

# **Satunnaiskävelyt ja niiden kasvattamat puut**

**Joonas Laukka**

**Perustieteiden korkeakoulu**

Kandidaatintyö  
Espoo 12.5.2018

**Vastuuopettaja**

TkT Riikka Korte

**Työn ohjaaja**

Prof. Lasse Leskelä



Copyright © 2018 Joonas Laukka



---

**Tekijä** Joonas Laukka

---

**Työn nimi** Satunnaiskävelyt ja niiden kasvattamat puut

---

**Koulutusohjelma** Teknillinen fysiikka ja matematiikka

---

**Pääaine** Matematiikka ja systeemitieteet

**Pääaineen koodi** SCI3029

---

**Vastuuopettaja** TkT Riikka Korte

---

**Työn ohjaaja** Prof. Lasse Leskelä

---

**Päivämäärä** 12.5.2018

**Sivumäärä** 25+4

**Kieli** Suomi

---

### Tiivistelmä

Nykymailma on täynnä erilaisia verkostoja Internetistä sosiaalisiin piireihin. Verkkojen kasvun ymmärtäminen ja mallintaminen onkin kiinnittänyt lukuisen tutkijoiden huomion kuluneiden vuosikymmenien aikana. *No Restart Random Walk* (NRRW) on uudenlainen malli satunnaisverkon generoimiseen. Mallissa satunnaiskävelijä kasvattaa suuntaamatonta verkkoa kulkissaan siinä. Kävelijä lähtee alkusolmesta, kulkee  $s$ :n kaaren läpi satunnaiseen solmuun, ja sen naapuriksi lisätään uusi solmu. Uuden solmun asteluku on aina yksi. Tätä prosessia jatkettaan satunnaiskävelijän viimeisimmästä tilasta, ja prosessin toistaminen kasvattaa satunnaisen puurakenteen. Mallin ominaisuudet määritetään siis askelparametri,  $s$ . NRRW-malli mukailee reaalimaailman verkoille tyypillistä suosivaa kiinnitymistä. Sen rakentaman verkon ominaisuudet riippuvat kuitenkin voimakkaasti askelparametrista. Malli ei generoi skaalautumattomia verkkuja, mutta se demonstroi kiinnostavan yhteyden satunnaisverkon astelujen jakauman ja sitä kasvattavan satunnaiskulun palautuvuuden välillä. Tämän yhteyden ymmärtäminen on askel kohti entistä realistisempia verkkojen kasvumalleja. Tässä kandidaatintyössä keskityn tarkastelemaan parillisen askelparametrin NRRW-malleja. Muodostan NRRW-prosessille erään stokastisen esityksen, todistan sen oikeaksi ja hyödynnän sitä parillisen askelparametrin NRRW-mallin ominaisuuksien tutkimiseen.

---

**Avainsanat** Satunnaiskävely, puu, verkkokaavio, NRRW

---

**Author** Joonas Laukka

---

**Title** Random Walks and the Trees they Grow

---

**Degree programme** Engineering Physics and Mathematics

---

**Major** Mathematics and System Analysis**Code of major** SCI3029

---

**Supervisor** DSc (Tech.) Riikka Korte

---

**Advisor** Prof. Lasse Leskelä

---

**Date** 12.5.2018

---

**Number of pages** 25+4

---

**Language** Finnish

---

**Abstract**

The modern world is full of networks from the Internet to social circles. Modeling the growth of graphs has caught the attention of many researchers during the last decades. One such growth model is a *No Restart Random Walk* (NRRW). In an NRRW, a random walker grows an undirected graph while it is traversing it. The walker starts from the root vertex and crosses  $s$  edges to end up in a random destination vertex. Then, a new vertex with degree 1 is added as its neighbour. This process is repeated starting from the destination vertex and consequently, a random tree is generated. The properties of the model are unambiguously determined by the step parameters,  $s$ . The NRRW model conforms to many properties that are common in real life graphs such as preferential attachment. The random walk process and the random graph process are mutually dependent. Examining these dependent processes can be simplified by proving that they are equivalent with a simpler stochastic process. I will show that the NRRW model is stochastically equivalent to a model seeded by a sequence of independent, uniformly distributed random numbers. Finally, I will use this alternative model to prove a theorem about the recurrence of the vertices in an NRRW generated graph.

---

**Keywords** Random walk, graph, tree, NRRW

## Esipuhe

Kiitän erityisesti tämän kandidaatintyön ohjaajaa, Professori Leskelää, kärsivällisyystä ja kannustavasta asenteesta. Hänen rohkaiseva ohjauksensa edesauttoi paitsi tämän tutkielman valmistumista niin myös omien kiinnostuksien kehitystä satunnaisprosessien suuntaan.

Otaniemi, 12.05.2018

Joonas Laukka

# Sisällysluettelo

|  |           |
|--|-----------|
| <b>Tiivistelmä</b>                       | <b>3</b>  |
| <b>Tiivistelmä (englanniksi)</b>         | <b>4</b>  |
| <b>Esipuhe</b>                           | <b>5</b>  |
| <b>Sisällysluettelo</b>                  | <b>6</b>  |
| <b>Symbolit ja lyhenteet</b>             | <b>7</b>  |
| <b>1 Johdanto</b>                        | <b>8</b>  |
| <b>2 Malli</b>                           | <b>10</b> |
| 2.1 Yleiskatsaus . . . . .               | 10        |
| 2.2 Kaksi prosessia . . . . .            | 11        |
| 2.2.1 Satunnaisverkko . . . . .          | 11        |
| 2.2.2 Satunnaiskulkku . . . . .          | 12        |
| 2.3 Kokonaisuus . . . . .                | 12        |
| 2.3.1 Tilajoukko . . . . .               | 13        |
| 2.3.2 Siirtymätodennäköisyydet . . . . . | 13        |
| <b>3 Stokastinen esitys</b>              | <b>15</b> |
| 3.1 Yleistä teoriaa . . . . .            | 15        |
| 3.2 Prosessin esitys . . . . .           | 15        |
| <b>4 Parillinen prosessi</b>             | <b>17</b> |
| 4.1 Astelukujen jakauma . . . . .        | 19        |
| 4.2 Palautuvuus . . . . .                | 20        |
| <b>5 Yhteenvetö</b>                      | <b>24</b> |
| <b>Viitteet</b>                          | <b>25</b> |
| <b>A Simulaatioiden lähdekoodi</b>       | <b>26</b> |

# Symbolit ja lyhenteet

## Symbolit

$\mathbb{N}_0$  ei-negatiiviset kokonaisluvut  $(0, 1, 2, \dots)$

## Operaattorit

$\mathbf{1}\{X\}$  indikaattorifunktio ehdolla  $X$   
 $\mathbb{P}[X]$  tapahtuman  $X$  todennäköisyys

## Lyhenteet

NRRW No Restart Random Walk  
i.i.d. riippumaton ja identtisesti jakautunut

# 1 Johdanto

Nykymaailma on täynnä erilaisia verkostoja Internetistä sosiaalisiin piireihin ja niiden syvällinen ymmärtäminen on tärkeämpää kuin koskaan. Verkkoteoria on matematiikan ala, joka pyrkii ymmärtämään solmuista ja niitää yhdistävistä kaarista rakentuvia abstrakteja verkkooja. Koska verkostoja tarkastellaan riittävän abstraktilla tasolla, verkkoteorian tulokset pätevät sovelluskohteesta riippumatta. Sama looginen rakenne, verkko, voi kuvata niin kaupunkien välisiä tieyhteyksiä kuin nettisivujen välisiä linkkejäkin, ja verkkoteorian tulokset ovat aina yhtä oleellisia. Vain verkon tulkinta vaihtuu.

Yksi tärkeä askel kohti kaikenlaisten reaalimaailman verkkojen ymmärtämistä on erilaisten kasvumallien luominen. Kun tiedämme, miten ja miksi verket kehittyvät sellaisiksi kuin ne ovat, pystymme simuloimaan niitä ja ennustamaan niiden kehitystä. Kuluneina vuosikymmeninä onkin kehitetty lukuisia erilaisia verkkojen kasvumalleja. Niiden generoimien verkkojen topologiassa on kiinnitetty huomioita moniin erilaisiin ominaisuuksiin, mutta yksi kiinnostavimmista piirteistä on solmujen astelukujen jakauma. Empiiriset kokeet ovat osoittaneet, että monet suuret ja kiinnostavat verket, kuten Internet, ovat skaalautumattomia (eng. scale-free) [1]. Skaalautumattoman verkon solmujen astelukujen jakauman määrittää jokin potenssilaki ( $\mathbb{P}[k] = k^\gamma$ ). Yksinkertaistaen tämä tarkoittaa sitä, että ääriarvot, eli sellaiset solmut, joilla on hyvin korkea astelu, ovat skaalautumattomissa verkoissa verrattain yleisiä. [2]

Monet tosielämän verket syntyvät näennäisesti hyvinkin toisistaan poikkeavista prosesseista, mutta niille on kuitenkin tyypillistä juuri skaalautumaton rakenne. Tämä viittaa siihen, että kyseisten verkkojen kasvu pohjautuu samankaltaisiin stokastiisiin mekanismiin. Näyttääkin siltä, että skaalautumattomia verkkooja syntyy erityisesti sellaisista prosesseista, jotka noudattavat kasvaessaan suosivaa kiintymistä (eng. preferential attachment) [3]. Suosivalla kiintymisellä tarkoitetaan satunnaisverkkojen yhteydessä sitä, että uusia solmuja liitetään todennäköisimmin sellaisiin solmuihin, joilla on jo ennestään korkea astelu. Koska suosiva kiintyminen näyttäisi olevan tosielämässä esiintyvien skaalautumattomien verkkojen kasvun taustalla, on siihen perustuvien kasvuprosessien kehittäminen kiintittänyt tutkijoiden huomion kuluneen vuosikymmenen aikana.

Yksi suosivaa kiintymistä ilmentävä kasvumalli on *No Restart Random Walk* (NRRW), jossa satunnaiskävelijä kasvattaa suuntaamatonta verkkoa kulkissaan siinä. Nimensä mukaisesti NRRW-prosessin satunnaiskulku ei ala koskaan alusta, vaan jatkuu katkeamatta verkon kasvaessa. Jokaisen  $s$  askeleen välein verkkoon lisätään uusi solmu, ja se liitetään siihen solmuun, johon satunnaiskulku on juuri siirtynyt. Prosesсин simulointi ja matemaattinen analyysi osoittavat, että parametrin  $s$  parillisuus vaikuttaa voimakkaasti satunnaiskävelyn ja satunnaisverkon luontesiin (taulukko 1). NRRW-prosessin luonnetta voidaan tarkastella joko satunnaisen verkon tai satunnaiskävelyn näkökulmasta. Satunnaiskävelyn palautuvuus ja sen kasvattaman verkon topologia ovat kuitenkin kietoutuneet selvästi toisiinsa. [4]

Koska NRRW-prosessin kasvattamat verket ilmentävät sekä skaalautumatonta että skaalautuvaa sisäsolmujen rakennetta, ovat ne epäilemättä kiinnostavia perinteisen verkkoonanalyysin näkökulmasta. Osuvimmat sovelluskohteet löytynevät kuitenkin

| Askelparametri | Satunnaiskulku | Satunnaisverkko   |
|----------------|----------------|---|
| $s = 1$        | Väistytävä     | Sisäsolmujen astelukujen jakauma on geometrisen jakauman ylhäältä rajoittama. |
| $s = 2n$       | Palautuva      | Sisäsolmujen astelukujen jakauma on potenssilain alhaalta rajoittama.         |

Taulukko 1: NRRW-prosessin askelparametrin vaikutus taustalla vaikuttavan satunnaiskulun ja sen generoiman satunnaisverkon ominaisuuksiin.

sellaisista systeemeistä, joissa satunnaiskulun ja verkon kasvun yhteys on prosessin sisäänrakennettu, kiistaton ominaisuus. Yksi tällainen systeemi esiintyy uudenlaisen kryptovaluutan, IOTA:n, toteutuksessa. Toisin kuin perinteiset kryptovaluutat, IOTA ei perustu lohkoketjuteknologiaan. Lohkoketjun sijaan transaktiot tallennetaan suuntaamattomaan, syklittömään verkoon. Kun uusi transaktio saapuu, on sen vahvistettava kaksi aiempaa transaktiota liittymällä niihin transaktiopuussa. Koska transaktioiden vahvistusmekanismi ei perustu louhimiseen (eng. mining), ei transaktioihin liity myöskään transaktiokustannuksia. Tämä on huomattava etu verrattuna perinteisiin kryptovaluuttoihin, sillä se mahdollistaa muun muuassa mikromaksujen maksamisen. [5]

IOTA:n transaktiopuun stabiilius ja turvallisuus ovat kuitenkin monimutkaisia kysymyksiä. Puuta mallinnetaan suuntaamattomia ja syklittömiä verkkoa rakentavalla stokastisella prosessilla, Tanglella, jossa uusien solmujen liitoskohdat valitaan satunnaiskulun avulla [6]. NRRW-prosessi on pohjimmiltaan satunnaiskulku satunnaisessa verkossa, ja siten sen ymmärtäminen on osa matemaattista kivijalkaa, jonka päälle Tanglen kaltaiset tietotekniset sovellukset rakennetaan.

Tässä kandidaatintyössä perehdyn sellaisiin NRRW-prosessin ominaisuuksiin, joita on käsitelty Giulio Iacobellin, Daniel Figueiredon ja Giovanni Neglian paperissa "Transient and Slim versus Recurrent and Fat: Random Walks and the Trees they Grow". Keskeytyn erityisesti parillisen askelparametrin prosessiin, sillä sen generoimien verkkojen sisäsolmut muodostavat monien kasvumallien sovelluksien kannalta kiinnostavan skaalautumattoman verkon. Formuloin NRRW-prosessille stokastisen esityksen, todistan sen oikeaksi ja hyödynnän sitä parillisen askelparametrin ominaisuuksien tarkasteluun.

## 2 Malli

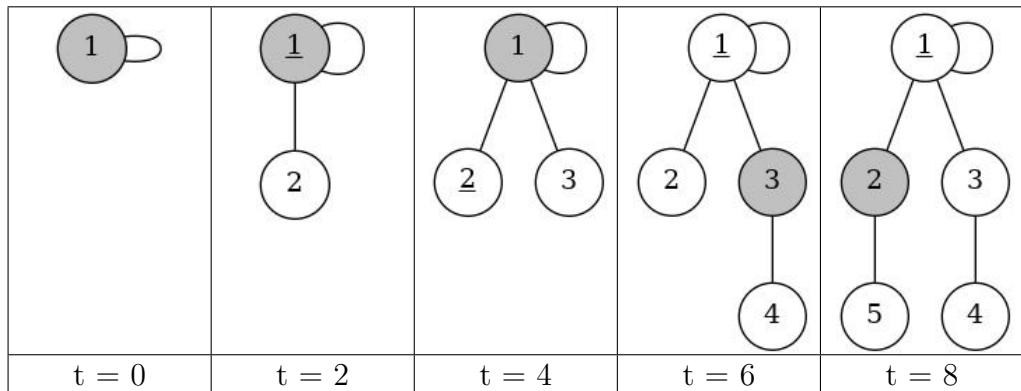
### 2.1 Yleiskatsaus

NRRW-prosessi askelparametrilla  $s$  käyttää seuraavanlaisen algoritmin mukaisesti:

0. Aluksi verkossa on yksi solmu, juuri, josta on kaari itseensä. Satunnaiskävelyn alkutila on juurisolmu.
1. Satunnaiskävely ottaa  $s$  askelta verkossa.
2. Verkkoon lisätään uusi solmu ja se liitetään satunnaiskävelyn senhetkiseen sijaintiin.
3. Palataan kohtaan 1.

Matemaattisesti prosessi on kahden toisistaan riippuvan stokastisen prosessin pari,  $(\mathcal{W}_t, \mathcal{G}_t^s)_{t \in \mathbb{N}_0}$ . Molemmat prosessit ovat kytkeytyneet vahvasti toisiinsa, mutta niitä voidaan jaksoittain tarkastella myös erillisinä. Taulukko 2 esittää yhden mahdollisen NRRW-prosessin realisaation askelparametrilla  $s = 2$  ja parillisilla ajanhetkillä. Askelparametri vaikuttaa huomattavasti siihen, millaisia verkkova NRRW-prosessi

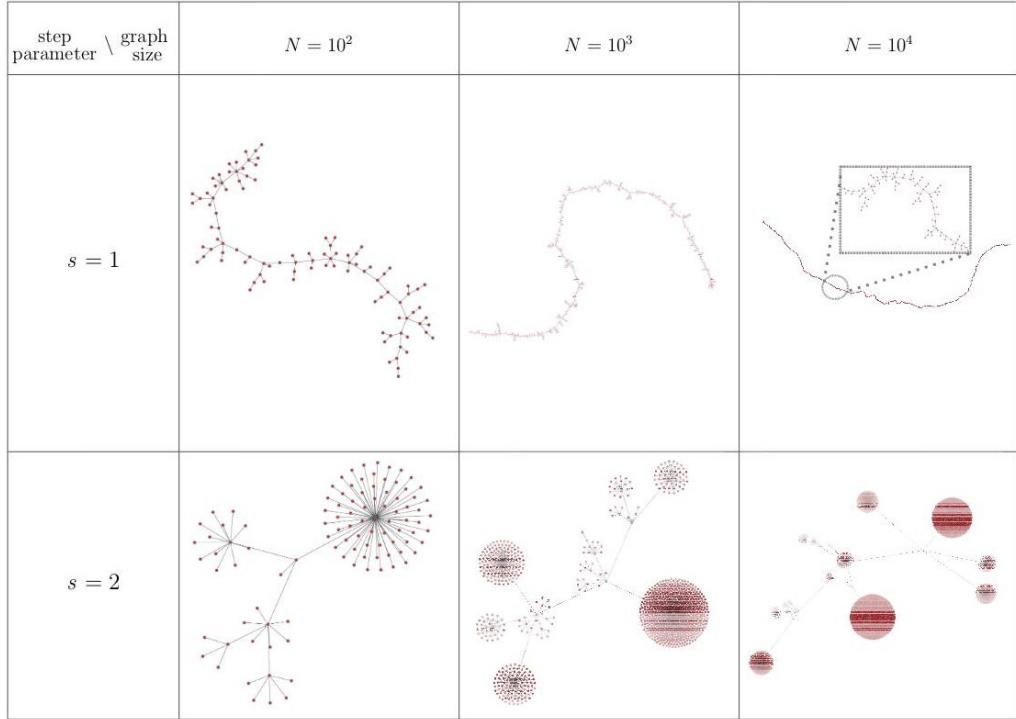
Taulukko 2: Yksi mahdollinen NRRW-prosessin realisaatio askelparametrilla  $s = 2$ . Satunnaiskävely on harmaan värin osoittamassa tilassa ja uusi solmu on juuri lisätty sen naapuriksi. Satunnaiskävely siirtyi nykyiseen tilaan alleviivatun solmun kautta.



kasvattaa (taulukko 1). Tämä ilmenee hyvin tietokonesimuloituista realisaatioista kuvassa 1. Parametrilla  $s = 1$  jokaiseen satunnaiskävelyn vierailemaan tilaan lisätään uusi solmu. Tämä johtaa korkeisiin puihin, joissa solmut jakutuvat tasaisesti parillisille ja parittomille tasoille. Satunnaiskävelyn kaikki tilat ovat väistyviä.

Sen sijaan parillinen askelparametri ( $s \in 2\mathbb{N}_0$ ) johtaa hyvin toisenlaisiin verkkoihin. Simulaatiotulokset viittaavat solmujen kasaantuvan suurissa määrin joidenkin yksittäisten solmujen jälkeläisiksi. Kyseessä on niin sanottu suosivan kiinnityksen ilmiö, jossa jotain mitattavaa suuretta kertyy sellaisille joukon alkioille, joilla on jo ennestään paljon kyseistä suuretta. Satunnaisverkossa tämä ilmenee siten, että uusia

solmuja liitetään todennäköisimmin sellaisiin verkon solmuihin, joilla on jo ennestään korkea asteluku. Suosiva kiinnityminen on yhteydessä skaalautumattomiin verkoihin [3].



Kuva 1: NRRW-prosessin realisaatioita tietokonesimulaatioissa. Kuva on kopioitu aiemmasta tutkimuksesta (Iacobelli et al.) [1].

## 2.2 Kaksi prosessia

NRRW-prosessin ominaisuudet ovat seurausta satunnaiskulun ja sen kasvattaman satunnaisen verkon välisestä vuorovaikutuksesta. Malli on monimutkainen ja yksinkertaisuuden vuoksi on ensin syytä tarkastella satunnaisverkkoa ja satunnaiskulkua erillisinä prosesseina. Tämä mahdollistaa erillisiin malleihin liittyvän terminologian ja yksityiskohtien esittelyn ennen NRRW-prosessin kokonaisuuteen paneutumista.

### 2.2.1 Satunnaisverkko

Olkoon  $(\mathcal{G}_t^s)_{t \in \mathbb{N}_0}$  satunnaisverkko, johon lisätään uusi solmu  $s$  aikayksikön välein. Jos verkon juurisolmussa esiintyvä silmukkaa ei huomioida, satunnaisprosessin tilajoukko on kaikkien suuntaamattomien, yhtenäisten ja syklittömiä verkkojen eli puiden joukko.

**Määritelmä 1.** *Suuntaamatona verkko  $G$  on pari  $(V, E)$ , missä joukko  $V$  on verkon solmujen joukko ( $v_i \in V$ ) ja  $E$  on suuntaamattomien kaarien joukko. Kaarien joukon alkiot ovat muotoa  $\{v_i, v_j\}$ . Kaari  $\{v_i, v_j\}$  yhdistää solmut  $v_i$  ja  $v_j$  toisiinsa.*

Kasvavan verkon käsittelyn helpottamiseksi satunnaisverkon  $(\mathcal{G}_t^s)_{t \in \mathbb{N}_0}$  solmut indeksoidaan kokonaisluvuilla  $(0, 1, 2, \dots)$  lisäsjärjestyksen mukaan. Juurisolmu on siis  $v_0$  ja hetkellä  $t = sn$  ( $n \in \mathbb{N}_0$ ) lisätään solmu  $v_n$ . Jos satunnaisverkkoa  $(\mathcal{G}_t^s)_{t \in \mathbb{N}_0}$  tarkastellaan itsenäisenä prosessina, sen siirtymätodennäköisyydet pois nykytilasta ovat nollasta poikkeavia vain ajanhetkillä  $s\mathbb{N}_0 = \{sn : n \in \mathbb{N}_0\}$ . Silloin ne määritetään satunnaiskulun  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  siirtymämatodennäköisyydet:

$$\begin{aligned} & \mathbb{P} [\mathcal{G}_{t+s}^s = (V_t \cup \{v_{j+1}\}, E_t \cup \{(v_i, v_{j+1})\}) \mid \mathcal{G}_t^s = (V_j, E_j), \mathcal{W}_t = v_j] \\ &= \mathbb{P} [\mathcal{W}_{t+s} = v_i \mid \mathcal{W}_t = v_n, G_t = (V_t, E_t)]. \end{aligned} \quad (1)$$

### 2.2.2 Satunnaiskulku

Vastaavasti satunnaiskulkuprosessin  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  tilajoukon ja siirtymätodennäköisyydet määritetään sen kasvattama satunnaisverkko. Sekä tilajoukko että siirtymätodennäköisyydet muuttuvat aina, kun verkkoon lisätään uusi solmu. Lisäyshetkien välillä tilajoukko on kuitenkin vakio ja satunnaiskävelyprosessi voidaan silloin nähdä tavallisena äärellisen tilajoukon Markov-ketjuna. Jos satunnaiskävely on solmussa  $v_i$  hetkellä  $t$ , määräytyy seuraava tila satunnaisesti valitsemalla yksi solmusta  $v_i$  lähtevä kaari ja siirtymällä sen osoittamaan solmuun. Todennäköisyysmassa on tasajakautunut kaikkien solmun  $v_i$  kaarien kesken.

Koska NRRW-prosessi rakentaa puita, ei verkossa ole muita syklejä kuin juuren kaari itseensä. Verkkoteoriassa kaaret solmesta itseensä lasketaan kahdesti, joten merkitään solmun  $v_i$  solmuun  $v_j$  yhdistävien kaarien määritetään verkossa  $G_i$

$$\psi_{G_i}(v_i, v_j) = 1 + \mathbb{1}\{v_i = v_0, v_j = v_0\}, \quad (2)$$

missä  $v_0$  on verkon  $G_i$  juurisolmu. Lauseke on perustuu siihen tosiasiin, että NRRW-prosessi on juurisolmun silmukkaa lukuunottamatta sykliton. NRRW-prosessin kasvattama verkko on sellainen. Nyt todennäköisyys siirtyy solmesta  $v_i$  solmuun  $v_j$  verkossa  $G_i$  on yksinkertaisesti

$$\mathbb{P} [\mathcal{W}_{t+1} = v_j \mid \mathcal{W}_t = v_i, \mathcal{G}_t^s = G_i] = \frac{\psi_{G_i}(v_i, v_j)}{d_{G_i}(v_i)}. \quad (3)$$

Yllä  $d_{G_i}(v_i)$  merkitsee solmun  $v_i$  astelukua verkossa  $G_i$ . Toisin sanoen

$$d_{G_i}(v_i) = \sum_{w \in V_i} \psi_{G_i}(v_i, w) \quad (4)$$

on kaikkien solmuun  $v_i$  yhdistyvien kaarien lukumäärä.

## 2.3 Kokonaisuus

Satunnaisverkon ja satunnaiskulun erillinen tarkastelu paljastaa, että näistä jälkimmäinen on NRRW-prosessin todellinen satunnaisuuden lähe. Satunnaiskävely muuttaa liikkeillään sen omaa tilajoukkoa eli satunnaisverkkoa  $\mathcal{G}_t^s$ . Kun verkko muuttuu, myös satunnaiskävelyn siirtymätodennäköisyydet muuttuvat. Satunnaisverkko

sen sijaan kasvaa satunnaiskävelyn realisaation ohjaamana. Kokonaisuutena NRRW-prosessi on siis ajassa epähomogeeninen äärettömän tilajoukon Markov-prosessi. Proseszin epähomogeenisuus juontaa siitä, että satunnaisverkko muuttuu vain  $s$ :n ajanhetken välein.

### 2.3.1 Tilajoukko

Olkoon  $V = \{v_1, v_2, \dots\}$  kaikkien solmujen numeroituvasti ääretön joukko. Jos  $n$ :n solmun joukkoa merkitään  $V_n = \{v_1, v_2, \dots, v_n\} \subset V$  niin  $n$ :n solmun suuntaamattonien verkkojen joukko on

$$\mathbb{G}_n = \{(V_n, E_n) \mid E_n \subset V_n^2\},$$

missä  $V_n^2 = \{e \subset V_n : |e| = 2\}$  on solmujoukon  $V_n$  järjestämättömien parien joukko. NRRW-prosessiin kuuluu myös satunnaiskävely, jonka tilajoukko on  $n$ :n solmun verkkoa kulkissa verkon solmujen joukko  $V_n$ . Selvästi  $n$ :n solmun NRRW-proseszin tilajoukko on siis karteesinen tulo  $S_n = V_n \times \mathbb{G}_n$ .

Merkitään kaikkien solmujen joukkoa numeroituvasti äärettömällä joukolla  $V$ . Vastaavasti kaikkien verkkojen joukkoa voidaan merkitä äärettömällä unionilla

$$\mathbb{G} = \bigcup_{n=1}^{\infty} \mathbb{G}_n. \quad (5)$$

Koska satunnaisverkko kasvaa NRRW-proseszin edetessä, täytyy sen tilajoukon sisältää kaikki eri solmulukumäärien mahdollistamat tilat. NRRW-proseszin tilajoukko onkin siis

$$S = V \times \mathbb{G}. \quad (6)$$

### 2.3.2 Siirtymätodennäköisyydet

NRRW-proseszin siirtymätodennäköisyydet ovat ajassa epähomogeenisia, sillä satunnaiskävelyn realisaatio kasvattaa sen omaa tilajoukkoa, eli NRRW-proseszin verkkoa vain kun  $t \bmod s = 0$ . Siirtymätodennäköisyyksiä on siten syytä tarkastella kahdessa eri tapauksessa ajanhetken jaollisuuden mukaan.

Olkoon  $(\mathcal{W}_t, \mathcal{G}_t^s)$  NRRW-prosessi hetkellä  $t$  ja  $G = (V, E) \in \mathbb{G}$  sekä  $G' = (V', E') \in \mathbb{G}$  joitain NRRW-proseszin kasvattamia verkkuja. Jos  $v \in V$  ja  $v' \in V'$  ovat solmuja noissa verkoissa, niin NRRW-proseszin siirtymätodennäköisyydet saavat esityksen

$$\begin{aligned} P((v', G'), (v, G)) &= \mathbb{P} [(\mathcal{W}_{t+1}, \mathcal{G}_{t+1}^s) = (v, G) \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v', G')] \\ &= \mathbb{P} [\mathcal{W}_{t+1} = v, \mathcal{G}_{t+1}^s = G \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \end{aligned} \quad (7)$$

**Kun**  $t + 1 \bmod s \neq 0$ , ei prosessin verkkoon lisätä uusia solmuja. Satunnaisverkko  $\mathcal{G}_{t+1}^s$  on siis varmasti sama kuin sen edeltäjä  $\mathcal{G}_t^s$ . Siten  $\mathbb{P} [\mathcal{G}_{t+1}^s = G \mid \mathcal{G}_t^s = G] =$

1 riippumattomasti satunnaiskävelyn tilasta. Selvästi tapahtuman komplementti ( $\mathcal{G}_{t+1}^s \neq G$ ) on mahdoton. Nyt yhtälön 7 mukaisesti

$$\begin{aligned}
P((v', G'), (v, G)) &= \mathbb{P} [\mathcal{W}_{t+1} = v, \mathcal{G}_{t+1}^s = G \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&= \mathbb{P} [\mathcal{G}_{t+1}^s = G \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&\quad \times \mathbb{P} [\mathcal{W}_{t+1} = v \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G', \mathcal{G}_{t+1}^s = G] \\
&= \mathbb{P} [\mathcal{G}_{t+1}^s = G \mid \mathcal{G}_t^s = G'] \mathbb{P} [\mathcal{W}_{t+1} = v \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&= \mathbb{1}\{G = G'\} \mathbb{P} [\mathcal{W}_{t+1} = v \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&= \mathbb{1}\{G_i = G_j\} \frac{\psi_{G_i}(v_i, v_j)}{d_{G_i}(v_i)}
\end{aligned}$$

missä viimeinen yhtäsuuruus seuraa lausekkeesta 3.

**Kun**  $t + 1 \bmod s = 0$ , satunnaisverkko  $\mathcal{G}_{t+1}^s$  määräytyy satunnaiskävelyn  $\mathcal{W}_{t+1}$  realisaatiosta.

$$\begin{aligned}
P((v', G'), (v, G)) &= \mathbb{P} [\mathcal{W}_{t+1} = v, \mathcal{G}_{t+1}^s = G \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&= \mathbb{P} [\mathcal{W}_{t+1} = v \mid \mathcal{W}_t = v', \mathcal{G}_t^s = G'] \\
&\quad \times \mathbb{P} [\mathcal{G}_{t+1}^s = G \mid \mathcal{W}_{t+1} = v, \mathcal{W}_t = v', \mathcal{G}_t^s = G'].
\end{aligned}$$

Uusi solmu lisätään nyt siihen verkon  $G'$  solmuun, johon satunnaiskävely siirtyy ajanhetkellä  $t + 1$ . Selvästi mahdollisia satunnaisverkon  $\mathcal{G}_{t+1}^s$  realisaatioita ovat vain sellaiset verkot, jotka on saatu liittämällä verkoon  $G'$  yksi uusi solmu,  $v^+ = v_{\lfloor \frac{t+1}{s} \rfloor}$ . Kun verkkona  $G$  ja  $G'$  merkitään verkon määritelmän (1) mukaisina solmujen ja kaarien pareina  $(V, E)$  ja  $(V', E')$ ,

$$\begin{aligned}
P((v', G'), (v, G)) &= \mathbb{P} [\mathcal{W}_{t+1} = v \mid \mathcal{W}_t = v', \mathcal{G}_t^s = (V', E')] \\
&\quad \times \mathbb{P} [\mathcal{G}_{t+1}^s = (V, E) \mid \mathcal{W}_{t+1} = v, \mathcal{W}_t = v', \mathcal{G}_t^s = (V', E')] \\
&= \mathbb{1}\{(V, E) = (V' \cup \{v^+\}, E' \cup \{\{v, v^+\}\})\} \\
&\quad \times \mathbb{P} [\mathcal{W}_{t+1} = v' \mid \mathcal{W}_t = v', \mathcal{G}_t^s = (V', E')] \\
&= \mathbb{1}\{G = (V' \cup \{v^+\}, E' \cup \{\{v, v^+\}\})\} \frac{\psi_{G'}(v, v')}{d_{G'}(v)},
\end{aligned}$$

missä viimeinen yhtäsuuruus perustuu kaavaan 3.

### 3 Stokastinen esitys

NRRW-prosessin tarkastelu osoittaa, että kyseessä on monimutkainen ja epähomogeeninen äärettömän tilajoukon stokastinen prosessi. Sen käsittelyä voidaan kuitenkin helpottaa tukeutumalla prosessin kanssa yhdenmukaiseen mutta yksinkertaisempaan malliin, NRRW-prosessin stokastiseen esitykseen.

#### 3.1 Yleistä teoriaa

Olkoon  $(\omega_t)_{t \geq 0}$  jono riippumattomia välin  $[0, 1]$  tasajakautuneita satunnaislukuja. Toisin sanoen  $\omega_t \sim \text{Tas}(0, 1)$ . Tällaista satunnaislukujen sarjaa voidaan käyttää monimutkaisemman stokastisen prosessin algoritmiseen simulointiin. Simulointi rakennetaan etsimällä deterministinen funktio  $\phi : S \times [0, 1] \rightarrow S$ , joka liittää stokastisen prosessin tilan ja välin  $[0, 1]$  reaaliluvun prosessin seuraavaan tilaan. Funktion  $\phi$  tulee toteuttaa  $\forall x, y \in S$  yhtälö

$$\begin{aligned} \mathbb{P}[\phi(x, \omega_t) = y] &= \int_0^1 \mathbf{1}\{\phi(x, w) = y\} dw \\ &= \mathbb{P}[X_{t+1} = y \mid X_t = x] = P(x, y). \end{aligned} \quad (8)$$

Yhtälön 8 toteuttavaa funktiota  $\phi$  ja satunnaislukujonoa  $(\omega_t)_{t \geq 0}$  kutsutaan siirtymämatruiisin  $P$  *stokastiseksi esitykseksi*. Jos stokastisen prosessin alkutila on  $X_0 = x_0$ , sitä voidaan simuloida sen stokastisen esityksen avulla rekursiivisesti:

$$X_{t+1} = \phi(X_t, \omega_t).$$

Mikäli stokastinen prosessi on epähomogeninen ajassa, vaaditaan stokastiseen esitykseen jono deterministisiä funktioita  $(\phi^{(t)})_{t \geq 0}$ , jotka ovat muotoa  $\phi^{(t)} : S \times [0, 1] \rightarrow S$ . Funktioiden  $\phi^{(t)}$  tulee toteuttaa yhtälöä 8 vastaava ehto  $\forall t \geq 0$  ja  $x, y \in S$ :

$$\mathbb{P}[\phi^{(t)}(x, \omega_t) = y] = \mathbb{P}[X_{t+1} = y \mid X_t = x] = P^{(t)}(x, y). \quad (9)$$

Prosessin simulointi tehdään, kuten homogenisessa tilanteessa, mutta rekursiofunktio valitaan funktiojonosta  $(\phi^{(t)})_{t \geq 0}$  järjestyksessä. [7]

#### 3.2 Prosesсин esitys

Edellä havaittiin, että NRRW-prosessi on epähomogeninen ajassa. Sen stokastinen esitys on siis muotoa  $(\phi_t, \omega_t)_{t \geq 0}$ , missä funktiot  $\phi_t$  toteuttavat ehdon 9. Muodostetaan sopiva jono deterministisiä funktioita NRRW-prosessille. Prosesсин tilat ovat muotoa  $(v, G)$ , missä  $G = (V, E)$  on verkko ja  $v \in V$  on verkon  $G$  solmu. Jakamalla rekursiofunktiot satunnaiskävely- ja satunnaisverkko-osiin saadaan niille korkeatasoinen muoto

$$\phi^{(t)}((v, G), w) = (\lambda((v, G), w), \mu^{(t)}(\lambda((v, G), w), G)), \quad (10)$$

missä  $\lambda$  on satunnaiskulun  $(\mathcal{W}_t)_{t \geq 0}$  stokastisen esityksen rekursiofunktio parametriksi annetussa verkossa  $G$ , ja  $\mu$  päivittää verkkoa  $G$  satunnaiskulun realisaation mukaisesti. Satunnaiskulun rekursiofunktio

$$\lambda((v_i, G), w) = \begin{cases} v_1 & w \in [0, \rho(G)_{i,1}) \\ v_2 & w \in [\rho(G)_{i,1}, \rho(G)_{i,1} + \rho(G)_{i,2}) \\ & \vdots \\ v_{|V|} & w \in [\rho(G)_{i,|V|-1}, \rho(G)_{i,|V|-1} + \rho(G)_{i,|V|}) \end{cases}, \quad (11)$$

missä  $\rho$  on funktio verkosta sen siirtymämatriisiin (12) ja  $|V|$  on verkon  $G$  solmujen lukumäärä. Hyödyntäen kaavan 2 esitystä solmuja yhdistävien kaarien määritelmä saadaan kuvaikselle  $\rho$  alkioittainen määritelmä

$$\rho(G)_{i,j} = \frac{\psi_G(v_i, v_j)}{d_G(v_i)} = \frac{1 + \mathbb{1}\{v_i = v_0, v_j = v_0\}}{d_G(v_i)}, \quad (12)$$

joka vastaa aiemmin laskettuja satunnaiskulun  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  siirtymätodennäköisyyksiä (kaava 3). Lopuksi stokastisen esityksen verkkoa päivittävä funktio

$$\mu^{(t)}(v, (V, E)) = \begin{cases} (V \cup \{v_{t/s}\}, E \cup \{\{v, v_{t/s}\}\}) & \text{kun } t \bmod s = 0 \\ (V, E) & \text{muulloin} \end{cases}. \quad (13)$$

Funktio lisää solmuun  $v$  uuden solmun  $v_{t/s}$  aina, kun aika,  $t$ , on jaollinen askelparametrillä  $s$ . Muulloin verkkoa ei muuteta.

**Lause 1.** Olkoon  $(\omega_t)_{t \geq 0}$  jono riippumattomia välin  $[0, 1]$  tasajakautuneita satunnaislukuja. Kaavojen 10-13 määrittelemä jono deterministisiä funktioita  $(\phi^t)_{t \geq 0}$  muodostaa yhdessä satunnaislukujonon  $(\omega_t)_{t \geq 0}$  kanssa NRRW-prosessin stokastisen esityksen.

*Todistus.* NRRW-prosessin siirtymätodennäköisyydet on laskettu kappaleessa 2.3.1. Proseszin käyttäytyminen ajanhetkellä  $t$  riippuu siitä, onko  $t$  jaollinen askelparametrilla  $s$ . Osoitetaan stokastisen esitykseen yhdenmukaisuus NRRW-proseszin kanssa molemmissa tapauksissa. Olkoon  $X_t = (\mathcal{W}_t, \mathcal{G}_t^s)$  NRRW-proseszin tila hetkellä  $t$ .

On näytettävä kaavan 9 mukaisesti, että

$$\mathbb{P}[\phi^{(t)}(x, \omega_t) = y] = \mathbb{P}[X_{t+1} = y \mid X_t = x]$$

kaikilla  $x, y \in S$  ja  $t \in \mathbb{N}_0$ , missä  $S$  on kaavan 6 määrittelemä NRRW-proseszin tilajoukko. Sijoittamalla aiemmin määritelty stokastinen esitys yhtälön vasempaan puoleen saadaan

$$\begin{aligned} \mathbb{P}[\phi^{(t)}(x, \omega_t) = y] &= \int_0^1 \mathbb{1}\{\phi^{(t)}(x, w) = y\} dw \\ &= \int_0^1 \mathbb{1}\left\{\left(\lambda((v_x, G_x), w), \mu^{(t)}(\lambda((v_x, G_x), w), G_x)\right) = (v_y, G_y)\right\} dw \\ &= \int_0^1 \mathbb{1}\{\lambda((v_x, G_x), w) = v_y\} \mathbb{1}\{\mu^{(t)}(\lambda((v_x, G_x), w), G_x) = G_y\} dw \end{aligned}$$

**Kun**  $t + 1 \bmod s \neq 0$  ,

$$\begin{aligned}\mathbb{P} [\phi^{(t)}(x, \omega_t) = y] &= \int_0^1 \mathbb{1}\{\lambda((v_x, G_x), w) = v_y\} \mathbb{1}\{\mu^{(t)}(\lambda((v_x, G_x), w), G_x) = G_y\} dw \\ &= \mathbb{1}\{G_x = G_y\} \int_0^1 \mathbb{1}\{\lambda((v_x, G_x), w) = v_y\} dw \\ &= \mathbb{1}\{G_x = G_y\} \rho(G_x)_{x,y} = \mathbb{1}\{G_x = G_y\} \frac{\psi_{G_x}(v_x, v_y)}{d_{G_x}(v_x)} \\ &= \mathbb{P}[X_{t+1} = y \mid X_t = x].\end{aligned}$$

**Kun**  $t + 1 \bmod s = 0$  ,

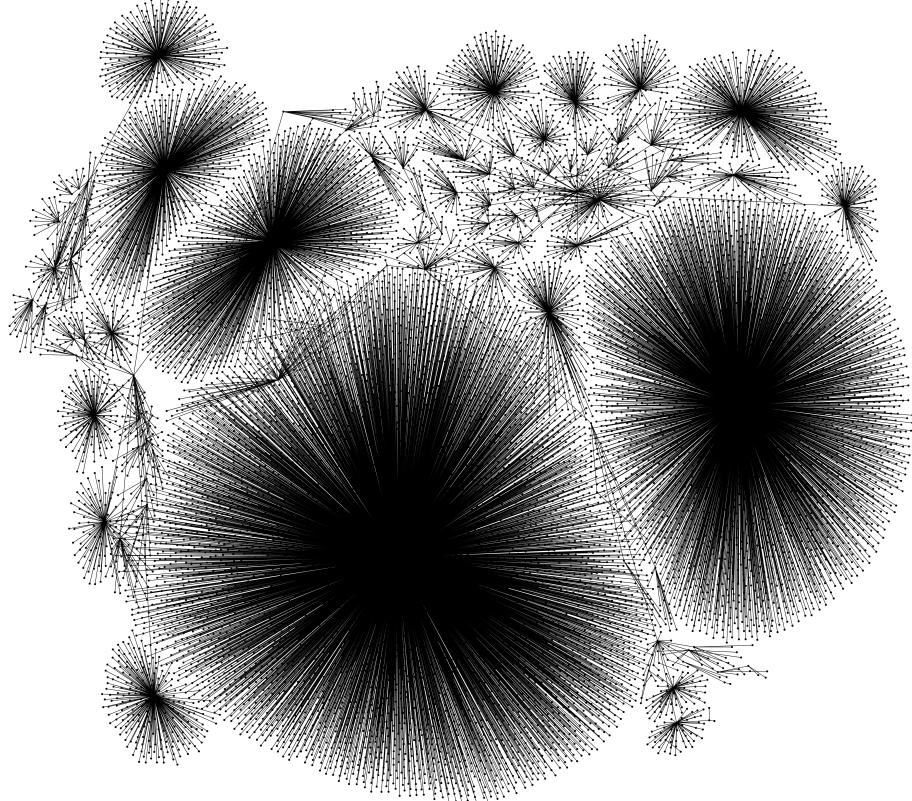
$$\begin{aligned}\mathbb{P} [\phi^{(t)}(x, \omega_t) = y] &= \int_0^1 \mathbb{1}\{\lambda((v_x, G_x), w) = v_y\} \mathbb{1}\{\mu^{(t)}(\lambda((v_x, G_x), w), G_x) = G_y\} dw \\ &= \mathbb{1}\{G_y = (V_x \cup \{v_y\}, E_x \cup \{\{v_x, v_y\}\})\} \int_0^1 \mathbb{1}\{\lambda((v_x, G_x), w) = v_y\} dw \\ &= \mathbb{1}\{G_y = (V_x \cup \{v_y\}, E_x \cup \{\{v_x, v_y\}\})\} \rho(G_x)_{x,y} \\ &= \mathbb{1}\{G_y = (V_x \cup \{v_y\}, E_x \cup \{\{v_x, v_y\}\})\} \frac{\psi^{(t)}(v_i, v_j)}{d_t(v_i)} \\ &= \mathbb{P}[X_{t+1} = y \mid X_t = x].\end{aligned}$$

Selvästi kaikilla  $t \in \mathbb{N}_0$  pätee, että  $\mathbb{P}[\phi^{(t)}(x, \omega_t) = y] = \mathbb{P}[X_{t+1} = y \mid X_t = x]$ . Siten rakentamani stokastinen esitys on ekvivalentti NRRW-prosessin kanssa.  $\square$

## 4 Parillinen prosessi

NRRW-prosessin käyttäytymiseen vaikuttaa voimakkaasti askelparametrin  $s$  parillisuus. Parillisen parametrin prosessissa satunnaiskävely  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  on palautuva, eli se vieraillee kaikissa sen tilajoukon solmuissa äärettömän monta kertaa lähes varmasti. Sen sijaan ainakin askelparametrin  $s = 1$  satunnaiskulku on väistytty. Askelparametrin parillisuus vaikuttaa yhtä radikaalisti satunnaisverkkojen ominaisuuksiin. Parillisen askelparametrin satunnaisverkon  $(\mathcal{G}_t^s)_{t \in \mathbb{N}_0}$  sisäsolmujen astelukujen jakausma on potenssilain alhaalta rajoittama, kun taas askelparametrin  $s = 1$  solmujen astelukujen jakausma on geometrisen jakauman ylhäältä rajoittama.

Erityisesti vain parillisen askelparametrin NRRW-prosessissa kaksi peräkkäistä uutta solmua voidaan lisätä samaan verkon solmuun. Parittoman askelparametrin prosessissa tämä ei ole mahdollista muualla kuin puun juuressa, sillä satunnaiskulku siirtyy  $s$ :ssä askeleessa puun parittomalta tasolta parilliselle ja toisinpäin. Merkitään satunnaiskulun tasoa ajanhetkellä  $t$  funktiolla  $d(\mathcal{W}_t, v_0)$ , missä  $v_0$  on juurisolmu.



Kuva 2: Tietokonesimuloitu NRRW-prosessi parillisella askelparametrilla  $s = 4$ .

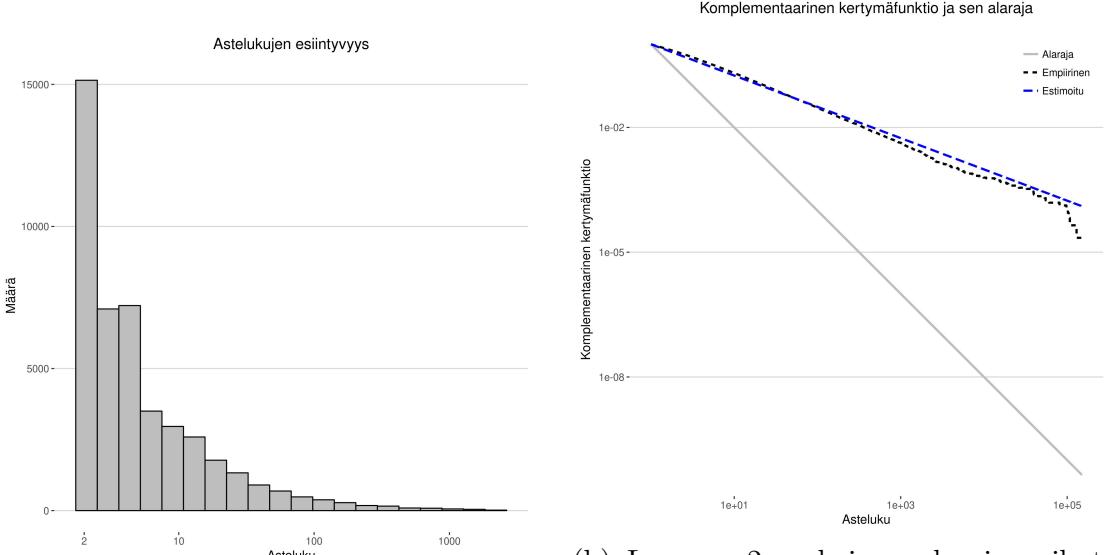
**Määritelmä 2.** *NRRW-prosessi ja sen satunnaiskulku  $\mathcal{W}_t$  ovat parillisia ajanhetkellä  $t \Leftrightarrow d(\mathcal{W}_t, r) + t$  on parillinen. Jos prosessi ei ole parillinen, se on pariton.*

Selvästi parillisen askelparametrin prosessin pariteetti muuttuu vain, kun satunnaiskulku kulkee puun juuren silmukan läpi.

Uuden, peräkkäisen solmun lisääminen samaan solmuun parillisessa NRRW-verkossa on paitsi mahdollista niin myös sitä todennäköisempää mitä suurempi kyseisen solmun asteluku on. Koska uusia solmuja lisätään erityisesti saman pariteetin tasolle puussa, jäävät uudet solmut lehdiksi (eng. leaf), kunnes prosessin pariteetti vaihtuu. Lehtisolmuista voi siirtyä vain takaisin lähtöpisteeseen, joten todennäköisyys palata johonkin satunnaisverkon solmuun suurenee, kun solmussa vierailaan ja sen asteluku kasvaa. Kuva 2 havainnollistaa, kuinka lehtisolmuja kasaantuu tähtimaiseen muodostelmaan joidenkin solmujen ympärille.

Tämä parillisen prosessin ominaisuus liittyy vahvasti suosivaan kiinnittymiseen, sillä uusia solmuja kertyy erityisen todennäköisesti sellaisille verkon solmuille, joilla on jo ennestään paljon naapureita. Kuten aiemmin on todettu, suosiva kiinnittyminen on tosielämän verkkorakenteiden kannalta kiinnostava ominaisuus. Käsite on yhteydessä sosiologiseen Matteus-efektiin, joka tarkoittaa etujen kasaantumista niille, joilla niitä on jo ennestään. Efektiin ydinajatus tiivistetään usein lausahdukseen: "Rikkaat rikastuu ja köyhät köyhtyy". Keskeytynkin tässä kandidaatintyössä parillisen NRRW-prosessin ominaisuuksia käsittelemiseen.

## 4.1 Astelukujen jakauma



(a) Sisäsolmujen astelukujen jakaumalla on raskaan näköinen häntä (eng. heavy tail). Histogrammin luokitus ei ole tasavälinen ja astelukuakseli on logaritminen.

(b) Lauseen 2 mukainen alaraja vaikuttaa simulaatioiden perusteella uskottavalta. Empiirisen kertymäfunktion lisäksi kuvaajaan on piirretty kertymäfunktion estimaatti. Akselit ovat logaritmisia.

Kuva 3: Kuusitoista simuloitua NRRW-prosessia ( $s = 4$ ) tuotti yhteensä 2400000 solmun otoksen. Solmuista 45077 oli sisäsolmuja.

Parillisen askelparametrin ja askelparametrin  $s = 1$  NRRW-prosessit generoivat hyvin erilaisia verkkoja. Ehkä tärkein ero prosessien rakentamien verkkojen välillä liittyy sisäsolmujen astelukujen jakaumiin. Parillinen prosessi nimittäin rakentaa verkkoja, joissa sisäsolmujen astelukujen jakauma on potenssilain alhaalta rajoittama (eng. heavy tail), kun taas parametrin  $s = 1$  vastaava jakauma on geometrisen jakauman ylhäältä rajoittama (eng. exponential tail). Parillisen prosessin jakauma mukailee erityisesti lauseen 2 alarajaa [4].

**Lause 2.** Olkoon NRRW-prosessin askelparametri  $s$  parillinen. Oletetaan, että  $T_j$  on se ajanhetki, jolloin solmuun  $j$  lisätään sen ensimmäinen naapuri. Tällöin kaikilla  $t \geq T_j$  ja  $k \in \{1, \dots, \lfloor \frac{t-T_j}{s} \rfloor + 1\}$  pätee, että

$$\mathbb{P}[d_t(j) \geq k+1 \mid T_j < \infty] \geq k^{-\frac{s}{2}}, \quad \text{jos } j \neq \text{juuri}$$

$$\mathbb{P}[d_t(j) \geq k+2] \geq \left(\frac{k(k+1)}{2}\right)^{-\frac{s}{2}}, \quad \text{jos } j = \text{juuri}$$

Arvioidakseni lauseen 2 uskottavuutta simuloin rinnakkain kuudentoista askelparametrin  $s = 4$  NRRW-prosessin kehitystä 600000 askeleen päähän alkutilasta. Simulaatio-ohjelman lähdekoodi löytyy liitteestä A. Kuvaaja 3 esittää simuloidun

otoksen sisäsolmujen astelukujen jakaumaa (3a) ja vertaa sisäsolmujen empiiristä kertymäfunktiota  $\bar{F}_n$  lauseen 2 mukaiseen alarajaan (3b). Huomaa, että kuvaajan 3b funktiot ovat komplementaarisia kertymäfunktioita ( $\bar{F}(k) = 1 - F(k)$ ) perinteisen kertymäfunktion  $F$  sijaan.

Koska kuvaajan 3b käyrät näyttävät potenssilain mukaisilta, sovitin empiiriseen kertymäfunktioon  $\bar{F}_n$  mallin  $\hat{\bar{F}}(k) = k^{-b}$ . Epälineaarinen regressio estimoi parametrille  $b$  arvon  $\hat{b} \approx \frac{3}{4}$ . Selvästiikään lauseen 2 alaraja ei ole kovin hyvä ainakaan NRRW-prosessin askelparametrilla  $s = 4$ . Yhden askelparametrin simulaatiotuloksista ei kuitenkaan voi tehdä sellaista johtopäätöstä, että alaraja olisi heikko kaikilla parillisen askelparametrin prosesseilla.

| $s$       | 2     | 4     | 6     | 8     | 10    |
|-----------|-------|-------|-------|-------|-------|
| $s/2$     | 1     | 2     | 3     | 4     | 5     |
| $\hat{b}$ | 0.645 | 0.752 | 0.910 | 0.814 | 0.870 |

Taulukko 3: Sisäsolmujen astelukujen jakauman alaraja näyttää heikkenevän askelparametrin kasvaessa.

Hahmottaakseni alarajan vahvuutta tarkemmin ajoin lyhyet simulaatiot viidellä eri parillisella askelparametrilla (2, 4, 6, 8 ja 10) ja vertasin lauseen 2 alarajaa estimoituihin, komplementaarisiin kertymäfunktioihin,  $\hat{\bar{F}}$ . Taulukko 3 esittää, miten estimoitu potenssi,  $\hat{b}$ , kehittyy alarajan potenssin,  $\frac{s}{2}$ , funktiona. Alaraja näyttää heikkenevän askelparametrin kasvaessa. Kaiken kaikkiaan vaikuttaakin siltä, että lauseen 2 alaraja on melko heikko. Vahvemman alarajan löytäminen ja sen oikeaksi osoittaminen on mahdollinen jatkotutkimuksen aihe.

## 4.2 Palautuvuus

Parillisessa NRRW-prosessissa pariteetin (määritelmä 2) vaihtumisella on tärkeä rooli prosessin rakentaman verkon kasvussa. Koska uusia solmuja lisätään vain parilliselle (tai parittomille) tasolle, kunnes prosessin pariteetti vaihtuu, näihin vasta lisättyniin solmuihin, jotka kuuluvat parittomalalle (tai parilliselle) tasolle, ei voida liittää uusia naapureita ennen kuin prosessin pariteetti vaihtuu. Tämä tarkoittaa sitä, että satunnaiskulun pariteetin on vaihduttava, jotta sen rakentaman puun korkeus kasvaisi.

Selvästi satunnaisverkon korkeus jää äärelliseksi, mikäli satunnaiskulun pariteetti vaihtuu vain äärellisen monta kertaa. Jotta satunnaiskulun pariteetti vaihtuisi äärettömän monta kertaa ja sen kasvattaman verkon korkeus olisi rajoittamaton, on satunnaiskulun vierailtava juurisolmussa äärettömän monta kertaa. Ainakin juurisolmun on siis oltava palautuva tila. Myöhemmin näemme, että peräti kaikki NRRW-verkon solmut ovat palautuvia. Ennen sitä osoitan kuitenkin lemmann 2, joka liittää satunnaiskulun vierailut johonkin solmuun satunnaiskulun vierailuihin sen naapureissa. Hyödynnän todistuksessa toista Borelin-Cantellin lemmaa, joka on yksi todennäköisyysteorian keskeisimpiä lauseita.

**Lemma 1** (Toinen Borelin-Cantellin lemma). *Olkoon  $E_1, E_2, \dots$  jono riippumattomia tapahtumia jossain todennäköisyysavaruudessa. Jos jonon tapahtumien todennäköisyksien summa on ääretön, eli*

$$\sum_{n=1}^{\infty} \mathbb{P}[E_n] = \infty,$$

*niin todennäköisyys, että tapahtumia sattuu äärettömän monta on yksi:*

$$\mathbb{P}\left[\limsup_{n \rightarrow \infty} E_n\right] = 1,$$

*missä  $\limsup_{n \rightarrow \infty} E_n = \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} E_k$  on tapahtumajonon  $\{E_n\}_{n=1}^{\infty}$  limit supremum.* [8]

Lemman 2 todistus mukailee aiemmassa tutkimuksessa (Iacobelli et al. [4]) rakennettua todistusta, mutta hyödynnän siinä itse rakentamaani NRRW-prosessin stokastista esitystä ja paikkaan aiemman todistuksen puutteita.

**Lemma 2.** *Olkoon  $v_i$  solmu ja  $\{v_i, v_j\}$  kaari NRRW-prosessin verkossa. Jos  $v_i$  on palautuva, niin satunnaiskävely  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  kulkee kaaren  $\{v_i, v_j\}$  läpi äärettömän monta kertaa lähes varmasti.*

*Todistus.* Määritellään aluksi apufunktioit

$$\begin{aligned} J_{t_1}^{t_2}(v_i) &= \sum_{k=t_1}^{t_2} \mathbb{1}\{\mathcal{W}_k = v_i\} \\ J_{t_1}^{t_2}(v_i, v_j) &= \sum_{k=t_1}^{t_2} \mathbb{1}\{\mathcal{W}_k = v_i\} \mathbb{1}\{\mathcal{W}_{k+1} = v_j\}. \end{aligned}$$

Funktion  $J_{t_1}^{t_2}(v_i)$  arvo kertoo, kuinka usein satunnaiskulkku  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  vierailee solmussa  $v_i$  ajanhetkien  $t_1$  ja  $t_2$  välillä. Vastaavasti  $J_{t_1}^{t_2}(v_i, v_j)$  ilmaisee, montako kertaa satunnaiskulkku siirtyy solmusta  $v_i$  solmuun  $v_j$  kaaren  $\{v_i, v_j\}$  läpi. Oletuksen mukaan solmu  $v_i$  on palautuva. Tämä tarkoittaa sitä, että NRRW-prosessin satunnaiskulkku palaa äärellisessä ajassa takaisin solmuun  $v_i$  riippumatta satunnaisverkon  $\mathcal{G}_t^s$  tilasta. Jos ensimmäinen siirtymähetki solmuun  $v_i$  määritellään

$$T_{v_i}^+ = \min\{t \geq 1 : \mathcal{W}_t = v_i\}$$

niin solmulle  $v_i$  päätee kaikilla oletuksen mukaisilla verkoilla  $G$ , että

$$\mathbb{P}\left[T_{v_i}^+ < \infty \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v_i, G)\right] = 1.$$

Jos paluuaiaka solmuun  $v_i$  on äärellinen lähes varmasti, niin on selvää, että solmusta  $v_i$  käynnistyvä satunnaiskulkku  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  myös vierailee rajoittamattoman ajanjakson aikana äärettömän monta kertaa solmussa  $v_i$ :

$$\mathbb{P}[J_0^\infty = \infty \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v_i, G)] = 1. \quad (14)$$

Olkoon  $(v', G')$  nyt mikä tahansa NRRW-prosessin tila, joka on saavutettavissa ajanhetkellä  $t$  ( $\mathbb{P}[(\mathcal{W}_t, \mathcal{G}_t^s) = (w, G)] > 0$ ). Koska NRRW-prosessin satunnaiskävely on yhtenäinen, satunnaiskulku  $(\mathcal{W}_t)_{t \in \mathbb{N}_0}$  vierailee solmussa  $v_i$  äärettömän monta kertaa lähes varmasti riippumatta siitä, mistä ajanhetkellä  $t$  saavutettavasta tilasta se käynnistyy. Toisin sanoen  $\mathbb{P}[J_t^\infty(v_i) = \infty \mid (\mathcal{W}_t, \mathcal{G}_t) = (v', G')] = 1$ . Jos näin ei olisi, niin

$$\begin{aligned} & \mathbb{P}[J_0^\infty < \infty \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v_i, G)] \\ & > \mathbb{P}[J_t^\infty(v_i) < \infty \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v', G')] \\ & \times \mathbb{P}[(\mathcal{W}_t, \mathcal{G}_t^s) = (v', G') \mid (\mathcal{W}_t, \mathcal{G}_t^s) = (v_i, G)] > 0, \end{aligned}$$

mikä on ristiriita sen oletuksen kanssa, että  $v_i$  on palautuva.

Olkoon  $t_0$  nyt sellainen ajanhetki, jolloin solmut  $v_i$  ja  $v_j$  sekä niitä yhdistää vä kaari  $\{v_i, v_j\}$  kuuluvat verkkoon  $\mathcal{G}_{t_0}^s = G_0$ . Lemman väite on nyt tosi, jos  $\mathbb{P}[J_{t_0}^\infty(v_i, v_j) = \infty] = 1$ . Todistetaan tämä hyödyntäen NRRW-prosessin stokastista esitystä.

Olkoon  $(\omega_t)_{t \geq t_0}$  jono riippumattomia välin  $[0, 1]$  tasajakutuneita satunnaislukuja. Kuten lause 1 osoittaa, tällaista satunnaislukujen jonoa voidaan käyttää koko NRRW-prosessin  $(\mathcal{W}_t, \mathcal{G}_t^s)$  simulointiin kaikilla  $t \geq t_0$ . Erityisesti sen avulla voidaan määrittää, siirryykö satunnaiskulku solmesta  $v_i$  solmuun  $v_j$  ajanhetkellä  $t$ . Määritellään tästä varten uusi jono satunnaislukuja:

$$\begin{aligned} \xi_t &= \mathbf{1}\{\lambda((v_i, G_t), \omega_t) = v_j\} \\ &= \mathbf{1}\{\omega_t \in [\rho(G_t)_{i,j-1}, \rho(G_t)_{i,j-1} + \rho(G_t)_{i,j}]\} \\ &= \mathbf{1}\left\{\omega_t \in \left[0, \frac{\psi_{G_t}(v_i, v_j)}{d_{G_t}(v_i)}\right]\right\} = \mathbf{1}\left\{\omega_t \in \left[0, \frac{1 + \mathbf{1}\{v_i = v_0, v_j = v_0\}}{d_{G_t}(v_i)}\right]\right\}, \end{aligned}$$

missä yhtäsuuruudet pohjautuvat kaavoihin 11, 12 ja 2. Lisäksi  $v_0$  on verkon juuri-solmu. Huomaa, että satunnaisluku  $\xi_t$  riippuu koko NRRW-prosessin satunnaiskulun historiasta hetkeen  $t$  asti. Nyt voidaan merkitä

$$J_{t_0}^t(v_i, v_j) = \sum_{k=t_0}^t \mathbf{1}\{\mathcal{W}_k = v_i\} \mathbf{1}\{\mathcal{W}_{k+1} = v_j\} = \sum_{k=t_0}^t \mathbf{1}\{\mathcal{W}_k = v_i\} \xi_k.$$

Olkoon  $t_h$  se satunnainen ajanhetki, jolloin satunnaiskulku  $\mathcal{W}_t$  palaa solmuun  $v_i$   $h$ :nnen kerran ajanhetken  $t_0$  jälkeen. Jos  $J_{t_0}^\infty < \infty$ , niin määritellään, että  $t_h = t_{J_{t_0}^\infty(v_i)} + h$ , kun  $h > J_{t_0}^\infty(v_i)$ . Formuloidaan jälleen uusi satunnaislukujen jono indikoimaan, kulkeeko satunnaiskulku  $h$ :nnen paluuhetken jälkeen kaaren  $\{v_i, v_j\}$  läpi:

$$\xi'_h = \mathbf{1}\left\{\omega_{t_h} \in \left[0, \frac{1 + \mathbf{1}\{v_i = v_0, v_j = v_0\}}{d_{G_0}(v_i) + h}\right]\right\}.$$

Kun  $h < J_{t_0}^\infty(v_i) + 1$ , niin  $\xi_{t_h}$  stokastisesti dominoi prosessia  $\xi'_h$ . Siis kaikilla  $h$  pätee, että  $\xi'_h \stackrel{(s.t.)}{\leq} \xi_{t_h}$ . Tämä johtuu siitä, että solmun  $v_i$  asteluku on kasvanut korkeintaan

$h$ :n verran  $h$ :n vierailun jälkeen eli  $d_{G_{t_h}}(v_i) \leq d_{G_0}(v_i) + h$ . Yläraja  $d_{G_{t_h}}(v_i) + h$  perustuu sellaiseen tilanteeseen, jossa jokainen vierailuhetki on ollut jaollinen NRRW-prosessin askelparametrilla ja solmuun  $v_i$  on siten lisätty aina vierailtaessa uusi solmu. Nyt pätee, että

$$J_{t_0}^t(v_i, v_j) = \sum_{k=t_0}^t \mathbb{1}\{\mathcal{W}_k = v_i\} \xi_k \geq \sum_{k=1}^{J_{t_0}^\infty(v_i)} \xi'_k$$

Tätä alarajaa hyödyntäen näemme, että

$$\begin{aligned} \mathbb{P}[J_{t_0}^\infty(v_i, v_j) < \infty] &\leq \mathbb{P}\left[\sum_{h=1}^{J_{t_0}^\infty(v_i)} \xi'_h < \infty\right] \\ &= \mathbb{P}\left[\left\{\sum_{h=1}^{J_{t_0}^\infty(v_i)} \xi'_h < \infty\right\} \cap \left\{J_{t_0}^\infty(v_i) = \infty\right\}\right] \\ &+ \underbrace{\mathbb{P}\left[\left\{\sum_{h=1}^{J_{t_0}^\infty(v_i)} \xi'_h < \infty\right\} \cap \left\{J_{t_0}^\infty(v_i) < \infty\right\}\right]}_{=0, \text{ koska } v_i \text{ on palautuva}} \\ &= \mathbb{P}\left[\left\{\sum_{h=1}^\infty \xi'_h < \infty\right\} \cap \left\{J_{t_0}^\infty(v_i) = \infty\right\}\right] \\ &= \mathbb{P}\left[\sum_{h=1}^\infty \xi'_h < \infty\right]. \end{aligned}$$

Toisessa yhtäsuuruudessa  $J_{t_0}^\infty(v_i)$  korvataan äärettömällä, sillä tarkasteltava tapahtuma on tapahtuman  $\{J_{t_0}^\infty = \infty\}$  osajoukko. Toinen tapahtuma, joka on joukon  $\{J_{t_0}^\infty < \infty\}$  osajoukko, on mahdoton, sillä  $v_i$ :n palautuvuudesta seuraa, että  $\mathbb{P}[J_{t_0}^\infty < \infty] = 0$ .

Tarkastelemalla riippumattomien tapahtumien  $\{\xi'_h = 1\}$  todennäköisyyksien summaa nähdään, että

$$\begin{aligned} \sum_{h=1}^\infty \mathbb{P}[\xi'_h = 1] &= \sum_{h=1}^\infty \mathbb{P}\left[\omega_{t_h} \in \left[0, \frac{1 + \mathbb{1}\{v_i = v_0, v_j = v_0\}}{d_{G_0}(v_i) + h}\right]\right] \\ &= \sum_{h=1}^\infty \frac{1 + \mathbb{1}\{v_i = v_0, v_j = v_0\}}{d_{G_0}(v_i) + h} \geq \sum_{h=1}^\infty \frac{1}{d_{G_0}(v_i) + h} \\ &= \sum_{h=1}^\infty \frac{1}{h} - \sum_{h=1}^{d_{G_0}(v_i)} \frac{1}{h} = \infty - H_{d_{G_0}(v_i)} = \infty, \end{aligned}$$

missä kaksi viimeistä yhtäsuurutta seuraavat siitä, että harmoninen sarja hajaantuu äärettömäksi ja  $H_{d_{G_0}(v_i)}$  on äärellinen, harmoninen numero.

Koska tapahtumajonon  $\{\xi'_h = 1\}_{h=1}^\infty$  alkiot ovat riippumattomia ja niiden todennäköisyyksien summa on ääretön, toisen Borelin-Cantellin lemmän (1) mukaan

$$\mathbb{P}\left[\limsup_{h \rightarrow \infty} \{\xi'_h = 1\}\right] = 1,$$

eli tapahtumia  $\{\xi'_h = 1\}$  sattuu äärettömän monta lähes varmasti. Siten todennäköisyyks, että tapahtumia tapahtuu vain äärellinen määrä

$$\mathbb{P} \left[ \sum_{h=1}^{\infty} \xi'_h < \infty \right] = 0$$

ja

$$\mathbb{P} \left[ J_{t_0}^{\infty}(v_i, v_j) < \infty \right] = 0 \Leftrightarrow \mathbb{P} \left[ J_{t_0}^{\infty}(v_i, v_j) = \infty \right] = 1,$$

mikä todistaa lemmantaan väitteen.  $\square$

Lemmaa 2 hyödyntäen lähdetutkimuksessa (Iacobelli et al.) todistetaan ehkä keskeisin parillisen askelparametrin NRRW-prosessia koskeva tulos, lause 3. Sen todistus sivuutetaan tässä kandidaatintyössä tilanpuutteen vuoksi. [4]

**Lause 3.** *Kaikki solmut ovat palautuvia parillisen askelparametrin NRRW-prosessissa.*

## 5 Yhteenveto

NRRW-prosessin tarkastelu paljastaa fundamentaalisen yhteyden satunnaiskulun palautuvuuden ja sen kasvattaman verkon rakenteen välillä. Mikäli prosessin askelparametri on yksi, satunnaiskulku on väistyvä, ja se rakentaa satunnaisverkkoja, joiden sisäsolmujen astelukujen jakauma on geometrisen jakauman ylhäältä rajoittama. Sen sijaan parillisen askelparametrin NRRW-prosessissa satunnaiskulku on palautuva (lause 3), ja prosessin generoimien verkkojen sisäsolmujen astelukujen jakauma on erään potenssilain (lause 2) alhaalta rajoittama.

Tässä kandidaatintyössä syvennytin NRRW-prosessin kasvumallin ymmärtämiseen ja esittelyyn. Formuloin prosessille siirtymätodennäköisyydet, tilajoukon ja erään stokastisen esityksen, jonka osoitin oikeaksi (lause 1). Hyödynsin kyseistä stokastista esitystä parillisen NRRW-prosessin ydinominaisuksien tarkastelussa. Ensin ohjelmoin stokastiseen esitykseen perustuvan simulaatio-ohjelman NRRW-prosessille ja arvioin lauseen 2 astelukujen jakauman alarajan hyvyyttä simulaatiotuloksilla. Alaraja osoittautui uskottavaksi, muttei vahvimmaksi mahdolliseksi. Lopuksi ehostin aiemmassa tutkimuksessa esitettyä lemmantaa 2 todistusta täydentämällä välivaiheita ja osoittamalla, että NRRW-prosessin stokastinen esitys, johon todistus perustuu, on todellakin olemassa.

Tämän lyhyen tarkastelun perusteella NRRW-malli on kiinnostava askel kohti syvälliempää verkkojen kasvun ymmärtämistä. Se yhdistää satunnaiskulun ja sen kasvattaman verkon ominaisuudet tavalla, joka voi parantaa käsitystämme tosielämän verkkojen kasvun taustalla olevista mekanismeista. Tämän fundamentaalisen kahtiajan ymmärtämislle on oma sijansa myös tulevaisuuden verkkopohjaisissa tietotekniikan sovelluksissa, kuten kryptovaluutta IOTA:ssa. Vain aika näyttää, mitä kaikkea NRRW-prosessin ja sen kaltaisten verkkojen kasvumallien tutkimus inspiroi ja mahdollistaa.

## Viitteet

- [1] Clauset, A., Shalizi, C. R. ja Newman, M. E. Power-law distributions in empirical data. Department of Computer Science, University of New Mexico, Albuquerque, USA, 2009.
- [2] Barabási, A. Linkit - verkostojen uusi teoria. Helsinki, Terra Gognita Oy, 2002. ISBN 952-5202-66-6.
- [3] Barabási, A., Albert, R. Emergence of Scaling in Random Networks. Department of Physics, University of Notre-Dame, Notre-Dame, France, 1999.
- [4] Iacobelli, G., Figueiredo, D. R., Neglia, G. Transient and Slim versus Recurrent and Fat: Random Walks and the Trees they Grow. Istituto de Matemática, Federal University of Rio de Janeiro, Brazil, 2017.
- [5] Popov, S. The Tangle. Viitattu 3.5.2018. Saatavissa: [https://iota.org/IOTA\\_Whitepaper.pdf](https://iota.org/IOTA_Whitepaper.pdf).
- [6] Popov, S., Saa, O., Finardi, P. Equilibria in the Tangle. Department of Statistics, Institute of Mathematics, Statistics and Scientific Computation, University of Campinas, Brazil, 2017.
- [7] Häggström, O. Finite Markov Chains and Algorithmic Applications. 4. painos. Cambridge, Cambridge University Press, 2007. ISBN 978-0-521-89001-4.
- [8] Klenke, A. Probability Theory. 2. painos. London, Springer-Verlag London, 2014. ISBN 978-1-4471-5360-3.

## A Simulaatoiden lähdekoodi

Simuloin NRRW-prosessia itse kirjoittamallani C++-kielisellä ohjelmalla. Lähdekoodin käänntäminen vaatii vähintään C++11-versiota tukevan käännejän, asennetun *Boost Graph Library*:n (BGL) ja *OpenMP*-tuen rinnakkaislaskennan mahdollistamiseksi.

Ohjelma tulostaa jokaista simuloitua NRRW-prosessia kohden kaksi tiedostoa työskentelykansion *output*-alikansioon. Tiedostot ovat CSV- ja DOT-formaateissa ja niiden nimet ovat muotoa *degrees\_N.csv* ja *graph\_N.dot*. Näistä ensimmäinen sisältää N:nnen simulaation kaikkien solmujen asteluvut listana. Jälkimmäinen tiedosto sisältää N:nnen simuloidun verkon esityksen DOT-kielellä. Verkon DOT-kielinen esitys voidaan muuttaa verkkoa esittäväksi kuvaksi esimerkiksi *sdfp*-työkalulla.

---

```

1 #include <iostream>
2 #include <random>
3 #include <iterator>
4 #include <fstream>
5 #include <memory>
6
7 #include <boost/config.hpp>
8 #include <boost/graph/adjacency_list.hpp>
9 #include <boost/graph/graphviz.hpp>
10
11 template <typename RandomGenerator = std::default_random_engine>
12 struct random_selector {
13     public:
14         random_selector(RandomGenerator g = RandomGenerator(std::random_device("dev/urandom")()))
15             : gen(g) {}
16
17         template <typename Iter>
18         Iter select (Iter start , Iter end) {
19             std::uniform_int_distribution<> dist(0, std::distance(start , end) - 1);
20             std::advance(start , dist(gen));
21             return start ;
22         }
23
24     private:
25         RandomGenerator gen;
26     };
27
28 template <typename Graph>
29 struct graph_writer {
30     public:
31         graph_writer(Graph& g) : graph(g) { }
32
33         void write_graphviz(const std::string& filename) const {

```

```

34     std::ofstream file (filename);
35     boost::write_graphviz(file, graph, v_writer, boost::default_writer(),
36                           g_writer);
37     file .close () ;
38 }
39
40 private:
41     const Graph& graph;
42
43     struct vertex_property_writer {
44         template <class VertexOrEdge>
45         void operator()(std::ostream& out, const VertexOrEdge& v) const {
46             out << "[label=\\\"\\\"";
47             }
48         } v_writer;
49
50     struct graph_property_writer {
51         void operator()(std::ostream& out) const {
52             out << "graph [ranksep=4 nodesep=1]" << std::endl;
53             out << "node [shape=point]" << std::endl;
54             }
55         } g_writer;
56     };
57
58 struct nrrw_process {
59     public:
60         typedef boost::adjacency_list<boost::listS, boost::vecS, boost::undirectedS,
61                               boost::no_property, boost::no_property> graph_t;
62
63         typedef boost::graph_traits<graph_t>::vertex_descriptor vertex_t;
64         typedef boost::graph_traits<graph_t>::edge_descriptor edge_t;
65
66         nrrw_process(const unsigned int& p) : parameter(p), counter(0), writer(graph) {
67             current = boost::add_vertex(graph);
68             boost::add_edge(current, current, graph);
69         }
70
71         std::vector<unsigned int> get_degrees() const {
72             typename boost::graph_traits<graph_t>::vertex_iterator start, end;
73             boost::tie(start, end) = boost::vertices(graph);
74
75             const unsigned int size = boost::num_vertices(graph);
76             std::vector<unsigned int> degrees(size);
77             std::transform(start, end, degrees.begin(),
78                           [this](&vertex_t v){return this->degree(v);});
79             return degrees;
80         }
81
82 }
```

```

80     void simulate(unsigned int steps) {
81         counter = 0;
82         while (step() < steps) {}
83     }
84
85     void write_graphviz(const std::string& filename) {
86         writer.write_graphviz(filename);
87     }
88
89     private:
90     graph_t graph;
91     vertex_t current;
92
93     random_selector<> selector;
94     graph_writer<graph_t> writer;
95
96     unsigned int counter = 0;
97     const unsigned int& parameter;
98
99     unsigned int step() {
100         // choose an outgoing edge randomly
101         typename boost::graph_traits<graph_t>::out_edge_iterator start, end;
102         boost::tie(start, end) = boost::out_edges(current, graph);
103
104         auto random_iter = selector.select(start, end);
105         const edge_t& random_edge = *random_iter;
106
107         // move to the random vertex
108         current = boost::target(random_edge, graph);
109         counter++;
110
111         // add a new vertex to the graph if necessary
112         if (counter % parameter == 0) {
113             vertex_t new_vertex = boost::add_vertex(graph);
114             boost::add_edge(current, new_vertex, graph);
115         }
116
117         return counter;
118     }
119
120     unsigned int degree(vertex_t vertex) const {
121         return boost::out_degree(vertex, graph);
122     }
123 };
124
125 int main(int argc, char* argv[]) {
126     if (argc < 3) {
127         std::cerr << "Usage: " << argv[0] << " PARAMETER STEPS [COUNT]" <<

```

```

128     std::endl;
129     return 1;
130 }
131 const std::string directory = "output";
132 const int parameter = atoi(argv[1]);
133 const int steps = atoi(argv[2]);
134
135 int count = 1;
136 if (argc == 4) {
137     count = atoi(argv[3]);
138 }
139
140 std::vector<std::unique_ptr<nrrw_process>> simulations;
141 for (int i = 0; i < count; i++) {
142     simulations.push_back(
143         std::unique_ptr<nrrw_process>(new nrrw_process(parameter)));
144 }
145
146 #pragma omp parallel for
147 for (int i = 0; i < simulations.size(); i++) {
148     auto& process = simulations.at(i);
149     process->simulate(steps);
150
151     auto degrees = process->get_degrees();
152
153     std::string degrees_path = directory + "/degrees_" + std::to_string(i) + ".csv";
154     std::ofstream degrees_file(degrees_path);
155     for (auto it = degrees.cbegin(); it != degrees.cend(); it++) {
156         degrees_file << *it << std::endl;
157     }
158     degrees_file.close();
159
160     std::cout << "FINISHED (" << i << ")" << std::endl;
161
162     std::string graph_path = directory + "/graph_" + std::to_string(i) + ".dot";
163     process->write_graphviz(graph_path);
164 }
165
166 return 0;
167 }
```

---