

Kollektív döntéseken alapuló alkalmazások megvalósíthatóságának algorithmikus és implementációs kérdései

c. TDK dolgozat

Készítette:

Tekán Dávid
Mérnök-informatikus BSc

Témavezető: Dr. Oláh András



Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar
2012

Absztrakt

Az infokommunikációs technológiák gyorsuló fejlődése és az azokon megvalósuló szolgáltatások iránti fokozódó igény egymásra hatásának eredménye, hogy a felhasználók valós időben óriási mennyiségű információhoz férhetnek hozzá, ugyanakkor a technológia lehetőséget biztosít arra is, hogy ők maguk is információt termeljenek, és közvetítsenek, valamint az őket körülvevő környezetüket preferenciájuk szerint közvetlenül formálják.

A közös érdekeltségek formálta (munkahelyi, egyetemi, lakó-, etc) közösségben számos kérdésben az individuális döntéseken alapuló kollektív döntést kell hozni. Ez támogatható infokommunikációs eszközökkel, ahol a kérdések meghirdetése, a szavazatok összegyűjtése és a kollektív döntés meghozatala (döntésméleti módszerekkel), majd annak kihirdetése a közvetlen mérnöki feladat. Az infokommunikációs technológiákon belül az okostelefonok fejlődése az elmúlt években robbanásszerű növekedésnek indult, új korszakot hozva a mobil infokommunikációs alkalmazások területén. Az okostelefon a felhasználó számára állandó hálózati kapcsolatot biztosít, melyen keresztül bárhol és bármikor közösségi szolgáltatásokhoz férhet hozzá, valamint aktív formálójává válhat az őt körülvevő tartalomnak, szolgáltatásnak, közösségnek, vagy lokális környezetének. A TDK dolgozat fókuszában a kollektív döntéseken alapuló rendszerek megvalósíthatóságának algoritmikus és implementációs kérdései állnak.

A dolgozat a kapcsolódó elméleti és technológiai háttér bemutatását követően két kollektív döntést támogató alkalmazáson keresztül (Timeocracy és Tempocracy) mutatja be a szavazat alapú rendszerek általános architektúráját valamint célalkalmazásként speciális algoritmikus és implementációs kihívásait. Az implementált alkalmazás valós környezetben végzett tesztelése során hozzáférhető szavazateredmények kiértékelését a dolgozat szintén tartalmazza.

Abstract

As a result of the interaction between the rapid improvement of the infocommunication technologies and the high demand for services based on these technologies users have gained access to an enormous quantity of information in real time. Moreover, the technology also allows them to produce and broadcast information, as well as to form their environment directly, based on their preferences.

In a community defined by common interests (workplace, university, home etc.), most of the collective decisions have to be based on individual decisions. This process can be supported by infocommunication devices, where the questions' publication, the collecting of the votes and making the collective decision is the engineer's task. Within the field of infocommunication technology, the development of the smartphones have achieved an explosive growth bringing the dawn of a new era in the world of mobile infocommunication applications. The constant network connection provided by smartphones does not only enable the user to have whenever and wherever access to the community services, but also to play an active part in shaping of the surrounding content, service, community or their local environment. This paper focuses on the implementation and algorithmic questions concerning the feasibility of the collective decision based systems.

Following the presentation of the related theoretical and technological background, the paper - using two collective decision support applications, Timeocracy and Tempocracy - demonstrates the general architecture of vote-based systems and the specific algorithmic and implementation challenges raised by certain applications. The analysis of the implemented application's test performance in a real-life environment is also included.

Tartalomjegyzék

Absztrakt.....	2
Abstract.....	3
Tartalomjegyzék.....	4
1. Bevezetés és motiváció	5
2. Kollektív döntés elméleti és gyakorlati megközelítése.....	7
2.1 Döntéshozatali folyamatok fejlődése és társadalmi relevanciája	7
2.2 A kollektív döntéselmélet algoritmikus háttere.....	10
2.2.1 Hagyományos döntéshozó algoritmusok	10
2.2.2 Fuzzy alapú döntéshozatal	12
2.2.3 Játékelmélet alapú döntéshozatal.....	18
2.3 Kurrens kutatás kihívásai.....	20
3. Kurrens infokommunikációs technológiák és elérhető megoldások.....	21
3.1 Pervazív számítástechnika és az okos platformok	21
3.2 Elérhető kollektív döntéstámogató alkalmazások	25
4. Új szavazat alapú kollektív döntéstámogató rendszer	28
4.1 Alkalmazás scenáriók	28
4.2 Univerzális architektúra	31
4.3 A felhasználható algoritmikus eszköztár	33
5. Konkrét alkalmazások implementációja	38
5.1 Demokratikus hőmérsékletbeállítás – Tempocracy	38
5.2 Szavazat alapú időpontegyeztetés – Timeocracy	40
5.3 Smart Grid.....	42
6. Teljesítőképesség analízis a Timeocracy alkalmazásban.....	44
7. Összefoglalás és továbblépési lehetőség.....	48
8. Köszönetnyilvánítás.....	49
Irodalomjegyzék.....	50

1. Bevezetés és motiváció

A XXI. század egyik társadalomformáló jelensége az információ túltermelés illetve ezek akadálymentes és azonnali hozzáférhetősége, amelyet az infokommunikációs technológiák (ICT) rohamléptékű fejlődésének köszönhető. Az ICT alapú alkalmazások lerövidített fejlesztési idő alatt érik el a felhasználókat, és az élet minden területén meghatározó és nélkülözhetetlen tényezővé válnak. A megjelenő alkalmazások területén az egyik legerősebb trend, hogy közösségek számára kínálnak szolgáltatásokat, amely közösségeket egy kollektív érdekeltség alakít ki (munkahelyi, egyetemi, lakóközösség, baráti közösség, etc).

Egy az érdekeltségek formálta közösségben számos kérdésben az individuális döntéseken alapuló kollektív döntést kell hozni. Ez különösen fontos akkor, hogyha az egyéni döntések jelentősen eltérnek egymástól, azonban a közösség érdeke, hogy egy, a tagok által elfogadott, konszenzusra jussanak.

A kollektív döntések támogathatók infokommunikációs eszközökkel, ahol a kérdések meghirdetése, a szavazatok összegyűjtése és a kollektív döntés meghozatala (döntésméleti módszerekkel), majd annak kihirdetése a közvetlen mérnöki feladat. Ilyen alkalmazások bizonyos részterületeken jelenleg is hozzáférhetőek, ugyanakkor nincs olyan általános architektúra (még a szakirodalomban sem), amely a célzottan a lehetséges scenáriók mindegyikére alkalmazható lenne. Mindemelllett az elérhető alkalmazások nem használják fel a kollektív döntésmélet algoritmikus eredményeit. Éppen ezért a TDK dolgozat fókuszában a kollektív döntéseken alapuló rendszerek megvalósíthatóságának algoritmikus és implementációs kérdései állnak. A dolgozat célkitűzései a következők:

- a kapcsolódó elméleti és technológiai háttér elemző bemutatása;
- általános architektúra kifejlesztése;
- az elérhető egyszerű döntéshozó algoritmusok bemutatása;
- a matematikatudomány egyéb területéről származó eljárások, módszerek alkalmazása döntéshozásra
- döntéshozó rendszerek implementációja a kidolgozott általános architektúrán (Timeocracy, Tempocracy);
- teljesítőképesség analízis a Timeocracy hatékonyságának igazolására.

A dolgozatban bemutatott eredmények a fenti célkitűzések szerint definiált mérföldkőnek tekinthetőek, ugyanakkor számtalan új kutatási utat nyitnak meg, amelyeken szándékaim szerint a későbbiekben elindulni kívánok.

A dolgozat felépítése a következő:

- **2. fejezet (Kollektív döntés elméleti és gyakorlati megközelítése):** röviden bemutatja a kollektív döntéselmélet fejlődésének történetét, a kollektív döntés hagyományos, fuzzy alapú és játékelméleti eszköztárát, valamint a terület kurrens kutatási kihívásait;
- **3. fejezet (Kurrens infokommunikációs technológiák és elérhető megoldások):** ismerteti a dolgozat szempontjából releváns infokommunikációs technológiák két legdinamikusabban fejlődő területének (mobil- és internet technológia) jellemzőit, valamint az elérhető kollektív döntéseket támogató alkalmazásokat;
- **4. fejezet (Új szavazat alapú kollektív döntéstámogató rendszer):** bemutatja a különböző alkalmazás scenáriókat, az univerzális architektúrát, valamint a felhasználható algoritmikus trükköket;
- **5. fejezet (Konkrét alkalmazások implementációja):** ismerteti a két elkészült alkalmazást, bemutatja a szavazás, döntéshozás folyamatát mindkét esetben, kiemelve a speciális algoritmikus részeket, valamint kiemeli a kollektív döntéselmélet használhatóságának vizsgálatának fontosságát a Smart Grid rendszerek esetében;
- **6. fejezet (A kollektív döntési algoritmusok valós teljesítőképesség analízise a Timeocracy alkalmazásban):** bemutatja a performanciateszt folyamatát, a közben felmerült nehézségeket, és az eredményét;
- és végül a **7. fejezet (Összefoglalás és továbblépési lehetőségek):** konkludálja a dolgozat eredményit és kijelöli a továbbfejlesztési irányokat.

2. Kollektív döntés elméleti és gyakorlati megközelítése

A kollektív döntések meghozatalának módszerei folyamatosan fejlődtek, változtak, valamint ma már nem feltétlenül ugyanolyan súlyú problémákra kell megoldást találni. A kollektív döntéselmélet hagyományos, illetve speciálisabb algoritmusainak áttekintése után, a fejezet a kurrens kutatás kihívásait, lehetőségeit tárgyalja.

2.1 Döntéshozatali folyamatok fejlődése és társadalmi relevanciája

Évezredekkel ezelőtt füstök és álmok értelmezése irányította az akkori ember döntéseit, a szavazat alapú demokratikus döntéshozatal kezdetleges formája először a görögöknél jelent meg Kr. e. XV. században, ez nem volt más, mint a cserépszavazás. Az ókori Athénban évente megrendezett szavazás, amelyen a polisz lakossága dönthetett arról, hogy akar-e valakit száműzni a városból. Amennyiben legalább hatezer polgár szavazott (megfestett cserépre írt névvel), akkor összeszámolták a szavazatokat, és a legtöbb voksot kapó polgárt száműzték 10 évre a városból, ha a 10 év lejártá előtt megpróbált visszatérni, halálbüntetés várt rá. A kicsit szofisztikáltabb módszerek csak a XVII. századtól kezdve alakultak ki [1].

René Descartes, a francia filozófus szerint a tudás megszerzésében fontosabbak az érvek, mint a tapasztalat, előtérbe kerül a racionalitás. Azonban kortársa, **Pascal** szerint a rosszul döntés következményei fontosabbak, mint maga a bekövetkezési valószínűség, ezért megfelelő figyelmet kell fordítani az esemény következményeinek vizsgálatára is, nem csak a bekövetkező valószínűség vizsgálatában. **Blaise Pascal** és **Pierre de Fermat** együtt alkotta meg az események bekövetkezési valószínűségének kiszámításának elméletét és alaptételeit. Azonban egészen 1738-ig kellett várni, míg **Bernoulli** megalkotta a kockázatomérés elméletét megfigyelve, abból a nézőpontból, hogy az egyedek az egyes lehetséges kimeneteket mennyire várják, illetve mennyire tartanak tőle. A létrehozott valószínűségi változót róla nevezték el.

Az **Abraham de Moivre** által leírt haranggörbét tanulmányozta **Carl Friedrich Gauss** és létrehozott egy módszertant a véletlen események bekövetkezésének kezelésére. Ez később alapvető fontosságú lett, gyakorlatilag a természetben kialakult véletlen jelenségek normális eloszlásúak.

Következő nagy ugrást **Francis Galton** jelentette, aki felfedezte, hogy bár a véletlen folyamatok durván eltérnek az átlagtól, de idővel mégis az átlaghoz tartanak, ezzel megalkotva

a sztochasztikus konvergencia fogalmát. Az átlaghoz való tartásról szóló elmélete nagy hatással volt a részvény- és üzleti matematikára.

A valószínűségszámítás fogalomrendszere akkor még eléggé hiányos volt, **Frank Knight** tett először különbséget a kockázat, amikor egy lehetséges kimenetel bekövetkezésének valószínűségét ismerjük, és a bizonytalanság, amikor egy lehetséges kimenetel bekövetkezésének valószínűségét nem ismerjük, között.

Ugyanakkor a döntéelmélet nem csak a valószínűségszámítás mellett fejlődött, hanem új irányt is vett a játékelmélet felé. **Neumann János** és **Oskar Morgenstern** a játékelméletről szóló könyvükben leírják a gazdasági alapú döntéshozatal matematikai alapjait, mint sok más tudóssal előttük, azt a nézőpontot feltételezték, hogy a döntéshozó mindig racionális és konzisztens gondolkodású. Itt olyan döntéshelyzetek kerülnek elsősorban előtérbe, amelyekben a döntéshozatal függ más emberek előre nem látható döntésétől. Világos, hogy ezt az elvet nem lehet sokáig tartani, **Herbert Simon** kifejtette, hogy elég gyakori az olyan eset, amikor az információszerzés jelentős költségei miatt a döntéshozók hajlandóak – korlátozott racionalitással – közel jó döntések meghozatalára is.

A döntéshozatali eszközök lassan elérték teljesítőképességük csúcsára, így a számítógép megjelenésével, és villámgyors elterjedésével arra jutottak, mint az 1950-es években MIT-n végzett kutatások addig jutottak, hogy elkezdték megalkotni az első számítógép alapú döntéstámogató eszközöket. Ezzel párhuzamosan **Kenneth Arrow** bebizonyította, hogy nem lehet olyan közösségi döntési szabályrendszert alkotni, amelynek segítségével a közösség összes követelményét ki lehet elégíteni.

A XX. század második felétől kezdve a döntéelmélet fejlődésében a tőzsde, és az üzleti élet robbanásszerű térnyerése játszotta a fő szerepet. Ekkor **Harry Markowitz** bemutatta, hogy matematikai módszerekkel lehetséges meghatározni, hogyan válasszunk a különböző tőzsdei portfólióink közül, hogy a profitunk egyenletes legyen.

Az **Albert Humphrey** által megalkotott SWOT analízis segítségével a gyors döntéshozatal olyan helyzetekben is elérhető vált, ahol kevés idő állt rendelkezésre, valamint a körülmények is összetettek voltak.

Irvine Janis vezeti be a csoportgondolat fogalmát, miszerint a közös megegyezéssel meghozott döntés fontosabb, mint az esetlegesen legjobb kimenetel, ezután a csoportos döntéseknél lehetőség nyílt arra, hogy vizsgálják az egyes résztvevők preferenciáját is, ne csak a konkrét szavazatok számítsanak.

Amos Tversky és **Daniel Kahneman** bizonyították, hogy az eddig ismert és használt racionális gazdasági modell nem veszi figyelembe az emberek döntéshozatalát abban az esetben, amikor szembesülnek az élet bizonytalanságaival.

A következőkben tekintsük át egy általános döntés meghozatalának folyamatát. **James Rest** a következő nyolc lépést határozta meg:

- **környezet kialakítása:** alapvető fontosságú, hogy a folyamat legelején definiáljuk azokat a kapcsolatokat, korlátokat és folyamatokat, amelyek mentén vizsgáljuk a problémát. Nem feltétlenül ér véget a megfigyelés kezdetével, folytatódhat tovább is, párhuzamosan a döntési folyamattal;
- **megfigyelés:** a probléma megfogalmazása;
- **értelmezés:** meg kell határozni a lehetséges megoldásokat a problémára, egyúttal kifejtve a motivációkat is;
- **vizsgálódás:** végignézve a lehetőségeket megállapítjuk, hogy melyik az elfogadhatóbb;
- **motiváció:** meg kell vizsgálni a kötelezettségvállalásokat, amik eltérhetnek a teendőktől, ezek után pedig ezeket a vállalásokat priorizálni, és az egyéni, szervezeti érdekek fölé helyezni;
- **akció:** döntés meghatározása az elérhető információk alapján, figyelembe véve a különböző zavaró körülményeket, akadályokat;
- **cselekvés közben:** döntés újragondolása, szükség esetén módosítása;
- **cselekvés után:** döntés és a cselekvés értékelése, konklúziók levonása.

Vannak más elképzelések a döntés optimális meghozatalának lépéseiről a szakirodalomban, ezek közül az egyik **Dr. Pam Brown** nevéhez fűződik:

- jelöljük ki a célt, és az eredményt;
- gyűjtsük össze az adatokat;
- készítsünk megoldási alternatívákat;
- érveljünk ezek mellett, illetve ellen;
- hozzuk meg a döntést;
- azonnal tegyünk lépéseket, hogy életbe lépjen a döntés;
- vonjunk le következtetéseket, és szükség esetén vegyük figyelembe a további döntéseknél.

Herbert Simon behatóan foglalkozott a döntéselmélettel, és tanulmányai/munkái szerint [2] a döntéshozás három fő fázisból áll:

- Keressük meg a döntéshozási lehetőségeket. (Intelligencia)
- Válasszuk ki a lehetséges kimeneteket. (Tervezés)
- Válasszunk ezek közül. (Választás)

A leghíresebb nem szekvenciális modell **Mintzberg, Raisinghani, and Théorêt** [2] nevéhez fűződik, akik megtartották Simon három pillérét, igaz átnevezve, de nincs közöttük egyszerű szekvenciális kapcsolat.

- **Azonosítás:** Problémák és lehetőségek felismerése, ezek diagnosztizálása, elemzése;
- **Fejlesztés:** Meglévő megoldások keresése, valamint módszerek felkutatása, amivel új megoldásokat lehet létrehozni, valamint a meglévőket továbbfejleszteni;
- **Választás:** A nyilvánvalóan csak szuboptimális megoldások elvetése, majd egy megoldás választása, végül e megoldás jóváhagyása.

A döntéshozást tekinthetjük leegyszerűsítve lényegében az összes lehetséges scenárió közül választásnak. Minden döntéshozó eljárásnak van végső választása [3]. Fontos megjegyezni, hogy a legtöbb döntést mégis öntudatlanul hozzuk meg. Jim Nightingale szerint „we simply decide without thinking much about the decision process.” [4].

Gyakori azonban az is, hogy egy-egy döntés előtt mérlegeljük ugyan az érveket, és az ellenérveket, de a mai világban, a naponta felmerülő döntések tekintetében nem mindig van idő erre a hosszú folyamatra, reflexszerűen dönt az ember. Így nem csoda, hogy a modern ember, az internetes vásárlások során, hozzá hasonló, de ismeretlen emberek véleményének hatására hoz döntéseket. Ugyanakkor Malcolm Gladwell kifejti Blink című könyvében, hogy a megérzés által vezérelt spontán döntések néha jobbak, mint azok, amelyeket hosszú, ésszerű analízis után hoztak.

2.2 A kollektív döntéselmélet algoritmikus háttere

Számtalan algoritmus létezik, amely segítségével az egyéni szavazatokból kollektív döntést lehet hozni. Ebben a fejezetben a hagyományos, egyszerű algoritmusok bemutatása után, a fuzzy, majd a játékelmélet alapú módszerek ismertetése található.

2.2.1 Hagományos döntéshozó algoritmusok

Ebben a fejezetben a kollektív döntéshozás alapvető építőkövei kerülnek bemutatásra. Az ilyen egyszerű döntéshozó algoritmusok pont az egyszerűségük miatt különösen hasznosak. Könnyű implementálhatóságuk miatt előszeretettel alkalmazzák ezeket kisebb rendszereknél, ahol még nincs szükség a későbbiekben tárgyalt algoritmikus arzenálra:

- **Átlag:** Ezen döntési folyamat során a szavazók értékelik az összes lehetséges opciót (kimenetelt) egy skálán. Miután leadták szavazatukat, vesszük az egyes lehetséges kimeneteknek hozzárendelt értékek számtani átlagát, és a legmagasabb értéket elérő kimenetel lesz a győztes. Előnye, hogy rettentő gyorsan kiértékelhető, nem igényel külön algoritmikus eszköztárat. Hátránya viszont, hogy egyszerűsége miatt nem nagyon alakítható az alkalmazáshoz.
- **Súlyozott átlag:** A döntési kiértékelés hasonlóan indul, mint az előzőnél, azonban lehetőség van arra, hogy az egyes csoporttagok szavazatai más súllyal számítsanak bele a döntésbe. Vállalati közgyűléseken történő döntéshozás a legkézenfekvőbb mód e döntési módszer alkalmazására. Általában egy vállalatban belül, különösen részvénytársaság esetén, nem egyenlő részesedéssel rendelkeznek a tulajdonosok, így valamilyen módon „jobban” figyelembe kell venni azok szavazatát, akik a szervezet nagyobb része fölött rendelkeznek.
- **Többségi döntés:** Végtelenül egyszerű döntési mód. Mindegyik csoporttag egy opciót választhat, amelyik opció a legtöbb szavazatot kapta, az a győztes. Előnye ennek a módszernek, hogy nagyon gyorsan végrehajtható. Hátránya viszont, hogy egyáltalán nem biztos, hogy optimális döntést hoz. Tekintsük a következő példát: az A opcióra 5-en szavaznak, a B valamint a C opcióra 3-3 felhasználó. Ekkor az A opció lesz a győztes, a probléma viszont az, hogy nagyon könnyen előfordulhat az a szituáció, hogy azoknak, akik a C opcióra szavaztak megfelelne a B opció is. Ha ezt figyelembe vennénk, akkor viszont a B opció lenne a döntés.
- **Borda:** A Borda módszer pont erre a többségi döntés problémájára kínál egyfajta megoldást. Ebben az esetben a szavazás során ugyanis nem egy opciót választunk ki, hanem egy preferenciasorrendet alakítunk ki az opciókból, majd az egyes lehetséges kimeneteknél összeadjuk az értéket, és végül a legkisebb értéket kapó kimenetel lesz a döntés. Előnye, hogy bizonyos tekintetben igazságosabb, itt ugyanis egy sorrendet alakítunk ki, azaz nem csak egy opcióról alkotunk, és fejezzük ki véleményünket, hanem az összesről. Hátránya azonban az előzőekhez képest, hogy jóval komplikáltabb algoritmusra van szükség.
- **Condorcet:** Ennél a döntési módszernél az opciók egymással versengenek. Egy-egy ellen, csak a rájuk adott szavazatok, illetve, hogy az adott szavazó egymással szemben melyiket preferálná. A végeredmény az az opció lesz, amelyik a legtöbbször győzött ezekben a „párbajokban”. Előnye a módszernek, hogy igazságos, mivel az egyes opciók között nincs kedvezményezett, amelyik előnyt élvezne, hanem mindegyiket a rájuk leadott szavazatok alapján rangsorolja. Ugyanakkor hátránya, hogy a futásidő a lehetséges opciók számával négyzetesen arányos.

A következőben egy olyan példát találhatók, amelyen jól látszik, hogy egyáltalán nem mindegy, hogy melyik döntési módszert választja ki a szavazás felelőse a szavazatok kiértékeléséhez [5].

Egy kilenctagú csoportnak öt jelöltet kell választania. A következő táblázatban láthatók a preferenciasorrendek.

1. táblázat A szavazáskor előforduló precedenciasorrendek

4 szavazó	3 szavazó	2 szavazó
A	B	C
E	C	D
D	E	E
C	D	B
B	A	A

Mint az látható 4 szavazó választotta az AEDCB, 3 a BCEDA és 2 CDEBA sorrendet.

- **Átlag/súlyozott átlag** döntésről itt nem érdemes beszélni, mivel nem az opciókat értékelték a szavazók, hanem egy sorrendet állapítottak meg, így az opciókat csak egymáshoz viszonyítva tudjuk vizsgálni;
- **Többségi döntés** esetén az **A** jelölt lesz a kiválasztott, mivel őrá 4-en szavaztak, míg a B, C jelöltre, 3 illetve 2 szavazat érkezett;
- A **Condorcet** módszer alapján végzett kiértékelésnél a C lesz a győztes, mivel minden egymás elleni (páros) szavazásnál több szavazatot kap. (C – A : 5 – 4, C – B : 6 – 3, C – D : 5 – 4, C – E : 5 – 4);
- **Borda** döntés esetén a következő pozícióösszegeket kapjuk (súlyozva a szavazók számával): A(29), B(31), C(24), D(28), E(23). Ebből pedig már látható, hogy itt az **E** lesz a győztes.

Látható, hogy fenti kiértékelő algoritmusok mindegyik esetben más opciót választanak. Ez erősíti azt az állítást, hogy a kiértékelő algoritmust a szavazat típusának, és rendeltetésének megfelelően kell kiválasztani.

2.2.2 Fuzzy alapú döntéshozatal

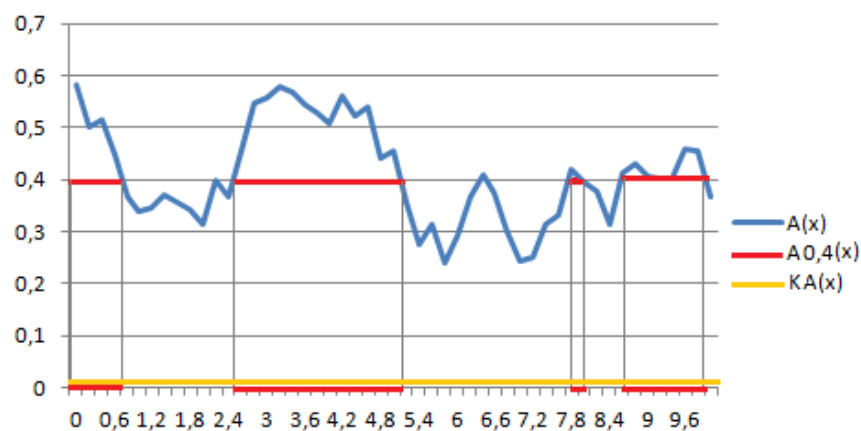
A fuzzy logika közvetlen előzményének Lukasiewicz többértékű logikája tekinthető [6]. Ez eredetileg háromértékű logika volt, később általánosították megszámlálhatatlanul végtelen

értékre. A tudományos életben meglehetősen vegyes fogadtatásban részesült, mivel sokan a valószínűségelmélet egy másik megfogalmazásaként értékelték, és mint olyan, feleslegesnek ítélték (ezt csak később, a fuzzy mértékelmélet pontosítása után tudták megcáfolni). Az arisztotelészi logika állításainak megkérdőjelezhetetlenségében hívók a „harmadik kizárása” elv miatt a fuzzy logikát eleve értelmetlennek minősítették. Azonban a negatív reakciók ellenére 1965-től kezdődően gyorsan növekedtek a fuzzy témájú publikációk, mind elméleti, mind konkrét alkalmazás témájúak [7].

Míg a standard halmazelméletben a halmazhoz tartozás mértéke (karakterisztikus függvénye) csak két értéket vehet fel (általában 0 – nem tartozik a halmazba, illetve 1 – beletartozik a halmazba), ezzel szemben a fuzzy halmazelmélet/logika esetében a karakterisztikus függvény egy előre meghatározott, rögzített tartományból (általában $[0,1]$ intervallumból) vehet fel értéket. Fontos, hogy a fuzzy nem kezelhető valószínűségként, itt nem a halmazba tartozás valószínűségéről van szó, hanem a mértékéről. Pont ez lesz fontos a döntéshozatalnál.

A fuzzy halmazok néhány jellemzőjét (amely eltér a hagyományos értelemben vett halmazokétól) fontos definiálni ($A(x)$ -szel jelöljük az A halmaz fuzzy karakterisztikus függvényét, X -szel pedig az alaphalmazt.):

- **α -vágat:** egy fuzzy halmaz α -vágata, A_α , minden $\alpha \in [0,1]$ értékre:
 $A_\alpha = \{x \mid A(x) \geq \alpha\}$, szigorú α -vágatról beszélünk, hogyha a definícióban szereplő egyenlőséget nem engedjük meg ($A_{\alpha+}$ -szal jelöljük).
- **szinthalmaz:** egy A halmaz szinthalmazán az összes egymástól különböző α -vágatának halmazát értjük: $\Lambda(A) = \{\alpha \mid \exists x \in X : A(x) = \alpha\}$
- **tartó:** az A fuzzy halmaz tartója a 0-nál nagyobb tagsági értékű pontjainak összessége.
- **mag:** az alaphalmaz 1 tagsági értékkel rendelkező elemeinek összessége.



1. ábra Példa fuzzy halmaztulajdonságokra

Az előző ábrán (1. ábra) látható egy példa egy fuzzy halmazra, és pár tulajdonságára. Kék színnel jelölt az alap fuzzy halmaz karakterisztikus függvény, pirossal a 0,4-es α -vágathoz tartozó elemek, valamint sárgával a tartóhoz tartozó értékek (mivel minden értékhez a karakterisztikus függvény értéke nagyobb, mint 0, ezért a teljes halmaz eleme a tartónak). Hasonlóan belátható, hogy a mag üreshalmaz lesz.

A halmaz jelölése is kicsit másképp történik: nem elég az elemek egyszerű felsorolása, valamilyen formában jelölni kell a halmazba tartozás mértékét. Ezt a következőképpen tesszük meg:

$$A = a_1/x_1 + a_2/x_2 + \dots + a_n/x_n$$

A három alap halmazművelet (Zedah féle) a következőképpen értelmezhető fuzzy halmazoknál:

- **komplement:** egy X alaphalmazon értelmezett A fuzzy halmaz Zedah féle komplemente $\bar{A}$, hogyha $\bar{A} = 1 - A(x)$. X azon értékeit, amelyre fennáll, hogy $\bar{A}(x) = A(x)$, egyensúlyi pontnak nevezünk.
- **metszet:** A, B fuzzy halmaz metszetén azt a fuzzy halmazt értjük, melynek tagsági függvénye a következőképpen írható: $(A \cap B)(x) = \min [A(x), B(x)]$
- **unió:** hasonlóan az előzőhöz, az unió tagsági függvénye: $(A \cup B)(x) = \max [A(x), B(x)]$

Ezek egyszerűen kiterjeszthetők több halmazra is.

Többféle integrált is lehet definiálni fuzzy halmazon, a legegyszerűbb a Choquet integrált. Legyen $A(x)$ egy fuzzy mérték X -en. Ekkor egy $f: X \rightarrow [0,1]$ függvény Choquet integrálja definíció szerint: $C_{A(x)}(f(x_1), \dots, f(x_n)) = \sum_{i=1}^n (f(x_i) - f(x_{i-1}))A(x_{(i)})$,

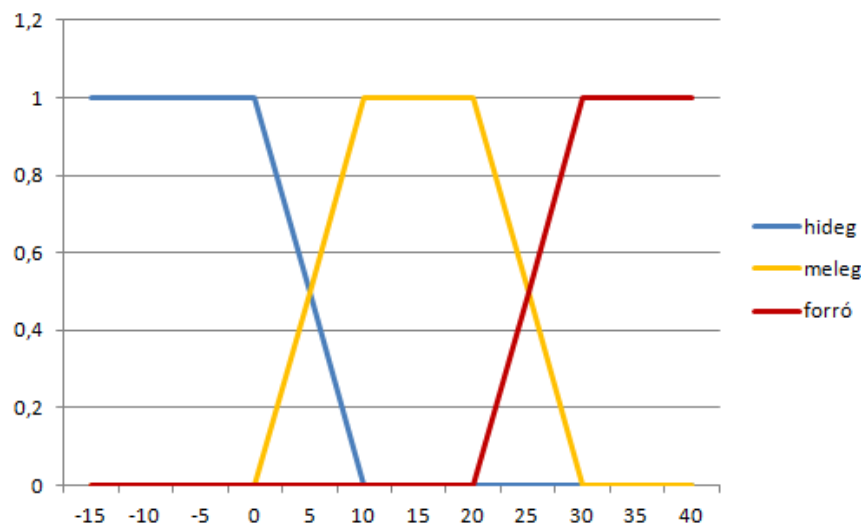
ahol az f függvény értékeit (halmazonként) növekvő sorrendbe rendezve (úgy, hogy egész belül pedig a halmazba tartozás mértéke szerint csökkenő sorrendben legyen) alakul ki a képletben lévő $f(x_1)$ sorozat.

A fuzzy logika egyik konkrét alkalmazásának bemutatása előtt célszerű áttekinteni egy példán keresztül a fuzzy elmélet alapjait.

A hőmérsékletek alapvetően három kategóriába sorolhatóak: hideg, meleg, forró. Ezek a kategóriák nem határolhatóak el egyértelműen, nagymértékben az egyéntől függ, hogy melyik csoportot hogyan definiálja. Ugyanakkor az egyénenkénti megadások valószínűleg nem térnek el egymástól nagyon jelentősen, például egy negatív hőmérsékletet vélhetően senki nem fog a

forró kategóriába sorolni. Ezen tulajdonságokat figyelembe véve adódik, hogy fuzzy eszközöket célszerű használni e csoportok leírásához.

A tartomány legyen -15°C és 40°C között (feltételezhető, hogy diszkrét és az elemei 5-ösével követik egymást). A három fuzzy halmaz egyik lehetséges karakterisztikus függvényén sorozata látható az 2. ábrán. A függvények felfutó, illetve lefutó élei áthelyezhetőek az ábrán, akkor egy másik, de még mindig fuzzy halmazcsoportot alkotnak. A meredekségeket is célszerű változtatni, hogyha az egyén úgy ítéli meg, hogy a határátmenet két kategória között élesebb vagy finomabb.



2. ábra Hőmérsékletek osztályozása

Az ábrán látható három fuzzy halmaz a következő módon írható le:

$$H(x) = -15/1 + -10/1 + -5/1 + 0/1 + 5/0.5,$$

$$M(x) = 5/0.5 + 10/1 + 15/1 + 20/1 + 25/0.5,$$

$$F(x) = 25/0.5 + 30/1 + 35/1 + 40/1.$$

Páronkénti metszeteket nézve a hideg-forró az üres halmazt adja, a meleggel vett esetekben pedig a következőket:

$$HM_{\cap}(x) = 5/0.5, \quad MF_{\cap}(x) = 25/0.5$$

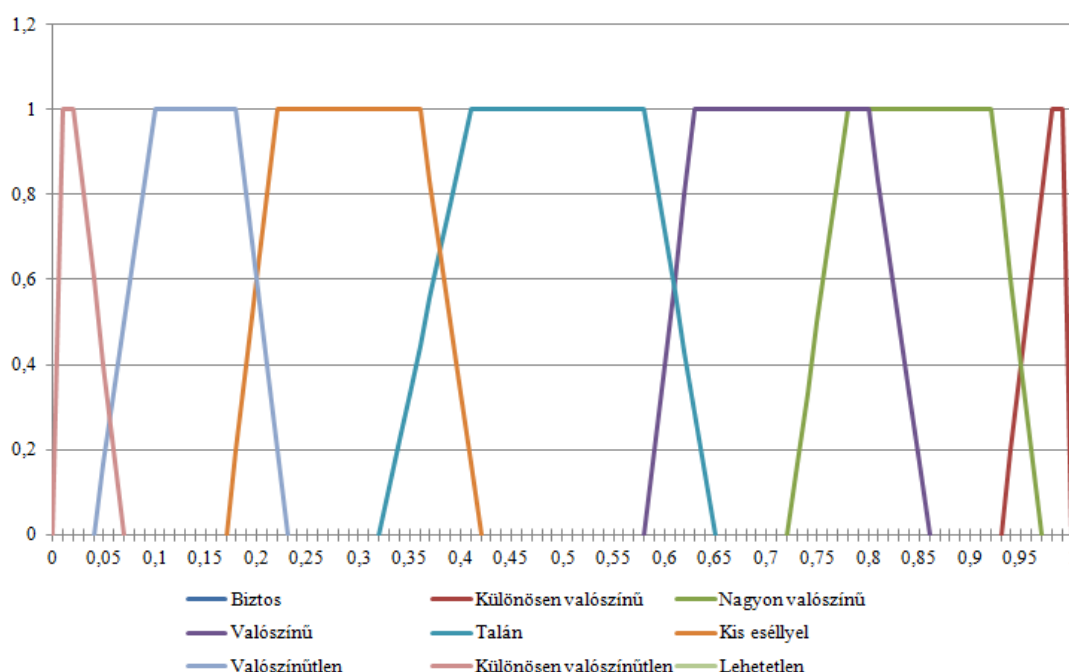
Természetesen finomabb felbontás esetén nem egyeleműek lennének ezek a halmazok. Az unió meghatározásához szintén egyszerűen csak alkalmazni kell a fentebb írt képletet.

A fuzzy halmazelmélet haszna két – kollektív döntéseknél – fontos alkalmazási lehetőségének bemutatásával szemléltethető a legjobban.

Gyakran előfordul, hogy a felhasználók nem tudnak egy opciót számszerűleg értékelni, sokkal könnyebb számukra, hogyha szövegesen megformálhatják véleményüket. Ekkor azonban kategóriákat (lényegében fuzzy halmazokat) kell definiálni, amik ugyan elsődlegesen szöveges jellemzésűek, hogy a felhasználó választani tudjon, viszont ugyanakkor pontosan meghatározott matematikai leírással is rendelkeznek, hogy a már meglévő rendszerben is használhatóak legyenek. Ennek létjogosultságát egy példán keresztül a legegyszerűbb bemutatni [8].

A meteorológusokból álló csoportnak arról kell szavaznia, hogy a következő napon melegebb lesz-e az idő. Ez tipikus esete azoknak a scenárióknak, amikor a kollektív döntésmélet eszközeinek segítségével szakértők hangolják össze véleményeiket. Ebben az esetben használhatjuk válaszként az alábbi szöveges véleményeket:

- Biztos (1, 1, 1, 1)
- Különösen valószínű (0.93, 0.98, 0.99, 1)
- Nagyon valószínű (0.72, 0.78, 0.92, 0.97)
- Valószínű (0.58, 0.63, 0.80, 0.86)
- Talán (0.32, 0.41, 0.58, 0.65)
- Kis eséllyel (0.17, 0.22, 0.36, 0.42)
- Valószínűtlen (0.04, 0.10, 0.18, 0.23)
- Különösen valószínűtlen (0, 0.01, 0.02, 0.07)
- Lehetetlen (0, 0, 0, 0)



3. ábra A szöveges valószínűségeknak megfelelő fuzzy mértékfüggvények

A sorok végén a négy szám a trapéz alakú fuzzy halmazba tartozási mérték négy jellemző értéke, ha a sor (x_1, x_2, x_3, x_4) , akkor x_2, x_3 jelzi az intervallumot, amiben a mérték értéke 1, x_1 és x_4 pedig a két határa a mérték értelmezési tartományának (ezeken kívül azonosan 0 az értéke). A 3. ábrán látható, hogy a két véglet csak egy-egy pont. A szavazás kiértékelése az egyes szavazatok átlagolásával történik, specializálható a módon, hogy lehetőség van rá, hogy ezen opciók közül ne csak egyet, hanem kettőt válasszon a szavazó, ekkor megadhatja, hogy milyen „arányban” szavaz a két válasza. A kiértékelésnél ekkor természetesen arányosan számolódik bele a véleményekbe.

Hasznos segítség a fuzzy halmazelmélet abban az esetben is, hogyha egy döntési szituációban több különböző opció közül egyet kell választani, vagy esetleg mindegyiket értékelni, de ugyanakkor figyelembe kell venni más – akár egymással összefüggésben is lévő – tulajdonságokat. Ekkor a fuzzy mértékfüggvények – így a halmazok – jellemzik az egyes tulajdonságok relevanciáját a döntéshez. Így lehetőség van szubjektív vélemények megadására is. Két, vagy több tulajdonság közös hatása is leírható az együttes fuzzy karakterisztikus függvényekkel.

Az előbbi esetre jó példa a diákok értékelése egy középiskolában [9]. A hagyományos módszer, hogy az egyes tárgyakból (mint attribútumok) elért jegyek számtani átlaga a tanuló eredménye. Legyen az iskola reál iskola, a tárgyak pedig a matematika, fizika és irodalom. Az osztályozás 1-es és 5-ös között lehetséges, értelemszerűen az 5-ös a legjobb jegy. Mivel az iskola inkább reál beállítottságú, az átlagot nem szokványos módon számítják, hanem súlyozzák, méghozzá a következő módon: matematika: 3, fizika: 3, irodalom: 2. Ezen átlag szerint a láthatóak a 2. táblázatban három tanuló eredménye.

2. táblázat Diákok értékelése hagyományos módon

	Matematika	Fizika	Irodalom	Súlyozott átlag
A	5	4	2,5	3,81
B	2,5	3	4,5	3,19
C	3,5	3,75	3,75	3,66

Ez teljesen jó módszer lenne, hogyha a tantárgyak egymástól teljesen függetlenek lennének, ekkor egyik eredménye nem befolyásolná a másikat. Ez azonban nem teljesül, ezért a tárgyak közötti kapcsolatot bele kell venni a döntéshozatalba. Látható a táblázatban az is, hogy a legjobb tanuló az A tanuló, de a C tanuló egyenletesebb teljesítményt nyújtott.

3. táblázat Diákok értékelése Chouquet integrállal

	Matematika	Fizika	Irodalom	Choquet integrálos átlag [9]
A	5	4	2,5	3,47 ↓
B	2,5	3	4,5	3,4 ↑
C	3,5	3,75	3,75	3,73 ↑

Az integrálban lévő f függvény itt az egyes rész fuzzy halmazokhoz tartozik (0, M, F, I, MF, MI, FI, MFI). A halmazba tartozás mértéke tükrözi az iskola eredeti szellemiségét, így a matematika és fizika 0,45-ös súlyúak, az irodalom pedig 0,3. A vegyeseket is érdemes még megemlíteni, MI, FI esetén 0,9, MF esetén 0,5. Ez abból fakad, hogy aki jó fizikából az általában jó matekból is, és ugyanez igaz fordítva is.

Ez természetesen egy mesterséges példa volt, azonban ugyanezen az elven használható valós döntések esetén is.

2.2.3 Játékelmélet alapú döntéshozatal

A játékelmélet tudománya stratégiai döntéshozatali kérdésekkel foglalkozik. Az első játékelmélettel foglalkozó munka James Waldegrave nevéhez fűződik, aki 1713-ban egy levélben [10] megadott egy kevertstratégiájú minimax alapú megoldást a kétjátékosú le Her kártyajátékra. Azonban külön tudományágként nem létezett, amíg Neumann János 1928-ban nem publikálta cikkét a témáról. A játékelmélet kutatásának igazán nagy lendületet Thomas Morgenstermmel közösen jegyzett cikke, The Theory of Games and Economical Behavior adta. A játékelmélet olyan komplex döntési problémákkal foglalkozik, amelyekben az egyes szereplők elsősorban a saját, egyéni érdekeiket figyelembe véve döntenek, de a döntésük kihatással van a többi szereplőre is.

A döntéshozatal játékelméleti megközelítéséhez fontos definiálni néhány alapfogalmat [11]. A játék a játékosok viselkedése és egyéb külső szabályok által leírt folyamat. Általános esetben egy játékot n játékos játszik. Az információs halmaz (szabályok, elképzelhető lépések, eddigi események) birtoklása különösen fontos. Teljesinformációs játékról akkor beszélünk, hogyha a játék véges és a játékosok tisztában vannak mindegyikük lépésével. Erre jó példa a sakk, ellenpélda a póker. A stratégia az a módszer, ami a szabályokat és a többi játékos hibáit felhasználva segíti győzelemre vagy legalábbis döntetlenre a játékost. Stratégiahalmaz elemei a játékos által választható stratégiák. A stratégiateret feszítik ki az játékosok stratégiái, így tulajdonképpen a stratégiák módosulása/változtatása felfogható ebben a térben való lépkedésnek. A győzelem elérésének legfontosabb eszköze a nyereség maximalizálása. Nash

egyensúlyba (NEP) ért a játék, hogyha egyik játékosnak sem érdeke módosítani a stratégiáját, hogyha a másik játékosok se módosítanak, tehát párhuzamosan az ő nyereségfüggvényének növekedésével ne csökkenjen játékosársai nyereségfüggvénye. Ezek a Nash egyensúlypontok a nyereségfüggvény lokális szélsőértékei a stratégiatéren. Nullösszegű, vagy zéróösszegű egy játék, hogyha a játékosok csak úgy tudják növelni a nyereségüket, hogyha másik játékosok nyeresége csökken. Nash bizonyította, hogy amennyiben a kevert stratégiákat is megengedjük, minden n szereplős véges stratégiaterű játéknak létezik Nash egyensúlya [12]. Kooperatív a játék, hogyha egyezség alakul ki a játékosok között, nem kooperatív, hogyha nem alakul ki.

Koalíció a játékosok valamely csoportja, a játékosok halmazának tetszőleges részhalmaza. Nagykoalíció, hogyha ez a részhalmaz a játékosok teljes halmaza, üres koalíció, hogyha egy játékos sincsen a koalícióban.

A játékosokról feltehető, hogy racionálisan gondolkoznak, vagyis a játékos célja, hogy a saját kifizető függvényét maximalizálja, így a viselkedése leírható matematikai értelemben vett optimális döntésként. A másik feltehető tulajdonság a köztudás. Ez semmi egyebet nem takar, mint a hogy a játékosok tudják egymásról, hogy ők is racionális gondolkoznak. Mivel zéróösszegű játékról van szó, így a másik játékosok kifizető függvényének módosulása szükségképpen módosítja a többi, esetleg koalícióból már kiszorult játékosárs helyzetét.

Az időpont meghatározó rendszernél annyira nem, de a demokratikus hőmérsékletbeállító rendszer esetén különösen jól használható ez a tulajdonság (részletes leírás található ezekről a rendszerekről az 5.1, illetve 5.2 fejezetekben). Ugyanis a Tempocracy felfogható egy zéróösszegű játékként, ahol a szavazók a saját kifizető függvényük maximalizálását próbálják elérni. A játékok majdhogynem mindig körökre vannak osztva, a Tempocracy pedig folyamatos kiértékeléssel működik hatékonyan (a felhasználói szavazat triggereli a kiértékelést). Így a körökre osztott játék mellett a Tempocracy kiértékelési ütemét is diszkretizálni kell. A felhasználók gondolkodását felfoghatjuk egy-egy stratégiaként, ennek segítségével tudjuk meghozni a kollektív döntést. Mindegyik felhasználónak a célja elérni az egyik NEP-et, elképzelhető, hogy nem mindig ugyanazon stratégia szerint dönt, hanem valamilyen valószínűséggel választ, ez a kevert stratégia. Ebben az esetben biztosan létezik NEP. Egy játékban több NEP is lehet ezért nem garantálható, hogy amelyik NEP-be került a játék, az a legjobb NEP, ahhoz hogy kimozduljon ebből a játék egyszerre több játékos (itt kerülnek előtérbe a játékosok által kialakított koalíciók) együttes stratégiaváltására van szükség.

Tempocracy esetében nagyon jól használható ez a módszer. A szavazók egy előre meghatározott tartományból választhatják ki a nekik megfelelő hőmérsékletet. Ezekből egy kiválasztott döntési algoritmussal megkapják az eredményt, majd a felhasználók ennek

tudatában újra szavazhatnak. NEP-be akkor kerül a játék, hogyha a szavazóknak nincs értelme az utolsó szavazatukon módosítani, mert a közös hőmérséklet nem fog közelebb kerülni a preferáltjukhoz. Természetesen a játékelméleti megközelítésnek jóval nagyobb léptékű és bonyolultabb rendszereknél van igazán létjogosultsága.

2.3 Kurrens kutatás kihívásai

A kollektív döntésselmelet jelenleg is aktív kutatási terület. Három fő kutatási irány él, az egyik a gazdasági, a másik a politikai, a harmadik pedig az orvosi. Egészen mások a motivációk az egyes irányokban. A gazdasági oldalon vállalatirányítási kérdések és tőzsdei, pénzügyi problémák a fő mozgatórugók, míg az orvosin inkább a pontos diagnózis, az emberélet megmentése kerül előtérbe.

Vállalati folyamatok kezelésének módjáról, és különösképpen ennek döntésselmeleti háttéréről ír Paul H. Robin [13] munkájában. A tőzsdén manapság már nem emberek, hanem gépeken futó szoftverek kereskednek, a lényeg, hogy mindig a jó részvényt vegye meg, pont jókor. Ehhez elengedhetetlen, hogy a döntéshozatali eljárások gyorsak és hatékonyak legyenek. A PROMETHEE módszer segítségével mutat be egy modellt és egy eljárást [14], ami használható a tőzsdei tranzakciók tervezéséhez. Az egyes részvények értékelésének módjáról, ennek döntéshozatali vonatkozásáról ír a [15].

A politikai életben is nagy szerepe van a különböző döntéshozatali formáknak. Különösen kutatott terület a választók szavazása, és a választási rendszerek elemzése [16]. Érdekes kérdés még a közösségi média hatása a politikára, különösképp a választók döntésére [17].

Orvosi, biológiai témájú cikkek is két irányból vizsgálják a kérdést. Egyik ág, amelyben a neurobiológia eszközeit felhasználva kutatják, és értelmezik az agy döntéshozásban részt vevő területeit [18], [19]. A másik ág a tényleges orvosi alkalmazások kutatása, elsősorban persze diagnosztikai alkalmazások. Diagnosztikai hibák csökkentésére, és a helyes diagnózis gyors felállításának módszertanára ad javaslatot [20]. Az eutanázia kérések tekintetében emeli ki és vizsgálja a döntéshozási mód fontosságát és megkerülhetetlenségét a [21].

Bár régóta kutatott terület a döntéshozatali folyamatok, kurrens kutatások még mindig folynak, és ezeknek a kutatásoknak van létjogosultsága.

3. Kurrens infokommunikációs technológiák és elérhető megoldások

Egyes infokommunikációs technikák (számítógép, mobiltelefon) a mindennapi élet megkerülhetetlen részévé léptek elő. A fejezet első felében ezen eszközök és technológiák fejlődésének folyamatát mutatja be, további részében pedig a jelenleg elérhető döntéshozó rendszerekről lesz szó.

3.1 Pervazív számítástechnika és az okos platformok

A számítógépek mára az emberek mindennap használt eszközeivé léptek elő. Sokáig az asztali számítógépek, és a hordozható laptopok jelentették a teljes felhasználói informatikai eszköztárat.

Habár már az 1970-es években foglalkoztak számítógépek és telefonok egybeépítésével, az okostelefon kifejezést mégis először csak az Ericsson használta GS 88 „Penelope” eszköze jellemzésére. Az első generációhoz tartozó okostelefonok a PDA-kból alakultak ki, a gyártók felszerelték a korábbi PDA-ikat cellás rendszerű mobilhálózatot használni képes egységgel.

Sokféle mobil operációs rendszert fejlesztettek ki, általában gyártónként különbözött. Ennek a versenynek az egyik első zászlóhajója a **Symbian** operációs rendszer. 2000-től gyakorlatilag az összes Nokia telefont Symbian operációs rendszerrel szállították, így elég hamar piacvezető mobil operációs rendszerré fejlődött. Eredetileg a Symbian Ltd. fejlesztette ki az EPOC rendszerből, de ma már az Accenture tartja karban. [22]

A felhasználói felülete barátságos, jól használható, joystick funkciójú gombokkal lehet navigálni a menüpontok között. Valamint két, az aktuális lehetőségekhez igazodó kiválasztó gomb is segíti a felhasználót a könnyebb irányítás érdekében. Már itt megjelenik az igény a navigálás kiterjesztésére, ami majd csak az érintőképernyők tömeges elterjedésével fog teljesülni.

Infokommunikációs szemszögből még meg kell említeni a WebKit alapú böngészőt, ugyanakkor sokáig az Opera mobiltelefonokra portolt változatát használták. Későbbi változatokban megjelent egy csökkentett funkcionalitású Flash bővítmény is, de se a telefonok sebessége, se az elérhető mobilinternet gyorsasága nem segített az elterjedésében. Az érintőképernyős használatra tervezett verziók követik az új trendeket mind a weboldalak által használt technológiákat illetően (HTML5, CSS3), mind a felhasználók igényeinek változását illetően (közösségi hálózatok elterjedése) [23].

Alkalmazásfejlesztés tekintetében is ígéretes volt a helyzet, 2010-ben a fő SDK a standard C++ lett Qt grafikus keretrendszerrel [24]. Ugyanakkor lehetőség van Python-ban, Adobe Flash Lite-ban, sőt még Java ME-ben is alkalmazást készíteni.

A Nokia alkalmazásboltjában (<http://store.nokia.com>) találhatóak fizetős valamint ingyenes alkalmazások is egyaránt. Vállalati környezetben különösen fontos, hogy az üzleti titoknak minősülő adatok biztonsága semmilyen esetben se forogjon kockán, valamint, hogy a munkavállaló a munkahelytől távol is tudjon dolgozni. A Symbian operációs rendszer már támogatja az u.n. remote-wipe/lock lehetőségét Microsoft Exchange ActiveSync-en keresztül [25]. Lehetőség van a ROM alapú alkalmazások, pluginek felülírásának letiltására is. Az alkalmazásboltból szintén letölthetőek irodai alkalmazások is.

A végső kegyelemdöfést a Symbiannak, mint mobil operációs rendszernek a Nokia adta meg 2011 februárjában, mikor bejelentette partnerségi megállapodását a Microsofttal, hogy következő okostelefonjainak nagy része Windows Phone operációs rendszert fog futtatni.

Vállalati piacon nagy vetélytársa a Symbiannak a kanadai Research In Motion (RIM) **BlackBerry** nevű operációs rendszere. Míg a Symbian elsősorban a lakossági felhasználókat célozta meg, a BlackBerry inkább az üzleti világban terjedt el. A BlackBerry telefonok döntő többségében teljesértékű QWERTY billentyűzettel rendelkeztek, amely jelentősen megkönnyítette a hosszabb szövegek gépelését. Egyéb funkcionális gombjai hasonlóak a Symbianos Nokiához.

A fő hangsúlyt az operációs rendszer tervezésénél a minél gyorsabb, és akadálymentesebb kommunikációra helyezték. A BlackBerry telefonokat a szintén az RIM által gyártott BES-eken (BlackBerry Enterprise Server) keresztül lehet központilag kezelni. (Ehhez természetesen létezik beépülő modul a Microsoft Exchange-hez, illetve a Lotus Domino-hoz). Minden BlackBerryhez tartozik egy egyedi azonosító PIN, ezt felhasználva közvetlenül is küldhetünk üzenetet másik BlackBerry tulajdonosnak. A felhasználók többsége nem a beépített internet böngészőt használja, hanem az Opera Minit.

A biztonságra, mivel üzleti operációs rendszernek tervezték, különösen nagy gondot fordítottak. Mind a tárolt, mind a továbbított adat eddigi ismereteink szerint feltörhetetlen titkosítással van ellátva [26].

Új fordulatot vett a verseny, amikor 2007-ben az Apple bejelentette az **iOS** operációs rendszert futtató első iPhone-t. Az mobil operációs rendszerek közül ez a legzártabb, nem hogy a forráskód nem nyilvános, de az Apple még nem is liszenszeli nem általa gyártott eszközökre.

Nem csak az érintőképernyőt, hanem a multi-touchot, a többszörös érintést is képes kezelni, ezáltal még természetesebbé téve a felhasználói élményt a különböző gesztusokkal (például:

nagyítás – két ujjunkat középről kifelé húzva). Egy fizikai gomb még megtalálható rajta, ami alapvető szerepet tölt be a navigálásban.

Beépített böngészője a Safari, amely szintén a WebKit alapjaira épül, így támogatja a legújabb webes technológiákat is. Az iOS egyik verziójában sincs natív Flash támogatás.

Az előzőleg bemutatott biztonsági funkciók nagy részét szintén gyárilag tartalmazza [27]. Részletes leírás olvasható a hivatkozott oldalon [28].

Az Apple alkalmazásboltja az AppStore, ahonnan iOS rendszerű készülékekre, tehát nem csak okostelefonokra lehet alkalmazásokat letölteni. Többségével fizetős alkalmazásokat találhatók, de a játékok kivételével szinte mindegyikre létezik ingyenes, jól használható alternatíva.

Természetesen van lehetőség saját alkalmazás készítésére. A fejlesztés Objective-C nyelven folyik, az Apple biztosít hozzá emulátort és fejlesztőkörnyezetet, így nem szükséges, bár kétségkívül hasznos, iOS rendszert futtató eszközzel rendelkezünk. Az alkalmazásboltba azonban csak fejlesztői liszensz megléte esetén publikálhatóak alkalmazások.

Az **Android** operációs rendszer eredetileg az Android Inc. fejlesztette, majd először a Google csak anyagilag támogatta, végül 2005-ben felvásárolta, és most a 2007-ben alapított Open Handset Alliance-szal [29] közösen fejleszti. Eltérően az előbbiektől nyílt forráskódú a rendszer. Rengeteg típusú mobiltelefonon fut Android operációs rendszer, de közös tulajdonságuk, hogy mindegyik érintőképernyős, kevés eltéréssel a későbbiekben ismertetett szenzorok megtalálhatóak bennük. Irányítása többnyire az érintőképernyőn keresztül történik, a régebbi telefonokon még megtalálhatóak voltak fizikai gombok is az előlapon, azonban az újabbnál ezeket már szinte teljesen felváltották a szoftveres billentyűk.

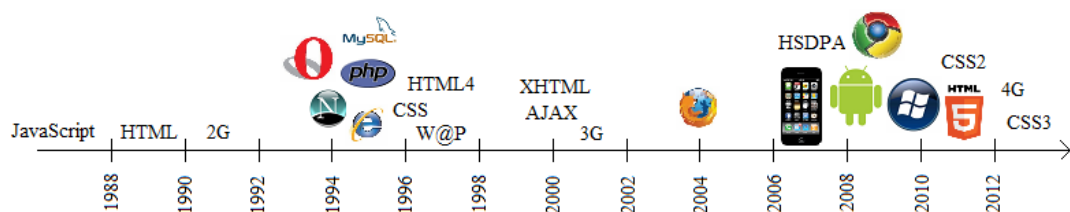
Többnyire a beépített böngészőt használják a felhasználók, azonban több másik is elérhető: Opera Mini, Dolphin. Az újabb (>4.0) Androidokhoz elérhető a Google Chrome Androidra portolt változata is. Biztonság tekintetében a vállalati igényeket is kielégíti, azonban a nyílt forráskód miatt a cégek ódzkodnak a használatától (így ugyanis könnyebb benne rejtett hibát keresni, ami nem került befoltozásra).

Android telefonra alkalmazásokat legegyszerűbben a Google saját, Google Play nevű online boltján keresztül szerezhetünk be. Újdonság az előzőekhez képest, hogy itt könyveket, zeneszámokat is vásárolhatunk, valamint magazinokra is előfizethetünk.

Hasonlóan az iOS rendszerhez, Android alkalmazásokat is fejleszthetünk. Többnyire Java nyelven íródnak ezek, saját körben szabadon terjeszthetjük, azonban a Google Play-be feltöltéshez itt is rendelkezünk kell fejlesztői liszensszel.

Már a BlackBerry-kben is megjelent néhány szenzor, azonban a szenzorhasználat igazán az iOS-es készülékeknél terjedt el. A kamera manapság már alapfelszereltségnek számít egy okostelefonnál. Ugyanakkor egyre nagyobb hangsúly helyeződik a GPS-re, a giroszkópra, az iránytűre, a fényérzékelőre, a gyorsulásmérőre, a közelségérzékelőre. Ezek mind-mind a felhasználói élmény minél teljesebb kiterjesztése miatt kerültek be az okostelefonokba. Néha olyan apróságokon múlik ez, mint amikor a fülünkhöz tartjuk a telefont, ezt a közelségérzékelő érzékeli, és elsötétíti a képernyőt, valamint blokkolja, hogy akaratlan érintésünket helyén kezelje az operációs rendszer. A fényérzékelő szenzor is az esetek többségében hasonlóan kis szerepet tölt be: a telefon ennek segítségével szabályozza a képernyő fényerejét, így napsütésben is megfelelően látszik, valamint sötétben se égeti ki ragyogásával a szemünket. Ezek apróságok, de mégis meg tudja könnyíteni a felhasználó életét.

Sokkal fontosabb szerep jut a GPS-nek, a hely-, illetve giroszkópnak, a helyzetmeghatározónak. Egyre jobban előtérbe került a broadcast jellegű tartalomszolgáltatás helyett a testreszabott, jelen esetben lokalizált szolgáltatás. Egy felhasználó számára sokkal nagyobb információértékkel bír az adott hely időjárása, mint egy tőle 100 km-re lévő településé. Hasonlóan számtalan példát lehetne találni a közéleti hírektől elkezdve a tömegközlekedésen át egészen a kávézóig, éttermekig. A helyzetmeghatározás a manapság egyre nagyobb teret nyerő ún. augmented reality (kiterjesztett valóság) alkalmazások miatt válik egyre fontosabbá.



4. ábra A web fejlődése

Mobiltelefonok internetes elérése sokat fejlődött az elmúlt időszakban, a kezdeti GPRS-es és WAP-os rendszerektől eljutottunk HSDPA-n keresztül a 4G LTE rendszerekig. (Ezek mindegyike aszimmetrikus sáv szélességű, downlink irányban gyorsabb.) Míg a mobilinternet megjelenésekor luxuscikknek számított, ma azonban gyakorlatilag szériatartozéka egy okostelefonnak. Ennek megfelelően a felhasználók mindig, minden körülmények között elérhetőek. Persze vannak hátrányai is, de az előnyök jelentősen kompenzálják.

Így nem csoda, hogy előtérbe kerül a felhasználók által szerkesztett tartalom, a közösségi oldalak gomba módjára nőnek.

A web2.0 filozófia (dinamikus oldalak) átveszi a hatalmat a hagyományos, hír jellegű oldalak felett. Megjelennek a dinamikus, tisztán web architektúrát használó alkalmazások, szolgáltatások, amiknek a használatához semmi másra nincs szükség, csak egy böngészőre. Gyakran olyan megoldásokra, képi-, grafikai elemek megjelenítésére képes a böngésző, amik sokáig csak és kizárólag desktop környezetben voltak elképzelhetőek. A böngésző válik a modern ember svájci bicskájává. Ennek megfelelően a felhasználó igénye is teljesül, hogy az őt körülvevő világ részese és egyben alakítója is legyen.

3.2 Elérhető kollektív döntéstámogató alkalmazások

Az alábbiakban néhány fontosabb jelenleg már elérhető és használható alkalmazás kerül bemutatásra különböző szempontok szerint értékelve.

- **When is good:** Regisztráció nélkül is lehet időpontokat egyeztetni, azonban ekkor csak 15, 30, 60 perces időintervallumok adhatóak meg. A szavazáshoz egy linket generál, amit el lehet küldeni a szavazásra jogosultaknak, az eredmények megtekintéséhez egy kódot kell beírni a felületen, ami azért egy kicsit körülményes. Van fizetős verzió is, az olcsóbban már nem csak “jó”, “nem jó” választási lehetőség van, hanem megjelenik a “tökéletes”, “nem ideális, de megoldható” opció is. A drágább változatban saját aldomaint kap a felhasználó, testreszabhatóvá, reklámmentessé válik a felület (saját logó, elrendezés), valamint a felhasználókezelésben is szabad kezet kap a szavazás gazdája.
- **Doodle:** Nagyon népszerű, egyszerűen használható. Szintén nem kötelező regisztrálni, az ingyenes változatban a regisztrációval ahhoz az előnyhöz jut a felhasználó, hogy egy oldalon tudja megtekinteni a szavazásait. A When is goodhoz hasonlóan itt is egy link segítségével tudnak szavazni, de felajánlja a rendszer, hogy a szavazásra jogosultak e-mail címének beírása után elküldi nekik a szavazást, valamint elküldi szavazás gazdájának is az adminisztráláshoz szükséges linket. Háromféle opció közül lehet választani, ezek az “igen”, “nem”, “talán”. Előnye még, hogy naptár nézetet is biztosít, valamint már meglévő (esetleg csoportos) naptárt megjelenítve segít kiválasztani az alkalmas időpontokat. Biztosít a rendszer még egy MeetMe funkciót is, ami az 1-1 találkozások tervezését segíti elő. Ez egy publikus felület, ahol megjelenik a naptár (természetesen csak az elfoglaltságot/nem elfoglaltságot jeleníti meg) és ismerősök be tudják jelölni a számukra is alkalmas időpontokat. A fizetős változat természetesen itt is reklámmentes, valamint mobil kliens is rendelkezésre áll. Testreszabható a szavazási oldal kinézete, valamint egyéni értesítések beállítására is

lehetőség van. Nincs korlát az egy időben futó szavazások számára, valamint a meghívott személyek számára sem.

- **Tungle.me:** Egyszerű regisztráció (OpenID támogatás) után írható ki a szavazást, de regisztráció nélkül is lehet szavazni. Számos naptárszolgáltatással tud szinkronizálni. Mobil alkalmazás egyedül Blackberry-hez létezik, de az online felület bármilyen böngészőben problémamentesen futtatható. Nincs fizetős változat, de az ingyenesben sincsenek reklámok. A Doodle MeetMe-jével megegyező funkció is megtalálható benne. Az angolon kívül francia nyelven is használható.
- **TimeBridge:** Regisztráció nélkül nem lehet szavazást létrehozni. Túl sok beállítási lehetőség létezik, ezért elvesznek a lényeges opciók. Tanácsokat, tippet is ad a szavazás kiírásánál, de ezek többnyire használhatatlanok. Előnye viszont, hogy csoportokat is létre lehet hozni, így nem kell mindig egyesével hozzáadni a felhasználókat. Az SMS értesítési lehetőség beállítása azonban kifejezetten hasznos. Mobil alkalmazás sajnos csak iOS-t futtató eszközökre létezik.
- **Meet-o-Matic:** Egyszerű, kicsit fapados az alkalmazás ingyenes verziója. Csak délelőtt/délutánra tervezhető találkozók, valamint a fizetős verzióban is csak negyed órás időközök lehet beállítani. Szavazni itt is lehet regisztráció nélkül azonban az e-mail címeket meg kell adni és a rendszer küldi ki az értesítőt. Az eredmények megjelenítése egy kicsit átláthatatlan, de jó ötlet, hogy egy gombnyomással lehet a résztvevőket értesíteni a végleges időpontról.
- **ScheduleOnce:** Nincs saját regisztrációs rendszerük, hanem a már meglévő Google vagy Facebook fiókunk segítségével tudunk bejelentkezni. Ez egy céges környezetben hátrány lehet, de magán felhasználásra kifejezetten előnyös. Itt is használható mind az ingyenes, mind a fizetős verzió. Már az ingyenes verzióban is megtalálható a csoportszavazások kiírási lehetősége valamint a MeetMe oldal, azonban e-mailes értesítést és reklámmentességet csak a fizetős verzióban kap a felhasználó. Sok világviszonylatban is ismert cég, egyetem használja például: IBM, NASA, Google, orange, Stanford University, Microsoft, Harvard University, WWF, Casio.

4. táblázat Elérhető alkalmazások összehasonlítása

	Kliensek/ Platformok	Algoritmus, döntés	Ár/Érték	Kényelmi funkciók	Összesen
When is good	3	1	2	3	9
Doodle	5	1	3	4	13
Tungle.me	4	2	3	5	14
TimeBridge	4	2	2	3	11
Meet-o-Matic	4	1	3	3	11
ScheduleOnce	5	2	4	5	16

A fenti táblázatban található egy összehasonlítás az alkalmazásokról. Leguniverzálisabbnak, legjobbnak a ScheduleOnce bizonyult, azonban ez sem teljesen megfelelő. Legnagyobb gyengeség mindegyik alkalmazás esetében a döntési résznél mutatkozott. Itt egyik alkalmazás sem tudott a sima többségi döntésnél kifinomultabb módot használni.

4. Új szavazat alapú kollektív döntéstámogató rendszer

A 3.2 fejezetben ismertetett alkalmazások hiányosságát orvosolandó mutatja be ez a fejezet az új kollektív döntéstámogató rendszert. A lehetséges alkalmazás scenáriók ismertetése után, az univerzális architektúra bemutatására kerül sor, ezt pedig a felhasználható speciális algoritmusok tárgyalása követi.

4.1 Alkalmazás scenáriók

Az egyes szavazás olykor gyökeresen eltérnek egymástól, mind a szavazás előkészítésében, mind a szavazatok értelmezésében, kiértékelésében különbözőek, így szükséges ezeket a szavazásokat típusokba sorolni, és pontosan megadni az egyes típusok sajátosságait, ezek kezelésének módjait is. Az alábbiakban látható hat csoportba az általam vizsgált használati esetek kategorizálhatók.

- **Opciók közötti egyszerű választás / opciók súlyozása:** A legegyszerűbb döntési scenárió, amikor semmilyen megkötés nincsen a szavazással kapcsolatban. A felhasználóknak N lehetséges opció közül kell egyet kiválasztaniuk, vagy az egyes lehetőségeket egy előre definiált skálán értékelni. A szavazás időtartama általában fix, van egy meghatározott vége, ami után már a szavazat leadás nem lehetséges. A legegyszerűbb példa erre egy időpont egyeztetés. A szavazás kiírója meghatározza egy eseményre lehetséges időpontokat, majd a kijelölt csoport szavazata leadásával alakítja ki a döntést.
- **Függvény által vezérelt döntés:** Egy olyan döntési scenárió, amelyben nem kizárólag a csoport akarata határozza meg a döntést, hanem egy külső vagy belső kényszer is. Ezt a legegyszerűbb egy függvénnyel reprezentálni. Célszerű ezt a függvényt úgy megválasztani, hogy az elérni kívánt optimális változat a függvény egyik szélsőértékébe essen. Sokszor nem egyértelmű ez a függvény, ugyanis elképzelhető, hogy több szempontot is figyelembe kell venni a döntés során. Erre jó példa a beruházások tervezése. Több beruházási terv esetén ki kell választani, hogy melyikeket valósítsa meg a vállalat. Ilyenkor nem csak a vállalat vezetése dönt, hanem figyelembe kell venni a ráfordítandó összeg korlátosságát is. Ennek szem előtt tartásával egész más eredményt szolgáltatathat a döntéshozó eljárás.
- **Előre szűrt döntés:** Ennél a döntési scenáriónál többdimenziós döntési tér van. Nevezetesen az X_i halmazok mindegyikéből pontosan egy opciót kell választani, és

mindegyik csoportból választani kell. $i = 5$ esetén is már rengeteg lehetőség van, ha csak mindegyik halmaz kételemű akkor is. Nem várható el a felhasználótól, hogy ő válassza ki a lehetőségek közül a legmegfelelőbbet. Sokszor nem is mindegyik kombináció lehetséges, így elveszhetnek az azokra leadott szavazatok. Erre egyfajta megoldás, amit ez a döntési scenárió is kínál, hogy nem halmazonként külön-külön szavaznak a csoport tagjai, hanem csak ezek kombinációjának bizonyos kritériumoknak megfelelően leszűkített változatának megmaradt elemeire szavazhatnak. Legkézenfekvőbb példa erre egy étteremi menü összeállítása. Az egyszerűség kedvéért csak levesből és főételből lehet választani. Ha a levesek között megtalálható a Zöldborsó leves, a főételek között meg a Zöldborsó főzelék, akkor ezt a lehetőséget ki kell venni, mert senki sem fogja választani.

- **Folyamatos/periodikus szavazás:** Folyamatos, periodikus kiértékelésű szavazásokra akkor van szükség, hogyha valamilyen szempontból fontos az eredmény, döntés frissessége, aktualitása. Gyors felhasználói vélemény változásakor különösen fontos, hogy ez minél előbb megjelenjen a tényleges döntésben. Periodikus kiértékelés esetén az előre meghatározott időtartam lejárta után határozzuk meg újra a döntést, míg folyamatos kiértékelés esetén elég megvárunk az első szavazat változását. Persze kivéve akkor, hogyha egy külső tényezőtől is függ a döntés. Egy példa erre a scenárióra egy hőmérséklet meghatározó alkalmazás. Egy irodában mindig gondot jelent a megfelelő hőmérséklet megtalálása, ugyan itt inkább célszerű lenne a folyamatos kiértékelést választani a periodikus helyett.
- **Kérdőívyszerű szavazás:** Ez a scenárió lényegében ugyanaz, mintha több Opciók közötti egyszerű választás típusú scenárió kerülne egymás után. Azonban fontos különbség, hogy itt lehet kezelni azt a problémát is, hogy egy kör (egy Opciók közötti egyszerű választás típusú szavazás) esetleg az előző válaszok függvényében irrelevánssá válik. Leginkább szolgáltatást nyújtó helyeken tudják ezt a scenáriot sikeresen alkalmazni, ugyanis az elégedettség-felmérés struktúrája nagyban hasonlít erre.
- **Egyéb:** Számos egyéb scenáriot lehetne még felsorolni, azonban a fent említettekből döntő többségük egyszerűen származtatható.

5. táblázat A különböző scenáriók összehasonlítása

	Egyéni szavazat	Csoportos döntés	Szavazási körök
Opciók közötti egyszerű választás	A szavazó mindegyik opciót értékeli, ezekből az értékekből alakul ki a szavazat.	A szavazatok alapján a választott algoritmusnak megfelelő döntés.	Egy szavazási kör, a lezárás előtt változtathat a döntésen.
Függvény által vezérelt döntés	A szavazat a lehetséges opciók sorrendjéből alakul ki.	Figyelembe véve a szavazók döntését, valamint a külső/belső kényszert.	Szintén egykörös szavazás, a megadott időlimiten belül akármikor szavazhat.
Előre szűrt döntés	A szavazó az előzetes szűrésnek köszönhetően már az amúgy is érvényes opciókra adhatja le a szavazatát, ezzel értékelve azokat.	A szavazatok alapján a választott algoritmusnak megfelelő döntés.	Akár több körös is lehet alkalmazástól függően.
Folyamatos/periodikus szavazás	A szavazás típusától függően a szavazó vagy a felsorolt N lehetőség közül egyet választ, vagy mindegyiket értékeli.	A szavazatok alapján történik szavazástípustól függő módszerrel.	Értelemszerűen többkörös, folyamatos kiértékelésű, az egyes körök között a felhasználóknak természetesen van lehetőségük módosítani a szavazaton.
Kérdőívyszerű szavazás	Az adott kérdés típusától függ, de általában egy opciót kell N lehetőség közül választani.	Nem mindegyik szavazó ugyanazokra a kérdésekre szavaz, ilyenformán csak valamilyen összesített értékből származtatható kollektív döntés.	Kialakításából adódóan többkörös, megvan a lehetőség rá, hogy a válaszokhoz adaptálódnak a szavazás további kérdései.

4.2 Univerzális architektúra

Az előző fejezetben ismertetett scénáriókban közös, hogy mindegyiknél a felhasználók előre jól definiált tulajdonságnak megfelelő csoportja vehet részt a szavazásban. Ebből adódik, hogy ennek a csoportnak a tagjai, a szavazás szempontjából, egyenrangúak (persze attól eltekintve, hogy a szavazatuk mekkora súllyal számít bele a végső eredménybe).

Mivel kiértékeléshez, döntéshozáshoz szükség van a csoport többi tagjának szavazatára is, így nem megengedhető, hogy ezeket a számítási feladatokat a szavazók eszközein végezzük, ugyanis ott mindig csak az adott eszközön szavazó szavazatát ismerjük. Ezért szükség van egy központra, ahol a szavazatok összegyűlnek és kiértékelődnek az adott szavazásnak és csoportnak megfelelően.

Így adódott, hogy a rendszert a kliens-szerver architektúra fölé, annak alapjaira épült. Kliensek a szavazók eszközei, a szerver pedig a központi számítógép, ami összegyűjti a szavazatokat.

A szavazatok eltárolásához adatbázist használ, a szerver ebbe menti le a kientől érkező szavazatokat, és innen kéri le a kiértékelés folyamán. Az adatbázisban el kell tárolni a felhasználók autentikációs és autorizációs információit, az azonosítás természetesen a kliensen keresztül történik. Fontos, hogy az adatbázis szerkezetét úgy legyen felépítve, hogy egyszerűen ki lehessen választani a felhasználók azon csoportját, akik jogosultak szavazni. Szükséges még rögzíteni az adatbázisban a szavazás adatait, típusát, a típushoz tartozó egyéb speciális tulajdonságokat is.

A rendszer modulokból épül fel. Két típusú modul különböztethető meg. Az első a kliensoldalon megtalálható modulok. Ezek sorra a következők:

- A **kliensoldali Megjelenítő** modulnak a feladata a szervertől érkező információk közvetítése a felhasználó felé az eszközön keresztül, kihasználva elsősorban az eszköz grafikus megjelenítési lehetőségeit.
- A **Kommunikációs** modul biztosítja a szerver és a kliens közötti kapcsolatot, a szavazás és az eszköz típusától függően más és más technológiát használva.

A szerveroldali modulok pedig a következők:

- A **Kommunikációs** modul tartja a kapcsolatot a klienssel.
- A **Szavazat** modul befogadja a Kommunikációs modultól a kientől érkező szavazatot, feldolgozza, és gondoskodik az Adatbázismodul felé továbbításáról.
- A **Szavazás** modul kezeli a szavazásokat, megfelelő jogosultság birtokában a kliens felé továbbítja őket a Kommunikációs modulon keresztül.

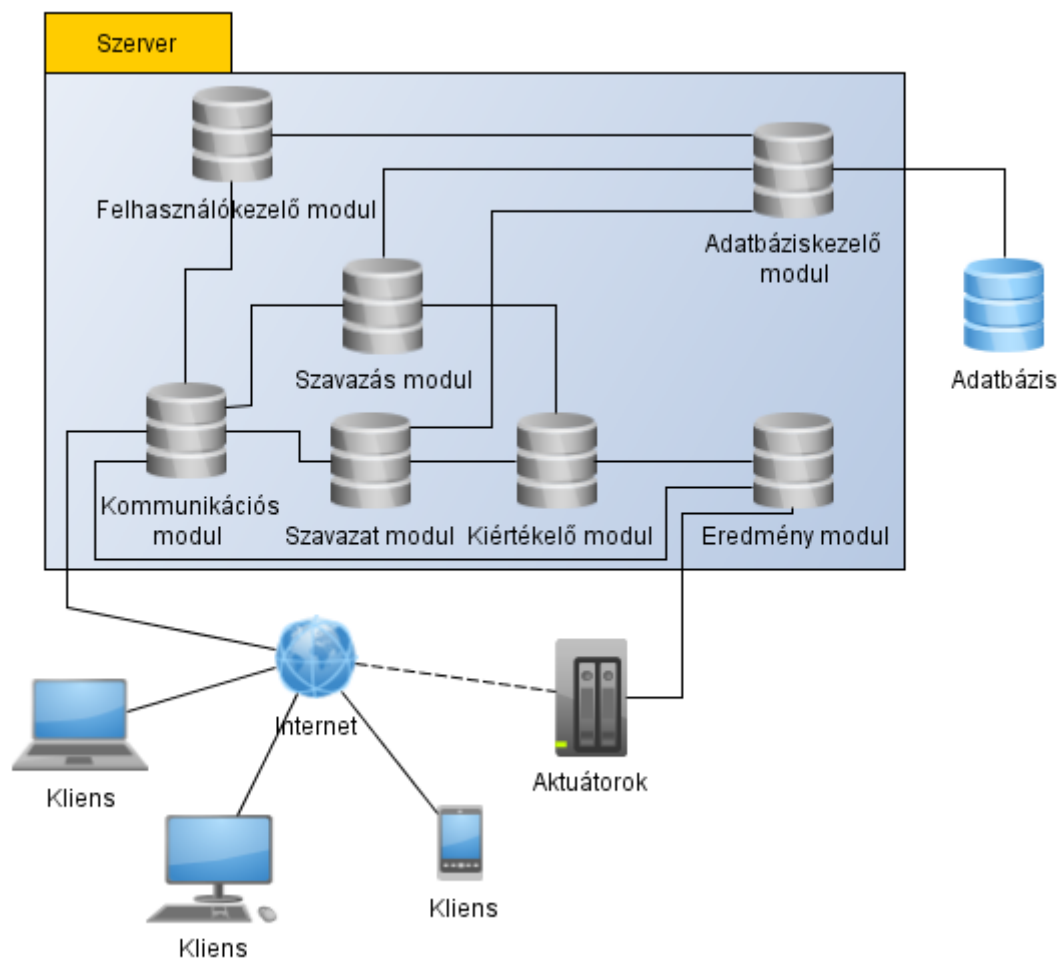
- Az **Adatbáziskezelő** modul kapcsolatban áll szinte az összes másik modullal. Egy interfészt valósít meg a tényleges adatbázishoz, ezen keresztül tudnak az egyes modulok módosításokat végezni az adatbázison.
- A **Felhasználókezelő** modul segítségével lehet az egyes felhasználók adatait módosítani, csoportokba szervezni őket, majd a csoportot egy vagy akár több szavazáshoz rendelni.
- A **Kiértékelő** modul valósítja meg a tényleges döntéshozást, a szavazás, a szavazatok, valamint a szavazó csoport elérhető adatainak felhasználásával. Így a Kiértékelő modulban pontosan definiált módon létrejön a bizonyos szempontból optimális döntés.
- Az **Eredmény** modul a Kiértékelő modulok eredményének felhasználásával alkotja meg a végleges döntést. Esetenként előfordulhat, hogy egy Kiértékelő modul nem egy konkrét lehetőséget jelöl ki győztesként, hanem az összeset súlyozza bizonyos értékekkel. Így a Kiértékelő modul eredménye lehet egy másik Kiértékelő modul bemenete, aminek viszont a kimenete már lehet egy döntés is. Ez az egyik nagy előnye ennek az univerzális architektúrának, hogy a kiértékelő modulokat, mint kockákat pakolhatjuk egymás mellé / alá / fölé, attól függően, hogy az adott szavazást milyen módon szeretné a felhasználó kiértékelni.
- Az **Aktuátor** modul felelős az aktuátorokkal való kapcsolattartásért, az Eredmény modul értékétől függően irányítja őket.

Általános esetben a legegyszerűbb, hogyha a kliens és a szerver az interneten keresztül kommunikál egymással (természetesen ez felvet biztonsági aggályokat is, ez esetben lehet korlátozni a helyi hálózatra). Ekkor két féle protokollsaládból választhatunk: TCP alapú protokollt akkor érdemes választani, amikor fontos, hogy a szavazat biztosan beérkezzen. UDP alapú protokoll használata akkor célszerű, hogyha a szavazást folyamatosan, vagy periodikusan kell kiértékelni, így a döntés meghozatalának fontossága felülírja az esetleges kimaradót, de szavazni óhajtó csoporttagokat. Őket természetesen a következő körben lehet kompenzálni.

Mivel a szerver nem biztos, hogy mindig helyi hálózaton elérhető, ezért célszerű egy interneten keresztüli kommunikációs lehetőséget is beleépíteni az alkalmazásba. Manapság már majdhogynem kötelező megfelelő titkosítással levédeni az ilyen csatornákat. Kliens oldalon természetesen szükség van felkészíteni az eszközöket a megfelelő kommunikációs mód használatára. Célszerű a szervert szolgáltatás részét egy felhőben elhelyezni, így némiképp egyszerűbb a kivitelezés, valamint az üzemeltetés is biztonságosabb.

Kliensoldalon elsősorban web böngésző képzelhető el eszközként, szerver oldalon pedig a megfelelő program futtatására képes szervergép. Azonban az informatikai rendszerek

megfelelően a kliensoldali eszközkinálat kiszélesedik az okostelefonokkal, PDA-kal. Szerver oldalon pedig megjelennek a felhő alapú megoldások.



5. ábra Az univerzális architektúra

4.3 A felhasználható algoritmikus eszköztár

Nem minden esetben határozható meg a kollektív döntés a szavazatokból, és a szavazók preferenciáiból (itt az egyéb tulajdonságok értendőek, amik rendelkezésre állnak a szavazóról, de ugyanakkor nem kapcsolódnak szorosan a szavazáshoz). Ekkor célszerű valamilyen algoritmikus segítséget igénybe venni. Törekedni kell arra, hogy a kollektív döntéssel a szavazók minél nagyobb elégedettségét érje el a döntés. Persze nem garantálható, hogy mindegyik felhasználó teljes mértékben elfogadja a döntést, és kedvére valónak véli, de bizonyos kompromisszumokat megengedve elérhető, hogy a többség akarata teljesüljön.

Egyik ilyen problémaforrás, amikor két szavazás nem teljesen független egymástól, ekkor a két szavazás kiértékelését össze kell kapcsolni és együtt kezelni. A konfliktusos szavazások

azonban nem mindig pontosan ugyanazon társaság számára vannak meghirdetve, előfordulhat, hogy egészen kisszámú közös tag van, valamint a kiértékelés is sokkal nehezebb [5].

Első lehetséges mód a probléma megoldására, hogy a két szavazás opciónak minden lehetséges kombinációjára szavaz mindkét csoport. Ezzel az a gond, hogy így olyanok szavazata is beleszámíthat a szavazásba, aki adott esetben csak az egyik szavazásra szavazhatna. Így ezt a megoldást el kell vetni. Egy másik célravezetőbb mód, hogy a konfliktus definiálását egészen az opciók szintjéig le kell vinni. Ez azt takarja, hogy minden egyes opciópár rendelkezni fog egy számmal, ami a közöttük lévő konfliktus mértéke. (Definíció szerint ez a szám 1, ha a két opció megengedhető együtt, azaz nincs közöttük konfliktus, és 0, hogyha teljesen kizárják egymást, kettő közötti értéket is felvehet természetesen, a konfliktus mértékének arányában.) Erre a későbbiekben $\Delta(O_1, O_2)$ -ként fogok hivatkozni.

Figyelembe kell ugyanakkor venni, hogy egy szavazásra a jogosultak mekkora része szavazott, ez egy α arányossági tényezővel jellemezhető, ami a ténylegesen szavazottak és a szavazásra jogosultak számának aránya. Ugyanakkor bizonyos szám alatt nem lehet megkülönböztetni a csoportos döntést a sima egyéni döntésektől, így ha az arány egy előre definiált küszöb alatt van, akkor az α 0-nak tekintendő.

Két szavazás egymásra hatásának, az opciók közötti konfliktuson kívül, másik nagy mérhető értéke a szavazások távolsága. Ezt nem az opciók segítségével lehet meghatározni, hanem a szavazásra jogosultak számával. Minél többen szavazhatnak mindkét szavazásra, annál közelebb van két szavazás egymáshoz. Ezt a $\delta(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$ együtthatóval írható le.

Látszik, hogyha a két szavazás szavazói teljesen megegyeznek, akkor a tört értéke 1, így a távolság 0. Abban az esetben, amikor csak kevés közös szavazó van, a tört értéke 0-hoz közelít, így a távolság 1 közelebb lesz. Akár felfogható egy egyszerű metrikaként a két szavazás között.

Ezen három paraméter, tulajdonság alapján a következő képlet segítségével meghatározható a kapcsolt vagy más néven konfliktusos szavazások esetén is az optimális döntés.

$$A(X_i, Y_j, \dots, Z_k) = \alpha(X) \cdot \omega(X_i) + \alpha(Y) \cdot \omega(Y_j) + \dots + \alpha(Z) \cdot \omega(Z_k)$$

Ez az A függvény adja meg a konfliktusok figyelembevétel nélkül számítva az opcióhalmaz mérőszámát. Az α együtthatókhoz így lehet akár szavazásonként is küszöböt rendelni, de célszerű azonban a későbbi tapasztalatok alapján egy intézményszintű, vagy legalábbis szavazási csoportszintű együtthatót megállapítani.

$$\begin{aligned}
B(X_i, Y_j, \dots, Z_k) \\
&= (\delta(X, Y) \otimes \Delta(X_i, Y_j)) \cdot (\delta(X, Z) \otimes \Delta(X_i, Z_k)) \cdot \dots \\
&\cdot (\delta(Y, Z) \otimes \Delta(Y_j, Z_k))
\end{aligned}$$

A B függvény pedig a konfliktust számszerűsíti. Nullát ad eredményként, hogyha valamely két opció konfliktusban vagy egymással (mivel szorzat elég egy pár opció, és az egész nulla). A B képletben látható műveletet a következőképpen lehet definiálni: Ha a két operandus közül az egyik 1, akkor függetlenül a másik értéktől a művelet értéke 1, egyébként az értéke a két operandus szorzata. A $\delta(X, Y)$ akkor 1, hogyha az X és Y szavazás hallgatóinak halmaza teljesen diszjunkt. A $\Delta(X_i, Y_j)$ pedig akkor 1, hogyha a két opció megengedhető együtt. Így érthető, hogy akármelyik feltétel teljesülése esetén az adott opciópár a konfliktus értékét nem befolyásolja, így a szorzatba szükségtelen belevenni. Ha két opció teljes konfliktusban van, tehát kizárják egymást, akkor a $\Delta(X_i, Y_j)$ értéke 0 lesz, így az egész szorzat 0.

$$D(X_i, Y_j, \dots, Z_k) = A(X_i, Y_j, \dots, Z_k) \cdot B(X_i, Y_j, \dots, Z_k) \cdot R(X_i, Y_j, \dots, Z_k)$$

Az összegzőképlet (D) három részből áll. Ebből az első kettő a fenti képlet, a harmadik pedig speciális szabályok rögzítésére szolgál (R), ilyenek lehetnek például a külső/belső kényszerek.

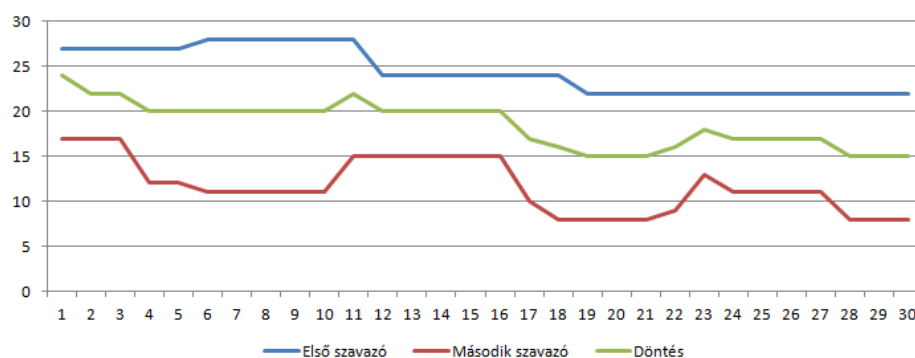
Esetenként előfordulhat az is, hogy a szavazók a döntési módszert ismerve igyekeznek a saját szavazatukkal túl- illetve alulszavazva az igazságosnál jobban befolyásolni a szavazás kimenetelét. Ez ellen a kiértékelés fázisában már nem lehet védekezni a felhasználó szavazatának részleges vagy akár teljes elvesztése nélkül. Ha azonban bent marad a kiértékelendő szavazatok között, akkor fals eredményt szolgáltatathat a döntési algoritmus. Rögtön adódik a lehetőség, hogy a szavazat beviteli módját korlátozva csökkenthető és csaknem teljesen megszüntethető a „csalás” lehetősége (szavazat- és ez által információvesztés nélkül).

Adott a példa, hogy egy iroda hőmérsékletét szeretnék az ott dolgozók demokratikusan beállítani. Ekkor szinte biztosan lesz olyan felhasználó, aki direkt „eltúlozza” a szavazatát, hogy akaratát jobban érvényesítse.

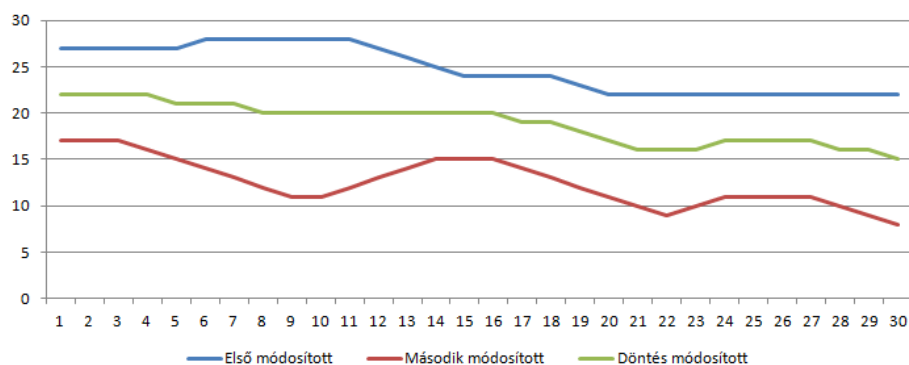
Az 5.1 fejezetben részletesen ismertetett módszer (a szavazatok, mint valószínűségi változók) itt nem jelent megoldást. A megoldás erre, hogy a felhasználók csak arra szavazhatnak, hogy melegebb legyen-e vagy hidegebb esetleg maradjon-e a jelenlegi hőmérséklet. Folyamatos kiértékelést választva, és szavazatot nem változtatva elérhető megfelelő után a jóval hidegebb hőmérsékletre való szavazás is. Nem utolsó sorban így a hőmérsékletigény változás is jóval egyenletesebb lesz, a klíma tervezhetőbb mennyiségű energiát fog felvenni. Ezáltal környezetbarátabbá válik, és az élettartama is nagyobb lesz.

A következő két ábrán látható az előbbi példa két szavazó esetében. Az első képen az az eset látható, amelyben a szavazók teljes szabadságot kaptak, nem voltak korlátozva. Megfigyelhető, hogy a szavazatokban ugrások találhatók, így a döntésben, a hőmérséklet függvényében is lesznek ugrások.

A második esetben az a megszorítás, hogy szavazók csak egy időegységben egy egységgel engedett meg a változás. Ezt kétféle módon lehet implementálni. Az egyik, hogy tetszőleges értékre szavazhatnak és minden időegységben csak a módosítás irányát veszi figyelembe az algoritmus. A másik lehetőség, hogy már csak arra szavazhat, hogy növekedjen, vagy csökkenjen a hőmérséklet. Érdekes azt is megvizsgálni, hogy mennyiben befolyásolja a szavazó döntését, hogy milyen formában szavazhat.



6. ábra Szavazatok és döntés, akkor hogyha nincs megkötés az értékekre



7. ábra Szavazatok és döntés, akkor hogyha van korlátozás

Az 7. ábra a 6. ábra adataiból készült, azzal a korlátozással, hogy a felhasználók csak egy egységgel módosíthatták szavazatukat (a plusz/mínusz szavazási rend szerint).

A szavazatok korlátozásának célja, hogy a szavazó a szükségesnél jobban ne tudja befolyásolni a szavazást. A fenti két ábra adatai mutatják, hogy mennyire szeszélyes szavazók vannak. A túl gyors véleményváltozás sehol sem jó, főleg akkor nem, hogyha valamilyen

elektromos eszközt kell működtetni a vélemény alapján. Ezért alkalmaztam egy szűrőt a másik esetben. Ez egy végtelenül egyszerű szűrő, megszabja csak, hogy egy időszakban hány egységgel módosíthatja el az eredményt. Látható a fenti képen, hogy azt az esetet implementáltam, amikor csak 1 egységgel változtathatja meg a szavazatát. Nagyon nagy haszna ennek a módszernek, hogy „simább” lesz mind a szavazatok grafikonja, mind a döntés grafikonja, így ha energiaként fogjuk fel, jelentős teljesítményingadozást akadályozhatunk meg, hűtés/fűtés le/fel kapcsolásával, ezzel nem csak a környezetnek, a felhasználóknak, hanem az elektromos hálózatnak is kedvez a döntés.

5. Konkrét alkalmazások implementációja

Az elméleti és algoritmikus bevezető után ebben a fejezetben két implementált alkalmazáson mutatom be a szavazás menetét. A fejezet utolsó felében az itt ismertetett módszerek, algoritmusok alkalmazásáról lesz szó Smart Grid rendszerek esetén.

5.1 Demokratikus hőmérsékletbeállítás – Tempocracy

Minden nagyobb légterű helyiségben előbb-utóbb felmerül az igény valamilyen hőmérsékletszabályozásra. Az ilyen irodák esetében ezt leggyakrabban légkondicionáló berendezésekkel oldják meg, amik a hűtésen kívül fűteni is tudnak. Ezeken általában egy hőmérsékletet lehet beállítani, valamint ezt szinten tartani.

A legnagyobb gondot az jelenti, hogy a helyiségben tartózkodó személyek nem mindig tudnak megegyezni, hogy ezt a hőmérsékletet mire állítsák be. Ebben nyújt segítséget a Tempocracy alkalmazás.

Első lépésként pontosan definiálni kell a szavazók körét. Mivel eléggé helyhez kötött felhasználásról van szó, így célszerű valamilyen fizikai korlátot meghatározni. Ilyen lehet például, hogy csak az adott helyiségben tartózkodók szavazhatnak. Ma már a technológia lehetőséget ad rá, hogy ezt egyszerűen megvalósítsuk (pl. QR-kód, WiFi alapú lokalizáció, NFC). Különösen fontos ez abban az esetben, hogyha a felhasználók nem asztali számítógépen keresztül szavaznak, hanem valamilyen hordozható eszközön (laptop, mobiltelefon). Célszerű akkor is figyelembe venni ezeket a fizikai korlátozó lehetőségeket, amennyiben az adott helyiségben több külön-külön vezérelhető légkondicionáló berendezés is található. Ebben az esetben a két eszköz által szabályozott légtereket, és így a szavazókat, szavazatokat is célszerű szétválasztani, külön-külön kezelni és kiértékelni.

A szavazás előkészítésének fontos szerepe a hőmérséklet-tartomány meghatározása. Ezt a helyiségben tartózkodóknak célszerű meghatározni. Értelemszerűen azért figyelembe kell venni a légkondicionáló teljesítményét, valamint energiagazdálkodási szempontokat is.

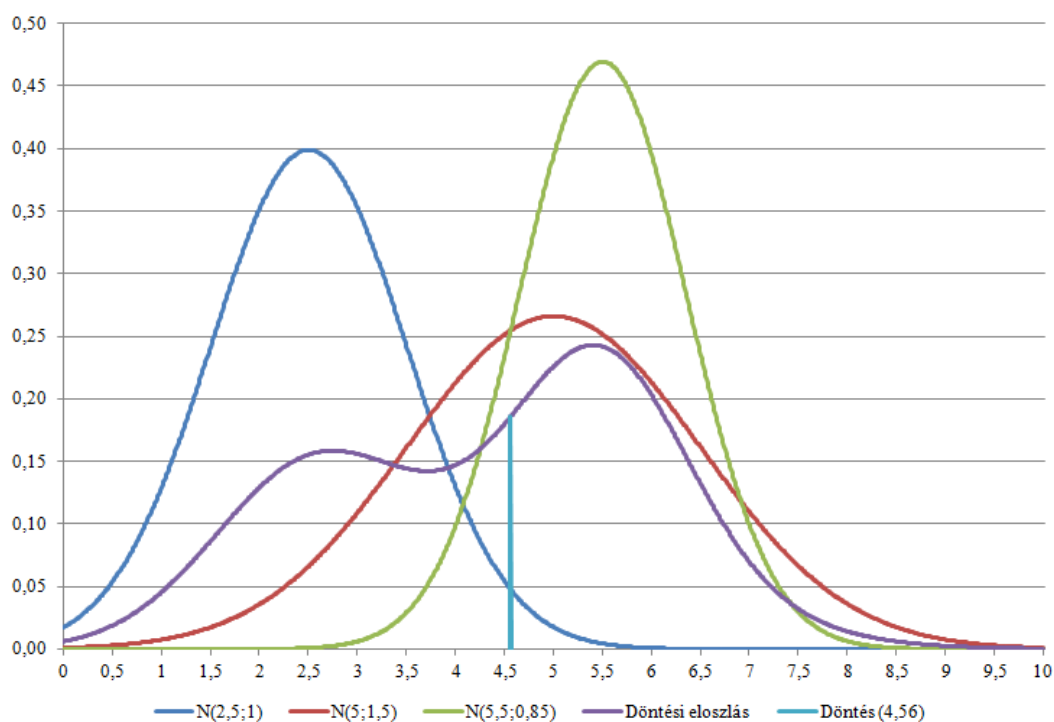
Szavazatoknál felmerül a kérdés, hogy mire is szavaz a felhasználó. Első megközelítésben az általa kívánt hőmérsékletre. Egy másik lehetőség lehet, hogy a változtatásra szavaz, nevezetesen, hogy hidegebb-, melegebb legyen vagy maradjon ugyanakkora a hőmérséklet. Hogy a felhasználó ne csak egy konkrét értékre tudjon szavazni, létezik olyan opció is, hogy a szavazat, mint valószínűségi változót jelenik meg a kiértékelésnél, amelynek van egy bizonyos sűrűségfüggvénye (ez a felhasználó elégedettségfüggvénye is egyben), várható értéke (a kívánt hőmérséklet a szavazó szempontjából) valamint szórása (a felhasználó

hőmérsékletváltozás tűrőképessége). Ezt a „folytonos” szavazatot először mintavételezéssel diszkrétizálni kell, majd ekkor már alkalmazhatóak rá a fentebb ismertetett eljárások.

A döntés megvalósításában fontos szerepe lesz annak, hogy melyik szavazati módot választja a szavazás kiírója.

Legkézenfekvőbb döntési eljárás, ami itt alkalmazható, az a sima átlagolás. (Bizonyítható, hogy az optimálistól való eltérés négyzetes hibája akkor lesz a legkisebb, hogyha egyszerű számtani átlaggal számítjuk ki a döntés eredményét.) A hőmérsékleti értékek egyben számértékek is, így matematikai akadály a műveletnek. Lehetőség van itt is súlyozásra, bizonyos szavazatok előnyben részesítésére.

Kicsit bonyolultabb a helyzet, hogyha a szavazat valószínűségi változó. Ekkor az egyes szavazatok sűrűségfüggvényeit mintavételezni kell, majd így a hőmérsékletekhez (tartományok) rendelt számokat (tartományba tartozás valószínűsége) lehet átlagolni. A döntés eredménye itt a hőmérsékletek súlyozott átlaga lesz, a súlyok pedig az adott tartományba esés valószínűsége (8. ábra). Jelen példában mindegyik valószínűségi változó normális eloszlású, de a módszer tetszőleges eloszlás esetén alkalmazható.



8. ábra Szavazás valószínűségi változóval

A felhasználók saját szavazatukkal akár a szükségesnél jobban is befolyásolhatják a szavazás eredményét. Ezt természetesen célszerű elkerülni, így át kell tekinteni a próbálkozások

kiküszöbölésének egyszerűbb módjait. Elképzelhető, hogy a szavazó jóval hidegebb/melegebb hőmérsékletre szavaz, mint ami neki az elvárt lenne, azért hogy az átlagos kiértékelésnél jóval nagyobbnak számítsa az ő szavazata. Például adott két szavazó, ők rendre a 20°C, 22°C-os hőmérsékletre szavaznak. A harmadik szavazó 18°C-ot szeretne, tisztában van vele, hogy a többi szavazó milyen hőmérsékletre adta le szavazatát, így egyszerűen ki tudja számolni, hogy 12°C-ra kell szavaznia, hogy döntés neki kedvezzen. Ez azonban 6°C-os eltérés az eredetihez képest. Ezt megakadályozni nem lehet teljes mértékben, maximum csökkenthető az előfordulása, például a szavazási mód változtatásával (hidegebbre, melegebbre vagy helybenhagyásra lehet szavazni), a szavazási tartományok meghatározásával vagy pedig a döntési eljárásoknál a kiugró szavazatok mellőzésével. Bővebben információ erről a 4.3 fejezetben található.

5.2 Szavazat alapú időpontegyeztetés – Timeocracy

A rohanó világban egyre fontosabbá válik a pontos időbeosztás. Az emberek nem érnék rá tervezgetni, szeretnék megspórolni azt az időt is. Ehhez már márig rengeteg segédeszköz áll rendelkezésükre, mint az a 3.2 részben olvasható. Azonban közös hiányosságuk, hogy nem használhatóak két részben egyező csoport egy-egy találkozásának egyszerre történő megszervezésére, azaz nem tudják az összefüggő szavazásokat kezelni. Ez az a kulcsponthely, ami miatt a Timeocracy-nak van helye a palettán. A Timeocracy egy időpont egyeztető, jelen speciális esetben ZH időpont egyeztető alkalmazás.

Minden félév végén, ZH időszakban, felmerül a probléma, hogy mikor is legyenek a ZH-k. Eddig az évfolyamfelelősök feladata volt az időpontszervezés/összeegyeztetés, immáron azonban rendelkezésükre áll egy intelligens alkalmazás is.

Egy-egy tárgynál nem ritkán több évfolyamot érintenek a ZH időpontok, így a kezelendő hallgatók száma is gyorsan százas, akár ezres nagyságrendeket vehet fel. Világos, hogy ennyi embert nem lehet közvetlenül megkérdezni az elfoglaltságukról, ha esetleg mégis, akkor is gondot okozna a rendszerezés, így kénytelen az ember valamilyen technikai megoldást bevetni.

Sok alkalmazásnál a szavazók két opció közül választhatnak, az igen illetve a nem közül. Ezzel a módszerrel az a legnagyobb baj, hogy a felhasználók nem tudnak kategorizálni az igen, illetve a nem csoporton belül. Ez adott esetben fals szavazatokkal is járhat, a két csoport határán található eseteknél. Ezért itt a jól megszokott igen/nem szavazás helyett lehetőség van arra, hogy az opciókat külön-külön is értékeljük. Ez azért fontos, mert így nem veszik el a

többi opcióra vonatkozó szavazat, információ. Minél több információ áll a döntő algoritmus rendelkezésére, annál jobb, a felhasználók szavazataihoz igazodó, döntést tud hozni.

A szavazáshoz nem kell külön jelentkezni, minden félév elején a NEPTUN rendszer adatainak felhasználásával egy automata hozza létre a szavazásokat, hozzárendeli a hallgatókat, testreszabható azonban a szavazások neve, leírása, lejárat ideje. Természetesen a lehetséges (szavazható) időpontokat a tárgyfelelős oktatók határozzák meg.

Amennyiben az egyes szavazások között kapcsolat van, akkor meg lehet adni a konfliktus mértékét is opciónként. Fontos, hogy opciónként és nem szavazásonként. Ugyan a konfliktus a két szavazás közös hallgatói miatt jelenik meg, ugyanakkor mégis az opciók (időpontok) szintjén lehet definiálni. Ehhez lehet automatikus kitöltőket használni (bizonyos szabály szerint számszerűsíti a konfliktust), de manuálisan beírva is megadhatóak a konfliktusértékek, mint ahogy azt a 9. ábra mutatja. Természetesen a szavazásra jogosult hallgatók száma is beleszámít a konfliktusba, de a fő szempont mégis az opciók közötti konfliktus (ez lehet akármilyen természetű, ebben az esetben időpontok közötti átfedés). Eddig tart az előkészítési fázis, látható, hogy ez rendkívül jól automatizálható, és gyorsan végrehajtható.

A következő fázisban a fő szerep a szavazókra hárul. Ez leghosszabb, hiszen nyilván elég időt kell hagyni a szavazásra. A hallgatók értesítést kapnak a számukra elérhető új szavazásokról, az egyetemi azonosítójukat használva bejelentkezhetnek a rendszerbe. Ha úgy ítéli meg a hallgató, akkor szavaz, ha nem akkor nem. Az egyéni szavazat itt a lehetséges időpontok (megfelelő opciók) súlyozása egy 0-tól 10-ig terjedő skálán (csúszka segítségével lehet szavazni). Többször is szavazhat egy hallgató, viszont mindig csak az időben legutolsó érvényes szavazata számít bele az eredménybe, szavazatot azonban nem tud törölni. A szavazás lezárultát követően értelemszerűen már nem tud szavazni. Az olyan hallgatók, akik nem tartoznak bele a csoportba (akiknek a szavazást kiírták) nem szavazhatnak, sőt nem is látják a szavazást.

A kiértékelés során többféle kiértékelési mód szerinti eredményt is meg lehet tekinteni. Szavazás közben is lehetőség van az alapértelmezett kiértékelési mód megváltoztatására (a 2.2.1 fejezetben említettek szerint), azonban az opciókat az első szavazat beérkezését követően már nem lehet módosítani. A szavazás lezárultával érdemes még megvizsgálni, hogy a hallgatóknak mennyire jó az eredmény. Ez akár alapot is adhat a következő ZH időpont tervezésénél. A szavazás a győztes időpont kihirdetésével ér véget.

A hallgató ezután természetesen megtekintheti másik szavazásait is, valamint szavazhat azokra is.

Ha az adott szavazás valamely más szavazásokkal kapcsolat/konfliktus áll fent, akkor természetesen kisebb lesz a hallgatók elégedettsége, de ezt a kompromisszumos árat kell azért

fizetni, hogy több szavazást egyszerre vizsgáljunk, és az eredményeket összehangoljuk. Azonban könnyen elképzelhető, hogy a hallgató összelégedettsége nagyobb, mint abban az esetben, hogyha konfliktusok nélkül vizsgálnánk a szavazásokat.

TIMEOCRACY
[szavazások](#) | [kijelentkezés](#)

Mobilhálózatok ZH

Kedves Hallgatók!

A gyakorlaton elhangzottaknak megfelelően összegyűjtöttem pár lehetséges időpontot a ZH megírására. Kérem szavazzanak.

Üdvözlettel:
 Oláh András

H 10-12
 (2012-05-21 10:00 - 2012-05-21 12:00)
 2.19

CS 10-12
 (2012-05-24 10:00 - 2012-05-24 12:00)
 7.32

P 8-10
 (2012-05-25 08:00 - 2012-05-25 10:00)
 3.41

[szavazok](#)

© 2012

TIMEOCRACY SERVER
[regisztráció](#) | [bejelentkezés](#) | [kijelentkezés](#)
[szavazás hozzáadása](#) | [szavazások](#)

Konfliktusok

Programozási nyelvek ZH

Ütközések elkerülése
Csak egy zh legyen egy nap

H 10-12 | CS 10-12 | P 8-10

H 8-10	☒	☐	☐
K 9-11	☐	☐	☐
Sz 10-12	☐	☐	☐
Cs 9-11	☐	☒	☐
P 8-10	☐	☐	☒

[Tovább](#)

© 2012

9. ábra Példa egy szavazásra, konfliktusok definiálása

5.3 Smart Grid

A hagyományos elektromos hálózatok egyetlen feladata volt, hogy néhány központi generátortól áramot szállítsanak nagyszámú előfizetőnek. Ezzel szemben a Smart Grid egy olyan intelligens hálózat, melyben a hatékony energiafelhasználás, a környezetkímélőbb energiatermelés modern infokommunikációs eszközökkel, és fejlett algoritmusokkal támogatott [30].

Az energetika, már az ipari társadalom kialakulása óta kiemelt szektor, amely jelenleg alapvető modernizációs változások előtt áll. Ezek a változások a következők: a centralizált energiatermelő-vezérelt elosztó hálózatból egy kevésbé centralizált fogyasztó fókuszú hálózattá való átalakulás. A formálódó új hálózati struktúra kommunikáló, intelligens elemei

egymással együttműködve biztosítják a rendszer hatékonyságát. A smart grid intelligenciáját a hálózathoz csatlakozó infrastrukturális elemeken (pl.: a fogyasztóknál smart meter-en) implementált algoritmusok jelentik. Az Smart Grid elosztott számítási és kommunikációs kapacitásában megjelenő intelligenciája, komplexitása (és idővel hozzáférhetősége is) hasonlatos az internetéhez, ezért várhatóan az energiatermeléshez, elosztáshoz és -fogyasztáshoz nem szorosan kapcsolódó alkalmazásoknak is teret fog adni (pl.: elosztott számításokon alapuló, cloud computing alkalmazások).

A döntéshozatali módszerek fontossága a kontrol rétegben kerül elő. Fontos része a kontrol réteg működésének a felhasználói visszajelzés is a költségoptimalizálás mellett. Nevezetesen, hogy mennyire volt elégedett a felhasználó a szolgáltatással, vagy adott esetben mennyire akadályozta a napi teendői elvégzésében (az eseteleges lassulások, energiahiányok miatt elcsúszó teendők, mint például az időzített mosás). Ennek a két szempontnak, amiknek segítségével értékelhető, továbbfejleszthető a Smart Grid, külön-külön nagy szakirodalma van (például: [31], [32]), azonban meglehetősen kevés kutató foglalkozott a két szempont egyesítésével, így jelenleg is aktuális téma, további kutatást igényel.

6. Teljesítőkéesség analízis a Timeocracy alkalmazásban

Egy alkalmazás elsődleges jellemzője a használhatósága. Ez a legobjektívebben egy teszttel mérhető le. A Timeocracy időpontok egyeztetésére használható, adódott a lehetőség, hogy a ZH időszak közeledtével az alkalmazás élőben is bizonyítson.

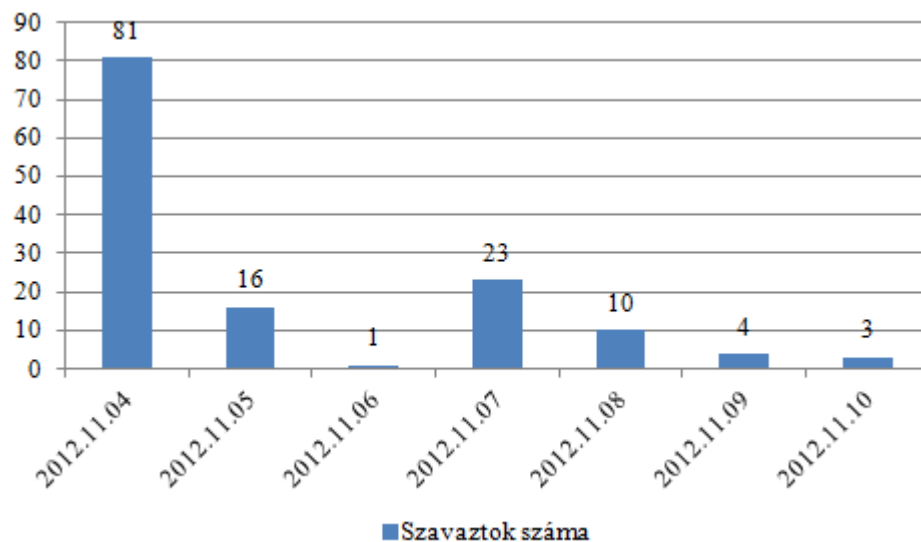
A szavazás előszítése egy kicsit tovább tartott, mint amire számítani lehetett. Az évfolyamfelelősökkel történő konzultáció után a következő öt tárgyat választottam ki a tesztre: Számítógépes hálózatok (SZH), Infocommunication Systems (ICS), Introduction to Neurobiology (ItN), Information and Coding Theory (IaCT) és a Neural Networks (NN). Ezek a tárgyak harmadéves tárgyak, azonban sokan hallgatják még ezeket negyedévben is. Így már rögtön adódott egy probléma. Valamilyen szinten azt is figyelembe kell venni, hogy a tárgyat hallgatók közül nem mindenkinek ugyanolyan az órarendje, adott esetben az egyéb órákon kívüli elfoglaltságai is más időpontban vannak. A tárgyfelelős oktatók négy-négy időpontot adtak meg mindegyik tárgyhöz december első hetére. Ezeket az időpontokat a rendszerbe betöltve létrehoztam a szavazásokat. A megadott időpontok két órás sávokban voltak, a következő módon:

6. táblázat A lehetséges ZH időpontok

	Hétfő 2012.12.03.	Kedd 2012.12.04.	Szerda 2012.12.05.	Csütörtök 2012.12.06	Péntek 2012.12.07
10:00 – 12:00	SZH, ICS	ItN	SZH, ICS, NN	ICS	ICS
12:00 – 14:00				ItN	
14:00 – 16:00	ICS, ItN	SZH, IaCT			IaCT, NN
16:00 – 18:00	NN		IaCT, SZH, ItN		NN

A következő lépés a konfliktusok definiálása. Egyértelmű, hogy egy hallgató egy időpontban nem tud két ZH-t írni, így az egy időpontban lévő ZH lehetőségek automatikusan kizárják egymást (így ezekhez a párokhoz 0-t célszerű rendelni). Abban az esetben, hogyha két opció egy napon is van, de nem ütköznek, akkor a konfliktusukat egy 0 és 1 közötti számmal jellemeztem. Jelen esetben függ ez a szám a ZH-k nehézségétől, valamint attól, hogy mennyi idő telik el a két ZH között. Mindegyikkel fordítottan arányos. Ebben a konkrét testben, az egyszerűség kedvéért 0.5 lett a konfliktus értéke, hogyha két opció ugyanazon a napon van.

A konfliktusok beállítása, és a hallgatók hozzáadása után elindulhatott az igazi szavazás. A hallgatóknak az első értesítést november 4.-én küldtem ki, a másodikat pedig november 7.-én. Az alábbi ábrán látható a szavazatok beérkezésének intenzitása, egyre csökkenő trendű, érvényesül az, hogyha nem azonnal szavaz a hallgató, akkor elfelejti, és egyáltalán nem szavaz. Azonban így is az értesített 150 hallgatóból 62-en szavaztak. Mivel ez egy tesztszavazás volt, így nem meglepő, hogy a hallgatóknak csak alig több mint 40% szavazott, és ez által vett részt a döntésben. Nincs arra lehetőség, hogy a nem szavazó hallgatók is figyelembe legyenek véve, ugyanis azon kívül, hogy nem szavaztak nem közvetítettek semmilyen értékelhető információt a döntéshozó algoritmus felé. Ez a 62 hallgató összesen 138 szavazatot adott le, ez a körülbelül kéttárgyas átlag megfelel az eredeti átlagnak, miszerint átlagosan két tárgyban érintettek a hallgatók. Így kijelenthető, hogy a szavazó hallgatók minden olyan tárgyra szavaztak, amiben érintettek voltak, azaz nem hagyták félben a szavazást.



10. ábra Szavazatok száma naponként

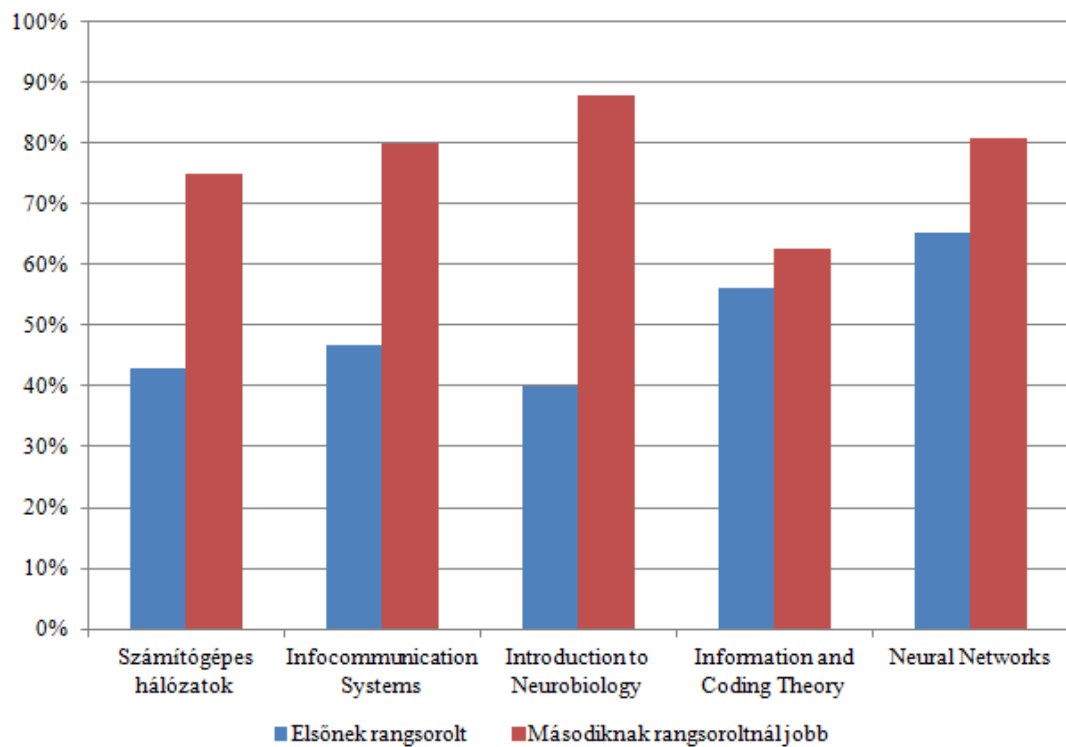
A szavazások november 13.-án kerültek lezárásra. Kiértékeléshez a 2.2.1 fejezetben tárgyalt döntéshozó algoritmusokat használtam. Nem túl meglepő módon ezek közül a sima átlagolási módszer, a Borda valamint a Condorcet a következő eredményt adta (a kiválasztott időpontot félkövér kiemelés jelzi):

7. táblázat Az átlagolási, Borda, Condorcet döntések eredménye

	Hétfő 2012.12.03.	Kedd 2012.12.04.	Szerda 2012.12.05.	Csütörtök 2012.12.06	Péntek 2012.12.07
10:00 – 12:00	SZH, ICS	ItN	<u>SZH</u> , ICS, NN	ICS	<u>ICS</u>
12:00 – 14:00				<u>ItN</u>	
14:00 – 16:00	ICS, ItN	SZH, IaCT			<u>IaCT</u> , NN
16:00 – 18:00	<u>NN</u>		IaCT, SZH, ItN		NN

Látható a táblázatban, hogy az algoritmusnak sikerült elosztania az öt időpontot négy napra, a szavazók preferenciájának megfelelően. A kimaradt módszer, a többségi módszer annyiban hozott más eredményt, hogy a csütörtöki Introduction to Neurobiology ZH-t keddre ütemezte (az ehhez tartozó elégedettség értékek rendre: 44%, 76%). Mivel ez egy tesztszavazás volt, így az eredmény kihirdetése elmaradt.

A szavazás lezárásának fontos lépése még az elégedettség mérése. Erre egy kézenfekvő módszer még egy szavazás kiírása, ahol a felhasználók értékelik a szavazást akár több szempont alapján (megfelel ennek a 4.1 fejezetben lévő egyik scenárió). Itt erre most nincs lehetőség így a felhasználók egyéni szavazatait kell összehasonlítani a döntéssel. A 11. ábra mutatja a felhasználók elégedettségét tárgyaként. Az első oszlop azt jelzi, hogy a felhasználók hány százaléka sorolta első helyre a később kiválasztott időpontot. Látható, hogy 50% környékén mozog, ami azt jelenti, hogy körülbelül a szavazók felének teljesült az akarata, ez egy demokratikus rendszerben általában elég. Hozzászámolva azt, hogy itt 5 szavazást kell összehangolni, nagyon jó eredmény. A második oszlopban a szavazók aránya látható, akik a döntési módszer által meghatározott opció legfeljebb a második helyen végzett. Ez az arány jóval jobb, némely esetben elérheti a 90%-ot is. Alapvetően elmondható tehát, hogy az alkalmazás figyelembe véve a szavazók preferenciáját és az opciók egymásra gyakorolt hatását, egy kompromisszumos, jó döntést adott.



11. ábra A szavazók elégedettsége

Végeztem egy terhelés tesztet is, 50-75 szavazat/másodperces gyorsaságnál kezdett el a rendszer lassulni, érezhetően visszaesett a teljesítménye. Ez részben betudható az alkalmazott technológiáknak (fájl alapú adatbázis, SQLite, PHP). Továbbfejlesztési lehetőség az eddigi szűk keresztmetszet (adatbázis) lecserélése valamilyen jobb, skálázhatóbb rendszerre.

A teszt sikeres volt, célját, hogy valós képet adjon az alkalmazás teljesítőképességéről, elérte. A hallgatók, illetve a Tanulmányi Osztály is nyitott erre az újfajta ZH összeegyeztető lehetőségre, így valószínűleg hamarosan élesben is működhet a rendszer.

7. Összefoglalás és továbblépési lehetőség

A dolgozat a kollektív döntéseken alapuló alkalmazások gyakorlati megvalósíthatóságának algoritmikus és implementációs kérdéseivel foglalkozott, felvonultatva a már elérhető kollektív döntést támogató algoritmikus módszereket, valamint a kommerciálisan elérhető rendszereket. A dolgozatban bemutatott konkrét eredmények a következők:

- a konszenzusos döntést igénylő szerteágazó **alkalmazások klasszifikálása**, a szcenárió osztályok kidolgozása;
- egy **univerzális** kollektív döntések meghozatalát támogató **architektúra** kifejlesztése;
- két kiválasztott szcenárió esetén a **döntéshozó rendszerek teljes implementációja** a kidolgozott általános architektúrán (Timeocracy, Tempocracy);
- az implementált rendszerekben a speciális alkalmazási követelmények szerint **módosított ill. kiegészített algoritmikusok** lettek kifejlesztve;
- felhasználók (ITK hallgatók) bevonásával valós körülmények **között átfogó teljesítőképesség analízis** készült a Timeocracy alkalmazáshoz.

A dolgozatban bemutatott eredmények egy átfogó kutatási projekt első lépcsőfokának tekinthetők, amelyről a továbblépés a következő irányokba teljeshet ki:

- **algoritmikus lehetőségek:** a dolgozatban bemutatott algoritmikus eszköztár tovább bővíthető a kollektív döntésmélethez ismert módszerekkel, kiegészítve az egyes szcenáriók speciális alkalmazási követelményeivel (pl.: költségfüggvény vezérelt döntés, szavazatsúlyozási módszerek);
- **szcenárió osztályok kiterjesztésének ill. pontosításának lehetőségei:** az egyik első ilyen feladat a Smart Gridekben alkalmazott kvadrátikus hő komfortmodell kiváltása egy szavazat alapú klímavezérlő rendszer integrálásával, amelyen keresztül a szolgáltatást használók konszenzus alapú döntése a smart metering eszközön keresztül közvetlenül vezérelheti az épületautomatikát.

implementációs lehetőségek: a szavazatgyűjtő kliens alkalmazás implementálása mobil platformokon (iOS, Android, Windows phone), a szerveroldalon teljesítményének növelése, hogy jóval magasabb felhasználószám esetén is megbízhatóan működjön;

- **döntéselőkészítéssel kapcsolatos lehetőségek:** egyes szcenáriók esetén szükséges döntéselőkészítő folyamat – az elérhető információk intelligens előfeldolgozásával – automatizálható.

8. Köszönetnyilvánítás

Köszönettel tartozom témavezetőmnek Dr. Oláh Andrásnak, hogy észrevételeivel, javaslataival segítette a szakdolgozat elkészülését, Csányi Márknak és Nemes Norbertnek, hogy a rendszer tervezésében segédkeztek, Mihálffy Andrea Tanulmányi Osztály vezető asszonynak, Nagy Balázs és Kovács Lóránt évfolyamfelelősöknek, hogy az alkalmazásanalízis gördülékeny lebonyolításában segítségemre voltak.

Irodalomjegyzék

- [1] L. Buchanan and A. O'Connell, "A brief history of decision making," *Harvard Business Review*, vol. 84, no. 1, pp. 32-41, Jan. 2006.
- [2] Sven Ove Hannon. (1984) Decision theory. [Online].
<http://www2.cs.siu.edu/~hexmoor/classes/CS539-F10/DT.pdf>
- [3] James T. Reason, *Managing Risks of Organizational Accidents.*: Ashgate Publishing, Limited, 1997.
- [4] J. Nightingale, *Think Smart - Act Smart: Avoiding The Business Mistakes That Even Intelligent People Make.*: John Wiley & Sons Inc., 2007.
- [5] Márk Csányi, "A kollektív döntésmélet gyakorlati alkalmazása," PPKE-ITK, 2012.
- [6] P. M. Larsen, "Industrial applications of fuzzy logic control," *International Journal of Man-Machine Studies*, vol. 12, no. 1, 1980.
- [7] László T. Kóczy and Domonkos Tikk, *Fuzzy rendszerek.*: Typotex Elektronikus Kiadó, 2000.
- [8] M. Delgado, F. Herrera, E. Herrera-Viedma, and E. Martínez, "Combining Numerical and Linguistic Information in Group Decision Making," ETS de Ingeniería Informática, Universidad de Granada, #DECSAI-96115, 1996.
- [9] Michel Grabisch, "The application of fuzzy integrals in multicriteria decision making," *European Journal of Operational Research*, 1996.
- [10] David Bellhouse, "The Problem of Waldegrave," *Journ@l Electronique d'Histoire des Probabilités et de la Statistique*, vol. 3, no. 2, December 2007.
- [11] Ferenc Forgó, Miklós Pintér, András Simonovics, and Tamás Solymosi. (2006, November) Kooperatív játékelmélet. [Online].
http://www.inf.unideb.hu/valseg/dolgozok/buraip/solymosi_jatekelmelet.pdf
- [12] John F. Nash, "Equilibrium Points in n-Person Games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, 1950.
- [13] Paul H. Robin, *Managing business transactions : controlling the cost of coordinating, communicating, and decision making.*: New York : Free Press , 2006.

- [14] Amir Albadvi, S. Kamal Chaharsooghi, and Akbar Esfahanipour, "Decision making in stock trading: An application of PROMETHEE," *European Journal of Operational Research*, vol. 177, no. 2, pp. 673-683, Mar. 2007.
- [15] Pavel Sevastjanov and Ludmila Dymova, "Stock screening with use of multiple criteria decision making and optimization," *Omega*, vol. 37, no. 3, pp. 659-671, June 2009.
- [16] James W. Dean and Mark P. Sharfman, "The Relationship between Procedural Rationality and Political Behavior in Strategic Decision Making," *Decision Sciences*, vol. 24, no. 6, pp. 1069-1083, June 2007.
- [17] Matthew James Kushin and Masahiro Yamamoto, "Did Social Media Really Matter? College Students' Use of Online Media and Political Decision Making in the 2008 Election," *Mass Communication and Society*, vol. 13, no. 5, pp. 608-630, 2010.
- [18] Paul W. Glimcher, Colin F. Camerer, Ernst Fehr, and Russell A. Poldrack, *Neuroeconomics: Decision making and the brain*, 1st ed.: Academic Press, 2009.
- [19] James K. Rilling and Alan G. Sanfey, "The Neuroscience of Social Decision-Making," *Annual Review of Psychology*, vol. 62, pp. 23-48, Jan. 2011.
- [20] Nimmo GR Croskerry P, "Better clinical decision making and reducing diagnostic error," *The Journal of the Royal College of Physicians of Edinburgh*, vol. 41, no. 2, pp. 155-162, 2011.
- [21] Marianne K Dees et al., "Perspectives of decision-making in requests for euthanasia: A qualitative research among patients, relatives and treating physicians in the Netherlands," Oct. 2012.
- [22] Ingrid Lunden. (2011, Sep.) Symbian now officially no longer under the wing of Nokia. [Online]. <http://paidcontent.org/2011/09/30/419-symbian-now-officially-no-longer-under-the-wing-of-nokia-2300-jobs-go/>
- [23] (2011, Apr.) Symbian Browser and Web Runtime 7.3 for Developers. [Online]. [http://www.developer.nokia.com/Community/Wiki/Symbian Browser and Web Runtime 7.3 for Developers](http://www.developer.nokia.com/Community/Wiki/Symbian_Browser_and_Web_Runtime_7.3_for_Developers)
- [24] [Online]. <http://www.developer.nokia.com/Devices/Symbian/>
- [25] [Online]. <http://www.nokia.com/gb-en/products/nokia-for-business/nokia-belle/security/symbian-security/>

- [26] Blackberry security. [Online].
http://uk.blackberry.com/business/topics/security.html#tab_wireless_data
- [27] [Online]. <http://www.apple.com/iphone/business/>
- [28] (2012, Oct.) iOS security. [Online].
http://images.apple.com/iphone/business/docs/iOS_Security_Oct12.pdf
- [29] Open Handset Alliance. [Online]. <http://www.openhandsetalliance.com/>
- [30] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang, "Smart Grid – The New and Improved Power Grid: A Survey," 2011.
- [31] Pedram Samadi, Amir-Hamed Mohsenian-Rad, Robert Schober, Vincent W.S. Wong, and Juri Jatskevich, "Optimal Real-time Pricing Algorithm Based on Utility Maximization for Smart Grid".
- [32] Chen Wang and Martin de Groot, "Managing end-user preferences in the smart grid," in *e-Energy '10 Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, New York, 2010, pp. 105-114.
- [33] Kathleen M. Eisenhardt and Mark J. Zbaracki, "Strategic decision making," *Strategic Management Journal*, Feb. 2007.