

Juho Salli

TYÖVOIMAN MÄÄRÄN ENNUSTAMINEN

Tietojenkäsittelyn koulutusohjelma
Sovellustuotannon suuntautumisvaihtoehto
2013

TYÖVOIMAN MÄÄRÄN ENNUSTAMINEN

Salli, Juho
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Maaliskuu 2013
Ohjaaja: Kyngäs, Jari
Sivumäärä: 19
Liitteitä: -

Asiasanat: työvoiman hallinta, optimointi, ennustaminen, simulointi

Opinnäytetyön aiheena on työvoiman hallinnan optimointi yleisellä tasolla sekä tarvittavan työvoiman määrän ennustaminen. Työvoiman määrän ennustaminen selitetään esittelemällä simulaattori-sovellus, jonka olen tehnyt simuloidakseni yhteyskeskuksen toimintaa. Työssä kuvataan myös PEAST -algoritmin toimintaa yleisellä tasolla.

Simuloinnilla pyritään selvittämään mahdollisimman tarkasti vaadittava työntekijöiden määrä eri ajanhetkinä. Simulaattori käyttää hyväkseen yhteyskeskuksen historiatietoja, joista selviää sekä saapuvien puheluiden määrä että työntekijöiden kyky palvella asiakkaita.

Esitän myös pohdintaa työvoiman hallinnan optimoinnin tulevaisuuden mahdollisuuksista. Näihin sisältyy esimerkiksi kustannussäästöjä yritykselle sekä työtyytyväisyyden parantuminen.

PREDICTING THE WORKLOAD

Salli, Juho

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technologies

March 2013

Supervisor: Kyngäs, Jari

Number of pages: 19

Appendices: -

Keywords: workforce management, optimization, prediction, simulation

The purpose of this thesis was to present a very high-level explanation of workforce management optimization with emphasis to predicting the needed amount of workers for a specific timeframe. To explain the process of predicting the amount of needed workers for a specific timeframe this thesis will expound on a simulation software that I have developed for simulating the operation of a contact center. The thesis will also describe the function of the PEAST-algorithm in a general.

Simulation is used to determine the needed amount of workers with as high an accuracy as possible. The simulation software uses the records of the contact center. The records show both the quantity of incoming calls and the workers' competence to serve different kinds of clients.

This thesis will also present some thoughts about the future applications of workforce management optimization. These include, for example, the lowering of salary expenses for the company and the betterment of job satisfaction.

SISÄLLYS

1	JOHDANTO.....	5
2	TYÖVOIMAN HALLINTA	6
2.1	Terminologia.....	6
2.2	Vuorojen muodostaminen käsin	7
2.3	Työvoiman hallinnan optimointi	8
3	TYÖVOIMATARPEEN ENNUSTAMINEN	11
3.1	Historiatiedot	11
3.2	Simulaattori.....	12
3.2.1	Asetukset	13
3.2.2	Toimintalogiikka	14
3.2.3	Tuloste-esimerkki.....	16
4	TULEVAISUUDEN MAHDOLLISUUDET	17
4.1	Integraatiot kaupallisiin sovelluksiin	17
4.2	Tyytyväisyys	17
4.3	Säästöt	18

1 JOHDANTO

Opinnäytetyössäni käsitellään työvoiman hallintaa ja sen optimointia, erityisesti työvoimatarpeen ennustamiseen paneutuen. Työvoiman hallinnalla tarkoitetaan tässä työssä työvuorojen tekemistä sekä työntekijöiden sijoittamista vuoroihin siten, että lakisäädökset, työnantajan vaatimukset ja työntekijöiden omat toiveet toteutuvat. Työvuoroja tehdään edelleenkin manuaalisesti, eli vuorojentekijät asettavat työntekijöille työpäivän sisällön edellä mainitut ehdot huomioiden. Tämä kuitenkin on erittäin aikaa vievää toimintaa, joka saattaa viedä useita päiviä. Yleensä vuorojentekijä tekee tätä muun työnsä ohessa.

Varsinkin nykyaikana, jolloin yritykset hakevat kustannustehokkuutta joka saralta, työvoiman hallinnan tärkeys on korostunut. Yritykset eivät halua maksaa palkkaa liialle työvoimalle, mutta toisaalta eivät halua saada lisäkustannuksia töiden viivästymisestä työvoiman puutteen takia. Juuri tähän ongelmaan työvoimatarpeen ennustaminen paneutuu; juuri oikean kokoisen henkilöstömäärän löytämiseen.

Työvuorojen, variaatioineen, tekeminen optimaaliseksi on NP-kova ongelma (Garey & Johnson 1979) (Lau 1996). Koska ongelmaan ei ole olemassa ratkaisua, avuksi ovat nousseet laskennallista älykkyyttä hyödyntävät algoritmit. Laskennallisen älykkyyden laajahko tutkimus on viime vuosina johtanut jo ensimmäisiin kaupallisiin sovelluksiin (Nameron 2012) (Ortec Harmony 2012).

Kerron tässä työssä yleisellä tasolla työvoiman hallinnan optimoinnin prosessista, jonka jälkeen paneudutaan tarkemmin työvoiman määrän ennustamiseen ja kirjoittamaani puhelinkeskuksen toiminnan simulointisovellukseen. Viimeisenä työssä on pohdintoja tulevaisuuden mahdollisuuksista työvoiman hallinnan optimoinnin avulla.

2 TYÖVOIMAN HALLINTA

Työvoiman hallinta käsitteenä sisältää monia asioita, kuten ihmissuhteiden hallintaa, työntekijöiden ja työnjohtajien koulutusta, palkka-asioita ja työntekijöiden etenemiskeskusteluja. Tässä työssä työvoiman hallinnalla tarkoitetaan keskitetysti tarvittavan työvoimamäärän ennustamista, vuorojen suunnittelua sekä työntekijöiden sijoittamista vuoroihin.

Tarkoituksena on siis luoda jokaiselle työntekijälle työvuorolistat sekä saada tekijä jokaiselle työlle.

2.1 Terminologia

Tässä kappaleessa kerron työvuorolistojen tekemisessä käytettävän terminologian. Käytän samoja termejä kuin (Kyngäs, Nurmi & Kyngäs 2013).

Työvuorojen suunnittelu koostuu työntekijöiden sijoittamisesta työtehtäviin ja vuoroihin annetun suunnittelujakson ajalle. *Suunnittelujakso* on aikajakso, jonka pituus on yleensä 2-6 viikkoa, mutta joissakin tapauksissa se voi olla pidempikin. Jokaisella työntekijällä on tietty työmäärä jonka tarvitsee täyttyä suunnittelujakson aikana. Jokaisella työntekijällä on *kompetensseja*, jotka kertovat mitä töitä työntekijä osaa tehdä. Kompetenssina voi olla esimerkiksi jokin vaadittava ajokorttiluokka tai työturvallisuuskortti. Päivät ovat jaettu *työpäiviin* ja *vapaapäiviin*. Jokainen päivä on jaettu *jaksoihin*. *Jakso* on pienin ajanyksikkö joka määrittää aikataulun *rakeisuuden* eli sen kuinka tarkka aikataulusta tulee. *Vuoro* on yhtäjaksoinen kokoelma työtunteja. Vuorolle määritellään aloitusjakso sekä vuoron pituus, joka kertoo kuinka monta jaksoa siinä on. Työvuoro voi esimerkiksi olla kahdeksan jakson mittainen ja yksi jakso voi olla tunnin pituinen, jolloin työvuoron kesto on kahdeksan jaksoa. Vuorot ovat yleensä jaettuna eri *vuorotyyppeihin*, kuten aamu-, päivä- ja yövuoroihin. Jokainen vuoro koostuu *tehtävistä* ja *tauoista*. Koko jakson mittainen tehtävä tai tauko on *aktiiviteetti*. Yhtäjaksoinen kokoelma aktiiviteetteja, jossa tehdään vain yhtä työtä, kutsutaan *venymäksi*. Vuoro tai tehtävä voi vaatia työntekijää, jolla on yhtä tai useampaa kompetenssia. Työaikasuunnitelmaa yhdelle työntekijälle

suunnittelujakson ajalle kutsutaan *rosteriksi*. Rosteri on yhdistelmä vuoroja ja vapaapäiviä, jotka kattavat kiinteän aikajakson.

2.2 Vuorojen muodostaminen käsin

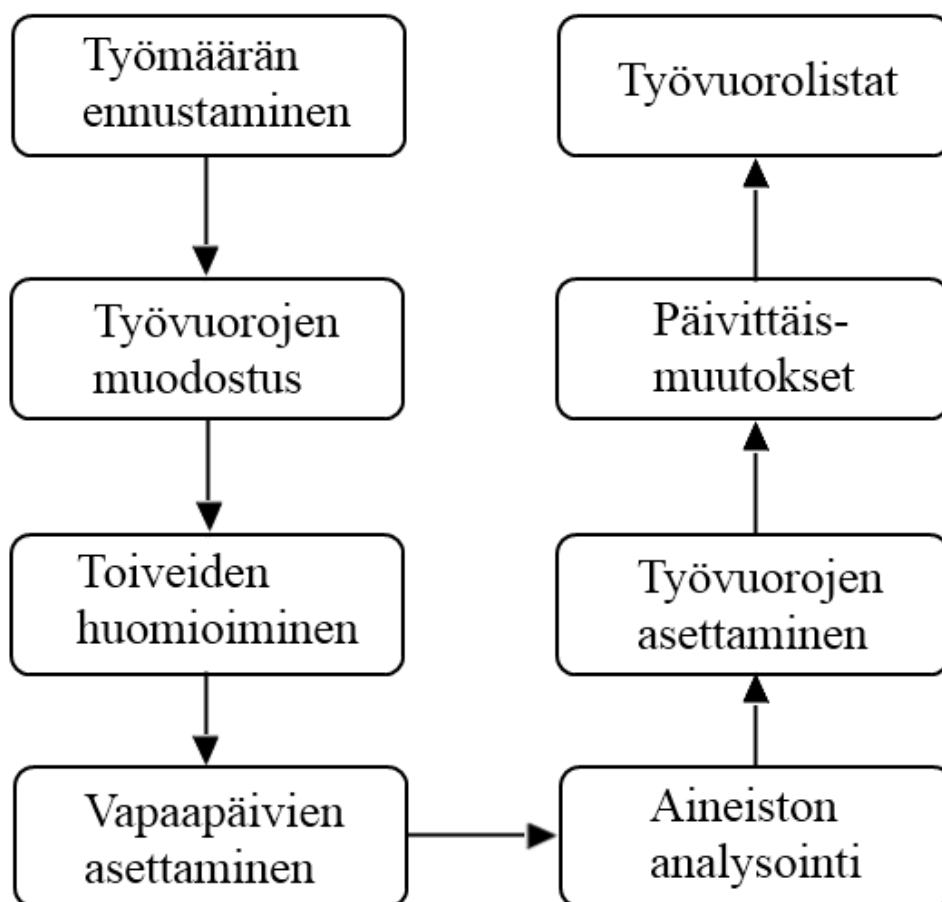
Työvuorojen muodostaminen käsin on erittäin haastavaa. Kun ottaa huomioon jokaisen työntekijän loma- ja vapaapäivät sekä toiveet ja kompetenssit, ongelman laajuuden voi ymmärtää helposti. Näiden lisäksi työehtosopimukset ja erilaiset lait ja säädökset laajentavat ongelmaa vielä entisestään. Käytännössä kukaan ihminen ei pysty muodostamaan parasta mahdollista vuorolistaa, koska erilaisia vaihtoehtoja on valtava määrä.

On kuitenkin mahdollista, että henkilö jolla on paljon kokemusta työvuorolistojen laatimisesta, pystyy toteuttamaan toimivan vuorolistan isollekin määrälle työntekijöitä. Esimerkiksi sairaalan apulaisosastonhoitaja teki hyväksyttävän vuorolistan 108:lla työntekijällä, kun taas pitkälle kehitetty algoritmi kehitti vuorolistan, joka oli hyväksyttävä 100:lla työntekijällä (Kyngäs, Nurmi, Ásgeirsson & Kyngäs 2012). Apulaisosastonhoitaja saa kuitenkin luultavasti laadittua työntekijöille ja työnantajalle miellyttävämmän työvuorolistan, sillä hän pystyy ottamaan huomioon esimerkiksi työntekijöiden väliset henkilökemiat (esimerkiksi ketä ovat ne henkilöt, joiden työskentely samassa tiimissä osoittautuu tehokkaimmaksi). Algoritmille on vaikea antaa säännöiksi esimerkiksi juuri henkilökemioihin liittyviä asioita.

Lisäksi työvuorolistojen laatiminen vie sitä tekevältä työntekijältä osan hänen työhönsä käyttämästään ajasta. Mitä suuremmalle joukolle työvuorot tulee tehdä, sitä enemmän työtunteja menetetään. Jos yritys tekisi työvuorot tietokoneavusteisesti, säästyisi useita miestyötunteja, sillä vuorojentekijä voisi käyttää tämänkin ajan omaan varsinaiseen työhönsä. Työntekijän vapauttaminen työvuorojen käsin tekemisestä johtaa suoraan kustannussäästöihin yritykselle.

2.3 Työvoiman hallinnan optimointi

Tämän luvun tarkoitus on selventää optimointiprosessia pääpiirteittäin. Kuvaan tässä työssä PEAST-algoritmin (Kyngäs 2011) toiminnan yleisellä tasolla. Algoritmi saa nimensä siinä käytetyistä menetelmistä: Population, Ejection, Annealing, Shuffling ja Tabu. Optimoinnilla tarkoitetaan jonkin numeerisen arvon minimoimista tai maksimoimista. Minimoitavia arvoja kutsutaan rajoitteiksi. Rajoitteita on kahdenlaisia, kovat ja pehmeät. Kovat rajoitteet on saatava pois, koska niillä määritellään ehdottomat säännöt. Näitä ovat esimerkiksi liian pitkä työpäivä työntekijälle tai se, että työntekijöitä pitää olla tarpeeksi työmäärään nähden tiettyinä aikana. Reaalimaailmassa kovia rajoitteita kuitenkin usein jää, mutta ne eivät välttämättä estä listan toteutumista. Pehmeät rajoitteet pyritään myös minimoimaan, mutta niiden olemassaolo ei estä ratkaisua olemasta hyväksyttävä. Pehmeisiin rajoitteisiin lasketaan esimerkiksi työntekijöiden toiveiden toteutumatta jääminen. Vastapainoisesti esimerkiksi työntekijöiden esittämien toiveiden täyttyminen pyritään maksimoimaan. Työvoiman työvuorolistojen tekeminen kuvataan kuten kuvassa 1.



Kuva 1: Työvoiman hallinnan optimoinnin prosessi

Ensimmäisenä pitää selvittää tarvittavan työvoiman määrä, eli montako työntekijää tarvitaan suoriutumaan kaikista töistä. Yksi tapa tämän tekemiseen on *työmäärää ennustamalla*, jos ei ole tiedossa tarvittavan työvoiman määrää. Usein ennustaminen tapahtuu simuloimalla. Jos työmäärä on tiedossa, tämä vaihe on mahdollista ohittaa. Luvussa kolme käsitellään ennustamista ja simulointia.

Kun on saatu tietoon tarvittava työmäärä, luodaan työvuorot (*työvuorojen muodostus*). Tässä vaiheessa määritellään vuoron rakenne sekä työt jotka on tehtävä vuoron aikana. Tarkoituksena on luoda vuorot siten, että työvoiman tarve saadaan tyydytettyä mahdollisimman hyvin. Jokaiselle vuorolle määritellään erikseen vaadittavat kompetenssit. Tämän vaiheen tarkoitus on luoda tiedot siitä, millaisia

töitä tarvitsee tehdä missäkin vuorossa minäkin päivänä. Tässä vaiheessa yhteenkään työhön ei sidota työntekijää.

Toiveiden huomioiminen on hyvin tärkeä osa työvuorolistojen tekemistä, sillä se lisää työntekijöiden työtyytyväisyyttä ja vähentää sairauspoissaoloja (Kellogg & Walczak 2007). Jokainen työntekijä määrittää omat toiveensa ja ne pyritään täyttää mahdollisimman hyvin. Tällaisia toiveita ovat esimerkiksi halu tiettyyn vapaapäivään tai halu tiettyyn vuoroon. Joissakin tapauksissa toiveena voi olla myös kahden (tai useamman) ihmisen halu samoihin vuoroihin, esimerkiksi yhteisen työmatkakuljetuksen takia.

Kun alustavat vuorot on tehty ja toiveet huomioitu, *asetetaan vapaapäivät* ja työpäivät parhaalla mahdollisella tavalla koko suunnittelujakson ajalle. Riippuen ongelmasta yksi tärkeimmistä optimoitavista kohteista on löytää tekijä jokaiselle työvuorolle, mutta pitää mahdollisimman vähän ylimääräisiä työntekijöitä töissä. Tässä vaiheessa tulee ottaa huomioon työehtosopimukset, työntekijän toiveet sekä muut mahdolliset rajoitteet. Esimerkkeinä muista mahdollisista rajoitteista toimii yksittäisten vapaapäivien minimointi sekä yksittäisten työpäivien minimointi vapaapäivien keskellä. *Vapaapäivien asettaminen* sisältää myös lomien ja erikoispäivien, kuten koulutusten, sijoittamisen.

Seuraavaksi tähän asti saatu *aineisto analysoidaan*, jossa tarkistetaan se, että pystyykö tällä työvoiman määrällä toteuttamaan tämän työvuorolistan. Tämä tarkoittaa sitä, että tarkistetaan muun muassa se, että löytyykö jokaiselle työlle tekijä, ilman että työntekijälle tulee päällekkäisiä töitä. Jos tähän astinen aineisto ei ole tyydyttävä, optimointiprosessi mahdollisesti aloitetaan alusta tai jostakin edellisestä vaiheesta.

Kun analysointi palauttaa tuloksen, joka kertoo aineiston toimivan, siirrytään *työvuorojen asettamiseen*. Työntekijät asetetaan mahdollisimman hyvin vuoroihin muun muassa siten, että työvuoro-toiveet pyritään maksimoimaan. Myös muut mahdolliset toiveet, esimerkiksi kaksi henkilöä aina samoihin vuoroihin, otetaan huomioon. Työehtosopimuksessa määritellyt ehdot ovat tässä vaiheessa hyvin tärkeitä ottaa huomioon. Tällä tarkoitetaan esimerkiksi sitä, että vuorot eivät ole liian

pitkiä ja vuorojen välinen lepotauko on vähintään ehdoissa määritellyn pituuden mittainen.

Päivittäisillä muutoksilla tehdään päiväkohtaiset muutokset, eli esimerkiksi sairauspoissaolojen aiheuttamat muutokset työvuoroihin tai muut tapaukset joissa työntekijä on estynyt tulemaan töihin. Nämä muutokset tehdään yleensä manuaalisesti.

3 TYÖVOIMATARPEEN ENNUSTAMINEN

Jotta työvuorojen optimointia pystyisi tekemään, tarvitsee tietää paljonko työtä on tehtävänä. Tuotantolinjamaisissa töissä, kuten tehtaat, tämä on helppoa; tehtävät ovat monesti hyvinkin tarkkaan määritelty ja tiedetään missä järjestyksessä ne pitää suorittaa. Asiakaspalvelumaisissa töissä, esimerkiksi yhteyskeskukset ja sairaalat, työn määrän määrittäminen ei ole läheskään niin suoraviivaista, sillä asiakkaita saapuu palveltavaksi eri määrät eri ajanhetkinä.

Eri määrät eri ajanhetkinä tarkoittaa sitä, että esimerkiksi maanantaina lounasaikaan voi tulla paljon enemmän asiakkaita palveltavaksi kuin perjantaina iltapäivällä. Eroja voi löytyä myös kuukausitasolla. Lähes itsestään selvänä esimerkkinä toimii kaupat, joissa joulukuu on vuoden kiireisin kuukausi.

3.1 Historiatiedot

Historiatiedoilla tarkoitetaan menneisyydessä tapahtuneita tapahtumia. Tapahtuma voi olla esimerkiksi kaupan tekemät myynnit tai asiakkaiden määrä tietyssä hetkenä. Se voi olla myös numeropalvelun tieto siitä, montako puhelua he ovat vastaanottaneet milloinkin.

Yritykseltä saatavat historiatiedot auttavat selvittämään eri aikoina tehtyjä työmääriä. Tiedon riittävä määrä takaa sen, että siitä pystytään tehdä tarkempia ennustuksia.

Ison ja tarkan tietomäärän ongelma voi kuitenkin olla puuttelliset tiedot. Pienen datamäärän kanssa isoksi ongelmaksi muodostuu epävarmuus. Jos tallella on vain yhden viikon ajalta tehtyjä työtunteja (tai esimerkiksi saapuneita puheluita), ei voida olla varmoja siitä, että tapahtuuko tietty pyrähdys puhelumäärissä joka maanantai lounasaikaan, vai oliko tämä vain poikkeus jota ei tapahtu koskaan muulloin. Vastaavasti jos tutkittavaa historiatietoa on paljon, poikkeukset pystytään huomioimaan selkeämmin, jolloin tarvittavan henkilöstömäärän ennustaminen on varmempaa.

3.2 Simulaattori

Työskentelen tämän kirjoittamisen aikana Satakunnan ammattikorkeakoulussa Laskennallisen älykkyyden tutkimuksessa, jossa eräs tutkimuskohteistamme oli yhteyskeskuksen työvuorojen optimointi. Yhteyskeskus hoitaa yritysten asiakassuhteita ulkoistetusti. Toiminta tapahtuu joko puheluin, tekstiviestein, sähköpostilla tai nykyään myös Internetin keskuskustelukanavia käyttäen. Yhteyskeskus toimii asiakaspalveluna, jolloin asiakaspalvelijat, joita kutsutaan agenteiksi, vastaavat keskukseen tuleviin puheluihin. Yhteyskeskus voi myös toimia olemassa olevien asiakassuhteiden ylläpitäjänä, jolloin agentit soittavat yhteyskeskuksen asiakasyrityksen asiakkaille.

Yhteyskeskustyö on siis asiakaspalvelutyötä, joten sen työmäärää ei voida yksiselitteisesti katsoa papereista. Tarvittavan henkilöstön määrittämiseksi tarvitsee simuloida yhteyskeskuksen toimintaa eri ajanhetkinä. Simulointi on mahdollista, koska yhteyskeskuksen historiatiedoista nähdään puheluiden saapumisintensiteetin jakauma sekä agenttien palvelunopeuden jakauma, joiden avulla simulointi on mahdollista toteuttaa realistisesti.

Keskus voi hoitaa useamman tyyppisiä puheluita, jotka kaikki vaativat eri osaamisia. Osa saapuvista puheluiden voi tulla esimerkiksi ruotsinkielisiltä asiakkailta, jolloin puheluun vastaavan agentin tarvitsee osata puhua ruotsia. Osa agenteista voi olla moniosaajia, jolloin he kykenevät palvelemaan useampia eri osaamisia vaativia puheluita. Tällaisella moniosaajalla on yleensä yksi vahva taito ja muut taidot ovat

heikompia. Esimerkiksi äidinkielenään ruotsia puhuva kykenee palvelemaan ruotsinkielisiä puheluita oikein hyvin, kun taas suomenkieliset puhelut eivät välttämättä hoidu yhtä nopeasti. Tämä ero on tärkeä havaita simulointia suorittaessa. Agenttien kyvyn palvella tiettyjä osaamisia vaativia puheluita määrittää palvelunopeus. Palvelunopeus kertoo sen, että millä nopeudella keskimäärin agentti palvelee eri osaamisen vaativia puheluita. Palvelunopeudet luetaan historiatiedoista. Simulaattori käsittelee eri osaamisia hallitsevia agenteja jakamalla ne omiin ryhmiin.

3.2.1 Asetukset

Käyttäjä määrittää ohjelman asetustiedostoon muun muassa halutut agenttiryhmät ja eri puhelutyypit sekä tavan jolla oikea agentti valitaan vastaamaan. Agenttiryhmistä määritellään agenttien minimi- ja maksimimäärät sekä palvelunopeus eli se, että kuinka nopeasti yksi agentti keskimäärin kykenee hoitamaan minkäkin tyyppisen puhelun. Jokaiselle ryhmälle määritellään yksittäisen agentin kustannus. Useaa eri osaamista hallitsevan agentin kustannus on isompi kuin vain yhden osaamisen agentin. Esimerkiksi ryhmä C voi koostua moniosaajista, jolloin sen ryhmän yksittäisen agentin kustannukseksi laitetaan 0,9 yksikköä. Tällöin ryhmän A agentin (joka hallitsee vain yhtä osaamista) kustannukseksi asetetaan 0,7 yksikköä. Moniosaajan isompi kustannus selittyy laajemmalla osaamisella.

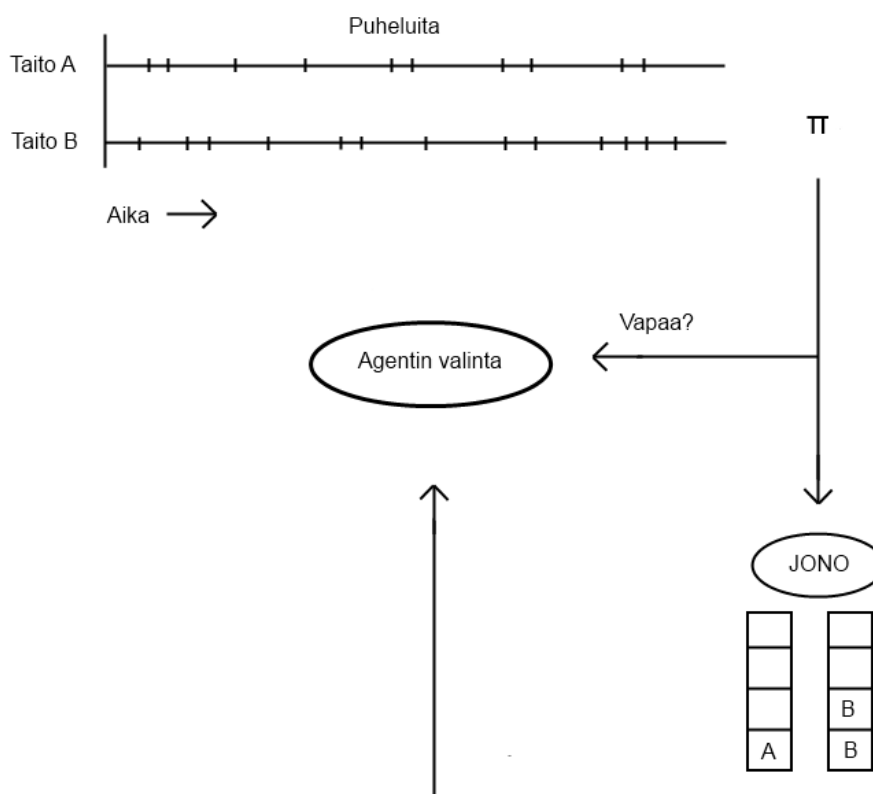
Puheluryhmistä kerrotaan historiatiedoista saatu keskiarvoinen saapumisväli kunkin tyyppin puheluille. Jokaiselle puheluryhmälle määritellään myös niin sanottu reitityspolitiikka, joka kertoo missä järjestyksessä puhelua tarjotaan vastattavaksi millekin agenttiryhmälle. Esimerkiksi reitityspolitiikka ”S1,S3,S2” tarjoaa ensimmäiseksi puhelua agenttiryhmälle S1 ja jos ko. ryhmässä ei ole vapaata agenttia, tarjotaan puhelua vastattavaksi ryhmälle S3 ja niin edelleen. Ideana reitityspolitiikassa on se, että ensimmäisenä halutaan tarjota puhelua vastattavaksi agentille joka nopeiten kykenee puhelun hoitamaan.

Simuloinnin kannalta asetustiedostossa olennaisena määritellään myös haluttu simulointiaika minuutteina sekä se, kuinka monta kertaa jokainen agenttikombinaatio testataan. Pääsääntöisesti kombinaatiot halutaan testata useammin kuin kerran, jotta saataisiin keskiarvosta realistisempi tieto kombinaation kyvystä suoriutua simulointijaksosta.

Sopivan agentin valintaa varten simulaattori osaa viisi eri vaihtoehtoa joista yksi valitaan asetuksissa. Ensimmäisenä on EAD (Expert Agent Distribution) jossa puhelu pyritään ohjaamaan agentille joka on pätevin hoitamaan kyseessä olevan tyyppin puheluita. Toisena on LC (Least Cost) jolla puhelu pyritään ohjaamaan agentille joka on edullisin niistä, jotka kykenevät hoitamaan puhelun. Kolmas vaihtoehto on LS (Least Skills) jossa puhelu pyritään ohjaamaan agentille joka pystyy hoitamaan vain ja ainoastaan tämän puhelun tyyppisiä puheluita. Tällä pyritään varmistamaan se, että puhelu palveltaisiin mahdollisimman nopeasti. Neljäs vaihtoehto on LOA (Least Occupied Agent) jossa valitaan agentti jonka käyttöaste on pienin. Tällä pyritään tasaamaan agenttien työmäärää. Viides ja viimeinen vaihtoehto on UCD (Uniform Call Distribution) jossa valitaan agentti joka on ollut vapaana kaikista pisimmän ajan.

3.2.2 Toimintalogiikka

Sovelluksen toiminta on peruslogiikaltaan hyvin yksinkertainen. Visualisointi toiminnasta nähdään kuvassa 2. Asetustiedoston lukemisen jälkeen ohjelma luo halutulle aikajaksolle puheluita listaan asetuksissa määriteltyjen ehtojen mukaisesti. Puheluille luodaan siis soittoaika, jonka jälkeen lista järjestetään aikaisimmasta ajasta alkaen. Seuraavaksi ohjelma tekee kaikki agenttikombinaatiot asetuksissa määriteltyjen minimi- ja maksimimäärien mukaan. Tämän jälkeen ohjelma alkaa ”soittamaan” puheluita, jolloin katsotaan onko sopivia agenteja vapaana vastaamaan niihin. Jos puheluun löytyy vastaaja, sille asetetaan kestoaja. Kesto saadaan agentin kyvystä palvella tämän taidon vaativia puheluita. Jos ei löydy sopivaa vapaata agenttia, puhelu asetetaan jonotukseen. Jonotuksessa olevilla puheluilla on suurempi prioriteetti tulla vastatuksi kuin uusilla puheluilla. Osa puheluista poistuu jonotuksesta tietyn ajan kuluttua.



Kuva 2: Simulaation toimintalogiikka

Kun määritelty aikajakso on kulunut, eli simulointi on suoritettu, lasketaan montako puhelua yhteensä tuli ja moneenko näistä pystyttiin vastaamaan määritellyssä ajassa. Saaduista tiedoista lasketaan palvelutaso. Palvelutaso on arvo, minkä yhteyskeskus on luvannut toteuttaa asiakkaillensa. Sitä kutsutaan myös palvelulupaukseksi. Se määritellään esimerkiksi seuraavasti: ”80 prosenttiin puheluita tulee vastata siten, että ne jonottavat enintään 20 sekuntia”. Palvelutaso tallennetaan muistiin myöhempää tarkastelua varten. Jos kombinaatioita on vielä jäljellä, aloitetaan ”soittaminen” alusta uudella agenttimäärällä.

Kun kaikki kombinaatiot on käyty läpi, tarkistetaan tulokset. Simulaattori laskee jokaisen yhdistelmän yhteishinnan agenttien määrästä ja yksittäisen agentin hinnasta. Tämän jälkeen simulaattori etsii sen yhdistelmän, jonka palvelutaso on se mitä vaaditaan, ja jos useampi kombinaatio pääsee samaan lukemaan, valitaan niistä yhteishinnaltaan halvin.

3.2.3 Tuloste-esimerkki

Seuraavaksi tarkastellaan lyhyesti simulaattorin tuottamaa tulostetta. Simulaattori tuottaa välitulostuksia koko suorituksensa aikana. Välitulosteiden intervalli on säädettävissä koodissa. Esimerkissä tuloste tulee tuhannen kierroksen välein jokaisella permutaatiolla. Kuvassa kolme rivien ensimmäiset numerot kertovat rivinumeron selvyiden vuoksi. Kuva on vain viimeinen osa mahdollista pitkää tulostetta.

```

1. 1152. permutaatio: S1 3 S2 3 S3 3 S4 5 S5 5 S6 2
2. Kierros: 1000
3. Puheluita: 48
4. 1000 KIERROSTA KESTO: 0.083
5. A AVG: 98.72600816701268
6. B AVG: 97.2921082522295
7. C AVG: 94.41574688674827
8. PERM tmpCombi 96.81128776866348
9. loppu
10. Paras permutaatio yhdistetyllä avg:lla: 3 3 2 3 3 1 16.0 80.26729875390009

```

Kuva 3: Osa simulaattorin tulosteesta

Ensimmäiseltä riviltä selviää kuinka mones yhdistelmä agentteja on tarkastelussa. Samalta riviltä nähdään myös yhdistelmän rakenne, joka luetaan seuraavasti: ryhmässä S1 on kolme agenttia, ryhmässä S2 on kolme agenttia ja niin edelleen. Rivi kaksi kertoo minkä kierroksen tiedot tulostuksessa on näkyvillä. Riviltä kolme nähdään, että tuhannes kierros sisälsi 48 puhelua yhteensä. Rivit 5-7 kertovat A, B ja C osaamista vaativien puheluiden toteutuneet palvelutasot. Esimerkiksi A-osaamista vaativat puhelut palveltiin 98,7 prosentin palvelutasolla. Riville kahdeksan tulostuu tämän yhdistelmän jokaiselta kierrokselta saadun keskiarvoisen palvelutason. Rivi kymmenen on viimeinen rivi, jonka simulaattori tulostaa. Siitä nähdään mikä yhdistelmä agentteja toteutti edullisimmin palvelulupauksen. Esimerkin tapauksessa haluttu palvelutaso oli 80 prosenttia ja tämä yhdistelmä oli edullisin kokonaiskustannuksella 16,0.

4 TULEVAISUUDEN MAHDOLLISUUDET

Tässä luvussa esittelen omia pohdintojani niistä mahdollisuuksista ja hyödyistä, joita työvoiman hallinnan optimoinnilla on mahdollista saavuttaa tulevaisuudessa.

4.1 Integraatiot kaupallisiin sovelluksiin

Yritysten alkaessa tiedostaa työvoiman hallinnan optimoinnin hyötyjä, alkaa myös sovellustarjonnalle olemaan enemmän kysyntää. Tulevaisuudessa yrityksille mahdollistuu työvoiman hallintaan tarkoitettujen sovellusten saaminen muualtakin, kuin erilaisten tutkimusten tuotteena. Työvoiman hallintasovelluksia tullaan myymään yrityksille yksilöllisesti räätälöityinä. Esimerkiksi kuljetusalan yritykset voivat parantaa kustannustehokkuuttaan ostamalla juuri heille suunnitellun sovelluksen. Sovellus luo yritykselle mahdollisimman tehokkaat reitit ajojen suorittamiseen ja samalla järjestää työntekijät sopiviin vuoroihin. Yhteyskeskuksissa vuorojen suunnittelutyö vastaavasti helpottuu heille suunnitellun ohjelmiston osatessa ennustaa tulevien viikkojen puhelumäärät entistä tarkemmin.

4.2 Tyytyväisyys

Työtyytyväisyys on yksi työntekijöiden kannalta olennaisimpia asioita, joihin työvoiman hallinnan optimointi tulee tulevaisuudessa vaikuttamaan. Työntekijöiden päästessä itse asettamaan toiveita, joiden toteutuminen on entistä todennäköisempää, työn tekemisen mielekkyys kasvaa (Kellogg & Walczak 2007). Optimointisovellusta käytettäessä työnantajan rooli muuttuu työntekijöiden silmissä: työvuorojen lopullinen muoto ei enää henkilöidy työnantajaan, vaan sovellus suorittaa vuorojen luomiseen liittyvät valinnat. Näin ollen asenne työntekoon muuttuu ja työn tekemisestä tulee tehokkaampaa. Työtyytyväisyys vaikuttaa myös suoraan henkilöstön vaihtuvuuteen: ihmiset viihtyvät työssä, joka on mielekästä. Tässä tapauksessa yritys säästää uusien työntekijöiden kouluttamiseen kuluvan ajan varsinaiseen työhön.

Työntekijöiden määrän ollessa oikein suhteutettu työmäärään nähden, muuttuu myös asiakkaiden palveleminen tehokkaammaksi. Palvelutahti saattaa esimerkiksi nopeutua, jolloin turha jonottaminen vähenee. Yhteyskeskuksessa jonottaminen on kaikille tuttua ja sen väheneminen vaikuttaa suoraan asiakastytyvyyteen. Asiakkailta on tapana palata kauppaan tai käyttää samaa palvelua johon he ovat aiemmin olleet tyytyväisiä.

4.3 Säästöt

Löytämällä optimaalisen henkilöstömäärän yritys pääsee eroon yli- ja alimiehityksestä. Ylimiehitys on tilanne, jossa työntekijöitä on tarpeeseen nähden töissä liian monta. Vastaavasti alimiehitys on tilanne, jossa työntekijöitä on tarpeeseen nähden liian vähän. Yritys saa aikaan huomattavia säästöjä palkkakustannuksissa, kun palkkaa ei tarvitse maksaa ylimääräisille henkilöille. Myös alimiehityksestä aiheutuu yritykselle ylimääräisiä palkkakuluja, kun kesken päivän töihin kutsutulle työntekijälle joudutaan maksamaan ylimääräisiä korvauksia esimerkiksi menetetyistä lomapäivistä. Työvoiman hallinnan optimointisovellusta käyttämällä on mahdollista eliminoida sekä yli- että alimiehitystilanteet.

Työvuorojen optimointi vaikuttaa myös suoraan sairauspoissaoloihin. Työntekijöiden päästessä tekemään heille itselleen mieleisiä vuoroja ja töitä, työn aiheuttama stressi vähenee. Töissä on myös luultavasti mukavampi olla kuin aiemmin. Sairauspoissaolojen vähentyminen tarkoittaa säästöjä yritykselle, kun palkkaa ei tarvitse maksaa sairaana olevan työntekijän lisäksi myös sijaiselle.

VIITTEET

1. Garey & Johnson. 1979. Computers and Intractability. A Guide to the Theory of NP-Completeness.
2. H.C. Lau. 1996. On the Complexity of Manpower Shift Scheduling. Computers and Operations Research 23(1), pp. 93-102.
3. Numeron WFM. Viitattu 12.12.2012. <http://www.numeron.com/numeron-wfm/>
4. Ortec Harmony. Viitattu 17.12.2012. http://www.ortec.com/products/ortec_workforce_scheduling.aspx
5. Kyngäs, N., Nurmi, K., Ásgeirsson, E. I. & Kyngäs, J. 2012. Using the PEAST Algorithm to Roster Nurses in an Intensive-Care Unit in a Finnish Hospital.
6. Kellogg & Walczak. 2007. Nurse Scheduling: From Academia to Implementation or Not?
7. Kyngäs, J., Nurmi, K. & Kyngäs, N. 2013. Solving the person-based multitask shift generation problem with breaks.
8. Kyngäs, J. 2011. Solving Challenging Real-World Problems. Väitöskirja. Turku: Turun Yliopisto.