- [10] Daniel Kahneman, Paul Slovic, and Amos Tversky, editors. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, 1982.
- [11] Denis Trček. A formal apparatus for modeling trust in computing environments. *Mathematical and Computer Modelling*, 2009.
- [12] Denis Trček. Ergonomic trust management in pervasive computing environments – qualitative assessment dynamics. In *5th International Conference on Pervasive Computing and Applications (ICPCA)*, 2010.
- [13] Denis Trček. Trust management methodologies for the web. In *Proceedings of the 7th international conference on Reasoning web: semantic technologies for the web of data*, 2011.
- [14] Isaac Pinyol and Jordi Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 2011.
- [15] Diego Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, 1990.
- [16] Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In Cristiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*. Springer Netherlands, 2001.
- [17] Harrison Mcknight and Norman Chervany. The meanings of trust. Technical report, University of Minnesota, 1996.
- [18] Daniel W. Manchala. Trust metrics, models and protocols for electronic commerce transactions. In *Distributed Computing Systems, 1998. Proceedings. 18th International Conference on*, 1998.
- [19] Audun Josang and Stéphane Lo Presti. Analysing the relationship between risk and trust. In Christian Jensen, Stefan Poslad, and Theo Dimitrakos, editors, *Trust Management*. Springer Berlin Heidelberg, 2004.
- [20] Damjan Kovač. *Obvladovanje zaupanja v storitveno usmerjenih arhitekturah*. PhD thesis, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2009.
- [21] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 2005.
- [22] Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, 2002.

- [23] Ramón Hermoso, Holger Billhardt, Roberto Centeno, and Sascha Ossowski. Effective use of organisational abstractions for confidence models. In *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS '06*, 2006.
- [24] Huaizhi Li and Mukesh Singhal. Trust management in distributed systems. *Computer*, 2007.
- [25] Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 2004.
- [26] Alférez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.
- [27] Audun Jøsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
- [28] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, 2003.
- [29] Audun Jøsang and Jochen Haller. Dirichlet reputation systems. In *The Second International Conference on Availability, Reliability and Security (ARES)*, 2007.
- [30] Nitin Chiluka, Nazareno Andrade, Dimitra Gkorou, and Johan Pouwelse. Personalizing eigentrust in the face of communities and centrality attack. In *IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)*, 2012.
- [31] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 2006.
- [32] Bin Yu, Munindar P. Singh, and Katia Sycara. Developing trust in large-scale peer-to-peer systems. In *IEEE First Symposium on Multi-Agent Security and Survivability*, 2004.
- [33] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science*, 1989.
- [34] Javier Carbó, José Molina, and Jorge Dávila. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems*, 2003.
- [35] Sarvapali D. Ramchurn, Nicholas R. Jennings, Carlos Sierra, and Lluís Godó. Devising a trust model for multi-agent interactions using confidence and reputation. *Applied Artificial Intelligence*, 2004.
- [36] Nathan Griffiths, Kuo-Ming Chao, and Muhammad Younas. Fuzzy trust for peer-to-peer systems. In *Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on*, 2006.
- [37] Siyuan Liu, Han Yu, Chunyan Miao, and Alex C. Kot. A fuzzy logic based reputation model against unfair ratings. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013.
- [38] Shanshan Song, Kai Hwang, Runfang Zhou, and Y-K Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *Internet Computing, IEEE*, 2005.
- [39] Jordi Sabater, Mario Paolucci, and Rosaria Conte. Repage: Reputation and image among limited autonomous partners. *Journal of artificial societies and social simulation*, 2006.
- [40] Ramón Hermoso, Roberto Centeno, Holger Billhardt, and Sascha Ossowski. Extending virtual organizations to improve trust mechanisms. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1489–1492. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [41] Michael Schumacher and Sascha Ossowski. The governing environment. In *Environments for Multi-Agent Systems II*, pages 88–104. Springer, 2006.
- [42] Damjan Kovač and Denis Trček. Qualitative trust modeling in soa. *Journal of Systems Architecture*, 2009.
- [43] Denis Trček. Towards trust management standardization. *Computer Standards & Interfaces*, 2004.
- [44] Denis Trček. An integrative architecture for a sensor-supported trust management system. *Sensors*, 2012.
- [45] Susan Jamieson. Likert scales: how to (ab)use them. *Medical Education*, 2004.
- [46] Thomas R. Knapp. Treating ordinal scales as interval scales: an attempt to resolve the controversy. *Nursing Research*, 1990.
- [47] Elke Kahler, Anja Rogausch, Edgar Brunner, and Wolfgang Himmel. A parametric analysis of ordinal quality-of-life data can lead to erroneous results. *Journal of Clinical Epidemiology*, 2008.
- [48] Robert Axelrod and William Hamilton. The evolution of cooperation. *Science*, 1981.
- [49] Stephen Marsh. *Formalising trust as a computational concept*. PhD thesis, University of Stirling, 1994.
- [50] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, 2002.

- [51] Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 2000.
- [52] Amirali Salehi-Abari and Tony White. Dart: A distributed analysis of reputation and trust framework. *Computational Intelligence*, 2012.
- [53] Partheeban Chandrasekaran and Babak Esfandiari. A model for a testbed for evaluating reputation systems. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011.
- [54] Karen K. Fullam, Tomas B. Klos, Guillaume Muller, Jordi Sabater, Andreas Schlosser, K. Suzanne Barber, Jeffrey S. Rosenschein, Laurent Vercouter, and Marco Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and MultiAgent Systems*, 2005.
- [55] Reid Kerr and Robin Cohen. Treet: the trust and reputation experimentation and evaluation testbed. *Electronic Commerce Research*, 2010.
- [56] Félix Gómez Mármol and Gregorio Martínez Pérez. Trmsim-wsn, trust and reputation models simulator for wireless sensor networks. In *Proceedings of the 2009 IEEE international conference on Communications*, 2009.
- [57] Félix Gómez Mármol and Gregorio Martínez Pérez. Trust and reputation models comparison. *Internet Research*, 2011.
- [58] Andrew G. West, Sampath Kannan, Insup Lee, and Oleg Sokolsky. An evaluation framework for reputation management systems. In Zheng Yan, editor, *Trust Modeling and Management in Digital Environments*. IGI Global, 2010.
- [59] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 2008.
- [60] Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 2000.
- [61] W. T. Luke Teacy, Michael Luck, Alex Rogers, and Nicholas R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, 2012.
- [62] Stanley S. Stevens. On the theory of scales of measurement. *Science*, 1946.
- [63] Didier Dubois and Henri Prade. Weighted minimum and maximum operations in fuzzy set theory. *Information Sciences*, 1986.
- [64] Francisco Herrera, Enrique Herrera Viedma, and Jose Luis Verdegay. A rational consensus model in group decision making using linguistic assessments. *Fuzzy Sets and Systems*, 1997.
- [65] Ronald Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 1995.
- [66] Lluís Godo and Vincenc Torra. On aggregation operators for ordinal qualitative information. *IEEE Transactions on Fuzzy Systems*, 2000.
- [67] M. J. North, T. R. Howe, N. T. Collier, and J. R. Vos. A declarative model assembly infrastructure for verification and validation. In *Advancing Social Simulation: The First World Congress*, 2007.
- [68] Michael North, Nicholson Collier, Jonathan Ozik, Eric Tatara, Charles Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with repast symphony. *Complex Adaptive Systems Modeling*, 2013.
- [69] Isaac Pinyol, Jordi Sabater-Mir, and Guifre Cuni. How to talk about reputation using a common ontology: From definition to implementation. In *Ninth Workshop on Trust in Agent Societies*, 2007.
- [70] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003.
- [71] Paul Resnick and Richard Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In *The Economics of the Internet and E-Commerce*. Emerald Group Publishing Limited, 2002.
- [72] Ravi Kumar and Sergei Vassilvskii. Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web*, 2010.
- [73] Oracle. Jar file specification – service provider. Java SE 6 Documentation, 2011.
- [74] Gnu general public license, 2007. URL <http://www.gnu.org/licenses/gpl.html>.
- [75] Bernard L Welch. The generalization of "students'" problem when several different population variances are involved. *Biometrika*, 1947.
- [76] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 2009.
- [77] John R. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*. Springer, 2002.

- [78] Bimal Viswanath, Mainack Mondal, Allen Clement, Peter Druschel, Krishna P. Gummadi, Alan Mislove, and Ansley Post. Exploring the design space of social network-based sybil defenses. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, 2012.
- [79] Lovro Šubelj and Marko Bajec. Group detection in complex networks: An algorithm and comparison of the state of the art. *Physica A: Statistical Mechanics and its Applications*, 2014.
- [80] Eric W. Weisstein. Statistical median. From MathWorld—A Wolfram Web Resource, 2013. URL <http://mathworld.wolfram.com/StatisticalMedian.html>. Sneto 21. 2. 2013.