

**Paavo Karppinen**

# **Matchmaking-järjestelmät videopeleissä**

Tietotekniikan kandidaatintutkielma

14. toukokuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Paavo Karppinen

**Yhteystiedot:** paavo.a.karppinen@student.jyu.fi

**Ohjaaja:** Antti-Jussi Lakanen

**Työn nimi:** Matchmaking-järjestelmät videopeleissä

**Title in English:** Matchmaking-systems in video games

**Työ:** Kandidaatintutkielma

**Opintosuunta:** Tietotekniikka

**Sivumäärä:** 21+0

**Tiivistelmä:** Ottelumuodostuksella (matchmaking) tarkoitetaan tapaa valita pelaajat pelamaan ottelua. Taitoihin perustuva matchmaking ei ole enää nykyaikana ainoa mahdollinen järjestelmä vaan parempia vaihtoehtoja on saatavilla, kuten edistyneet järjestelmät, tai pelaajien säilyttämiseen perustuva matchmaking. Tässä tutkielmassa käydään läpi mitä tarkoittaa matchmaking videopeleissä, mitä matchmaking-järjestelmiä on olemassa, kuinka ne toimivat ja mitä ne pyrkivät saavuttamaan kyseisellä toteutuksella. Aihe on valittu sen ajankohtaisuuden takia, sillä erityisesti taitoihin perustuvaa matchmaking-tapaa on kritisoitu, ja täten pyritään löytämään mahdollisesti korvaavia järjestelmiä. Tutkimusmenetelmä on kirjallisuuskatsaus.

**Avainsanat:** matchmaking, lobby, videopelit, algoritmi

**Abstract:** Skill-based matchmaking is no longer the only nor best matchmaking system for modern games, and there are better options available to manage a game's matchmaking, such as advanced ones, or systems based on player retention. In this document are gone through what matchmaking in video games mean, what kind of matchmaking systems there are, how they work, and what they try to reach with such implementation. Matchmaking is an event in which the players are chosen to a single match. The topic is chosen for its timeliness as especially skill-based matchmaking (SBMM) has been talked a lot recently. Chosen research method is literature review.

**Keywords:** matchmaking, lobby, video games, algorithm

## Sisällys

1	JOHDANTO .....	1
2	MATCHMAKING VIDEOPELEISSÄ .....	3
2.1	Matchmaking yleisesti .....	3
2.2	Taitoihin perustuva matchmaking .....	5
2.3	Sitoutumiseen perustuva matchmaking .....	8
2.4	MOBA-peleihin suunnitellut järjestelmät .....	10
2.5	Edistynyt matchmaking .....	11
3	YHTEENVETO.....	15
	LÄHTEET .....	16

# 1 Johdanto

Eri videopelit ovat aikojen myötä omaksuneet erilaisia tapoja muodostaa otteluita online videopeleissä saavuttaakseen erilaisia tavoitteita. Matchmaking videopeleissä on hyvin vanha aihe, ja erilaisia järjestelmiä on kehitetty ajan myötä. Eri järjestelmät keskittyvät tiettyihin puoliin ja jotkin ovat kehittyneitä variaatioita aikaisemmista järjestelmistä. Järjestelmien tavoitteet ovat moninaisia ja voivat olla mm. tasapainoisten otteluiden muodostaminen, kuten taitoihin perustuvassa on tapana (Herbrich, Minka ja Graepel 2007), tai muodostaakseen otteluita, jotka maksimoivat sitoutumisen, useimmiten tarkoittaen pelaajan säilyttämistä, kuten sitoutumiseen perustuvassa matchmaking-järjestelmissä (Chen ym. 2017). Tutkielmassa käydään läpi eri matchmaking-järjestelmiä ja tämän lisäksi onko järjestelmä toimiva testien perusteella. Matchmaking on ollut hyvin puhuttu aihe pelaajien keskuudessa internetissä viime vuosien aikana, etenkin taitoihin perustuva matchmaking, joten tutkielman tarkoituksena on esitellä eräs taitoihin perustuva matchmaking järjestelmä, ja taitoihin perustuvan matchmaking-järjestelmän mahdollisia korvaavia järjestelmiä.

Luvun 2 ensimmäisessä kappaleessa kerrotaan mitä matchmaking tarkoittaa, ja mitä varten se on. Tässä luvussa käydään läpi myös lyhyesti mitä eri matchmaking-tapoja on olemassa. Luvun 2 toisessa kappaleessa käydään läpi ensin perinteinen, ja useimmiten käytetty taitoihin perustuva matchmaking. Luvun 2 kolmannessa kappaleessa käydään läpi Electronic Artsin EOMM, engagement-optimised matchmaking. Luvun 2 neljännessä kappaleessa käydään läpi taisteluareenamoninpeleihin (eng. Multiplayer online battle arena, MOBA) kehitelty, ja *League of Legends (LoL)*-pelin avulla testattu puhtaasti rooleihin perustuva matchmaking. Itse matchmaking-järjestelmän lisäksi luvussa käydään läpi miksi tämä on kehitelty, sekä testien tulokset. Lopuksi, Luvun 2 viidennessä kappaleessa käydään läpi edistynyt matchmaking-tapa, joka on kehitetty taitoihin perustuvan pohjalta parantaakseen otteluiden laatua ja pelaajien hauskuuden saantia hyödyntäen koneoppimista ja enemmän parametrejä kuin pelkkä taitoluokitus joka on myös paljon monimuotoisempi kuin esim. TrueSkill-järjestelmässä.

Jokaisessa kappaleessa käydään läpi ensin kyseinen matchmaking-järjestelmä läpi yleisellä tasolla, jonka jälkeen käydään läpi tarkemmin kuinka se toimii. Tutkielman lopussa on

yhteenveto tutkielmassa käydyistä asioista.

## 2 Matchmaking videopeleissä

### 2.1 Matchmaking yleisesti

Matchmaking on tapa, jolla videopelien pelaajat ”liitetään” toisiinsa, eli kuinka ottelut muodostetaan. Yleisesti ottaen matchmaking videopeleissä pyrkii muodostamaan otteluita, jotka ovat rehellisiä, ja joissa pelaajilla on hauskaa. Matchmaking on tarpeellista, jotta pelaajat saadaan otteluihin, sillä matchmaking etsii pelaajia, jotka voidaan liittää otteluun. Ilman matchmakingia otteluita ei voitaisi muodostaa. Yleisimpänä tapana on edelleen perinteinen taitoihin perustuva matchmaking, mutta tästä on kehitelty edistyneempiä järjestelmiä, joissa taitoluokitus, eli numeerinen arvo pelaajan taidoille laskettu ottelun tuloksen ja pelaajan suorituksen avulla matemaattisella kaavalla, on paljon monimuotoisempi. On kuitenkin olemassa puhtaasti satunnainen matchmaking, jonka avulla otteluun otetaan satunnaisesti pelaajia, jotka ovat ilmoittaneet haluavansa otteluun. Tämä ilmoitus pelaajalta, että hän haluaa otteluun on useimmiten pelin aulassa (engl. lobby) ”Play”-napin painaminen. Taitoihin perustuvan matchmakingin lisäksi on olemassa puhtaasti sitoutumiseen perustuva järjestelmä. Näiden lisäksi sankaripeleihin (engl. hero games) on olemassa tapa, jossa pelaajan valitsema hahmo tai rooli otetaan huomioon. Joissain peleissä matchmaking tarkoittaa vain yhden pelaajan etsimistä, eli 1v1 pelimuodot, mutta se ei ole ainoa tilanne, jossa matchmakingia tarvitaan. On olemassa pelejä, jotka tarvitsevat 60, 100 tai jopa 150 pelaajan etsimistä muodostaakseen ottelun. Jokaisella pelintekijällä on oma matchmaking-järjestelmä saavuttaakseen heidän omat tavoitteensa valitsemallaan tavalla, oli se sitten taitoihin perustuva tai jokin muu, sekä tästä oleva heidän oma toteutuksensa.

Eräs yleisimmistä matchmaking-järjestelmistä on taitoihin perustuva, (engl. skill-based matchmaking, SBMM), mutta tämä ei ole suinkaan ainoa järjestelmä. Tästäkin on monella pelillä omanlaisensa toteutus. Taitoihin perustuvilla matchmaking-järjestelmillä pelintekijät pyrkivät luomaan rehellisiä otteluita, kun kaikilla ottelun pelaajilla on samankaltainen todennäköisyys voittaa (Graepel ja Herbrich 2006). Ehkäpä tunnetuin taitoihin perustuva järjestelmä on TrueSkill, joka kehitettiin Microsoftin Xbox Live-palvelua varten, jotta moninpeleihin on tapa, jolla monen pelaajan vaatimat ottelut voidaan muodostaa. Taitoihin perus-

tuivissa matchmaking-järjestelmissä yleisiä parametrejä räiskintäpeleissä, (engl. FPS, first-person shooter), muodostaakseen jokaisen pelaajan taitoluokituksen (engl. skill-rating), ovat tappo/kuolema-aste, (engl. kill/death rate/ratio), voittoprosentti, kohtaamisten voittoprosentti.

Taitoihin perustuvasta järjestelmästä on kehitetty laajempia ja enemmän parametrejä huomioon ottamia järjestelmiä, eli matchmakingia tehdessä ei ole otettu huomioon vain pelkkä taitoluokitus, vaan myös muita pelaajan ominaisuuksia, joita hän on näyttänyt aikaisemmin. Edistyneissä tavoissa on mukana taitoluokitus, mutta myös esim. pelaajan pelityyli, tiimipeleissä mikrofonin käyttäminen/yleisesti kommunikointi, jotta oman joukkueen kanssa kommunikoiivat, joko mikrofonin kanssa tai muilla pelin tarjoamilla tavoilla, pelaajat laitetaan samaan joukkueeseen. ”Sankaripeleissä” (engl. hero games), joissa on erilaisia hamoja joilla voi pelata, hahmovalinnan huomioon ottaminen ja tähän liittyen kuinka pelaajan pelityyli tällaisella hahmolla. Mikrofonin hyödyntäminen tai muulla tavalla kommunikointi pelissä on hyvä ottaa huomioon, sillä toistensa kanssa kommunikoiivat, erityisesti battle royale-videopeleissä, voivat tuntea ottelut paremmiksi ja hauskemmiksi, ja heillä on mahdollista voittaa ottelu todennäköisemmin kuin heillä, jotka eivät kommunikoi tiimensä kanssa. Näillä tavoilla pyritään muodostamaan rehellisiä/hauskoja otteluita paljon paremmin kuin perinteisellä taitoihin perustuvalla järjestelmällä. Delalleau ym. (2012) hyödyntävät järjestelmässään pelaajaprofilia, joka koostuu monimuotoisesta taitoluokituksesta, joka sopii nykyajan monitahoisiin videopeleihin. Profilissa voisi teoriassa olla mukana kuinka pelaaja kommunikoi. Münnich (2015) puolestaan väittää, että myös pelaajat ovat monimuotoisempia pelien lisäksi kuin ennen.

Eräs, ehkäpä kiistanalaisin järjestelmä, on Electronic Artsin sitoutumiseen perustuva matchmaking, (engl. engagement optimised matchmaking, EOMM). Tällä tavalla otteluiden muodostaminen ei perustu kokonaan taitoluokitukseen, vaan ottelut muodostetaan maksimoimaan pelaajan sitoutumisen, eli jotta pelaaja pelaa enemmän, käyttää enemmän rahaa peliin, jne. Tällä tavalla ottelun pelaajat eivät ole välttämättä saman tasoisia, vaan niissä voi olla parempia ja heikompia pelaajia, jotta paremmat pelaajat pelaisivat enemmän.

Puhtaasti viiveeseen perustuvissa matchmaking-järjestelmissä ottelut muodostetaan pelkäämään pelaajan viiveen peliin mukaisesti. Tämä ei tarkoita suoraan vain fyysisen sijainnin mu-



kaista matchmakingiä, sillä internetyhteys on suuressa roolissa pelaajan viiveessä. Näissä tavoissa ottelut muodostetaan siten, että jokaisella pelaajalla on samanlainen viive muihin verrattuna. Viive on myös mukana monessa muussaa matchmaking-järjestelmässä parametriminä. Puhtaasti tällaiset tavat ovat kuitenkin erittäin harvinaisia, vaan tämä on usein yhtenä osa-alueena, jonka matchmaking järjestelmä ottaa huomioon. Eräs puhtaasti viiveeseen perustuva on kuitenkin Wang ja Wu (2018) kehittämä ULM. Toinen tällainen järjestelmä on Agarwal ja Lorch (2009) kehittämä Htrae.

On olemassa myös puhtaan roolin mukaan matchmaking-järjestelmiä ”hero-peleissä” kuten *League of Legends*. Tällaisissa tavoissa matchmaking ottaa huomioon vain pelaajan valitseman hahmon tai roolin, ja pyrkii muodostamaan ottelun, jossa pelaajien valitsemat hahmot/roolit toimivat hyvin yhteen, ja ettei pelaajien ennalta valitsemisissa hahmoissa/rooleissa ole päällekkäisyyksiä, kuten Myślak ja Deja (2015) kehittämä järjestelmä *League of Legends*-videopelille. Tämä on myös mukana edistyneissä matchmaking-järjestelmissä yhtenä parametriminä.

Erilaiset järjestelmät toimivat toisia paremmin tässä, mutta jokainen peli on erilainen, ja täten jokaisen pelin käyttämä järjestelmä on erilainen. Tutkielmassa käydään läpi tarkemmin edellä mainitut järjestelmät läpi tarkemmin, pl. viiveeseen perustuva.

## 2.2 Taitoihin perustuva matchmaking

Taitoihin perustuvassa matchmakingissä peli muodostaa jokaiselle pelaajalle oman taitoluokituksen, joka voi muuttua jokaisen ottelun jälkeen. Taitoluokitus muodostetaan erilaisten seurattujen arvojen mukaisesti, ja jokaisella pelillä on omat tilastot, joilla luokitus muodostetaan. FPS-peleissä monesti luokitukseen otetaan mukaan voittoprosentti, kohtaamisten voittoprosentti, tappo/kuolema-aste (t/k-aste), otteluissa tehdyn vahingon määrä keskiarvona. Voittoprosentti on eräs arvo, joka on universaalisti mahdollinen arvo luokituksen muodostamiseen pelistä riippumatta. Taitoluokituksen muodostamiseen pelintekijät voivat myös antaa joillekin arvoille suuremman painotuksen, esim. voittoprosentti on määräävämpi kuin t/k-aste. Taitoihin perustuva matchmaking-järjestelmien perimmäisenä tavoitteena on muodostaa otteluita, jotka koostuvat saman tasoisista pelaajista, täten otteluiden tulisi olla tekni-

sestä näkökulmasta reiluja.

TrueSkill on bayesilainen malli taitoluokituksen laskemiseen. Tämä malli kehitettiin Microsoftin toimesta Xbox Live-järjestelmää hyödyntäville peleille (Herbrich, Minka ja Graepel 2007), ja jota he testasivat *Halo 2*-videopelin avulla (Makarov ym. 2018). TrueSkill omaa seuraavat ominaisuudet: tuki tiimipeleille; TrueSkill on kehitetty toimimaan peleille, jotka tarvitsevat enemmän kuin kaksi pelaajaa. TrueSkill pystyy muokkaamaan pelaajan taitoluokitusta pelattujen otteluiden määrästä huolimatta, se on yhteensopiva muiden taitoihin perustuvien järjestelmien kanssa. TrueSkill omaa jokaisen pelin haluamat tavoitteet, se ei tarvitse suurta määrää dataa kouluttaakseen sitä, ja lopuksi sen laskenta ei vie paljoa resursseja. *Gears of War*-pelin tekijä kuitenkin tarvitsee vielä yhden ominaisuuden: matchmaking-järjestelmän on kyettävä automaattisesti sopeutua peliin tehtyihin muutoksiin, kuten lisättyihin pelimuotoihin, TrueSkill ei pysty tähän (Minka, Cleven ja Zaykov 2018).

TrueSkill omaa viisi oletusta, jotka ovat muutettu TrueSkill 2-järjestelmässä. Nämä oletukset ovat seuraavanlaiset: (1) Pelaajan taitoluokitus edustaa hänen oletettua kontribuutiota joukkueelle, (2) joukkueen suoriutuminen on sen pelaajien suoritusten kautta laskettu summa. (3) Joukkue voittaa, kun sen suoriutuminen on tietyn marginaalin verran parempi kuin toisen joukkueen. (4) Pelaajan taito muuttuu ajan myötä ja sen kasvu ja lasku ovat yhtä todennäköiset, ja lopuksi (5) pelaajan taito yhdessä pelimuodossa on eri kuin toisessa (Minka, Cleven ja Zaykov 2018). *Halo 3* oli ensimmäinen peli joka hyödynsi TrueSkill-järjestelmää. Tämän lisäksi *Titanfall 2*-pelin kehittäjä on sanonut sen matchmaking-järjestelmän olevan lähellä TrueSkill-järjestelmää.

TrueSkill-järjestelmän datan on oletettavan koostuvan otteluiden tulosten järjestyksestä, joissa jokaisessa on listaus jokaisesta pelaajasta, heidän joukkuesijoituksesta, jokaisen ottelun alkuaika ja pituus, pelaajan peliaika, ja molempien joukkueiden lopputulos, esim. kuinka monta tappoa ne saivat. TrueSkill tulkitsee tätä dataa hyödyntämällä todennäköisyyden lisääntyvää mallia otteluiden tuloksista. Pelaajien taitotasojen luominen on ensimmäinen vaihe prosessissa. Jokaisella pelaajalla  $i$  on oletettavan omaavan laskettu taito, merkittynä  $taito_i^t$  ajanmääreessä  $t$ . Minka, Cleven ja Zaykov (2018) käyttävät normaalijakaumaa laskeakseen alkuperäinen pelaajan taitotaso, odotusarvolla  $m_0$  ja varianssilla  $v_0$ . Otantaprosessi merkitään

seuraavasti

$$taito_i^{t_0} N(m_0, v_0)$$

missä  $t_0$  on pelaajan ensimmäisen ottelun ajankohta ja  $(m_0, v_0)$  ovat parametrejä, joita on mahdollista muokata. Jokaisen ottelun jälkeen pelaajan taito muuttuu, joka lasketaan myös normaalijakaumaa hyödyntäen

$$taito_i^{t+L} N(taito_i^t, \gamma^2)$$

jossa  $L$  on ottelun pituus.

Seuraava vaihe prosessissa on ottelun tulosten muodostaminen pelaajien taidon mukaan. TrueSkill olettaa jokaisen pelaajan omaavan mitattavan suorituksen jokaisessa ottelussa, joka voidaan merkitä seuraavanlaisesti

$$suoritus_i^t N(taito_i^t, \beta^2)$$

jossa  $\beta$  on muokattava ottelun satunnaisuutta edustava arvo. Jokaisen joukkueen suoritus ottelussa saadaan pelaajien suoritusten painotetusta summasta, jossa painotetaan pelaajan pelaiaikaa tutkittavassa joukkueessa

$$suoritus_{joukkue} = \sum_{i \in \text{joukkue}} suoritus_i \frac{\text{pelattuAika}_i}{L}$$

tällä tavalla saadaan kahden joukkueen välinen tulos (Minka, Cleven ja Zaykov 2018).

TrueSkill 2 muokkaa edellisen TrueSkill-mallin oletuksia: Pelaajan taito saadaan sen yksittäisistä tilastoista kuten tapot ja kuolemat, pelaajan poistuessa pelistä ennen kuin se on lopunut käsitellään tappiona, pelaajan taito yhdessä pelimuodossa riippuu muissa pelimuodoissa olevista taidoista, satunnainen kävely pelaajan taidon kehittämisessä on puolueellistettu parantuvaan taitoon etenkin pelaamisen alkuvaiheilla, pelaajan suorituksen on oletettavan parempi kun hän on osa ennalta määrättyä ryhmää.

TrueSkill 2 -järjestelmän taustalla on ajatus, että sen on oltava mahdollisimman yksinkertainen sen taitoluokituksen muodostamiseen, sekä kuitenkin sen on pystyttävä saavuttamaan eri pelistudioiden vaatimukset. Tämän lisäksi jokaisen muutoksen järjestelmään on oltava perusteltavissa metriikan avulla, joka näyttää että toteutuksessa ja todellisuudessa on eroavaisuus. Metriikalla tarkoitetaan tässä kontekstissa funktiota, joka on keskiarvoistettu testidatan osa-alueiden avulla. Datan osa-alueet ovat taas saatu hyödyntämällä ominaisuuksia.

TrueSkill 2-järjestelmää kehittämistä varten Minka, Cleven ja Zaykov (2018) käyttivät dataa *Halo 5*-pelistä saatujen ottelujen vuodesta 2015 lähtien. Data koostuu miljoonista pelaajista ja otteluista erilaisista pelimuodoista. Näistä he käyttivät 23 miljoonaa ottelua juuri julkaisun jälkeen kouluttamista varten ja 3 miljoonaa ottelua koulutusvaiheen loppupuolella olevaa ottelua käytettiin testaamiseen. Testausten tuloksena TrueSkill saavutti 52% ja TrueSkill 2 68% ennustuksen tarkkuuden, kun mitattiin ottelun voittajaa. Kun testataan pelaajan taitojen kehittymistä, TrueSkill ei pysty ennustamaan voiton todennäköisyyttä tarkasti, kun taas tehtyjen muutosten kautta TrueSkill 2 pystyy ennustamaan ne tarkemmin kun taidot kehittyvät.

### **2.3 Sitoutumiseen perustuva matchmaking**

Sitoutumiseen perustuva matchmaking on uudempi tapa toteuttaa pelin matchmaking. Tällaisessa tavassa otteluiden muodostaminen, jotta ne olisivat tasapainoisia, ei ole pääpainona, vaan tarkoituksena on muodostaa otteluita, jotta pelaajat jatkavat pelaamista pidempään, käyttävät enemmän rahaa pelin sisäisiin esineisiin, kuten ostamalla eri hahmojen ”skinejä”. Kuten perinteisessä taitoihin perustuvassa matchmaking-järjestelmässä, taito on eräs asia joka otetaan huomioon, mutta matchmakingiä ei suoriteta pelkästään sen perusteella.

EOMM:n kehittämisen taustalla on eräs hyvin yleinen syy: SBMM:n tavoin luodut reilut ottelut eivät ole optimaalisia otteluiden muodostamiseen, ja erityisesti sitoutumiseen (Chen ym. 2017). Tästä huolimatta samantasoisista pelaajista koostuvat ottelut on eräs erittäin epätavallinen tapaus EOMM:stä. EOMM:n tavoitteena on muodostaa otteluita, jotka ovat niimensä mukaisesti optimaalisia sitoutumiseen.

EOMM pyrkii liittämään pelaajat, jotta pelaajien sitoutuminen on maksimaalista. Tässä hyödynnetään laskemalla jokaisen pelaajan irrottautumisen (engl. disengagement) ”poistumisriski” (engl. churn risk), joka tarkoittaa tässä tapauksessa pelaamattomuutta jossakin tietyssä aikavälissä. Jokainen pelaaja, joka odottaa matchmaking-altaassa, muodostetaan graafiksi, jossa jokainen pelaaja on yksi solmu ja kahden pelaajan välinen raja on heidän yhteenlaskettu riski, jos heidät liitetään toisiinsa. Matchmaking suoritetaan ratkaisemalla MWPM-ongelma (minimum weight perfect matching) löytämällä pareja, joiden minimisumma ei ole päällekkäinen graafissa. EOMM koostuu kolmesta osa-alueesta: taitoluokitus, ennustusmalli

poistumisriskille ja liittämismalli graafina.

Chen ym. (2017) kuvaa luomansa järjestelmän erityisesti 1 vs 1 -pelimuotoja varten. Järjestelmässä hyödynnettävä poistumisriski on tietyn ajan aikainen irrottautumisen status, eikä se välttämättä tarkoita loppullista irrottautumista. Poistumisriski merkitään  $c_{i,j}$  kun matchmaking on suoritettu pelaajien  $i$  ja  $j$  jälkeen. Tämä on puolestaan funktio, joka sisältää molempien pelaajien statuksen, eli  $c_{i,j} = \Pr(p_i \text{ poistumiset} \mid s_i, s_j) = c(s_i, s_j)$ . Pelaajan status on profiili, joka sisältää pelaajan ominaisuudet kuten taitotaso, pelaamisen tiheys, suoritustaso, jne. Järjestelmässä hyödynnetään listaa pelaajamonikkoja,  $M = \{(p_i, p_j)\}$ , merkitäkseen matchmaking-tulosta, toisin sanoen parien jakoa, jossa pelaajat  $P$  on liitetty toisiinsa vain kerran. Yksittäisten poistumisriskien mukaan muodostettu pelaajien irrottautumisen määrittelyä varten EOMM pyrkii optimaaliseen parien jakoon  $M^*$ , jotta

$$M^* = \operatorname{argmin}_M \sum_{p_i, p_j \in M} c(s_i, s_j) + c(s_j, s_i).$$

Solmut graafilla  $G$  ovat pelaajia  $p_i$ , joilla on tila  $s_i$  ennen kuin matchmaking suoritetaan. Kahden pelaajan  $p_i$  ja  $p_j$  välillä oleva raja liitetään painoon  $c_{i,j} + c_{j,i}$ , joka on puolestaan pelaajien yhteenlaskettu irrottautumisen mittari, jos heidät liitetään toisiinsa. Jokaisen graafin  $G$  pelaajat voidaan liittää toisiinsa, ja kun kaikki  $c_{i,j}$  ovat laskettu,  $M^*$  muodostetaan MWPM-ongelmaksi.

EOMM-järjestelmän toimivuutta Chen ym. (2017) testasivat EA:n eräässä pelissä, vaikka hyödynnettävää peliä ei manittu itse tutkimuksessa, EOMM-järjestelmää testattiin 1v1-pelimuodon avulla. Koska otteluiden mahdollisina tuloksina olivat voitto, tasapeli tai häviö, on turvallista olettaa että testattu peli oli *FIFA*. Sillä data otteluista kerättiin vuoden 2016 ensimmäisellä puoliskolla, on turvallista olettaa, että kyseessä oleva peli oli *FIFA 16*, sillä *FIFA 16* julkaistiin vuoden 2015 toisella puoliskolla. Jokaisen pelaajan taitotaso laskettiin Glicko-järjestelmän avulla. Poistumisen laskemista varten tutkimuksessa hyödynnettiin logistista regressiota, selvittääkseen onko pelaaja 8 tuntinen poistuja ottelun jälkeen.  $c(s_i, s_j) + c(s_j, s_i)$  arvoimiseen tutkimuksessa käytettiin pelaajan tilaa sekä muita syötteitä, jotka tässä tapauksessa olivat pelaajan 10 viimieisintä ottelua, tuleva ottelu, sekä pelattujen otteluiden määrä jaettuna eri aikaväleihin. Ennustetut mahdollisuudet ovat linjassa todellisten lukujen kanssa. Simulaatiossa hyödynnetyt pelaajien tilat ovat koostettu todellisista pelaajien tiloista

perustuen kerättyyn dataan.

Simulaatiossa EOMM verrattiin kolmeen eri matchmaking-tapaan, jotka olivat tässä tapauksessa satunnainen, taitoihin perustuva, ja huonoin mahdollinen matchmaking, joka on täysin EOMM:n vastainen. Metodeja käytettiin samaa pelaaja-allasta hyödyntäen. Pelaajien määränä käytettiin 100-500 pelaajaa. Tulosten perusteella, kun pelaajien määrä on enemmän kuin 100, säilytettyjen pelaajien määrä on parempi kun hyödynnetään EOMM-järjestelmää, ja sen tehokkuus kasvaa sitä mukaan mitä enemmän pelaajia on. 100:n pelaajan kohdalla paras järjestelmä pelaajien säilyttämiseen on taitoihin perustuva. Huonoin mahdollinen matchmaking pärjää jokaisessa tilanteessa huonoiten, kuten odottaa saa. Mielenkiintoisin huomio on kuitenkin siinä, että satunnainen matchmaking pärjää hyvin, ja jopa on parempi joissain tapauksissa, kuin taitoihin perustuva, joka puolestaan puoltaa Chen ym. (2017) väitettä, että tasapainoiset ottelut eivät ole aina optimaalisia sitoutumisen kannalta.

## 2.4 MOBA-peleihin suunnitellut järjestelmät

Myślak ja Deja (2015) mukaan sen aikainen *League of Legends (LoL)* ”rankkaus” (engl. ”ranked”) pelimuotoon tarkoitettu matchmaking järjestelmä ei ole sopiva, kun verrataan empiiriseen dataan. Tämä heidän mukaan vähentää pelimuodon tehokkuutta, ja vaikuttaa negatiivisesti pelaajien kokemukseen. Tutkimuksessa kehitetty järjestelmä hyödyntää pelin ominaisuuksia. He myös mainitsevat, että vaikka tuloksia annetaan *LoL*-pelin mukaan, samaa ratkaisua voidaan hyödyntää muissa MOBA-peleissä, kuten *Dota 2*.

*League of Legends*-pelin ottelut koostuvat kahdesta viiden hengen joukkueesta, jossa jokainen pelaaja valitsee pelattavan hahmon. Ennen ottelua pelaajat voivat valita mieluisensa roolin, mutta tämä ei ole käytössä rankkaus-pelimuodossa (Myślak ja Deja 2015). Pelin yksi puoli on META (Most Effective Tactic Available), joka *LoL*-pelissä tarkoittaa, että yksi pelaaja on kartan yläosassa, yksi keskiosassa, yksi kartan ”viidakko”osissa, (engl. jungle), ja kaksi alhaalla. Heikommat pelaajat yhä useammin (melkein kolmasosa) eivät hyödynnä tätä taktiikka, kun taas paremmat pelaajat hyödyntävät sitä useammin (vain alle 10% ei hyödynnä). Tämän lisäksi joukkueet jotka käyttävät epästandardia metaa, omaavat suuremman riskin, että joukkueessa on AFK (Away From Keyboard)-pelaajan, 23% vs. 6% metaa hyö-

dyntävillä. Tämä attributoidaan matchmaking-järjestelmän syyksi.

Tutkimuksessa esitelty järjestelmä on pelkästään rooleihin perustuva. Tämän järjestelmän ytimenä on etsiä pelaajia, joiden valitsema rooli ei ole ollenkaan päällekkäinen. Tämän takia pelaajan taitotaso on oltava vektori taitojen arvio jokaiselle eri roolille, eikä skalaari. Matchmaking järjestelmä määrää pelaajan otteluun käyttämällä tietoa pelaajan ensisijaisen roolin arvioidun taitovektorin kautta. Tämän avulla roolista taistelemisen loppuisi ja tuloksena olisi pelistä saadun nautinnon kasvaminen. Eräs potentiaalinen ongelma tässä on, että jotkin roolit ovat suositumpia kuin toiset, mutta tutkimuksen mukaan on olemassa erillisiä ryhmiä pelaajista joilla on mieluisin rooli, ja nämä ryhmät eivät ole päällekkäisiä (Myślak ja Deja 2015). Tämän lisäksi pelaajan taidon kasvaessa, hän pelaa yhä useammin tietyssä roolissa.

Toimivuuden testaamiseksi tutkimuksessa hyödynnetään virallista LoL API:ta, vaikkakin he eivät voineet valita rooleja, ottelun voittaja voidaan ennustaa käyttämällä oletuksia. Myślak ja Deja (2015) kehittämän järjestelmän avulla ottelun voittaja voidaan ennustaa paljon paremmalla todennäköisyydellä kuin 50%, joka on LoL-pelin matchmaking-järjestelmän tavoite. Heinäkuusta elokuuhun 2014, Myślak ja Deja (2015) keräsivät yli 200 000 satunnaisesti valittua rankkaus-pelimuodon ottelua. Jokaisen pelaajan LoL-data kerättiin, kuten taso ja divisioona, jotka käyvät taidon arvioimiseen. Näistä otteluista poistettiin standardista poikkeavan strategian käyttämät ottelut, ja joissa oli ottelun poistuvia pelaajia. Tuloksena oli 2000 ottelua uniikeilla pelaajilla. Tarkkoja arvioita pelaajan taitotasosta ei ollut saatavilla, sillä koko otteluhistoriaa pelaajille ei ollut saatavilla. Tämän takia Myślak ja Deja (2015) tekivät oletuksen, että pelaaja saavuttaa parhaat tulokset, kun hän pelaa mieluisessa roolissaan. Näiden oletusten tuloksena tulee, että mitä useampi pelaaja sinisessä joukkueessa pelaa mieluisessan roolissa, sitä korkeampi voittoprosentti on. Kun sinisellä joukkueella on kaikki viisi pelaajaa mieluisessa roolissa ja violetilla joukkueella ei yhtään, sinisen joukkueen voittoprosentti on yli 65%.

## **2.5 Edistynyt matchmaking**

Edistyneiden matchmaking-järjestelmien pohjana käytetään monesti koneoppimista muodostaakseen otteluita, jotka ovat reilumpia ja hausempia kuin mitä perinteinen SBMM pys-

tyisi muodostamaan. Näissä järjestelmissä huomioon otetaan enemmän kuin pelkkä taitoluokitus, mutta se voi olla mukana yhtenä parametrinä. Taitoluokitus onkin hyvä olla mukana, jotta ottelut ovat hauskoja paremmille pelaajille, mutta kuitenkin tarpeeksi haastavia, jotta he eivät tylsistyisi. Muita parametrejä ovat pelaajan pelityyli, hero-peleissä pelaajan valitsema hahmo/rooli ja valittuun hahmoon mukautuva pelityyli, pelaajan kommunikointi ottelussa, joko mikrofonin avulla, pelin sisäisellä chat-palvelulla tai esim. pelissä olevan ping-järjestelmän hyödyntäminen. Edistynyt matchmaking-järjestelmä koneoppimista hyödyntämällä on kehitetty sen takia, sillä taitoluokitus on monesti hyvin yksiulotteinen kuvaamaan pelaajan taitoa. Nykyajan pelit ovat kuitenkin paljon monimutkaisempia, ja ne vaativat useita erilaisia taitoja kuten refleksit, tiimitaito, pelisilmä (Delalleau ym. 2012). Delalleau ym. (2012) väittävät, että taitoihin perustuvan matchmaking-järjestelmien luomat ottelut, jotka koostuvat samantasoisista pelaajista, täten ottelun olevan tasapainoinen, ei ole hauska, kuten tasapainoton ottelu, sillä sen tuloksena voi olla kaksi joukkuetta, jotka eivät halua liikkua pois aloituspaikastaan. Tämä ei kuitenkaan välttämättä pidä paikkaansa, erityisesti esim. pelissä *Apex Legends* sillä jokaisella hahmolla on omat taidot (engl. abilities), joista monet ovat vahvimmillaan kun pelaajat pelaava aggressiivisesti etsien vihollisia. Toisaalta samassa pelissä on hahmoja jotka loistavat ”camperinä”, ja jos ottelut koostuvat vain korkeatasoisista pelaajista, nämä hahmot ovat kaikista vahvimpia.

Delalleau ym. (2012) kehittämä edistynyt matchmaking-algoritmi perustuu koneoppimiseen, tarkemmin ottaen neuroverkkoihin, jonka tarkoituksena on muodostaa otteluita, jotka ovat hauskoja. Näiden algoritmien avulla he pyrkivät ennustamaan ottelun voittajan ja laskemaan pelaajan nautinnon. Nämä ennustukset pyritään muodostamaan mahdollisimman tarkasti neuroverkkojen avulla käyttäen hyödyksi informaatiota ottelun sisältämisestä pelaajista, sekä erinäisistä asetuksista, jotka ovat erityisiä joka ottelua varten. Yksittäisen pelaajan informaatio kerätään historiallisesta datasta, jossa otetaan huomioon sekä edellisten otteluiden tulokset, että pelaajan käytös ajan myötä.

Delalleau ym. (2012) hyödyntämä neuroverkko koulutetaan siten, että tasapainoinen ottelu on saavutettu silloin, kun joukkueen A todennäköisyys voittaa joukkuetta B vastaan on 50%. Neuroverkon syötteenä on pelaajien profiilit, sekä kaksi vektoria, upotettu ja attribuutit. Upotusvektori  $e_i$ , joka on  $n_e$  ulottuvuuskellinen, opetellaan automaattisesti harjoitusvaiheessa.



Attribuutitvektori  $a_i$ , joka on  $n_a$  ulottuvuuskellinen koostuu pelistä saaduista tilastoista, jotka on sitten normalisoitu. Nämä tilastot kooostuvat esim. pelaajan tappo/kuolema-asteesta, otteluiden määrästä, jne. Pelaajan ominaisuudet-vektori  $P$  saadaan yhdistämällä pelaajan tiedot lineaarisesti seuraavanlaisesti:

$$P_i = e_i + Wa_i$$

$W$  on tässä tapauksessa  $(n_e \times n_a)$  matriisi. Näitä ominaisuuksia voidaan kutsua profiilin koosteeksi, jossa on pelaajan taidot eri osa-alueilla, tämän laajuus riippuu tietenkin siitä, kuinka monimuotoinen pelissä olevat taidot voivat olla.

Jokaiselle joukkueelle lasketaan joukkueen ominaisuudet  $t_j$  jotka koostuvat pelaajien ominaisuuksista.

$$t_j = \sum_{i \in \text{team } j} P_i$$

Joukkueen ominaisuudet sitten verrataan ja summautetaan ei-lineaarisella transformaatiolla piilotettuun kerrokseen  $h$ , joka määritellään seuraavasti

$$h = \tanh(b + \sum_{j \in \{A, B\}} V_j t_j)$$

jossa  $b$  on  $n_h$  ulottuvuuskellinen vektori ja  $V_A$  ja  $V_B$  ovat  $(n_h \times n_e)$  matriiseja. Yksi sigmoid yksikkö on laskennan viimeisessä vaiheessa, jotta saadaan todennäköisyys  $\alpha$ , että joukkue  $A$  on voittaja

$$\alpha = \sigma(u * h + c)$$

tässä  $u$  on  $n_h$  ulottuvuuskellinen vektori,  $c$  on skalaari ja  $\sigma$  on sigmoid funktio  $\sigma(x) = 1/(1 + e^{-x})$ . Delalleau ym. (2012) mainitsevat, että edellä esitelty tapa on yksinkertainen, joka sopii heidän tarkoitukseen, mutta monimutkaisemmilla tavoilla voi saavuttaa parempia tuloksia. He esittävätkin mahdollisia tapoja, joilla tätä voi parantaa.

Ottelun hauskuuden mittaamiseen tutkimuksessa käytetään tavoitevektoria  $f$ , jolloin  $f_i$  on 1 jos pelaaja  $i$ :llä oli hauskaa, ja 0 jos ei. Numeroiden 1 ja 0 käyttäminen on hyvin yleistä ohjelmoinnissa, joten näiden lukujen merkitys on erittäin helppo ymmärtää. Saatu arvo voidaan saada selville eri tavoilla, kuten pelin sisäisellä kyselyllä, joita on käytetty muissakin peleissäkin kuin *Ghost Recon*. Eräs toinen mahdollinen tapa on hyödyntämällä aikaisempaa tietoa, mikä tekee ottelusta hauskan, FPS-peleissä tämä voi olla korkea tappomäärä ja/tai korkea

vahingon (engl. damage) määrä, näistä vastakohtat tekevät tietenkin ottelusta mahdollisesti ei-hauskan. Toisin kuin ottelun voittajan ennustavassa neuroverkossa, josta tulee yksi globaali luku näyttääkseen todennäköisyyden joukkueen voittamiseen, hauskuuden mittaama neuroverkko on erilainen, sillä se ennustaa pelaajasta riippuvaa lopputulosta, ja jokaiselle pelaajalle tulee eri lopputulos. Eli  $P_1$  voi saada eri tuloksen kuin vaikka  $P_3$ , mutta se voi saada saman tuloksen, jonkun toisen pelaaja kanssa. Tässä arkkitehtuurissa piilotettu kerros ottaa syötteen mitattavan pelaajan ominaisuusvektorin, lopun joukkueen ominaisuusvektorin, sekä molempien kokonaisten joukkueiden ominaisuusvektorit. Käytettävät profiilit ja attribuutit ovat samat kuin ottelun tasapainoisuuden mittaamisessa, mutta kun mitataan tiettyä pelaajaa  $i$  joukkueessa  $A$ , piilotettu kerros  $h$  lasketaan seuraavanlaisesti

$$h = \tanh(b + Yp_i + V_1t_A + V_2t_B + U \sum_{k \in \text{team } j, k \neq i} p_k).$$

Tuloksen todennäköisyys saadaan ennalta määrättyllä tavalla sigman avulla. Jokainen esitelty matriisi voidaan kouluttaa erikseen jokaiselle kartalle ja pelimuodolle (Delalleau ym. 2012).

Esitetyt neuroverkkoihin perustuvat algoritmit toimivat käytännössä seuraavanlaisesti: järjestelmä etsii pelaajia, jotka ovat yhteensopivia taitojen perusteella, tämän lisäksi pelaajilla tulisi olla yhtä hyvä yhteys palvelimeen, ja sen tulisi suosia pelaajia, jotka ovat odottaneet pidempään. Jokainen näistä tietenkin vaikuttaa siihen, että onko pelaajalla hauskaa, sillä ottelun odottaminen pitkään vaikuttaa negatiivisesti kokemukseen. Kun jokaisella pelaajalla on samankaltainen yhteys, minimoidaan lagin mahdollisuus, joka tietenkin aiheuttaa ärtymystä, joka puolestaan vaikuttaa negatiivisesti hauskuuteen. Pelaajien taitoluokitus, erityisesti kun se on monimuotoinen, vaikuttaa tietenkin ottelusta saatuun hauskuuteen, sillä liian hyvä vastustaja aiheuttavat turhautumista, kun pelaaja ei voi suoriutua hyvin, ja erittäin heikot viholliset voivat aiheuttaa tylsyyttä, kun ottelut ovat liian helppoja. Heikot pelaajat omassa joukkueessa voivat aiheuttaa turhautumista, kun pelaaja kokee, että hän tekee kaiken itse.

Tulosten mukaan tasapainoisuuteen kehitetty neuroverkko ”BalanceNet” suoriutuu paremmin TrueSkill-järjestelmää vastaan tasapainoisuutta mittaamisessa. Hauskuuteen kehitetty ”FunNet” suoriutuu paljon paremmin TrueSkill-järjestelmään verrattuna, ja BalanceNet ei ole paljoa huonompi.

### 3 Yhteenveto

Tutkielmassa esiteltiin käytännössä neljä eri tapaa muodostaa videopelin matchmaking. Tämän lisäksi TrueSkill 2 käytiin läpi ja kuinka se eroaa edeltäjästään. Kun mietitään eri järjestelmien toimivuutta, voidaan todeta, että jokainen niistä saavuttaa tavoitteensa vähintään kohtalaisesti. Kuten on jo aiemmin mainittu, perinteinen taitoihin perustuva matchmaking ei enää nykyajan monimutkaisissa peleissä ole kovinkaan pätevä, vaan edistynyt matchmaking hyödyntäen koneoppimista on paras mahdollinen tapa muodostaa ottelut, sillä se ottaa huomioon useampia osa-alueita kuin pelkkä taitoluokitus, ja sen käyttämä taitoluokitus on moniulotteisempi kuin perinteisen taitoihin perustuvan. Pelaajan roolin ja siihen liittyen taidon mukaan perustuva matchmaking on sekin parempi tapa kuin perinteinen taitoihin perustuva peleissä, joissa on eri hahmoja, jotka omaavat erilaisia kykyjä. Sitoutumiseen perustuva EOMM on sekin tulosten valossa parempi valinta kuin perinteinen taitoihin perustuva, kun mittarina on pelaajien säilyttäminen.

Kun ajatellaan tämän tutkielman heikkouksia, niin ensimmäisenä tietenkin tulee esille esitelyjen järjestelmien vähyys, mutta tästä huolimatta kolme niistä on järjestelmiä, jotka ovat yleisesti käytössä, etenkin taitoihin perustuva jota pidetään kaikista yleisimpänä järjestelmänä. Tutkittujen järjestelmien toimivuutta voitaisiin paremmin testata, jos ne toteutetaan olemassa olevissa peleissä ja niiden pelaajista kerätään dataa.

## Lähteet

Agarwal, Sharad, ja Jacob R. Lorch. 2009. "Matchmaking for Online Games and Other Latency-Sensitive P2P Systems". *SIGCOMM Comput. Commun. Rev.* (New York, NY, USA) 39, numero 4 (elokuu): 315–326. ISSN: 0146-4833. <https://doi.org/10.1145/1594977.1592605>. <https://doi.org/10.1145/1594977.1592605>.

Chen, Zhengxing, Su Xue, John Kolen, Navid Aghdaie, Kazi A Zaman, Yizhou Sun ja Magy Seif El-Nasr. 2017. "Eomm: An engagement optimized matchmaking framework". Teoksessa *Proceedings of the 26th International Conference on World Wide Web*, 1143–1150.

Delalleau, O., E. Contal, E. Thibodeau-Laufer, R. C. Ferrari, Y. Bengio ja F. Zhang. 2012. "Beyond Skill Rating: Advanced Matchmaking in Ghost Recon Online". *IEEE Transactions on Computational Intelligence and AI in Games* 4 (3): 167–177. <https://doi.org/10.1109/TCIAIG.2012.2188833>.

Graepel, Thore, ja Ralf Herbrich. 2006. "Ranking and matchmaking". *Game Developer Magazine* 25:34.

Herbrich, Ralf, Tom Minka ja Thore Graepel. 2007. "TrueSkill(TM): A Bayesian Skill Rating System". Teoksessa *Advances in Neural Information Processing Systems 20*, 569–576. MIT Press, tammikuu. <https://www.microsoft.com/en-us/research/publication/trueskilltm-a-bayesian-skill-rating-system/>.

Makarov, Ilya, Dmitry Savostyanov, Boris Litvyakov ja Dmitry I. Ignatov. 2018. "Predicting Winning Team and Probabilistic Ratings in "Dota 2" and "Counter-Strike: Global Offensive" Video Games". Teoksessa *Analysis of Images, Social Networks and Texts*, toimittanut Wil M.P. van der Aalst, Dmitry I. Ignatov, Michael Khachay, Sergei O. Kuznetsov, Victor Lempitsky, Irina A. Lomazova, Natalia Loukachevitch ym., 183–196. Cham: Springer International Publishing. ISBN: 978-3-319-73013-4.

Minka, Tom, Ryan Cleven ja Yordan Zaykov. 2018. *TrueSkill 2: An improved Bayesian skill rating system*. Tekninen raportti MSR-TR-2018-8. Microsoft, maaliskuu. <https://www.microsoft.com/en-us/research/publication/trueskill-2-improved-bayesian-skill-rating-system/>.

Münnich, Sven. 2015. "Advanced Matchmaking for Online First Person Shooter Games using Machine Learning". Tutkielma, KTH, School of Information ja Communication Technology (ICT).

Myślak, Mateusz, ja Dominik Deja. 2015. "Developing Game-Structure Sensitive Matchmaking System for Massive-Multiplayer Online Games". Teoksessa *Social Informatics*, toimittanut Luca Maria Aiello ja Daniel McFarland, 200–208. Cham: Springer International Publishing. ISBN: 978-3-319-15168-7.

Wang, Ning, ja Jie Wu. 2018. "Latency minimization through optimal user matchmaking in multi-party online applications". Teoksessa *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 1–10. IEEE.