

## A RANG-64 MŰKÖDÉSÉNEK ALAPJAI\*

Dobó Andor – Szajcz Sándor

Prodinform Műszaki Tanácsadó Vállalat

A minősítő, rangsoroló programrendszernek ez az áttekintése elsősorban azok számára készült, akik már használják a RANG-64-et, a modellrendszer felépítésének és működésének elvét, alapjait azonban közelebbről is szeretnék megismerni. A leírás kerüli annak a bonyolult és összetett matematikai eszköztárnak a bemutatását, melynek algoritmusai a rendszert működtetik. Célja csupán nagyvonalú összefüggésben láttatni, érzékeltetni azokat a lényegesebb momentumokat, melyek a széles körű alkalmazhatóság sikerét elősegítik, szolgálják.

Adva van az  $O_1, O_2, \dots, O_m$  objektum  $T_1, T_2, \dots, T_n$  tulajdonságát jellemző számszerű adatok halmaza. Jelölje  $x_{ik}$  a  $k$ -adik objektum  $i$ -edik tulajdonságához rendelt számértéket, mutatót.

Legyen az  $i$ -edik tulajdonságra,

$$\delta_i = \begin{cases} 0, \text{ vagy } m, & \text{ha az a jó, ha } x_{ik} \text{ minél kisebb} \\ & \text{(minimális)} \\ 1/2, \text{ vagy } \bar{x}_i, & \text{ha az a jó, ha } x_{ik} \text{ az } \bar{x}_i \text{ értéket} \\ & \text{veszi fel (optimális)} \\ 1, \text{ vagy } M, & \text{ha az a jó, ha } x_{ik} \text{ minél nagyobb} \\ & \text{(maximális)} \end{cases}$$

(Ha az  $i$ -edik tulajdonság megadott értéke  $\bar{x}_i$ , akkor, ha  $\bar{x}_i < x_{ik}$ ,  $\delta_i = 0$ ; ha pedig  $x_{ik} < \bar{x}_i$ ,  $\delta_i = 1$ .)

A  $\delta_i$ -t az értékelésben játszott szerepe miatt *értékelési vezérlő* vagy röviden *vezérlő vektornak* nevezjük. A  $\delta_i$ -nek alapvető szerepe van a *bázishasonlóság* kifejező "transzformációs formula" megválasztásában. A felhasználóra mindebből csupán annyi tartozik, hogy a mutatókat

- ◆ minél kisebb (minimális =  $m$ ),
- ◆ minél nagyobb (maximális =  $M$ ),
- ◆ adott érték körül mozgó (optimális =  $\bar{x}_i$ )

kategóriába kell sorolni. (Az utóbbi esetben az optimális értéket is meg kell adni!)

\* A *Minőség és Megbízhatóság* c. folyóiratban ugyanezen a címen megjelent cikk alapján (1986. 6. sz. p. 459–461.)

Az eddig közöltek az 1. ábra szerint kiinduló mátrixban foglalhatjuk össze.

Az  $X$  alapadatmátrixban a mutatók már közös mértékegységben vannak megadva, ami azt jelenti, hogy *a dimenzió minden sorban azonos*.

A rendszer elvi működését az alábbi lépések jellemzik. (Ezek részletes leírása az irodalomban megtalálható [1], [2], [3], [4].)

1. Az  $x_{ik}$  alapadatok bevitelle, a mutatók kategóriába sorolása a vezérlő vektor ismeretében, illetve megadása mellett.

2. Ha az  $x_{ik}$  értékek között negatív érték is van, akkor abban a sorban a program egy *lineáris transzformációt* hajt végre. Ezzel az  $X$  mátrix az  $Y = \{y_{ik}\}$  mátrixba megy át. *Jele:  $X \rightarrow Y$* . Hogy a transzformáció *megengedett*, az [1]-ben van bizonyítva. (A bizonyítás 3 lemma és 5 tételre keresztül történik, leírása 40 oldal terjedelmű.)

$$X = \{x_{ik}\}$$

Tulajdonságok (mutatók)	Objektumok				$\delta$
	$O_1$	$O_2$	$O_k$	$O_m$	
$T_1$	$x_{11}$	$x_{12}$	$x_{1k}$	$x_{1m}$	$\delta_1$
$T_2$	$x_{21}$	$x_{22}$	$x_{2k}$	$x_{2m}$	$\delta_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$T_i$	$x_{i1}$	$x_{i2}$	$\rightarrow  x_{ik}  \leftarrow$	$x_{im}$	$\delta_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$T_n$	$x_{n1}$	$x_{n2}$	$x_{nk}$	$x_{nm}$	$\delta_n$

1. ábra Az alapadatmátrix

3. Ha az  $Y$  mátrix elemei között hiányzóknak vannak, akkor a program azokat pótolja, s így  $Y \rightarrow Y^*$ -ba. Itt feltételezzük, hogy nincs olyan objektum, melynek minden mutatója hiányzik, továbbá, hogy létezik legalább egy olyan objektum, melynek

minden mutatója – komponensként – adott. A program az összes rendelkezésre álló alapadatot felhasználja becslésre, a következő három módszer valamelyikét választva:

1. módszer: lényegében egy iteráció első lépése.
2. módszer: iteratív eljárás, melynek segítségével ha már egy adatot pótolunk, akkor a továbbiakban az eljárás azt ismertnek (adottnak) tekinti. (Kezdetben az 1. módszer első lépését jelenti!) Ezt az eljárást használja a RANG-64.
3. módszer: a hasonlósági transzformáció alkalmazásán, felhasználásán alapszik.

A módszerek értékelése, összehasonlítása az irodalomban megtalálható [4].

4. A szükségesnek bizonyuló lineáris transzformáció elvégzése után még egy újabb, ún. *hasonlósági transzformációt* alkalmazunk többek között azért is, hogy a mutatók dimenzióit "eltüntessük". Erre két módszer is kínálkozik.

*Nullmódszer:* alkalmazása előnyös akkor, amikor a mutatónak csupán azt kell kifejezni, hogy egy tulajdonság létezik-e, vagy sem. (Például a termeléshez kell-e import, vagy sem.) Hátránya, hogy nem eléggé érzékeny és árnyalt.

*Pozitív módszer:* érzékenyebb bizonyos tulajdonságok számszerű alakulására. Az olyan mutató esetén, amikor az a jó, ha annak értéke minél kisebb – és ezen értékek minimuma "előfordultan" nulla – ez az eljárás figyelembe veszi azt is, hogy a mutatósor pozitív értékei közül melyik nagyobb, mennyivel.

A RANG-64 ezt a változatot használja. Ezáltal  $X \rightarrow Y \rightarrow Y^* \rightarrow Q$  alakú a transzformációs lánc.

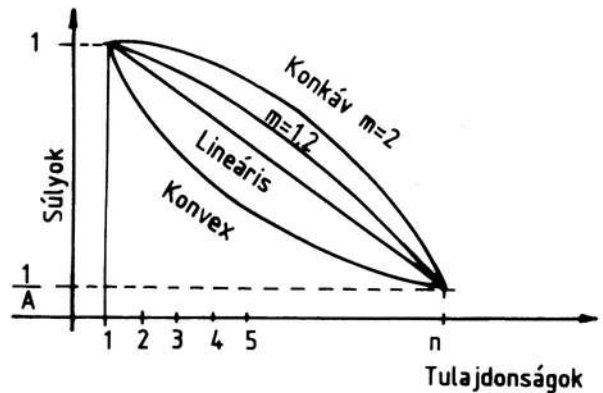
5. A tulajdonságokat (mutatókat) *súlyozhatjuk*. A súlyokat – melyek 0 és 1 közé eső értékek – alkalomszerűen is megválaszthatjuk, de alkalmazhatunk olyan módszert is, melynek segítségével szisztematikusan állíthatjuk elő azokat. A súlyoknak számítógepen való előállítása történhet úgy, hogy:

- ◆ minden tulajdonságot különböző súlyokkal látunk el,
- ◆ több tulajdonságnak azonos súlyai is lehetnek.

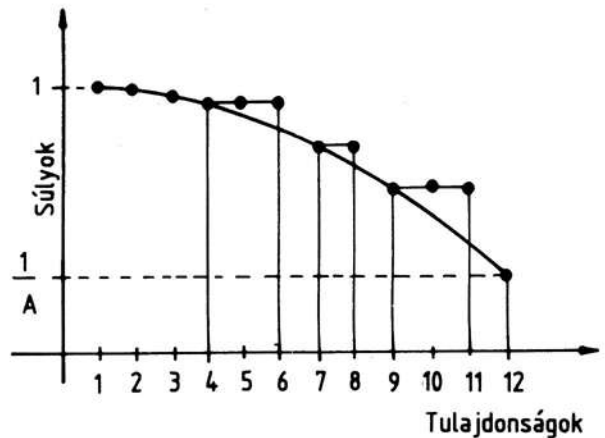
Ilyenkor a tulajdonságokat először fontosságuk szerint monoton csökkenő sorba kell rendezni, s dönteni kell afelől, hogy milyen típusú és mekkora  $m$  paraméterű (meredekségű) súlyfüggvényt alkalmazunk, s mekkora legyen a *legkisebb súlyérték* ( $= \frac{1}{A}$ , ahol  $A > 1$ ).

A program szerint *konvex*, *konkáv* vagy *lineáris* súlyfüggvény közül választhatunk (lásd a 2. és 3.

ábrát). A súlyfüggvényt előállító matematikai formulák az irodalomban [4] találhatóak.



2. ábra Különböző típusú súlyfüggvények



3. ábra Több tulajdonság azonos súllyal

Bizonyítható (lásd [6] 2. tétel), ha az alapadatok mátrixának soraiban az adatok "szóródása" (egy mástól való eltérése) nem nagy (homogenitás), vagyis ha az alapadatok átlagos relatív hibája kicsiny, akkor a súlyoknak a rangsorolásnál nincs számottevő szerepük. Más szóval a súlyozás a rangsorban nem hoz létre változást, s ha netán mégis, az nem számottevő. A súlyozásnak – és így a szakértők véleményének – csak akkor van jelentősége, ha inhomogenitással állunk szemben, vagyis ha az azonos tulajdonságokat reprezentáló számértékek, illetve mutatók számszerű értékeinek "szóródása" nagy.

6. A program minden objektumra kiszámítja a hasonlósági függvény számszerű értékét, vagyis előállítja a hasonlósági mutatót ( $H$ ), majd a  $k$ -adik objektumra számított  $H_k$  értékek monoton csökkenő sora alapján az objektumokat rangsorolja.

A hasonlósági transzformáció alkalmazása folytán a hasonlósági mutató tulajdonképpen a hasonlóságok hasonlóságát méri, illetve fejezi ki, ezért ez az

érték mindig közel esik 1-hez. Ha a hasonlósági mutatóban a tizedespont után megszakítás nélkül következő 9-esek számát  $\nu$ , az alapadatok átlagos relatív hibáját pedig  $\bar{\Delta}$  jelöli, akkor  $\nu$  és  $\bar{\Delta}$  összefüggése jó közelítéssel:

$\nu$	$\bar{\Delta}\%$
0	100
1	72,5
2	50,0
3	32,5
4	20,0
5	12,5
6	10,5

Ez egyben azt is jelenti, hogy a 9-esek száma azonnal informál bennünket az alapadatok szóródásáról is. Ezáltal egyetlen mutató kétféle információt hordoz:

- ◆ méri a hasonlóságot,
- ◆ kifejezi az alapadatok "szóródását".

Ha pl.  $H=0,9994768$ , akkor  $\nu=3$ , és így  $\bar{\Delta}\sim 32,5\%$ .

Gyakorlati megfontolások szükségessé tették a  $\gamma$ -szinten normált hasonlósági mutató fogalmának a bevezetését. A "normálás" széthúzza a hasonlósági skálát, méghozzá úgy, hogy a tizedespont után egymást követően lényegesen kevesebb 9-es számjegy jelenik meg. A gyakorlatban *gyengébb* homogenitás esetén  $\gamma=1$ , *erősebb* homogenitás esetén a  $\gamma=2$  választással célszerű élni. ( $H$ -ban az előbbit többnyire 3–4, az utóbbit 5–6 9-es jellemzi.)  $\gamma=1$  esetén az egyszerűen normált,  $\gamma=2$  esetén a kétszeresen normált elnevezést is használhatjuk. Mondhatjuk úgy is: a normálás egyszeres, illetve kétszeres. A RANG-64 ezzel a két "szinttel" számol. Ide vonatkozóan további információk az irodalomban található [6].

7. A programmal hátrább álló objektumot előre léptethetünk. Ha a felhasználó megadja, hogy az elérendő pozíció a rangsorban hányadik helynél ne legyen rosszabb, akkor a program megmondja (kijelzi), hogy melyik tulajdonság számszerű értékét mennyivel kell csökkenteni, vagy növelni ahhoz, hogy a rangsorban a megadott helyre, vagy annál előbbre a legkisebb ellenállás irányából jusson az objektum. Itt alkalmazzuk az irodalomban [5] található *optimalizálás minimális információval elvet*, illetve eljárást, pontosabban az ott bizonyított *alaptételt*.

Az előreléptetéskor két esetet különböztetünk meg:

- ◆ A szóba jövő objektum minden mutatója megváltoztatható a "kedvező" irányban.
- ◆ Az előreléptetendő objektumnál bizonyos mutatók értékeit nem lehet kedvező irányban változtatni. (Ez a helyzet előállhat azért, mert a változtatás eleve reménytelen, vagy mert más megfontolásból nem akarjuk őket változtatni.) Ilyenkor ezeket a mutatókat a többi közül elhagyjuk, s a maradék mutatórendszerrel végezzük el az előreléptetéseket, ha lehetséges.

Amikor korlátozás nélkül, minden mutatóra kiterjesztjük a léptetést, akkor azokat a mutatókat, amelyek értékei változatlanul maradnak, *domináns* vagy *ható*, esetleg *lényeges* mutatóknak nevezzük. Ezeknek a mutatóknak a kellő értéken való megléte esetenként – a rendszer szempontjából – döntően fontos lehet.

Előreléptetést a mutatókra súlyozás és egyedi alsó vagy felső korlátok alkalmazása mellett is végezhetünk. Ekkor bizonyítani kell (lásd [4]) azt is, hogy a "visszatranszformálás" után kapott új mutatóérték nem lépi túl az előre megadott alsó vagy felső egyedi korlátot. Ha a mutatók számítható módon összefüggésben állnak egymással, akkor az előreléptetéskor a túldetermináltság miatt az összefüggésekkel is számolni kell. Ilyenkor a mutatók közül – kellő körültekintés mellett – bizonyos számút el kell hagyni. A probléma részletesebb bemutatása az irodalomban található [4].

\*

## Irodalom

- [1] DOBÓ A.–SZAJCZ S.: Alap- és hasonlósági transzformáció. Kézirat, 1983.
- [2] DOBÓ A.–SZAJCZ S.: Vállalatok összehasonlítása mutatóik alapján. Kézirat, 1983.
- [3] DOBÓ A.–SZAJCZ S.: Mutatókat összehasonlító és kiértékelő eljárás matematikai programjának leírása. Kézirat, 1983.
- [4] DOBÓ A.–SZAJCZ S.: A becslési módszer és az előreléptető algoritmus továbbfejlesztése. Kézirat, 1984.
- [5] DOBÓ A.–SZAJCZ S.: A hasonlóságok növelése. Kézirat, 1982.
- [6] DOBÓ A.–SZAJCZ S.: A hasonlósági függvény újabb tulajdonságai. Kézirat, 1986.

*DOBÓ Andor—SZAJCZ Sándor: A RANG—64 működésének alapjai*

Az összemérhető termékek paraméterértékeinek "hasonlósága", illetve a "különbözőségük" is lehet a termékek összehasonlításának alapja. A program az objektumok (termékek) tulajdonságait (paraméterek) olyan kiinduló mátrixba rendezi, ahol a mátrix adatai az egyes termékek meghatározott tulajdonságaihoz tartozó számértékek, és ezt kiegészíti egy ún. vezérlő vektor. A fontosságuk szerint csökkenő sorba rendezett tulajdonságok termékenkénti paraméterértékei alapján a program minden objektumra kiszámítja a hasonlósági függvény számszerű értékét, és ennek alapján rangsorolja az objektumokat. A paraméterértékek változtatásának az objektumnak a rangsorban elfoglalt helyére vonatkozó hatása értékelhető, illetve elemezni lehet az előreleptetéshez szükséges paraméterváltoztatásokat.

\* \* \*

*ДОБО, А. — САЙЦ, Ш.: Основные принципы функционирования программной системы РАНГ-64*

Схожесть и различие сравнимых параметров изделий может послужить основой для сравнения изделий. Программная система РАНГ-64 распределяет свойства (параметры) изделий и объектов в такой исходный матрикс, где данные матрикса — цифровые показатели определяемого изделия и их дополнением является т. н. „управляющий вектор”. На основе параметров, распределенных по своей важности в уменьшающейся степени, программная система дает цифровой подсчет функционального ряда и таким образом классифицирует изделия. Можно сделать оценку изменения параметров или анализировать параметры для изменения.

*DOBÓ, A. — SZAJCZ, S.: The basics of RANG—64 operation*

The "similarity" or the difference of the parameter values of commensurable products can be the basis of product comparison. The RANG—64 program sorts the properties (parameters) of the objects (products) into a starting matrix where the elements of the matrix are numerical values belonging to specific properties of particular products, supplemented by a so-called control vector. The program calculates the numerical values of the similarity function for each object, based on the parameter values of the products sorted in descending order of their importance, and it produces a ranking of the objects. The effects of parameter value alteration on the ranking order of the objects can be evaluated, or the parameter alteration needed for higher ranking can be analyzed.

\* \* \*

*DOBÓ, A. — SZAJCZ, S.: Die Grundlagen der Funktion des RANG—64*

Die "Ähnlichkeit" der Parameterwerte der kommensurablen Produkte bzw. deren Unterschied kann auch die Grundlage des Vergleichs der Produkte sein. Das Programm ordnet die Eigenschaften (Parameter) der Objekte (Produkte) in eine Ausgangsmatrix, in der die Angaben der Matrix die zu den bestimmten Eigenschaften der einzelnen Produkte gehörenden Zahlwerte sind, und das ein sogenannter "Steuerungsvektor" ergänzt. Aufgrund der Produktenparameterwerte der nach ihrer Wichtigkeit in eine degressive Reihe geordneten Eigenschaften berechnet das Programm auf jeden Objekt den zahlenmässigen Wert der Ähnlichkeitsfunktion, und aufgrund dessen bestimmt es die Rangordnung der Objekte. Die Wirkung der Änderung der Parameterwerte auf die Stelle des Objekts in der Rangliste kann bewertet werden, bzw. die zum Vortreten nötigen Parameterabänderungen können analysiert werden.