SZEGEDI TUDOMÁNYEGYETEM
Természettudományi és Informatikai Kar
Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék

Informatika Doktori Iskola

Döntési modellek és fuzzy optimalizálás

Doktori értekezés tézisei

Vincze Nándor

Témavezető:

Dr. Dombi József

Szeged, 2018

1 Bevezetés

A kockázatokat és bizonytalanságokat kezelő preferenciastruktúrákat széles körben alkalmazzák a különböző döntési problémák megoldása során. Ezek kidolgozásakor komoly figyelmet kapnak azok a döntéshozói megfontolások, amelyek figyelmen kívül hagyják a hasznosságelmélet Neumann-Morgenstern féle megalapozásának elveit, így például a preferenciák tranzitivitása. Ez utóbbi modellek kidolgozása más megközelítést igényel mint a klasszikus hasznosságelmélet.A többtényezős döntések elmélete a humán döntések modellezésére is szolgál és a szociológiai és pszichológiai vizsgálatok azt mutatják, hogy humán viselkedés korlátozottan racionális. Például az döntési eljárás eredménye sok esetben nemtranzitív. Fishburn [14] konstruált először a preferencia tranzitivitásának elvét elvető hasznosságelméleti modellt. Itt a preferenciát egy ferdén szimmetrikus bilineáris funkcionállal adja meg.
A dolgozat első fejezetében megadjuk a nemtranzitív preferenciastruktúrák általános jellemzését a Fishburn-féle modellben.
A lexikografikus döntési módszert nagyon elterjedt mivel a döntési szabályok nagyon egyszerűek és jól interpretálhatóak. Egyik alkalmazási területe pályázatok elbírálása [34], fő tulajdonsága a döntési sorrend gyors kialakíthatósága. A lexikografikus döntés koncepciója a kritériumok rendezett halmazát feltételezi. A döntési alternatívákat először az első rendezésben legelső kritérium alapján vizsgáljuk meg. Ha alternatívák közötti preferencia sorrend ezen kritérium alapján nem dönthető el, akkor a rendezésben második kritérium alapján hasonlítjuk össze. Ezt az eljárást folytatjuk míg az alternatívák a teljes rendezését el nem érjük. A lexikografikus döntési módszert a "rendezés az első differencia alapján" elvének is nevezzük. A dolgozat

második fejezetében a lexikografikus döntési módszert egy általános keretrendszerbe beágyazva konstruáljuk meg, amelyet Dombi [4][5] adott meg az outranking módszerekre. Ebben a lexikografikus döntést általános preferenciafüggvény és egy egyváltozós függvény súlyozott kompozíciójával modellezzük, beillesztve így az outranking módszerek közé.
A harmadik fejezetben a lexikografikus döntést vizsgáljuk [13]. A többtényezős optimalizálási modelleknél a kritériumok lineárisan rendezettek, az optimális megoldást pedig a lehetséges megoldások lexikografikus rendezése alapján definiáljuk. Használják ezenkívül a lineáris programozás módszerében a szimplex algoritmus egyik hatékonyan alkalmazható változatában. A döntéstámogató modellek és keretrendszerek [22][23][24][25][26][27][28] különféle paraméterei esetében fontos ezen paraméterek pontos jelentése és könnyű kezelésének lehetősége a felhasználó számára. A lexikografikus döntés a kritériumok lineáris sorrendjén alapul. A dolgozat harmadik fejezetében arra adunk algoritmikus megoldást hogyan lehet megtanulni a lexikografikus döntési módszer kritériumainak sorrendjét és vizsgáljuk a tanulás feltételeit.
A kritikus út módszere (CPM) a projekttevékenységek ütemezésének széles körben használatos algoritmikus megközelítése. A CPM módszer a tevékenységek megvalósítási idejét és a tevékenységek egymás közötti logikai kapcsolatait használja fel. Ezekből a paraméterekből adható meg az időbeli ütemezést reprezentáló ütemterv háló. A folyamathálózatok módszertanát Friedler és mások [18][19] vezették be vegyipari gyártási folyamatok modellezésére. A dolgozat második felében megadjuk a CPM módszertan és a folyamathálózat módszertan közötti transzformációt, mellyel CPM problémák folyamathálózati reprezentációjának lehetőségét adjuk meg. Tevékenységekhez

adunk meg alternatívát a CPM probléma folyamathálózatban történő reprezentációjában, így a proléma kiterjesztését adjuk meg.
Megadunk időoptimális, költségoptimális, időkorlátos költségoptimális, költségkorlátos időoptimális matematikai modellt, melyeket egy példán szemléltetünk.
A CPM problémákhoz hasonlóan a folyamathálózatokban is sok esetben kell kezelnünk bizonytalan idő vagy költségparamétereket. A bizonytalanság kezelésének a fuzzy elmélet jól kidolgozott és egyik leggyakrabban használt módszere. Olyan területeken bizonyított az alkalmazhatósága, hatékonysága mint az irányításelmélet [7]. Egy új két lépéses fuzzy lineáris optimalizálási koncepció kidolgozásával dolgozatunk ötödik fejezetében megmutatjuk, hogy a fuzzy elmélet sikeresen alkalmazható bizonytalan idő és költségparaméterű folyamathálózatok optimalizális feladatainak megoldására.
Három különböző modell megadásával, a modelleken belül időoptimális, költségoptimális, időkorlátos költségoptimális, költségkorlátos időoptimális feladatok megoldásával egy konkrét példán mutatjuk be a módszer hatékonyságát.

2 A disszertáció eredményei

2.1 Nem-tranzitív preferenciák általános reprezentációja

A Neumann-Morgenstern által kidolgozott hasznosságelméletben az alternatívák rendezése mindig tranzitív. A nem-tranzitív struktúrák megalapozása a klasszikus hasznosságelmélettől eltérő megközelítést igényel. Fishburn volt aki először konstruált nem-tranzitív hasznossá-

gon alapuló eljárást.
Ebben az koncepcióban Fishburn elveti a Neumann-Morgenstern hasznosságelmélet néhány tulajdonságát. Bevezet egy $\phi(x, y)$ ferdén szimmetrikus bilineáris funkcionált(SSB) [15][16][17] ami preferenciaként értelmezhető. Az így kapott eredmények nem feltétlenül tranzitívak. Fishburn feltételezte, hogy a ferdén szimmetrikus bilineáris funkcionál a következő alakú $\phi(x, y) = h(x - y), \ \ x \geq y$, ahol h, egyváltozós szigorúan monoton folytonos függvény, amely mint már említettük preferenciát generál. Fishburn azt mutatta meg, hogy speciális esetben ez az egyváltozós függvény ciklikus preferenciát eredményez. Az azonban nyitott kérdés maradt hogy hogyan jellemezhetők ezek a nem-tranzitív struktúrák a h függvény megválasztásával. A dolgozat első fejezetében megadjuk a ciklikus preferenciák általános jellemzését a Fishburn-féle SSB hasznossági modellben. Azt az esetet vizsgáljuk, amikor a döntési kimenetek X halmaza diszkrét és az X-en megadott valószínűségi mértékek kétértékűek. Definiáljuk a preferencia k-ciklikusságát minden pozitív egész k-ra és megmutatjuk, hogy minden k-ra létezik k-ciklikus preferencia h megfelelő választásaival. Megmutatjuk továbbá hogy a k-ciklikusság előállítható ha h konkáv, konvex, és egy lineáris függvényhez tetszőleges közeli ε távolságra van. Az alábbi eredmények a következő cikkben találhatók [9].

Eredmények

1. Definíció *Legyen* $X = \{j, j + 1, ..., n\}, j, n \in N_+$. *A preferencia* X*-en* $k-$*ciklikus, ha*

$$[m, p(m)] < [m + rk, p(m + rk)]$$

és

$$[m, p(m)] < [m + i, p(m + i)], \qquad ha \qquad i \neq rk$$

teljesül minden $m, r, i \in N$ *egészre, melyre*

$$j \leq m, m + rk, m + i \leq n, k < n - j.$$

A dolgozatban a másodrendű homogén differencia egyenletek elméletének alapján a k-ciklikusságot megvalósító h függvényt az alábbi egyenletből adjuk meg:

$$g(m + 2) - sg(m + 1) + g(m) = 0$$

speciális s paraméterre.
Megadjuk a preferenciát generáló h függvényt az egyenlet megoldásából konkáv és konvex formában. Megmutatjuk, hogy k-ciklikusságot nem generálhat lineáris h függvény, de megadható k-ciklikusságot generáló lineárishoz tetszőlegesen közeli h függvény. Ezért definiáljuk az ε-linearitás fogalmát mint lineáris függvénytől való eltérés mértékét ("kvázi-linearitás"). Egy ilyen függvény nem lesz sem konvex sem konkáv. Végezetül egy konkrét példán k-ciklikus preferenciát generáló h függvényt adunk meg konvex konkáv és 0.4-lineáris formában.
A preferenciát generáló h függvényt $h(m) = v(m)g(m)$ formában adjuk meg, ahol $g(m)$ az alábbi függvény egyenlet rendszer megoldása:

$$F(m + i)g(m) - g(m + i)F(m) = g(i) \qquad (1)$$

ahol
$F(m) = \dfrac{f(m)}{v(m)}$, $1 \leq m, i, m + i \leq n$ és $g(m) > 0$

ha $m \in \{1, 2, ..., n\}$
Az $F(m)$ definíciójában szereplő $v(m)$ függvényre teljesül, hogy $v(m) > 0$ ha $m \in \{1, 2, ..., n\}$ és

$$\begin{array}{lll} v(i) < v(m)v(m+i) & ha & i \neq rk \\ v(i) > v(m)v(m+i & ha & i = rk \end{array} \tag{2}$$

ahol $r \in N$, $r \leq \left[\frac{n}{2k}\right]$, $j \leq m, m+i, rk \leq n$

Az fejezet legfontosabb eredményét az alábbi tétel mondja ki:

1. Tétel: *Minden $k, n \in N$-re amelyre $2k \leq N$ és minden $\varepsilon > 0$-ra létezik $j \in N$ úgy hogy a preferencia k-ciklikus a $[j, n]$ intervallumon és minden k-ra létezik olyan $h(m)$ k-ciklikus preferenciafüggvény amely felírható $h(m) = g(m)v(m)$ alakban, ahol $g(m)$ a (1) olyan megoldása, ahol $g(m)$ pozitív $\{1, 2, ..., n\}$-en és $v(m)$ megoldása (2)-nek és $h(m)$ lehet konkáv, konvex és ε-lineáris.*

A tétel következménye, hogy k-ciklikus preferencia minden pozitív egész k-ra létezik. egy rögzített k-ra preferenciát generáló h függvényre különböző függvényosztályokból adható megoldás.

2.2 Lexikografikus döntési függvény analitikus konstrukciója

A dolgozat második fejezetében a preferencia alapú döntésekkel foglalkozunk (azaz az úgynevezett outranking modellekkel). Megmutatjuk, hogy ha az alternatívák fölötti rendezés lexikografikus [13][35] akkor ez a lexikografikus döntés megkapható az outranking eljárásokra kidolgozott általános modell segítségével [4][5]. Azaz a lexikografikus döntés is outranking eljárás. Ismeretes, hogy az minden outranking módszer egy rögzített preferenciafüggvény megfelelő módosításával állítható elő. Megmutatjuk, hogy a lexikografikus döntés megkapható egy speciális módosító függvény és a így kapott preferenciák megfelelő súlyozásával. Dolgozatunkban megmutatjuk, a súlyok megválasztásának eljárását. Az alábbi eredmények a következő cikkekben találhatók [10][11].

Eredmények

Legyen $a = (x_1, x_2, ..., x_n)$ és $b = (y_1, y_2, ..., y_n)$ két alternatíva, az x_k, y_k hasznossági értékekkel megadva, és $w_1, w_2, ..., w_n$ súlyok. A preferencia a következő módon számítható:

$$p(a, b) = \sum_{i=1}^{n} w_i \tau_i(p_i(x_i, y_i))$$

ahol a preferenciafüggvény

$$p(x, y) = \tfrac{y-x+1}{2}$$

és legyen $\tau_i : [0, 1] \to [0, 1]$ egyváltozós monoton függvény az alábbi:

$$\tau(x) = \begin{cases} 0 & ha \quad 0 \le x < \frac{1}{2} - \delta \\ \frac{1}{2} & ha \quad \frac{1}{2} - \delta \le x \le \frac{1}{2} + \delta \\ 1 & ha \quad \frac{1}{2} + \delta < x \le 1 \end{cases}$$

A fejezet legfontosabb eredményét mondja ki az alábbi tétel:

2. Tétel *Legyen* $A = \{a_1, a_2, \cdots, a_m\}$ *az alternatívák halmaza,* $C = \{c_1, c_2, \cdots, c_n\}$ *a kritériumok halmaza, a döntéshozó által megadott fontossági sorrend szerint. Jelölje* x_{ij} *az* a_i *alternatíva* c_j *kritérium*

szerinti kiértékelését (hasznosságát), és feltesszük, hogy a kiértékelés értékei normáltak, azaz $0 \leq x_{ij} \leq 1$.
Azaz a döntési mátrix:

	c_1	c_2	$\ldots$	c_n
a_1	x_{11}	x_{12}	$\ldots$	x_{1n}
a_2	x_{21}	x_{22}	$\ldots$	x_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_m	x_{m1}	x_{m2}	$\ldots$	x_{mn}

A $p(x, y)$ *preferenciafüggvény és* $\tau(x)$ *módosító vagy küszöbérték függvény legyen olyan, ahogyan azt fentebb megadtuk.*
Ekkor léteznek olyan w_k *(*$k = 1, 2, ...n$*) súlyok hogy az* m *elemű*

$$l_i = \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (\tau(\sum_{k=1}^{n} w_k \tau(p(x_{ik}, x_{jk}))))$$

pozitív valós számhalmazra teljesül:

$$l_i > l_j \text{ pontosan akkor } a_i <^L a_j.$$

A lexikografikus döntési függvény konstrukciójához egy súlyozás megvalósításával jutunk el, mely megtartja a a nem-kompenzatórikus tulajdonságot. Ezt a súlyozást a következőként konstruáljuk meg: Legyen a c_i kritérium fontossági súlya:

$$w_i = \frac{1}{2^i} + \frac{1}{n2^n} \; .$$

Továbbá igaz az alábbi:

$$\sum_{k=1}^{n} w_k = 1$$

feltétel. A lexikografikus döntési függvényt ekkor a következő függvénykompozíció valósítja meg:

$$\tau(\sum_{k=1}^{n} w_k \tau(p(x_{ik}, x_{jk}))) = \begin{cases} 0 & ha \quad a_i >^L a_j \\ 1 & ha \quad a_i <^L a_j \end{cases}$$

Ez a konstrukció leírja a PROMEETHE és ELECTRE [33][36] módszereket. A döntési függvényt normalizálva kapjuk a valós l_i értékeket a [0,1] intervallumon:

$$l_i = \frac{1}{m} \sum_{j=1}^{m} \tau(\sum_{k=1}^{n} w_k \tau(p(x_{ik}, x_{jk}))), \; i = 1, 2, ..., m$$

melyekre:

$l_i < l_j$ akkor és csak akkor ha $a_i >^L a_j$.

Ezzel a konstrukcióval beilleszthető a lexikografikus döntési függvény az outranking módszerekre megkonstruált általános döntési keretrendszerbe.

2.3 A lexikografikus döntési eljárás tanulása

A dolgozat harmadik fejezetében a lexikografikus döntési eljárás taníthatósága volt a célunk. Tanulás szempontjából ez a döntési kritériumok rendezési sorrendjének - másképpen fogalmazva fontossági sorrendjének - megtanulását jelenti. A kritériumok fontossági sorrendjét döntési minták alapján kaphatjuk meg. A konstrukció során feltesszük, hogy a minták "excehange value (EV)" vektorok által adottak, amelyek egy "excehange value evaluation (EVE)" függvény által kiértékeltek. Az EV vektor n-dimenziós és a vektor elemei +1, 0, -1 értéket vehetnek fel. A +1 az i-edik koordinátában azt jelenti hogy a tanulás során az i-edik kritérium szerint javítjuk a döntési alternatívát, az i-edik kritériumot tekintve, -1 azt jelenti, rontjuk a döntési alternatívát, az i-edik kritériumot tekintve, 0 azt jelenti nem változtatjuk az alternatívát. Az EVE függvény egy EV vektoron +1 kiértékelési értéket ad ha az EV vektor általi eredmény alternatíva megelőzi az eredeti alternatívát a lexikografikus döntési sorrendben, ellenkező esetben -1-et. Az optimális eset vizsgálatánál két modellt adunk meg és a megfelelő EV vektorokat generáljuk. A dolgozatban megadunk egy algoritmust amely legfeljebb $n \lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1$ EV vektort használ. Legrosszabb eset elemzésekor foglalkoztunk az "ellenfél alapú" modellekkel. Ezeken a modellekben az EV vektorok listája "ellenfél" által generáltak. Az Ellenfélnek az a célja hogy olyan a lehető leghosszabb listát generálja. Különböző tipusú "ellenfeleket" vizsgálhatunk a használt távolságfogalmak alapján (a távolság az EV vektorok halmazán értelmezett). Két EV vektor távolsága például lehet a különböző komponensek száma. Továbbá definiálunk egy megszorítást, a szigorú 4-távolságot, amely arra kényszeríti az "ellenfelet", hogy legfeljebb $O(n^2)$ hosszú EV vektor listát használjon. Az alábbi eredmények a következő cikkben találhatók [6].

Eredmények

A dolgozatnak ebben a részében egy minta kiértékelő (Sample Evaluation) algoritmust mutatunk be, amely megadja az EV vektorok sorozatának kiértékelését az EVE függvénnyel és eredményként kapjuk a kritériumok fontossági sorrendjének meghatározását ha lehetséges, vagy annak eldöntét, hogy a minta elégtelen ehhez (azaz más kritériumsorrend által is generálható) vagy inkonzisztens (azaz nem kapható meg lexikografikus sorrend eredményeként). Az algoritmusnak n fázisa van. Az i-edik fázisban a fontossági sorrendben i-edik kritérium lesz meghatározva. Ha az i-edik lépésben nincs meghatározva az i-edik kritérium akkor adódik a következtetés, hogy a minta elégtelen vagy ellentmondásos. Az algorimust a következőkben írjuk le:

SamEv Algorithm

Initialization

$S_1 := \{1 \ldots, n\}$

Iteration part

for $i = 1$ to n do

$S := S_i$,

for every $p_k \in L$ do

if $i > 1$ and $p_k(l_{i-1}) \neq 0$ **then**

delete p_k from L,

else if $EVE(p_k) = 1$ **then**

delete each j with $p_k(j) = -1$ from S,

else if $EVE(p_k) = -1$ **then**

delete each j with $p_k(j) = 1$ from S,

endfor

if $|S| = 0$ **then**

stop, the sample is inconsistent,

if $|S| \geq 2$ then

note that the sample is not sufficient,

delete arbitrary $|S| - 1$ elements from S.

Let $S_{i+1} := S_i \setminus S$,

let l_i be the index which is contained in S,

endfor

Output: Egy fontossági sorrend $l_1, l_2, ..., l_n$ és ha valamely fázisnál megjegyezzük, hogy ha a minta elégtelen, akkor más fontossági sorrendek is konzisztensek lehetnek a mintával.

3.Tétel *Ha a SamEv algoritmus nem áll meg inkonzisztens mintával, akkor a mintával konzisztens fontossági sorrendet ad eredményül. Továbbá jól határozza meg azt, hogy a minta inkonzisztens, illetve azt hogy a minta elégtelen.*

A következőkben azt vizsgáljuk, hogy a legjobb esetben milyen hosszú EV vektor lista kell a kritériumok fontossági sorrendjének meghatározásához. Ebben a modellben (Oracle modell) feltesszük, hogy az EVE függvényt úgy tudjuk használni mint egy jóslót, azaz meg tudjuk kérdezni egy általunk generált EV vektor kiértékelését. Meg szeretnénk találni EV vektorok legrövidebb sorozatát ami meghatározza a kritériumok fontossági sorrendjét. Megadunk az EV vektorok sorozatának generálására egy algoritmust. Az i-edik lépésben az ii-edik legfontosabb kritériumot határozzuk meg. Az algoritmus leírásánál a következő jelölést használjuk. A kritériumok tetszőleges C_1, C_2 halmazainál az $EV(C_1, C_2)$ jelöli azt az EV vektort, mely $+1$ a C_1-beli koordinátákban, -1 a C_2-beli koordinátákban és 0 bármely más koordinátában.

EV sequence generating algorithm

for $i := 1$ **to** n
  $C := S$
  while $|C| > 1$ do
    let C_1 be the set of first $\lfloor |C|/2 \rfloor$ elements of C
    let C_2 be the set of the other elements of C

let $V = EV(C_1, C_2)$
if $EVE(V) = +1$ **then** $C := C_1$
else $C := C_2$
endwhile
C contains the i-th criterion in the importance order
delete the element of C from S
endfor

Az algoritmus műveletigényéről azt mondhatjuk, hogy legfeljebb: $\lceil \log_2 n \rceil + \lceil \log_2 (n-1) \rceil + ... + \lceil \log_2 2 \rceil = n \lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1.$ elemű EV vektor listát használ.
A legrosszabb esetet is megvizsgáltuk. Vizsgálatunkban feltesszük, hogy az EV vektorok listája "ellenfél" által generált, akinek célja, hogy olyan hosszú listát generáljon, amilyet csak lehet. Az "ellenfél" által generált EV vektor listára megszorításokat alkalmazunk. Első lépésben a redundáns vektorok generálásának lehetőségét tiltjuk meg. Megtiltjuk egy már generált EV vektor ellentettjének generálását, egy már generált EV vektorhoz képest minden komponensben legalább olyan jó vektor generálását. Mivel minden nem olyan EV vektor amely nem a zéró vektor és nem tartalmaz -1 komponenst, +1 kiértékelést kap, azt a megszorítást tesszük, hogy -1-nek lennie kell a generált vektorban. Megmutatjuk, hogy ezen megszorítások mellett az "ellenfél" tud exponenciális hosszú listát generálni. A továbbiakban egy távolságfogalmat definiálunk és annak segítségével alkalmazunk megszorításokat az "ellenfélre". A Hamming távolság általánosítását tekintjük távolságfogalomnak, azaz két EV vektor távolsága a különböző komponenseik száma. Gyenge 1-távolság megszorított sorozatnak nevezzük az olyan sorozatot, mely esetén az "ellenfél" által generált vektor az azt megelőzőtől 1 távolságra van. Megmutatható, hogy

ezen megszorítás mellett az "ellenfél" tud exponenciális hosszú listát megadni. Bevezetjük a szigorú k-távolság megszorított sorozat fogalmát, ami azt jelenti nincs a sorozatban két egymástól k-nál nagyobb távolságra lévő vektor. Ekkor igazak a következők:

4.Tétel *Ha $n > 3$ nem létezik szigorú 1-távolság megszorított sorozat mely elegendő információt tartalmazna a fontossági sorrend meghatározásához. Ha $n \geq 6$ nem létezik olyan szigorú 2-távolság megszorított sorozat mely elegendő információt tartalmazna a fontossági sorrend meghatározásához. Ha $n \geq 8$ nem létezik olyan szigorú 3-távolság megszorított sorozat mely elegendő információt tartalmazna a fontossági sorrend meghatározásához. Tetszőleges n-re létezik olyan szigorú 4-távolság megszorított sorozat mely elegendő információt tartalmaz a fontossági sorrend meghatározásához.*

5.Tétel *Egy szigorú 4-távolság megszorított vektorsorozat elemszáma legfeljebb $O(n^2)$.*

Megmutattuk, hogy létezik az "ellenfél" olyan megszorítása, a szigorú 4-távolság megszorítás, amely mellett legfeljebb $O(n^2)$ hosszú sorozatot generálhat, és ez elegendő a kritériumok fontossági sorrendjének meghatározásához.

2.4 Alternatívákkal bővített CPM problémák megoldása folyamathálózatokban

Folyamathálózatok [1][12][21][29][30][31][32][37][38][40] döntési problémáival foglalkozunk a továbbiakban. A dolgozat negyedik fejezetében a kritikus út módszere (CPM) a projektmenedzsmentben, tevékenységek egy halmazának ütemezésére bevezetett eljárás. A

CPM módszertan a tevékenységek elvégzéséhez szükséges idő mellett azok logikai összefüggéseit is felhasználja és ezek alapján építi fel a tevékenységek ütemterv hálóját [2]. A dolgozat negyedik fejezete egy új módszert mutat be mely a CPM problémák kiterjesztését teszi lehetővé. Megadjuk első lépésben a CPM probléma folyamathálózatokra történő transzformációját. Megmutatjuk, hogyan generálható a folyamathálózatban a CPM problémát megoldó optimális struktúra. Négy különböző modellt vizsgálunk, **időoptimális**, **költségoptimális**, **időkorlátos költségoptimális**, **költségkorlátos időoptimális**. Ha egy adott problémát egynél több tevékenységgel vagy egynél több tevékenységsorozattal lehet megoldani, ezek a tevékenységek vagy tevékenységsorozatok mint egymás alternatívái jelennek meg. Modellünkben a megadott folyamathálózat további alternatívákkal bővíthető. A folyamathálózat reprezentációban az alternatívák problémája új élek hozzáadásával jelenik meg. Ebben a fejezetben példán mutatjuk meg, hogy az alternatívákkal bővített CPM probléma miként jelenik meg a folyamathálózatokban. A példa jól szemlélteti a kidolgozott módszer hatékonyságát. Az alábbi eredmények a következő cikkben találhatók [39].

Eredmények

Megadjuk a CPM probléma folyamathálózat reprezentációjának matematikai modelljét. Legyenek A, E, D véges halmazok, ahol A jelölje a tevékenységek halmazát, E jelölje az események halmazát és D jelölje az élek halmazát.
A $G(A, E, D)$ folyamathálózat gráfra teljesülnek az alábbi tulajdonságok:

$A \cap E = \emptyset, D \subseteq (A \times E) \cup (E \times A).$

$A = \{i \in N\}$ tevékenységek
$E = \{j \in N\}$ események
Jelölje x_i az i-edik tevékenységet a folyamathálózatban.

$$x_i = \begin{cases} 1 & \text{ha az i-edik tevékenység végrehajtódik} \\ 0 & \text{ha az i-edik tevékenység nem hajtódik végre} \end{cases}$$

t_i=a folyamat kezdő időpontjától az ii-edik esemény bekövetkezéséig eltelt idő

T_i=az i-edik tevékenység végrehajtási ideje

T=felső korlát a folyamat(projekt) teljes idejére

C_i=az i-edik tevékenység költsége

C=felső korlát a folyamat(projekt) teljes költségére

Mivel a projekt befejező eseménypontját két különböző tevékenység előzi meg, a projekt folyamathálózat modellezésében egy technikai operációs egységet vezetünk be, melyet $Close$ jelöl.

Az alternatívákkal bővített CPM probléma időoptimális projekttervének matematikai programozási modelljét adjuk meg az alábbiakban:

$$x_{Close} = 1 \qquad Close \in A \tag{3}$$

$$\sum_{\{i:i \in A\ (i,j) \in D\}} x_i = 1 \qquad \forall j \in E \quad and \quad j \neq Start \tag{4}$$

$$t_{Start} = 0 \qquad Start \in E \tag{5}$$

$$t_k + \sum_{\{i:i \in A\ (i,j) \in D\}} x_i T_i \leq t_j \quad \forall k, j \in E \setminus Start$$

$$where \quad \exists i:(k,i) \quad and \quad (i,j)\in D \tag{6}$$

$$t_{End} \longrightarrow min \tag{7}$$

A matematikai programozási modell célja meghatározni a folyamathálózat struktúrából az **időoptimális** megoldást. Emellett megadunk még **költségoptimális**, **költségkorlátos időoptimális** és **időkorlátos költségoptimális** megoldást.

2.5 Folyamathálózatok optimalizálása új fuzzy lineáris programozási eljárással

A folyamathálózatokban [18][19] sok esetben nem tudjuk meghatározni az idő vagy költség paraméterek értékét. A fuzzy elmélet [20] egyik legfontosabb fogalma a halmazhoz tartozási függvény, amivel a fent megfogalmazott bizonytalanságot tudjuk modellezni. A halmazhoz tartozási függvény a klasszikus "karakterisztikus függvény" folytonos függvénnyel való helyettesítése. A bizonytalanság pedig a halmazhoz tartozási függvény és a karakterisztikus függvény "távolsága" ami a fuzziság mértéke [3]. A fuzzy elmélet jól alkalmazható amikor paraméterek bizonytalanságát modellezzük, ami egyben lehetőséget teremt arra, hogy a folyamathálózatok paramétereinek bizonytalanságát matematikailag egzakt módon kezeljük a folyamatok ütemezésének és optimalizálási feladatainak megoldásaiban. A dolgozat ötödik fejezetében ezért új fuzzy lineáris programozási eljárást vezetünk be a folyamathálózatok optimalizálási feladataihoz, mellyel fuzzy időket és költségeket tudunk számítani. A bizonytalanság csökkentésének eléréséhez a "legélesebb" (a karakterisztikus függvényhez legközelebbi) fuzzy megoldást keressük, azaz a legközelebb szeretnénk kerülni a klasszikus éles megoldáshoz. A dolgozatban kifejtett eljárásban a tagsági függvények élessége megjelenik a célfüggvényben. Modellünkben az egyszerűség kedvéért trapezoid fuzzy tagsági függvényt használunk, melyet bal és jobb oldali támasztó egyenesekkel írunk le. A trapezoid fuzzy tagsági függvények speciális reprezentációját használjuk [7] normalizált vágófüggvénnnyel(amit a szögletes zárójellel reprezentálunk).

$$L(x) = \left[m_l(x - a_l) + \frac{1}{2}\right] \quad and \quad R(x) = \left[m_r(x - a_r) + \frac{1}{2}\right]$$

A fuzzy tagsági függvények ezen reprezentációjában az a_l és az a_r paramétereket bal és jobb oldali közép paramétereknek nevezzük, az m_l és az m_r paramétereket bal és jobb oldali tangens paramétereknek nevezzük. A fuzzyság mértékét az m_l és az d m_r paraméterek reprezentálják. Az alábbiakban az új fuzzy optimalizációs módszer logikai koncepcióját foglaljuk össze. A három új koncepcióban különböző fuzzy optimalizálási célfüggvényt használunk mind a négy optimalizálási modellre, **időoptimalizálás**, **költségoptimalizálás**, **időoptimalizálás költségkorláttal** és **költségoptimalizálás időkorláttal**. Elsőként megadunk egy új két lépéses fuzzy lineáris optimalizálási koncepciót. Első lépésben a fuzzy eseményidők baloldali és a jobboldali közép paramétereinek optimalizálását végzi az algoritmus. A második lépésben a fuzziságot minimalizáljuk azon optimális struktúrára melyet az első lépéssel generáltunk. Második koncepciónkban megadunk egy egylépéses fuzzy lineáris optimalizálási módszert, melyben egy lépésben definiáljuk a közép és fuzziság paraméterekre a feladatot és a célfüggvényben a közép paraméterek átlagát mimimalizáljuk. Harmadik koncepcióként megadunk egy kétlépéses fuzzy lineáris optimalizálási módszert új célfüggvénnyel. Ebben a koncepcióban a projektmenedzsment ütemezési problémájában alkalmazott standard optimalizálási logikát követtük, annyiban, hogy az utolsó esemény fuzzy idejére optimalizálunk, az új két lépéses módszer fuzzy optimalizálási logikáját követve. Végezetül példán illusztráljuk a módszerek hatékonyságát és megmutatjuk, hogy az új kétlépéses módszer adja a "legélesebb" megoldást. A kapott eredmények bizonytalansági paraméterei mutatják, hogy az új kétlépéses fuzzy optimalizálási koncepcióban megfogalmazott feltételek szigorúsága szükséges a "legélesebb" megoldás megtalálásához. Az alábbi eredmények a következő cikkben találhatók [8].

Eredmények

Ahhoz hogy leírjuk a folyamatot az alábbi jelöléseket használjuk.
A T_i fuzzy akitivitás idő bal (jelölésben l) és jobb (jelölésben:r) közép és tangens paramétereit jelölje:

$$a_{A,i,l}, \quad m_{A,i,l}, \quad a_{A,i,r}, \quad m_{A,i,r}$$

A t_i, fuzzy eseményidő bal (jelölésben l) és jobb (jelölésben:r) közép és tangens paramétereit jelölje:

$$a_{E,i,l}, \quad m_{E,i,l}, \quad a_{E,i,r}, \quad m_{E,i,r}$$

Az ii-edik tevékenység fuzzy költségének bal (jelölésben l) és jobb (jelölésben:r) közép és tangens paramétereit jelölje:

$$a_{C,i,l}, \quad m_{C,i,l}, \quad a_{C,i,r}, \quad m_{C,i,r}$$

A matematikai modell felírásához vezessük be a következő jelölést:

$$M_{A,i,r} = \frac{1}{m_{A,i,r}} \quad M_{E,i,r} = \frac{1}{m_{E,i,r}}$$

Step 1.

$$x_{Close} = 1 \qquad Close \in A \tag{8}$$

$$\sum_{\{i:i\in A\ (i,j)\in D\}} x_i = 1 \qquad \forall j \in E \qquad j \neq Start \tag{9}$$

$$a_{E,1,l} = a_{E,1,r} = 0 \tag{10}$$

$$a_{E,k,l} + \sum_{\{i:i\in A\ (i,j)\in D\}} x_i a_{A,i,l} \leq a_{E,j,l} \quad \forall k,j \in E \quad j \neq Start$$

$$where \quad \exists i : (k,i) \in D \quad and \quad (i,j) \in D \tag{11}$$

$$a_{E,k,r} + \sum_{\{i:i\in A\ (i,j)\in D\}} x_i a_{A,i,r} \leq a_{E,j,r} \quad \forall k,j \in E \quad j \neq Start$$

$$where \quad \exists i : (k,i) \in D \quad and \quad (i,j) \in D \tag{12}$$

$$\sum_{\forall s\in E} \frac{a_{E,s,l} + a_{E,s,r}}{2} \longrightarrow min \tag{13}$$

Az első lépésben a matematikai modell kiválasztja az optimális struktúra lehetséges alternatíváit. A_T jelölje az output tevékenység halmazt, D_T az optimális struktúra élhalmazát. Az E eseményhalmaz nem változott az első lépésben.

Step 2.

$$x_{Close} = 1 \qquad Close \in A_T \tag{14}$$

$$\sum_{\{i: i \in A_T \ (i,j) \in D_T\}} x_i = 1 \qquad \forall j \in E \qquad j \neq Start \tag{15}$$

$$a_{E,1,l} = a_{E,1,r} = 0 \tag{16}$$

$$M_{E,k,l} + \sum_{\{i: i \in A_T \ (i,j) \in D_T\}} x_i M_{A_T,i,l} \leq M_{E,j,l}$$
$$\forall k, j \in E \quad j \neq Start$$

$$where \quad \exists i : (k,i) \in D_T \quad and \quad (i,j) \in D_T \tag{17}$$

$$|M_{E,k,r}| + \sum_{\{i: i \in A_T \ (i,j) \in D_T\}} x_i |M_{A_T,i,r}| \leq |M_{E,j,r}|$$
$$\forall k, j \in E \quad j \neq Start$$

$$where \quad \exists i : (k,i) \in D_T \quad and \quad (i,j) \in D_T \tag{18}$$

$$\sum_{\forall s \in E} \frac{M_{E,s,l} + |M_{E,s,r}|}{2} \longrightarrow min \tag{19}$$

Három koncepciót adunk meg és mindhárom koncepción belül vizsgáljuk az időoptimalizálás, költségoptimalizálás, időoptimalizálás költségkorláttal és költségoptimalizálás időkorláttal modelleket. A "legélesebb" megoldást keressük, és a modellekből kapott eredmények bizonytalansági paraméterei mutatják, hogy az új kétlépéses fuzzy optimalizálási módszerben megfogalmazott feltételek szigorúsága szükséges a "legélesebb" megoldás megtalálásához.

References

[1] M. Barany, B. Bertok, Z. Kovacs, F. Friedler, and L. T. Fan. Optimization software for solving vehicle assignment problems to minimize cost and environmental impact of transportation. Chemical Engeneering, 21, 2010.

[2] S. Chanas and P. Zielinski. Critical path analysis in the network with fuzzy activity times. Fuzzy sets and systems, 122(2):195-204, 2001.

[3] J. Dombi. Membership function as an evaluation. Fuzzy sets and systems, 35(1):1-21, 1990.

[4] J. Dombi. Fuzzy Logic and Soft Computing Advances in Fuzzy Systems - Application and Theory, chapter A general framework for the utility/based and outranking methods, pages 202-208. World Scientific Publishing, 1995.

[5] J. Dombi. Principles of Fuzzy Preference Modelling and Decision Making, chapter A Common Preference Model for Various Decision Models. Academia Press, Gent, Belgium, 2003.

[6] J. Dombi, Cs. Imreh, and N. Vincze. Learning lexicographic orders. European Journal of Operational Research, 183(2):748-756, 2007.

[7] J. Dombi and T. Szépe. Arithmetic-based fuzzy control. Iranian Journal of Fuzzy Systems,14(4):51-66, 2017.

[8] J. Dombi and N. Vincze. Optimization the process network using new concept of fuzzy linear programming. under review.

[9] J. Dombi and N. Vincze. Universal characterization of non-transitive preferences. Mathematical Social Sciences, 27(1):91-104, 1994.

[10] J. Dombi and N. Vincze. The lexicographic decision function. Acta Cybernetica, 17(1):95-106,2005.

[11] J. Dombi and N. Vincze. Lexikografikus döntések egy általános döntési modellben. Szigma, XXXVI.(3-4.):149-161, 2005.

[12] L. T. Fan, Y. Ch. Lin, S. Shafie, B. Bertok, and F. Friedler. Exhaustive identification of feasible pathways of the reaction catalyzed by a catalyst with multiactive sites via a highly effective graph-theoretic algorithm: application to ethylene hydrogenation. Industrial & Engineering Chemistry Research, 51(6):2548-2552, 2012.

[13] P. C Fishburn. Lexicographic orders, utilities and decision rules. Management science, 20(11):1442-1471, 1974.

[14] P. C. Fishburn. Nontransitive measurable utility. Journal of Mathematical Psychology, 26(1):31-67, 1982.

[15] P. C. Fishburn. SSB utility theory: An economic perspective. Mathematical Social Sciences, 8(1):63-94, 1984.

[16] P. C. Fishburn. SSB utility theory and decision-making under uncertainty. Mathematical social sciences, 8(3):253-285, 1984.

[17] P. C. Fishburn. Non-linear preference and utility theory. Wheatsheaf Books Ltd., Brighton,1988.

[18] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan. Combinatorial algorithms for process synthesis. Computers & chemical engineering, 16:S313-S320, 1992.

[19] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan. Graph-theoretic approach to process synthesis: axioms and theorems. Chemical Engineering Science, 47(8):1973-1988, 1992.

[20] R. Fullér and R. Mesiar. Special issue on fuzzy arithmetic. Fuzzy Sets and System, 91(2), 1997.

[21] J. C. Garcia-Ojeda, B. Bertok, and F. Friedler. Planning evacuation routes with the p-graph framework. Chemical Engineering, 29, 2012.

[22] E. Jacquet-Lagreze. Binary preference indices: A new look on multicriteria aggregation procedures. European Journal of Operational Research, 10(1):26-32, 1982.

[23] E. Jacquet-Lagreze. Interactive assessment of preferences using holistic judgments the PREFCALC system. In Readings in multiple criteria decision aid, pages 335-350. Springer, 1990.

[24] E. Jacquet-Lagreze. An application of the uta discriminant model for the evaluation of R & D projects. In Advances in multicriteria analysis, pages 203-211. Springer, 1995.

[25] E. Jacquet-Lagreze, R. Meziani, and R. Slowinski. MOLP with an interactive assessment of a piecewise linear utility function. European Journal of Operational Research, 31(3):350-357,1987.

[26] E. Jacquet-Lagreze and M. F. Shakun. Decision support systems for semi-structured buying decisions. European Journal of Operational Research, 16(1):48-58, 1984.

[27] E. Jacquet-Lagreze and J. Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the UTA method.

European journal of operational research, 10(2):151-164,1982.

[28] E. Jacquet-Lagreze and Y. Siskos. Preference disaggregation: 20 years of MCDA experience. European Journal of Operational Research, 130(2):233-245, 2001.

[29] K. Kalauz, Z. Süle, B. Bertok, F. Friedler, and L. T. Fan. Extending process-network synthesis algorithms with time bounds for supply network design. Chemical Engineering, 29, 2012.

[30] J. Klemess, F. Friedler, I. Bulatov, and P. Varbanov. Sustainability in the process industry:integration and optimization (Green Manufacturing & Systems Engineering). McGraw-Hill Professional, 2010.

[31] Z. Kovács, Zs. Ercsey, F. Friedler, and L. T. Fan. Exact superstructure for the synthesis of separation-networks with multiple feedstreams and sharp separators. Computers & Chemical Engineering, 23:1007-1010, 1999.

[32] Z. Kovács, Zs. Ercsey, F. Friedler, and L. T. Fan. Separation-network synthesis: global optimum through rigorous super-structure. Computers & Chemical Engineering, 24(8):1881-1900, 2000.

[33] D. L. Olson. Decision aids for selection problems. Springer Science & Business Media, 1996.

[34] T. Rapcsák. Többszempontú döntési problémák, AHP módszertan. MTA SZTAKI, 2003.

[35] T. Solymosi. Adat, modell elemzés, chapter Ordinális mérési skálák lexikografikus aggregációja, pages 119-126. Aula, Budapest, 2001.

[36] J. Temesi. A döntéselmélet alapjai. Aula, Budapest, 2002.

[37] J. Tick. P-graph-based workflow modelling. Acta Polytechnica Hungarica, 4(1):75-88, 2007.

[38] J. Tick, Cs. Imreh, and Z. Kovács. Business process modeling and the robust PNS problem. Acta Polytechnica Hungarica, 10(6):193-204, 2013.

[39] N. Vincze, Zs. Ercsey, T. Kovács, J. Tick, and Z. Kovács. Process network solution of extended CPM problems with alternatives. Acta Polytechnica Hungarica, 13(3),101-117, 2016.

[40] Ch. Yun, T. Y. Kim, T. Zhang, Y. Kim, S. Y. Lee, S. Park, F. Friedler, and B. Bertok. Determination of the thermodynamically dominant metabolic pathways. Industrial & Engineering Chemistry Research, 52(1):222-229, 2012.