Sztochasztikus folyamatok kezelése a városi koncentrált igénypont-halmazok szimulációs modellezésében

Dr. Bóna Krisztián*, Sárdi Dávid Lajos**

** Budapesti Műszaki és Gazdaságtudományi Egyetem, Közlekedésmérnöki és Járműmérnöki Kar, Anyagmozgatási és Logisztikai Rendszerek Tanszék, tanszékvezető, egyetemi docens (e-mail: krisztian.bona@logisztika.bme.hu)*

*** Budapesti Műszaki és Gazdaságtudományi Egyetem, Közlekedésmérnöki és Járműmérnöki Kar, Anyagmozgatási és Logisztikai Rendszerek Tanszék, PhD-hallgató (e-mail: david.sardi@logisztika.bme.hu)*

Absztrakt: Kutatásunkban a városi koncentrált igénypont-halmazok city logisztikai rendszereinek felmérésével, elemzésével, modellezésével, valamint lehetséges jövőbeli koncepcióinak kidolgozásával foglalkozunk. Ezek olyan igénypont-halmazok, amelyekben kis területen nagyszámú igénypont található, ez a koncentráltság pedig jó lehetőséget rejt magában a konszolidáció alapú fejlesztésekhez. Ebben a cikkünkben azt mutatjuk be, hogy az általunk kidolgozott szimulációs modelljeinkben hogyan kezeltük a különböző sztochasztikus folyamatokat, hogyan generáltuk például a szállítási tranzakciókat és paramétereikeit. Elsőként ismertetjük a sztochasztikus rendszereket általános jelleggel, majd néhány, a modelljeinkben is alkalmazott eloszlást is bemutatunk, egy 2015-ös tanulmány példáján keresztül szemléltetjük a city logisztikai modellek sokféleségét, majd pedig bemutatjuk, hogy négy különböző, általunk kidolgozott szimulációs modellben hogyan kezeltük a sztochasztikus city logisztikai folyamatokat.

1. BEVEZETÉS

Cikkünk az úgynevezett városi koncentrált igénypont-halmazok city logisztikai rendszereinek szimulációs modellezésével foglalkozunk. Ezek olyan igénypont-halmazok, ahol relatíve kis területen relatíve nagy számú igénypont helyezkedik el (ezek lehetnek ipari, kereskedelmi, vagy szolgáltató jellegű igénypontok is). Ezeken belül is megkülönböztetünk nyílt infrastruktúrájú, illetve zárt infrastruktúrájú halmazokat. Előbbiek esetén a városi közlekedési infrastruktúra (pl. utak, terek) jelölik ki a halmaz határait. Ilyen például egy bevásárlóövezet vagy bevásárlóutca. Zárt infrastruktúra esetén egy adott épület adja meg a halmaz határait. Ilyen egy bevásárlóközpont vagy egy zárt piac (vásárcsarnok), de például az egyetemek is számos szempontból hasonlóak ezekhez az igénypont-halmazokhoz. Kutatásunkban ezekkel az igénypont-halmazokkal foglalkozunk elsősorban. A BME Anyagmozgatási és Logisztikai Rendszerek Tanszékének City Logisztikai Kutatócsoportjában 2015 óta fókuszálunk ezekre (BME-ALRT, 2020; Bóna és Sárdi, 2019, 2020a; Sárdi és Bóna, 2019a; Bóna et. al., 2020).

Első lépésben tekintsük át a különböző modelltípusokat. A különböző, a közlekedéstudományban, azon belül is a logisztikában (valamint azon belül értelemszerűen a city logisztikában is) alkalmazott modellek csoportosítása esetén is kiemelt fontosságú szempont a véletlenszerűség. Ezeket a modelleket általában az alábbiak szerint csoportosíthatjuk (Bóna, 2015):

- **Modell realizációja**: folyamatmodellek, fizikai modellek, analitikus modellek, **szimulációs** modellek
- **Az elérni kívánt cél**: leíró modellek, heurisztikus jellegű modellek, optimalizáló modellek
- Számszerűsíthetőség szerint: kvalitatív modellek, kvantitatív modellek
- **Idő kezelése**: diszkrét modellek, folytonos modellek
- **Bizonytalanság kezelése**: determinisztikus modellek és **sztochasztikus** modellek (Bóna, 2015)

Esetünkben a sztochasztikus jellegű modellek, azok közül is a **szimulációs modellek** lesznek a legfontosabbak. Ezek lehetnek leíró jellegűek, lehetnek heurisztikus jellegű modellek, amikor egy elegendően jó megoldást szeretnénk találni a szimuláció során, valamint lehetnek optimalizáló modellek is. Ezen felül a szimulációs modelljeink szolgáltathatnak minőségi és mennyiségi adatokat is a vizsgált sztochasztikus rendszerrel kapcsolatosan, miközben vizsgálhatjuk a folyamatokat minden időpillanatban (folytonosan) és mintavételezve is (diszkréten).

2. SZTOCHASZTIKUS RENDSZEREK

A determinisztikus eseményekkel szemben a sztochasztikus rendszerekben olyan hatásokról, eseményekről és folyamatokról beszélhetünk, amelyek egyértelműen definiált körülmények között is csak valamilyen valószínűséggel következhetnek be. Ha ezek a **sztochasztikus események** bekövetkeznek, akkor értékük sem adható meg előre, az is bizonytalansággal terhelt. Az ilyen hatások kialakulásának és az ilyen jellegű események bekövetkezésének ideje, illetve értéke tehát **bizonytalan meghatározottságú**. Ezeket az

Online: ISBN 978-963-88875-6-6
Paper 18

Editor: **Dr. Péter Tamás**

eseményeket és folyamatokat **véletlen eseményeknek** és **folyamatoknak** nevezzük (Korondi et. al., 2014). Egy véletlen kísérlet bármelyik lehetséges eredményét **elemi eseménynek** nevezzük, és az elemi események halmaza alkotja az **eseményteret**. A **szimulációs modellezésben** ezeknek a véletlen kísérleteknek kiemelkedő szerepe van, mivel sztochasztikus rendszerek modellezése esetén az egyes szimulációs kísérletek során véletlen kísérleteket kell végrehajtanunk, melyek során a rendszer bemeneti paraméterei is lehetnek véletlenszerűek (például city logisztikai rendszerek esetén az adott üzletbe adott beszállítótól adott napon szállított árumennyiség), illetve az egyes folyamatok bekövetkezése is lehet véletlenszerű (például véletlenszerű lehet az, hogy egy adott üzletből adott napon adott városi területre történik-e házhozszállítás vagy nem). A **sztochasztikus szimuláció** az elmúlt évtizedekben kiemelt figyelmet kapott a modellezés területén (Asmussen és Glynn, 2003).

A **sztochasztikus folyamat** fogalmát akkor érthetjük meg jól, ha a valószínűségi változókra és ezek kiterjesztésére alapozzuk a megközelítést. Sztochasztikus rendszerek vizsgálata esetén az eseményekhez rendelt bekövetkezési esélyek meghatározásával értelmezhetjük a **valószínűség** fogalmát (Korondi et. al., 2014), ezekben a rendszerekben kiemelt fontosságú a valószínűségi változó fogalmának értelmezése. A valószínűségi változók olyan függvények, amelyek egy véletlen kísérlethez tartozó eseménytér minden kimeneteléhez, elemi eseményéhez egy valós számot rendelnek. Az X valószínűségi változó lehet **diszkrét vagy folytonos eloszlású**. Egy vizsgált city logisztikai rendszerben például diszkrét valószínűségi változó lehet az adott napon az üzletbe beérkező beszállító járművek száma. Ekkor az X értékkészlete megszámlálható számosságú számhalmaz. Ha a valószínűségi változó értékkészlete folytonos, akkor bármilyen valós számértéket tartalmazhat (Závoti, 2010). Az általunk vizsgált rendszerek sztochasztikus rendszerek, mert bizonytalanok és nem előre jelezhetők azok az időpontok, amikor az igények beérkeznek, és a csatornával szemben támasztott igények nagyságát sem lehet előre pontosan megmondani.

A műszaki, azon belül is a jelenleg elsősorban vizsgált közlekedési és logisztikai rendszerek jelentős része nem modellezhető statikus rendszerekként, ezek jellemzően dinamikus rendszerek, ezért nem lehet elhanyagolni az időbeli változékonyságukat, véletlenszerűségüket (Korondi et. al., 2014). Ezen rendszerek vizsgálata kapcsán fontos, hogy jól értsük a sztochasztikus folyamatot, amely mindig a realizációk összessége. Sztochasztikus folyamaton az elemi események halmazán értelmezett t paramétertől függő függvényt értünk. A **sztochasztikus folyamat egy realizációja** alatt egy az eseményhez tartozó t paramétertől függő függvényt értünk. Ezeknek a realizációknak az összessége alkotja magát a sztochasztikus folyamatot.

A sztochasztikus rendszerek elemzése esetén felmerül az a probléma, hogy a vizsgálat számára fontos és meghatározó valamilyen $\{X_t, t \in \mathcal{T}\}$ mennyiségek véletlenszerűek. Ezek a mennyiségek leggyakrabban adott objektumokra vonatkozó adatok időbeli és/vagy térbeli változását írják le, de akár mennyiség-beli változás is előfordulhat. Ha a $\mathcal{T}$ paraméterhalmaz része a pozitív valós számok halmazának, akkor a $\{t \in \mathcal{T}\}$ halmaz felfogható **időparaméterként** is, és ebben az esetben az $\{X_t, t \in \mathcal{T}\}$ valószínűségi változók együttesét a valószínűségelméletben megszokott módon sztochasztikus folyamatnak szokás nevezni. Az **időbeli változékonyságra** jó példa az igények véletlenszerű beérkezése, a **térbeli változást** pedig a házhozszállítási igények megjelenésének változékonyságával lehet szemléltetni egy city logisztikai rendszerben. **Mennyiség-beli** változás előfordulhat például a vevői igények esetén. A keresletelőrejelzés és a készlettervezés kapcsán kiemelt fontosságú a véletlenszerű vevői igények kezelése azáltal, hogy a mennyiség-beli változásának véletlenszerűségét jellemezni tudjuk Keresletelőrejelzésben például ilyen helyzet megoldását kezeli az úgynevezett Newsboy-modell.

Az eddigiek alapján **sztochasztikus folyamatok** alatt olyan véletlen jellegű, időben lejátszódó jelenségeket (folyamatokat) értünk, amelyeknek tetszőleges, a $\mathcal{T}$ paraméterhalmazhoz tartozó időpontokban felvett értékei valószínűségi változók. Attól függően, hogy ezeknek a valószínűségi változóknak mi az értékkészlete, beszélhetünk **valós értékű**, **komplex értékű** vagy éppen **többdimenziós sztochasztikus folyamatokról** (Michelberger et. al., 2003). Az általunk elsősorban vizsgált city logisztikai rendszerek is jellemzően valós értékű rendszerek.

A sztochasztikus folyamatotok osztályozásánál a leggyakrabban alkalmazott fogalmak **a stacioner sztochasztikus folyamatotok** és ezeken belül az **ergodikus sztochasztikus folyamatotok**. A $\xi(t, \omega)$ sztochasztikus folyamatot elsőrendben stacionernek nevezzük, ha minden ε értékre $F(x, t) = F(x, t + \varepsilon)$. Ekkor feltételezzük, hogy ez időtől függetlenül teljesül. Ilyenkor $f(x, t) = f(x, t + \varepsilon) = f(x)$. A $\xi(t, \omega)$ sztochasztikus folyamat **másodrendben stacioner** folyamat, amennyiben minden ε értékre $F(x_1, x_2, t_1, t_2) = F(x_1, x_2, t_1 + \varepsilon, t_2 + \varepsilon)$, valamint $f(x_1, x_2, t_1, t_2) = f(x_1, x_2, t_1 + \varepsilon, t_2 + \varepsilon)$ teljesül. Hasonlóan értelmezhetjük az **n-ed rendben stacioner** folyamatokat is. A $\xi(t, \omega)$ sztochasztikus folyamat **szigorúan stacioner** lesz, amennyiben minden ε értékre stacioner. Ha ez nem teljesül, akkor a folyamat **instacioner**. A $\xi(t, \omega)$ elsőrendben stacioner sztochasztikus folyamatot **elsőrendben ergodikusnak** nevezzük, ha 1 valószínűséggel meghatározható az egydimenziós eloszlás a folyamat bármely realizációjából.

Amennyiben a vizsgált idősor önmagában nem stacioner, de valamilyen determinisztikus trend kiszűrése után az lesz, akkor azt mondjuk, hogy az idősor egy **trendstacioner folyamat**, ez pedig a logisztika területén ismét a keresletelőrejelzésben lehet fontos, ahol kiemelt fontosságú az, hogy az elemzett idősorban be tudjuk azonosítani az alapjelt, a trendet, a szezont és a zajt is, annak érdekében, hogy megfelelő megbízhatósággal előre tudjuk jelezni a jövőbeli igényeket, melyek ismerete nélkülözhetetlen egy rendszer működtetéséhez. Ha feltételezhető az, hogy a vizsgált idősor mögött ilyen determinisztikus trend van, akkor annak eltávolításához először meg kell becsüljük annak paramétereit, majd egyszerűen ki tudjuk vonni az idősorból, ha pedig helyesen feltételeztük azt, hogy determinisztikus trenddel van dolgunk, akkor a folyamat során egy stacioner idősort kaptunk, azaz stacionarizáltuk a vizsgált idősorunkat (Winston, 2011).

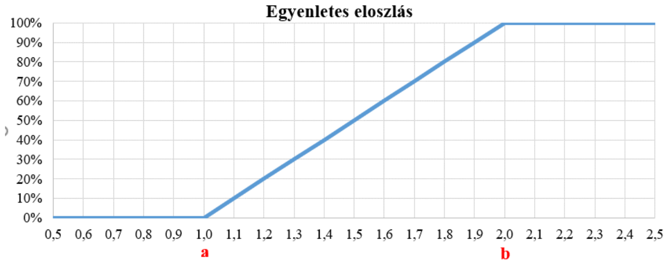
Sztochasztikus rendszerek ilyen jellegű vizsgálata során a másik alapvető trend-fogalmunk a **sztochasztikus trend**, melynek az idősorból történő kiszűréséhez a differenciálás nevű folyamattal kell úgynevezett **transzformált idősorokat** képeznünk. Ennek során mindig az eredeti idősor t. időpillanat beli és t-1. időpillanat beli értékének a különbségét kell vennünk. Amennyiben egy idősor önmagában nem stacioner, de a fentiekben ismertetett transzformáció során azzá válik, akkor a vizsgált idősor egy **differenciastacioner folyamat** (Winston, 2011). Ez a folyamat a logisztikában ismételten a keresletelőrejelzés során fontos, ahol a modellidentifikáció egyik legfontosabb lépése az, amikor transzformált idősorokat állítunk elő, melyekre a későbbiekben autokorrelációs függvényeket írunk fel (Clements és Hendry, 2001).

3. SZTOCHASZTIKUS FOLYAMATOK ELOSZLÁSA

Sztochasztikus rendszerek szimulációs úton történő vizsgálata során kiemelt fontosságú a rendszer jellemzőinek fő paramétéreinek (**várható érték**, **szórás**, **eloszlás)** megadása. Az eloszlás és annak paramétereinek ismerete nélkülözhetetlen egy szimulációs modell felépítése és futtatása során sztochasztikus rendszerek vizsgálata esetén, mivel ezen eloszlások alapján tudjuk felépíteni a véletlenszerűséget megvalósító **véletlenszám generátorokat**, a szimulációs modelljeinkben pedig ezek alapján tudjuk előállítani. Ezek a véletlenszám generátorok a különböző szimulációs eszközökben számos esetben a 0 és 1 között egyenletes eloszlású véletlenszám generátorra épülnek, így van ez például a logisztikai tervezésben gyakran alkalmazott MS Excel használata során is, ahol a VÉL() / RAND() függvényre tudjuk felépíteni a generátorainkat, valamint a kutatásunk során szintén alkalmazott Python-alapú szimulátor eszközben is a Python-ban elérhető 0 és 1 között egyenletes eloszlású véletlenszám generátorra épülnek a különböző generátorok.

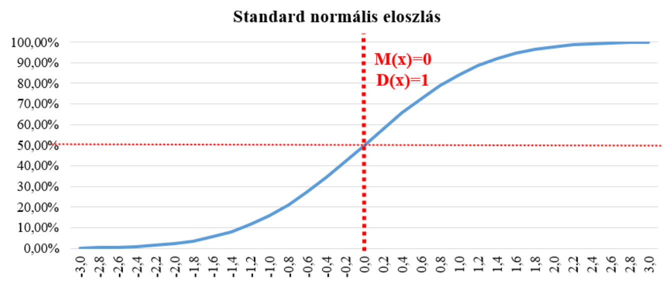
A leggyakoribb, logisztikai rendszerek vizsgálata kapcsán alkalmazott folytonos eloszlásfüggvények az egyenletes eloszlás, normális eloszlás (Gauss-eloszlás), exponenciális eloszlás és Weibull-eloszlás. Ezeket szeretnénk részletesebben is ismertetni, mivel sztochasztikus rendszerek szimulációs vizsgálata során kulcsszerepük van.

A **folytonos** eloszlású valószínűségi változót **egyenletes eloszlást követő**nek nevezzük az [a, b] intervallumon akkor, ha a valószínűségi változó]a, b[bármely részintervallumába az intervallum hosszával arányos valószínűséggel esik. A **diszkrét egyenletes eloszlásnál** pedig a valószínűségi változó minden lehetséges értékét ugyanakkora valószínűséggel veszi fel. A folytonos egyenletes eloszlást valósítja meg például Excelben is 0 és 1 között a VÉL() függvény, a diszkrét egyenletes eloszlást tetszőleges egész a és b között pedig a VÉLETLEN.KÖZÖTT(a;b) függvény. Az általunk felépített modellekben többnyire egyenletes eloszlással modelleztük a várhatóan megtett utat, az anyagmozgatási időszükségleteket és a szállítandó árumennyiségeket is. Az egyenletes eloszlásfüggvényt az alábbi 1. ábra mutatja meg, ahol a és b, az eloszlás alsó, illetve felső határa.

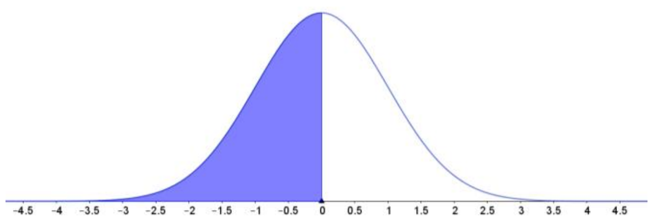


1. ábra: A folytonos egyenletes eloszlásfüggvény (a=1, b=2)

A **normális eloszlásnak** a valószínűségszámításban meghatározó szerepe van. A gyakorlatban, illetve a természetben előforduló számos valószínűségi változó normális eloszlású, vagy normális eloszlással közelíthető. Az általunk felépített city logisztikai modellekben például a házhozszállítási folyamatok vizsgálata során alkalmaztuk a normális eloszlást. A normális eloszlás sűrűségfüggvényének grafikonját a nagyon kifejező haranggörbének is szoktuk nevezni. A normális eloszlás eloszlásfüggvényt az alábbi 2. ábra mutatja meg, a haranggörbe pedig a 3. ábrán látható. Két paramétere a várható érték és a szórás. Az ábrán a standard normális eloszlás látható, melynek várható értéke 0, szórása pedig 1.



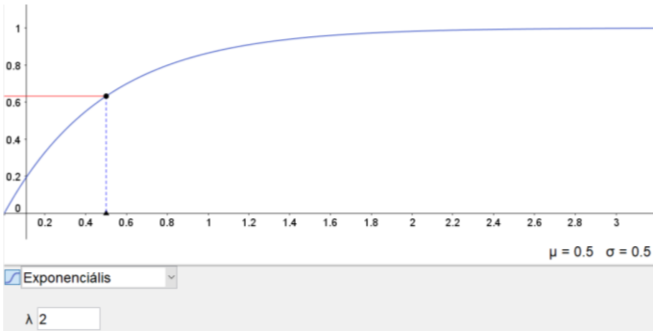
2. ábra: A standard normális eloszlásfüggvény



3. ábra: A standard normális eloszlás sűrűségfüggvénye GeoGebrában

A **háromszög eloszlás** a normális eloszlású valószínűségi változó egyik közelítése, melynek három fő paramétere van: „a" jelenti a minimum értéket, „b" a szokásos értéket (usual), „c" pedig a maximum értéket mutatja.

A következő fontos vizsgálandó eloszlás az **exponenciális eloszlás**. Exponenciális eloszlásúnak tekinthetők például a következő időtartamok: egy üzletbe érkező vásárlók között eltelt időtartamok vagy éppen várható várakozási időtartamok. Logisztikai rendszerek szimulációs úton történő vizsgálata során rendszeresen szoktunk azzal a feltételezéssel élni, hogy egy olyan folyamat, melynek nem ismert az eloszlása, csak a várható értéke, az exponenciális, így készülve fel a legrosszabb esetre és adva kellő biztonságot a rendszernek. Az exponenciális eloszlást elsősorban "örökifjú" tulajdonsága miatt használjuk. Az exponenciális eloszlás egyparaméteres, ez pedig a λ paraméter, mely a várható érték reciprokját adja. Az exponenciális eloszlásfüggvényt az alábbi 4. ábra mutatja meg.



4. ábra: A λ=2 paraméterű exponenciális eloszlás eloszlásfüggvénye GeoGebrában

A különböző eloszlások vizsgálata során érdemes még kitérni a **Weibull-eloszlásra** is, amelyet szintén rendszeresen alkalmazunk logisztikai folyamatok vizsgálata során. Ez egy kétparaméteres eloszlás, ahol „a” az alak- és „b” pedig a skálaparaméter. Ennek vizsgálata azért fontos számunkra, mert megfelelő paraméterekkel más eloszlásokat is megadhat, pl. a normális vagy az exponenciális eloszlást is, így nagy segítség lehet ismert adatsorok eloszlásának felírása során.

Sztochasztikus rendszerek szimulációs vizsgálata során szükséges lehet a fenti eseteken felül tetszőleges eloszlásfüggvényeket generálni. Ekkor többnyire a 0 és 1 közötti egyenletes eloszlású véletlenszám generátor szükséges. Az alkalmazott Python-alapú szimulációs környezetben például, amikor létrehozzuk a kívánt **eloszlásfüggvény**t, minden bemenő értékhez megadjuk annak előfordulási valószínűségét. Ezután képezzük ennek az eloszlásfüggvénynek az **inverz függvényét**, a véletlenszám generátort pedig az inverz függvény független változójaként alkalmazva, létrehoztuk a tetszőleges eloszlású véletlenszám generátort.

4. SZTOCHASZTIKUS FOLYAMATOK SZIMULÁCIÓS VIZSGÁLATA A CITY LOGISZTIKÁBAN

A city logisztikai területén számos különböző modell kidolgozásra került már különböző kutatási projektekben, illetve kutatásunk során mi is több különböző matematikai és szimulációs modellt is kidolgoztunk különböző eszközök alkalmazásával, különböző rendszerek vizsgálatára. Ebben a pontban először egy 2015-ös, a *Transport Reviews* folyóiratban megjelent cikk alapján szeretnénk szemléltetni a city logisztikai célú modellek sokféleségét, bemutatva ezáltal azt is, hogy a city logisztikai szimulációs modellekben hány féle jellemző véletlenszerűségét kéne biztosítani (ismertetve egyúttal azt is, hogy az általunk készített modellekben mely jellemzők jelentek meg a felsoroltak közül), majd bemutatjuk azt is, hogy a kutatásunk során elkészült eddigi modellekben hogyan is kerültek kezelésre a különböző sztochasztikus city logisztikai folyamatok.

4.1. Szimulációs modellek a city logisztikai rendszerek kutatásában

Nilesh Anand, Ron van Duin, Hans Quak és Lori Tavasszy 2015-ös munkájukban (Anand et. al., 2015) azt vizsgálták, hogy milyen jellegű modellek készültek el a korábbiakban a city logisztika területén. Ezen tanulmány eredményei alapján szeretnénk szemléltetni a city logisztikai modellek és az azokban kezelendő feladatok sokféleségét. A modelleket négy fő szempont szerint vizsgálták: a modellben vizsgált érdekelt felek (Stakeholders), a modellben alkalmazott meghatározó leíró jellemzők (Descriptor), a modell célfüggvénye (Objective) és az alkalmazott megoldási megközelítés (Solution approach) szerint csoportosították a vizsgált munkákat (Anand et. al., 2015). A kutatás során 31 modell került elemzésre.

Érdekelt felekként a projekt során a modellben definiált résztvevők, a vizsgált szereplők típusait definiálták (Anand et. al., 2015). Négy fő kategóriát határoztak meg, ezen szereplők különböző, többnyire sztochasztikus jellegű folyamatait kell kezelnie a különböző modelleknek:

- **adminisztrátor**: ide tartoznak a különböző hatóságok, melyek érdekeltek a városi logisztikai folyamatokban; ezeken keresztül képződnek le a lakók, és az utakat nem áruszállítási céllal használók szempontjai; a kutatásunk során fejlesztett modellekben ezeket a szereplőket nem vizsgáltuk;
- **beszállítók**: gyártók, nagykereskedők stb.; az általunk kidolgozott modellekben ezen szereplőktől indult el a city logisztikai folyamat;
- **fuvarozók**: a szállítási feladatokat ellátó harmadik felek, például logisztikai szolgáltatók; az általunk kidolgozott modellekben a jelenlegi rendszerben a beszállítók látták el az összes feladatot mezoszkopikus szinten, a jövőbeli, konszolidáció alapú rendszerekben pedig a beszállítók a konszolidációs központig és egy city logisztikai szolgáltató a konszolidációs központtól;
- **vevők**: az árut átvevő szereplők, üzletek, éttermek, szolgáltatók, háztartások; a modelljeinkben üzleteket, valamint a házhozszállítások végpontját jelentő háztartásokat modelleztünk.

A modellek **célfüggvénye** alatt értelemszerűen azt vizsgálták ebben a projektben is, hogy mi is az a probléma, amelynek megoldására az adott modellt létrehozták (Anand et. al., 2015).

A kutatás során hat különböző jellegű célfüggvényt definiáltak: **gazdasági, hatékonysági, közlekedésbiztonsági, környezeti, infrastruktúra és menedzsment,** valamint **városszerkezeti.**

Látható, hogy a különböző city logisztikai modelleknek számos különböző célja lehet, a célfüggvény értékeit pedig számos esetben különböző sztochasztikus folyamatok lefutása során kapjuk. Az összköltséget jelentősen befolyásolja a véletlenszerűen alakuló forgalom, a hatékonyság függ szintén a forgalomtól és a véletlenszerűen megjelenő igényektől. A közlekedésbiztonsági kérdésekben jelentős szerepe van a véletlenszerűen viselkedő szereplőknek (pl. gyalogosoknak, sofőröknek), a környezeti kérdéseket megint a véletlenszerűen alakuló forgalom, illetve az attól függő haladási sebesség is befolyásolja. Az infrastruktúra kérdéseire például a számos véletlenszerű tényezőtől függő rakodási idők fejtenek ki jelentős hatást, ezek pedig a városszerkezeti kérdéseket is befolyásolhatják. Kutatásunk során elsősorban gazdasági jellegű, hatékonysági és környezeti célokat vizsgálunk. Modelljeinkben vizsgáltuk, hogy a konszolidáció alapú rendszerek bevezetése hogyan hat **az összes logisztikai költségre**, amely például függ a véletlenszerűen alakuló szállítási időtől és a véletlenszerűen alakuló igényektől. Vizsgáltuk azt is, hogy a konszolidáció alapú rendszerek bevezetése hogyan hat a **szállítási tranzakciók szükséges számára**, amely függ a véletlenszerűen alakuló igényektől. Ezeken felül pedig vizsgáltuk azt is, hogy a konszolidáció alapú rendszerek bevezetése hogyan hat a **káros anyag kibocsátás alakulására**, amely szintén függ a véletlenszerűen alakuló szállítási időtől és igényektől.

A vizsgált kutatásban **leíró jellemzőkként** a modellépítés során figyelembe vett jellemzőket definiálták (Anand et. al., 2015), ezek az alábbiak voltak és az alábbi módokon jelennek meg az általunk elkészített szimulációs modellekben:

- **áruszállítási feladat generálása** (kiindulási hely és cél szerint): a modelljeinkben a feladatok véletlenszerűen jelentek meg, véletlenszerű árumennyiségekkel, ismert határértékek alapján (minimum és maximum mennyiségek, szállítások napi és havi maximális száma);
- **áruáramlás** (alkalmazott közlekedési alágazatok szerint): az áruáramlási folyamatot a vizsgált koncepció egyértelműen definiálta, azonban annak időtartama sztochasztikus volt modelljeinkben;
- **jármű kialakítása** (technológiai megoldás szerint, például kisméretű, alacsony kibocsátású jármű): modelljeinkben bemeneti adatként adott volt;
- **jármű megrakodása**: modelljeinkben időtartama véletlenszerű volt, végrehajtója azonban bemeneti adatként adott;
- **elhelyezkedés** (a városi területen belül): a résztvevők elhelyezkedése bemeneti adatként adott volt;
- **épület és telephely** (járművek és az épület/telephely kompatibilitása): a járművek és a telephelyek kompatibilitását feltételeztük, adott épületekhez;
- **járatok generálása**: a járatok generálása véletlenszerűen, a megjelenő véletlen igények függvényében történt;
- **különböző áruszállítási módok** megjelenése: az áruszállítási módokat a vizsgált koncepció definiálta a modelljeink esetén;
- **közlekedési infrastruktúra**: az infrastruktúrát a vizsgált koncepció definiálta a modelljeink esetén;
- **járműforgalom**: a járműforgalmat külön nem vizsgáltuk, az áruáramlási folyamatot a vizsgált koncepció egyértelműen definiálta, azonban annak időtartama sztochasztikus volt modelljeinkben;
- **ipar szerkezete** (a vizsgált city logisztikai rendszer résztvevői és kapcsolataik): a résztvevőket a vizsgált koncepció, illetve az adott bementi adatok definiálták.

Negyedik fő jellemzőként azt vizsgálták meg ebben a kutatásban, hogy milyen city logisztikai **megoldási megközelítést** vizsgáltak a megoldás során (Anand et. al., 2015). Ezek az alábbiak voltak:

- **Szabályozási**: azt vizsgálják ezekben a modellekben, hogy különböző szabályozási megoldások milyen hatásokat jelentenének (például súly és méret korlátok bevezetésének milyen hatása lenne).
- **Tervezési**: azt vizsgálják ezekben a modellekben, hogy a különböző tervezési döntéseknek milyen hatása van az áruszállításra (például az úthálózat, a rakodási lehetőségek, várakozási lehetőségek változása milyen hatást fejt ki).
- **Technológiai és információs**: azt vizsgálják ezekben a modellekben, hogy különböző technológiák bevezetése milyen hatással van az áruszállítási folyamatra (például intelligens közlekedési rendszerek alkalmazása, valós idejű követés alkalmazása milyen hatást fejt ki).

Modelljeinkben mi elsősorban technológiai megoldásokat vizsgáltunk, elsősorban a konszolidációs bevezetésének hatásait, konszolidációs központ és áruforgalmi zsilipek létesítésének hatásait, a városi kötött pályás áruszállítás bevezetésének hatásait, valamint cargo kerékpárok alkalmazásának hatásait.

A vizsgált kutatás során arra jutottak, hogy a legtöbb city logisztikai modell a fuvarozókat (16/31 modell) és az adminisztrátorokat (21 modell) vizsgálta, kevesebbet foglalkoztak a beszállítókkal (9 modell) és a vevőkkel (7 modell), melyek a mi modelljeinkben kulcsszerepet töltöttek be. A célfüggvény az esetek nagy többségében a hatékonyság volt (16 modell), de környezeti és infrastrukturális kérdésekkel is többen foglalkoztak (8, illetve 7 modell). Az alkalmazott leíró jellemző az esetek nagy többségében a járműforgalom (21 modell), a járatok generálása (18 modell) és a rakodás voltak (10 modell), ezekkel mi is foglalkoztunk modelljeinkben. A vizsgált modellek közül 9 vizsgálat a tervezési kérdések hatásait, 8 a szabályozási kérdéseket, 4 pedig a mi modelljeinkhez is hasonlóan a technológiai kérdésekkel foglalkozott (Anand et. al., 2015). Látható, hogy számos különböző city logisztikai modell létezik, különböző jellegű megoldásokkal és célokkal, azonban ez azt is jelenti, hogy számos különböző sztochasztikus folyamatot is kezelni kell: változhatnak a vevői igények, a szállítási és rakodási idők, az alkalmazott járművek, az útvonalak, a forgalom résztvevői is, ezek pedig hatnak a költségekre, a hatékonyságra, a káros anyag kibocsátásra, így pedig azok is véletlenszerűen fognak változni, számos paraméter függvényében.

A kutatás során a szerzők arra jutottak, hogy fontos lenne minden résztvevőt vizsgálni a city logisztikai modellekben, ki kell bővíteni az alkalmazott leíró jellemzők körét, valamint mind a hat lehetséges célt együttesen kellene vizsgálni annak érdekében, hogy környezetileg és gazdaságilag is fenntartható city logisztikai megoldásokat dolgozhassunk ki, valamint a jelenlegi modellek sokszor elhanyagolják a technológiai kérdéseket is, pedig sok esetben ezek jelenthetnék a megoldást (Anand et. al., 2015).

4.2. A kutatásunk során alkalmazott szimulációs megoldások

A városi koncentrált igénypont-halmazok jelenlegi és tervezett jövőbeli, konszolidáció alapú megoldásainak vizsgálata során az eddigiekben az alábbi szimulációs modellek kerültek kidolgozásra:

- **Mezoszkopikus szintű szimulációs modell MS Excelben** (Sárdi és Bóna, 2017, 2019a, b; Bóna és Sárdi, 2019, 2020a): a jelenlegi city logisztikai rendszer szimulációs modellje; valamint a jövőbeli, konszolidáció alapú city logisztikai rendszer szimulációs modellje (tehergépkocsik; illetve áruszállító villamos szerelvények alkalmazásával).
- **Makroszkopikus szintű szimulációs modell MS Excelben** (Sárdi és Bóna, 2018; Bóna és Sárdi, 2020b): a jelenlegi city logisztikai rendszer szimulációs modellje; valamint a jövőbeli, geometriai modell alapú koncepció alapján kidolgozott, konszolidáció alapú city logisztikai rendszer szimulációs modellje, tehergépkocsikkal és cargo kerékpárokkal.
- **Mezoszkopikus szintű pilot szimulációs modell AnyLogic-ban** (Bóna és Sárdi, 2020a; Bóna et. al., 2020): a jelenlegi city logisztikai rendszer pilot szimulációs modellje; valamint a jövőbeli, konszolidáció alapú city logisztikai rendszer pilot szimulációs modellje (tehergépkocsik alkalmazásával).
- **Mikroszkopikus szintű pilot szimulációs modell Pythonban** (Bóna és Sárdi, 2020a; Bóna et. al., 2020): a jelenlegi city logisztikai rendszer pilot szimulációs modellje (jelenleg tesztelés alatt); valamint a jövőbeli, konszolidáció alapú city logisztikai rendszer pilot szimulációs modellje (tehergépkocsik alkalmazásával; jelenleg fejlesztés alatt)

Ebben a fejezetben elsősorban a szimulációs modellekből a sztochasztikus folyamatok kezelését, azon belül is a beszállítások generálást szeretnénk bemutatni, mivel a szimulációs eredményekkel kapcsolatosan már számos publikációt készítettünk a korábbiakban.

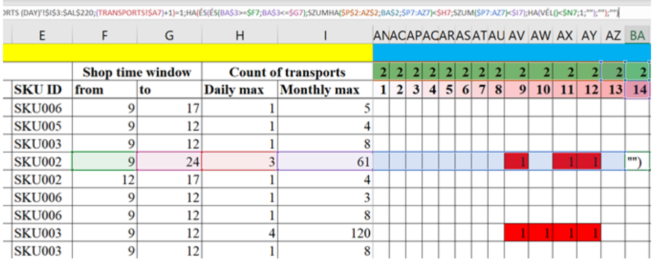
Az MS Excel alapú, mezoszkopikus szintű szimulációs modell alapeleme a beszállítás-generátor, amely azt mutatja meg, hogy adott beszállítási tranzakció mikor valósul meg, erre épül a szimulációs modell összes többi része. A jelenlegi rendszer beszállításai esetén ez a generátor két fő komponensből áll. A beszállítás-generátor első része azt határozza meg, hogy adott napon lesz-e beszállítás a vizsgált koncentrált igénypont-halmaz adott üzletébe, a második komponens pedig megmutatja, hogy adott napon belül mely órában valósul meg a beszállítás (Sárdi és Bóna, 2017) (Sárdi és Bóna, 2019b) (Bóna és Sárdi, 2019) (Sárdi és Bóna, 2019a) (Bóna és Sárdi, 2020a). Ekkor $x_{i,j}$ lesz az a változó az (1) formula szerint, amely megmutatja, hogy a j. napon megvalósulhat-e az i. típusú beszállítási tranzakció (amely egy adott üzlethez tartozik, de egy üzlethez tartozhat több különböző beszállítási tranzakció is). A beszállítási napok száma ekkor: $N_i^{sup\_day} = \min\left\{N_i^{max\,(month)}; 30\right\}$, ahol $N_i^{max\,(month)}$ a beszállítások havi maximális száma $\mathbf{r_{i,j}}$ **0 és 1 közötti egyenletes eloszlású véletlen szám** (ennek köszönhetjük a beszállítási tranzakciót véletlenségét), q_i pedig az adott beszállításhoz tartozó valószínűségi érték.

$$x_{i,j} = \begin{Bmatrix} 0,\text{ha} \sum_{k=1}^{j-1} x_{i,k} = N_i^{sup\_day} \\ 0,\text{ha } x_{i,j} - 1 = 1 \text{ és } N_i^{sup\_day} < 7 \\ 0,\text{ha } \sum_{k=1}^{j-1} x_{i,k} < N_i^{sup\_day} \text{ és } r_{i,j} \geq q_i \\ 1,\text{ha } \sum_{k=1}^{j-1} x_{i,k} < N_i^{sup\_day} \text{ és } r_{i,j} < q_i \end{Bmatrix} \tag{1}$$

A beszállítás-generátor második komponense azt fogja megadni, hogy az adott nap adott órájában lesz-e beszállítás. Ekkor $x_{i,j}^l$ azt mutatja meg a (2) egyenlet szerint, hogy a j. nap l. órájában megvalósul-e az i. típusú beszállítási tranzakció, $\mathbf{R_{i,j}^l}$ **0 és 1 közötti egyenletes eloszlású véletlen szám**, Q_i az adott beszállításhoz tartozó valószínűség, TW-vel a vizsgált üzletek időablakát, illetve annak alsó és felső határát jelöltük, $N_i^{max\,(day)}$ pedig a beszállítások napi maximális száma. Az adott órában csak akkor lehet beszállítás, ha a generátor első komponense azt mutatja, hogy az adott napon lehet beszállítási tranzakció. Amennyiben az első komponens alapján nincs beszállítás, akkor a második komponens egyértelműen 0 értéket ad vissza.

$$x_{i,j}^l = \begin{Bmatrix} 1,\text{ha} \begin{Bmatrix} x_{i,j} = 1 \\ TW_i^{CSDL(UL)} \leq k \\ TW_i^{CSDL(LL)} \geq k \\ \sum_{k=1}^{l-1} x_{i,j}^k < N_i^{max\,(day)} \\ \sum_{h=1}^{j-1} \sum_{g=1}^{24} x_{i,h}^g + \sum_{k=1}^{l-1} x_{i,j}^k < N_i^{max\,(month)} \\ R_{i,j}^l < Q_i \end{Bmatrix} \\ 0,\text{különben} \end{Bmatrix} \tag{2}$$

Ennek a generátornak egy részletét mutatja meg az 5. ábra.



RTS (DAY)'!$I$3:$AL$220;(TRANSPORTS!$A7)+1)=1;HA(ÉS(ÉS(BA$3>=$F7;BA$3<=$G7);SZUMHA($P$2:AZ$2;BA$2;$P7:AZ7)<$H7;SZUM($P7:AZ7)<$I7);HA(VÉL()<$N7;1;"");"");"")

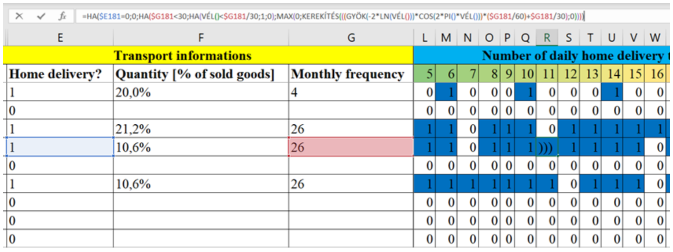
E	F	G	H	I	AN	AC	AP	AC	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA
	Shop time window		Count of transports		2	2	2	2	2	2	2	2	2	2	2	2	2	2
SKU ID	from	to	Daily max	Monthly max	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SKU006	9	17	1	5														
SKU005	9	12	1	4														
SKU003	9	12	1	8														
SKU002	9	24	3	61									1		1	1		"")
SKU002	12	17	1	4														
SKU006	9	12	1	3														
SKU006	9	12	1	8														
SKU003	9	12	4	120									1	1	1	1		
SKU003	9	12	1	8														

5. ábra: Beszállításgenerátor a jelenlegi rendszer mezoszkopikus szintű szimulációs modelljében, MS Excelben

Természetesen a modell további részeiben is kulcsszerepet játszik a véletlen számok generálása: az egy tranzakció során kezelendő árumennyiség, a szállítási idő, az anyagmozgatási idő, illetve egyes költségelemek is megadott minimum és maximum határok között véletlenszerűen alakulnak, mivel a valós adatok alapján közülük számos csak intervallumosan ismert. A szimulációs modellben a göngyölegek kezelése a beszállítási folyamat inverzeként lett leképezve. Az inverz logisztikai jellegű tranzakciók közül a göngyölegkezelési feladatok mellett a házhozszállítási tranzakciókat modelleztük még a valós adatok alapján megadott házhozszállítási részarányból kiindulva. A házhozszállítási tranzakciók (azaz házhozszállítást végző körjáratok száma a j. napon, normális eloszlást feltételezve) az alábbi lesz. Ebben a (3) formulában **$r_{i,j}$ 0 és 1 közötti egyenletes eloszlású véletlen szám, $r^*_{i,j}$ pedig normális eloszlású, 0 várható értékű, 1 szórású véletlen szám**, ezáltal pedig itt az egyenletes eloszlás és a normál eloszlás is megjelenik a házhozszállítási tranzakciók generálása során. A formulában p_i^{hd} a házhozszállítási tranzakciók várható részaránya (ami azt adja meg, hogy az összes áru mekkora aránya jut célba házhozszállítással), N_i^{hd} pedig a házhozszállítási napok száma.

$$N_{i,j}^{hd} = \begin{cases} 0, \text{ha } p_i^{hd} = 0 \\ 0, \text{ha } p_i^{hd} > 0, N_i^{hd} < 30 \text{ és } r_{i,j} \geq \frac{N_i^{hd}}{30} \\ 1, \text{ha } p_i^{hd} > 0, N_i^{hd} < 30 \text{ és } r_{i,j} < \frac{N_i^{hd}}{30} \\ \max\left\{0; \frac{N_i^{hd}}{30} + \frac{N_i^{hd}}{60} * r^*_{i,j}\right\}, \text{ha } p_i^{hd} > 0 \text{ és } N_i^{hd} \geq 30 \end{cases} \quad (3)$$

A házhozszállítási napok generálását a következő 6. ábra mutatja meg.



=HA($E181=0;0;HA($G181<30;HA(VÉL()<$G181/30;1;0);MAX(0;KEREKÍTÉS(((GYÖK(-2*LN(VÉL()))*COS(2*PI()*VÉL()))*($G181/60)+$G181/30);0))))

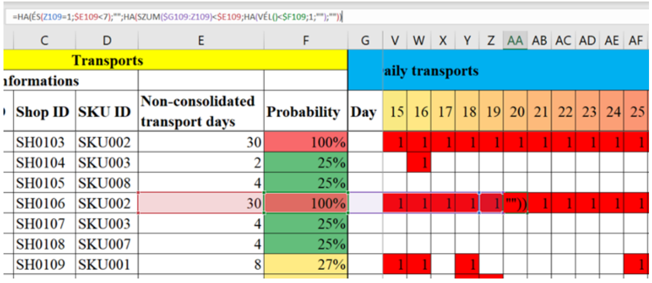
E	F	G	L	M	N	O	P	Q	R	S	T	U	V	W
Transport informations			Number of daily home delivery											
Home delivery?	Quantity [% of sold goods]	Monthly frequency	5	6	7	8	9	10	11	12	13	14	15	16
1	20,0%	4	0	1	0	0	0	1	0	0	0	1	0	0
0			0	0	0	0	0	0	0	0	0	0	0	0
1	21,2%	26	1	1	0	1	1	1	0	1	1	1	1	1
1	10,6%	26	1	1	0	1	1	1	)))	1	1	1	1	0
0			0	0	0	0	0	0	0	0	0	0	0	0
1	10,6%	26	1	1	1	1	1	1	1	0	1	1	1	0
0			0	0	0	0	0	0	0	0	0	0	0	0
0			0	0	0	0	0	0	0	0	0	0	0	0
0			0	0	0	0	0	0	0	0	0	0	0	0

6. ábra: Házhozszállítási napok generálása a jelenlegi rendszer mezoszkopikus szintű szimulációs modelljében, MS Excelben

A mezoszkopikus szintű modellben vizsgáltuk a jövőbeli city logisztikai rendszert is, ahol konszolidációs központot helyezünk el a beszállítók és a vizsgált koncentrált igénypont-halmazok között, az igénypont-halmazok beszállítói udvarait pedig áruforgalmi zsilipekkel helyettesítettük. A beszállítóktól érkező tranzakciók generálása teljes mértékben megegyezik a jelenlegi rendszer kapcsán bemutatott megoldással (annyi különbséggel, hogy a koncentrált igénypont időablaka helyett a konszolidációs központ időablakához kell igazodni), azonban külön kellett generálni a konszolidációs központból induló szállítási feladatokat, azaz azt, hogy adott napon adott üzletbe kell-e beszállítást végezni a konszolidációs központból. A konszolidációs központ és a koncentrált igénypont-halmazok közötti folyamatokat teljes mértékben napi bontásban írtuk fel, így a generátornak (esetünkben kiszállításgenerátornak) egy komponense volt csak. A következő (4) formulában $x_{i,j}^{CSDL}$ az a változó, amely megmutatja, hogy a j. napon megvalósulhat-e az i. konszolidált beszállítási tranzakció a koncentrált igénypont-halmazhoz. **Az $r_{i,j}$ 0 és 1 közötti egyenletes eloszlású véletlen szám** a korábbi megoldásokhoz hasonlóan, $N_i^{max(month),CC}$ a konszolidált beszállítások havi maximális száma, q_i^{CSDL} pedig az adott beszállításhoz tartozó valószínűségi érték, amelyet a korábbiakhoz hasonlóan kísérleti úton határoztunk meg.

$$x_{i,j}^{CSDL} = \begin{cases} 0, \text{ha } \sum_{k=1}^{j-1} x_{i,k}^{CSDL} = N_i^{max(month),CC} \\ 0, \text{ha } x_{i,j-1}^{CSDL} = 1 \text{ és } N_i^{max(month),CC} < 7 \\ 0, \text{ha } \sum_{k=1}^{j-1} x_{i,k}^{CSDL} < N_i^{max(month),CC} \text{ és } r_{i,j} \geq q_i^{CSDL} \\ 1, \text{ha } \sum_{k=1}^{j-1} x_{i,k}^{CSDL} < N_i^{max(month),CC} \text{ és } r_{i,j} < q_i^{CSDL} \end{cases} \quad (4)$$

Az új rendszer kiszállításgenerátornak egy részletét mutatja meg a 7. ábra.



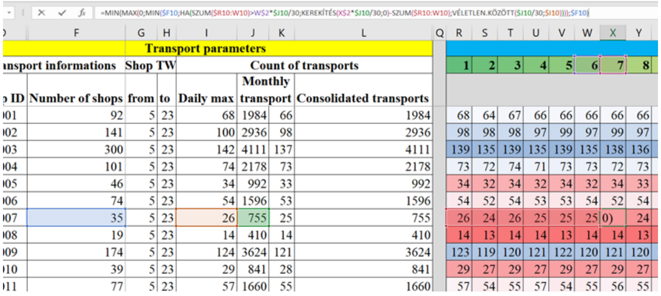
=HA(ÉS(Z109=1;$E109<7);"";HA(SZUM($G109:Z109)<$E109;HA(VÉL()<$F109;1;"");""))

C	D	E	F	G	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
Transports				aily transports											
nformations															
Shop ID	SKU ID	Non-consolidated transport days	Probability	Day	15	16	17	18	19	20	21	22	23	24	25
SH0103	SKU002	30	100%		1	1	1	1	1	1	1	1	1	1	1
SH0104	SKU003	2	25%			1									
SH0105	SKU008	4	25%												
SH0106	SKU002	30	100%		1	1	1	1	1	""))	1	1	1	1	1
SH0107	SKU003	4	25%												
SH0108	SKU007	4	25%												
SH0109	SKU001	8	27%		1	1		1							1

7. ábra: Kiszállításgenerátor a jövőbeli rendszer modelljében, a konszolidációs központból az üzletek felé, MS Excelben

A házhozszállítási tranzakciók generálását is kis mértékben át kellett alakítani a jelenlegi rendszer modelljéhez képest, de jelentős változás nem történt, hasonló struktúrában, hasonló eloszlásokat alkalmaztunk az i. típusú konszolidált beszállítási tranzakcióhoz tartozó i. típusú házhozszállítási tranzakció generálása kapcsán. A göngyölegkezelési tranzakciókat az új rendszer modelljében is a beszállítások inverzeként kezeltük.

Kutatásunk során, az előbbiekben bemutatott mezoszkopikus szintű MS Excel alapú modell alapján elkészült egy kevésbé részletes (magasabb szintű aggregációval dolgozó), makroszkopikus szintű szimulációs modell is MS Excelben (Sárdi és Bóna, 2018) (Bóna és Sárdi, 2020b). Ebben a szimulációs modellben egy speciális, cargo kerékpáros city logisztikai rendszert vizsgáltunk, ahol a koncentrált igénypont-halmazok beszállításait szintén konszolidációs központból bonyolítjuk le, innen tehergépkocsikkal szolgáljuk ki a városi belső gyűrű koncentrált igénypont-halmazait, a belső gyűrű áruforgalmi zsilipjeiből pedig cargo kerékpárokkal (illetve egyes koncepciókban elektromos kistehergépkocsikkal) látjuk el a gyűrűről induló sugarak mentén található további igénypont-halmazokat. A beszállításgenerátor alapvetően a beszállítási napokat generáló megoldáshoz hasonló, annak alapján készült el. A fő különbség az, hogy míg mezoszkopikus szinten üzletek szintjén aggregáltuk a feladatokat, itt már koncentrált igénypont-halmaz szinten vizsgálódtunk, így pedig már nem szállítási napokat, hanem adott napi beszállítási igények számát generáltuk, azaz lényegében azt, hogy adott napon, adott koncentrált igénypont-halmazból hány üzlet igényel beszállítást a konszolidációs központból. Ennek a generátornak egy részletét mutatja be a 8. ábra, a matematikai háttér teljesen hasonlóan alakul a mezoszkopikus modellhez, egyenletes eloszlás szerint történik az igények generálása.


=MIN(MAX(0;MIN($F10;HA(SZUM($R10:W10)>W$2*$J10/30;KEREKÍTÉS(X$2*$J10/30;0)-SZUM($R10:W10);VÉLETLEN.KÖZÖTT($J10/30;$I10))));$F10)

D	F	G	H	I	J	K	L	Q	R	S	T	U	V	W	X	Y
	Transport parameters															
ansport informations		Shop TW		Count of transports					1	2	3	4	5	6	7	8
p ID	Number of shops	from	to	Daily max	Monthly transport		Consolidated transports									
001	92	5	23	68	1984	66	1984		68	64	67	66	66	66	66	66
002	141	5	23	100	2936	98	2936		98	98	98	97	99	97	99	97
003	300	5	23	142	4111	137	4111		139	135	139	135	139	135	138	136
004	101	5	23	74	2178	73	2178		73	72	74	71	73	73	72	73
005	46	5	23	34	992	33	992		34	32	34	32	34	32	34	33
006	74	5	23	54	1596	53	1596		54	52	54	53	53	54	52	54
007	35	5	23	26	755	25	755		26	24	26	25	25	25	0)	24
008	19	5	23	14	410	14	410		14	13	14	14	13	14	14	13
009	174	5	23	124	3624	121	3624		123	119	120	121	122	120	121	120
010	39	5	23	29	841	28	841		29	27	29	27	29	27	29	27
011	77	5	23	57	1660	55	1660		57	54	55	57	54	55	56	55

8. ábra: Beszállításgenerátor a cargo kerékpárokat alkalmazó city logisztikai rendszer modelljében, MS Excelben

A modell további részeiben a szállított árumennyiségek generálása szintén véletlenszám generátorok alkalmazásával, megadott minimum és maximum mennyiségek alapján történik.

Az eddigiekben bemutatott MS Excel alapú (mezoszkopikus szintű) szimulációs modell jól használható és jól fejleszthető. A mezoszkopikus szintű modell utolsó változatában 4 bevásárlóközpont teljes logisztikai rendszerét vizsgáltuk, több, mint 200 üzlet adataival, így pedig a modell túl nagyméretűre nőtt. Méretéből adódóan a megfelelő számú szimulációs futtatás végrehajtása rendkívül lassúvá vált. Emiatt arra jutottunk, hogy a további fejlesztésekhez és szimulációs vizsgálatokhoz vagy teljesen át kellene alakítani a jelenlegi modell struktúrát, vagy más szimulációs megoldásokat kell keresnünk. Első lépésben az AnyLogic szimulációs szoftvert választottuk ki (AnyLogic, 2016). Ebben az eszközben 1 bevásárlóközpont folyamataira dolgoztunk ki egy mezoszkopikus szintű pilot szimulációs modellt (Bóna és Sárdi, 2020a) (Bóna et. al., 2020), következőkben pedig a véletlenszerűség kezelését szeretnénk ismertetni ezzel a megoldással kapcsolatosan. A modellépítés során az első lépés a megfelelő objektumok felvétele volt, melynek eredményeképpen megkaptuk azt a logisztikai hálózatot, amin dolgoznunk kellett.

Kiemelendő, hogy az AnyLogic esetén komplexebb feladat volt a modellépítés: különböző objektumokat (ágenseket) kellett felvenni, GIS-alapú felületet alkalmaztunk a logisztikai hálózat leképezéséhez, a folyamatokat állapotábrák és folyamatábrák képezték le, SQL-alapú adatbázis kapcsolatokat kellett létrehozni, a folyamatok működtetését pedig JAVA-alapú kódolással valósítottuk meg. Mivel egy valós hálózatot modelleztünk AnyLogic-ban, térkép alapon, így a logisztikai hálózaton való haladásban is szükség volt bizonyos szintű véletlenszerűségre, a haladási sebességekkel kapcsolatosan véletlen értékeket kellett generálnunk, nem tudtuk egyszerűen a szállítási időket véletlenné tenni, mint az MS Excel alapú megoldásban.

Az objektumok létrehozása után létre kellett hoznunk a járművek populációját, előbbiek szállították a beszállítók és a koncentrált igénypont-halmaz között az árut (illetve hasonlóan az új rendszerben a beszállítók és a konszolidációs központ, illetve a konszolidációs központ és a koncentrált igénypont-halmaz között). A felvett járművekhez a már ismertetett okok miatt sebességértékeket is fel kellett vennünk. A városon belülről érkező szállítások esetén 25 km/h várható értékű (10 minimumú, 50 maximumú, 5 szórású) **normális eloszlást**, városon kívülről érkező szállítások esetén pedig 70 km/h várható értékű (50 minimumú, 110 maximumú, 5 km/h szórású) normális eloszlást vettünk fel az alábbi módon, JAVA-alapú kód alkalmazásával:

```
(this.supplier1 == 1) ? normal(10, 50, 25, 5)
: normal(50, 110, 70, 5);
```

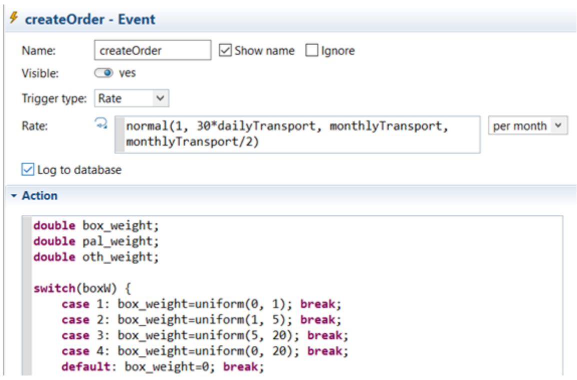
Az objektumok felvétele, a megfelelő adatbázis kapcsolatok megvalósítása, valamint az idő kezelésének megvalósítása után tudtuk végrehajtani a beszállítási folyamat modellezését. Ehhez létre kellett hozni egy megrendelés-generáló parancsot a megrendeléseket generáló eseménynél. Itt meg kellett adni egy havi igénykeletkezési gyakoriságot, és az AnyLogic beépített eloszlásait használva az Excel-alapú modelltől eltérve **normális eloszlást alkalmaztunk** (melynek várható értéke a havi beszállítások számának várható értéke, minimuma 0, maximuma a napi beszállítások maximális számának harmincszorosa, szórása pedig a havi beszállítások maximális számának fele):

```
normal(1, 30*dailyTransport, monthlyTransport,
monthlyTransport/2)
```

A keletkező igényekhez ezen felül egy mennyiséget is meg kellett határozni, ezt a matematikai modellnek megfelelően hajtottuk végre a különböző szállítási egységek számának és méretének megfelelően, megadott minimum és maximum határok között, **egyenletes eloszlás** alapján. A megfelelő paraméterű véletlen megrendeléseket generáló eseményt a következő oldal 9. ábrája mutatja be.

A jövőbeli city logisztikai rendszer pilot modelljéhez a beszállítások generálását át kellett alakítanunk, mivel az MS Excel alapú modellhez hasonlóan, a konszolidációs központ készletezési feladatokat is átvállal, így egyszerre nagyobb mennyiségek is beszállíthatók, így a beszállítások száma is megváltozik. A beszállítások napi számát így nem a beszállítások havi maximális számából határoztuk meg, hanem az abból meghatározott havi összevont beszállítási tranzakciók számából (ennyiszer kell a beszállítóknak a készletezési feladatokat átvevő konszolidációs központba szállítani egy hónapban, ez lett a várható érték, a maximum pedig a beszállítások korábbi havi száma, vagy amennyiben az havi 30-nál több, akkor 30) az alábbi módon:

```
normal(1, min(30,monthlyTransport),
suptransportDays, suptransportDays/2)
```



9. ábra: Véletlen megrendelések generálása AnyLogic-ban

Ebből adódóan az egy szállításhoz generált árumennyiség is megváltozott, figyelembe kellett vennünk, hogy egy összevont beszállításban hány egyszerű beszállítási tranzakció került összevonásra.

Az AnyLogic alapú megoldásban ezek az elemek biztosították a véletlenszerűséget. A kapott eredmények alapján ez is egy jó fejlesztési irány lehet, azonban ennek kapcsán is megfogalmazódott néhány probléma. Legfontosabb ezek közül az, hogy az elérhető verzióban limitált a felhasználható objektumok száma. Másik jelentős probléma volt az a modellépítési feladat túlzott komplexitása volt, egy ilyen részletes modellnél pedig ez jelentősen nehezíti a modell utólagos módosítását, átalakítását (pl. bevásárlóközpontról bevásárlóövezetre való átalakítását). Emiatt úgy döntöttünk, hogy elkezdünk egy olyan irányban gondolkozni, ami rugalmasabb, mint az Excel, egyszerűbben módosítható, mint az AnyLogic-alapú modellek, valamint tetszőleges nagyságú modelleket tudunk felépíteni benne. Ez vezetett minket a **Python-programozási nyelv** alkalmazásához (Python, 2020). Jelenleg ennek a modellnek a fejlesztése a kutatásunk legfontosabb feladata. A Python-alapú szimulációs modellünket a Dr. Lipovszki György által fejlesztett, Python-alapú DES-szimulátor (diszkrét esemény alapú szimulátor) környezetben (Lipovszki, 2020) kezdtük felépíteni. Ebben a környezetben a korábbi mezoszkopikus szintnél már egy lépcsővel mélyebbre mentünk, és nem csak üzletek szerinti bontásban, hanem termék szerint vizsgáljuk a folyamatokat, egy mikroszkopikus szintű szimulációs modellben. Jelenleg a jelenlegi rendszer pilot modelljét vizsgáljuk, amely két beszállítót tartalmaz, négy termékfajtával, valamint egy koncentrált igénypont-halmaz két üzletével, keresztkapcsolatokkal (azaz mindkét beszállító szállít mindkét üzletbe) (Bóna és Sárdi, 2020a) (Bóna et. al., 2020). Első feladatként a futási idők alakulásával kapcsolatban szeretnénk eredményeket kapni, megpróbálva előre jelezni a teljes modellel kapcsolatos futási időket. A tesztelés során kulcskérdés az alkalmazott eloszlások paraméterezése. A modell fejlesztését és tesztelését a Spyder fejlesztőkörnyezetben bonyolítjuk le (Spyder, 2020). Az egyes paraméterek eloszlásainak beállítását az alábbi 11. ábrán látható módon tudjuk végrehajtani. A modell jelenlegi pilot változatában mindössze a rendszerbe kerülő áru mennyisége véletlenszerű, a további paraméterek a jelenlegi fázisban konstans értékekként vannak kezelve.

```
### SKU001 Source paraméterek

SKU001_Termeles__Time_Step   = minutes(16)/n
SKU001_Termeles_Min__Time_Step   = minutes(14)/n #várható érték 15, az
SKU001_Termeles_Max__Time_Step   = minutes(16)/n #várható érték 15, az
SKU001_Termeles_Distribution = 'TDistribution_Uniform_Minimum_Maximum'
SKU001_Termeles_Batch_Size = 10000000*n #3000 #tetszőleges hosszúságú

SKU001_SUP01_INIT_Batch_Size = 200*n #200 darabos kezdőkészlettel indí

SKU001_SH001_INIT_Batch_Size = 300*n #300 darabos kezdőkészlettel indí

### SKU002 Source paraméterek

SKU002_Termeles__Time_Step   = minutes(13)/n
SKU002_Termeles_Min__Time_Step   = minutes(10)/n #várható érték 11,25,
SKU002_Termeles_Max__Time_Step   = (minutes(12)+seconds(30))/n #várhat
SKU002_Termeles_Distribution = 'TDistribution_Uniform_Minimum_Maximum'
SKU002_Termeles_Batch_Size = 10000000*n  #4000 #tetszőleges hosszúságú
```

10. ábra: A forrás (Source) objektumok paraméterezése a Python-alapú szimulátorban

Az egyes objektumoknál értelemszerűen megjelenik az aktuálisan beállított eloszlás is, mint az a 11. ábrán is látható. Az ábrán látható Source objektum felparaméterezése a 10. ábra alsó részén látható, ezekkel a paraméterekkel dolgozik a szimulátor a továbbiakban.

```
of  DES_OBJECT_7 = 'SUP01_present.SKU002_in_S'
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 DES_OBJECT_7  = TDES_OBJECT(
     'TSOURCE',        #Object_Type
     'SUP01_present.SKU002_in_S',        #Object_Name_str
     0,         #Inputs
     6,         #Outputs
     )


 DES_OBJECT_7.DESObject_OutputChannel_Index_Selection_Strategy.Set_Item_str_to('TS
 DES_OBJECT_7.DESObject_OutputChannel_Index = 0

 DES_OBJECT_7.SOURCE_Minimum_Of_Time = SKU002_Termeles_Min__Time_Step
 DES_OBJECT_7.SOURCE_Usual_Of_Time = 2.000
 DES_OBJECT_7.SOURCE_Maximum_Of_Time = SKU002_Termeles_Max__Time_Step
 DES_OBJECT_7.SOURCE_Deviation_Of_Time = 1.000
 DES_OBJECT_7.SOURCE_Mean_Of_Time = SUP01_present.SKU002_in_S__Time_Step
 DES_OBJECT_7.SOURCE_Constant_Value_Of_Time = SUP01_present.SKU002_in_S__Time_Step

 DES_OBJECT_7.SOURCE_Distribution.Set_Item_str_to(SKU002_Termeles_Distribution)
```

11. ábra: Source objektum, egyenletes eloszlással a Python-alapú szimulátorban

A szimulációs modell jelenlegi pilot verziójában ezen megoldások biztosítják a véletlenszerűséget, ezeket is vizsgáljuk a modell tesztelése során. A kutatásunk következő lépéseinek legfontosabb feladata ezen szimulációs modell részletesebb elemzése, ezáltal működésének alaposabb megismerése lesz, különböző véletlengenerátorok alkalmazása mellett is, valamint a vizsgálatok végrehajtása, a jövőbeli, konszolidáció alapú pilot szimulációs modellre is.

5. ÖSSZEFOGLALÁS

Cikkünkben elsősorban a városi koncentrált igénypont-halmazok szimulációs modellezésével foglalkoztunk, azon belül is a sztochasztikus folyamatok kezelésével. Munkánk első részében ismertettük a koncentrált igénypont-halmazok fogalmát, melyek vizsgálatával kapcsolatosan az IFFK konferencián már a 2019-es évben publikáltuk eredményeinket. Ezt követően áttekintettük a sztochasztikus folyamatokkal kapcsolatos alapfogalmakat, valamint a city logisztikai szimulációs modellezés szempontjából legfontosabb eloszlásokat is bemutattuk. Ezt követően egy 2015-ös tanulmány példáján szemléltettük a city logisztikai modellek sokféleségét, valamint az általunk elkészült modellek fő jellemzőit is ismertettük itt. Ez a tanulmány a city logisztikai modelleket négy fő szempont szerint csoportosította: a vizsgált érdekelt felek, az alkalmazott meghatározó leíró jellemzők, a modell célfüggvénye és az alkalmazott megoldási megközelítés. Ezt követően pedig cikkünk utolsó részében ismertettük a kutatásunk során eddigiekben kidolgozott szimulációs modelleket és az azokban alkalmazott megoldásokat a véletlenszerűség kezelésére, egyenletes és normális eloszlással, MS Excelben, AnyLogic-van és egy Python-alapú DES-szimulátorban, mezoszkopikus szintű, illetve makroszkopikus szintű bontásban. Mivel a szimuláció során kapott eredményeink az eddigiekben rendkívül kedvezőek voltak (a jövőbeli rendszerekben konszolidáció alkalmazásával jelentősen, akár felére is csökkenthető a teljesítmények és kibocsátások, illetve negyedével csökkenthető az összes logisztikai üzemeltetési költség is), ezért a továbbiakban is fogunk a koncentrált igénypont-halmazok city logisztikai rendszereinek modellezésével foglalkozni, különböző eszközökben, különböző eloszlások alkalmazása mellett keresve azt a megoldást, amely a lehető legrészletesebb adatokat szolgáltatja, a lehető leggyorsabban futtatva a modelljeinket, megvalósítva a bemeneti jellemzők véletlenszerűségét.

HIVATKOZÁSOK

Anand, N, van Duin, R., Quak, H., Tavasszy L. (2015). Relevance of City Logistics Modelling Efforts: A Review. *Transport Reviews*, **35/6**, 701-719.

AnyLogic (2016). Operations and Supply Chain Management with AnyLogic 7.2. URL: https://www.anylogic.com/upload/pdf/operations-and-supply-chain-simulation-with-anylogic72.pdf

Asmussen, S., Glynn, P. W. (2007). Stochastic Simulation: Algorithms and Analysis. *Springer Science & Business Media.*

BME-ALRT (2020). City Logisztikai Kutatócsoport honlapra. URL: https://www.logisztika.bme.hu/citylog/

Bóna, K. (2015). Modellezési alapok. *BME-ALRT Folyamatmodellezés és szimuláció.*

Bóna, K., Lipovszki, Gy., Sárdi, D. L. (2020). A városi koncentrált igénypont-halmazok city logisztikai rendszereinek Python-alapú szimulációs modellezése. *Logisztikai Évkönyv 2021.* Megjelenés előtt.

Bóna, K., Sárdi D. L. (2019). A városi koncentrált igénypont-halmazok áruellátási rendszerének új koncepciói a különböző közlekedési alágazatok lehetőségeinek kihasználásával. *XIII. IFFK Konferencia (Innováció és fenntartható felszíni közlekedés) 2019, Budapest.*

Bóna, K., Sárdi D. L. (2020). Mesoscopic simulation model of the logistics system of concentrated sets of urban delivery locations. *International Journal of Simulation and Process Modelling.* Megjelenés előtt.

Bóna, K., Sárdi, D. L. (2020). Áruszállító kerékpárok alkalmazási lehetőségeinek értékelése a bevásárlóközpontok logisztikai rendszerében a hálózat geometriai struktúrája alapján (2019). *Logisztikai Évkönyv 2020*, 163-173.

Clements, M. P., Hendry, D. F. (2001). Forecasting with difference-stationary and trend-stationary models. *The Econometrics Journal*, **4/1**, 1-19.

Korondi, P., Huba, A., Graff, J., Aradi, P., Czmerk, A., Bojtos, A., Fekete., R., Lakatos, B. (2014). Rendszertechnika. *BME-MOGI.* URL: https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0042_rendszertechnika/adatok.html

Lipovszki, Gy. (2020). Discrete Event Simulation Documentation (Python). Rendszerfejlesztési dokumentáció. Nem publikált.

Michelberger, P., Szeidl, L., Várlaki, P. (2003). Alkalmazott folyamatstatisztika és idősor-analízis. *Typotex Kiadó.*

Python (2020). The Python Tutorial. URL: https://docs.python.org/3/tutorial/index.html

Sárdi, D. L., Bóna, K. (2017). Developing a mesoscopic simulation model for the examination of shopping mall freight traffic in Budapest. *Smart City Symposium Prague 2017, Prága.*

Sárdi, D. L., Bóna, K. (2018). Macroscopic simulation model of a multi-stage, dynamic cargo bike-based logistics system in the supply of shopping malls in Budapest. *Smart City Symposium Prague 2018, Prága.*

Sárdi, D. L., Bóna, K. (2019). Examination of the logistics systems of concentrated sets of urban delivery points by simulation. *The 21th International Conference on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, 2019, Lisszabon*, 1-10.

Sárdi, D. L., Bóna, K. (2019). Simulation modelling in the sizing of city logistics systems – a study for concentrated delivery points. *International Journal of Engineering and Management Sciences*, **4 (1)** 1-11.

Spyder (2020). Spyder: The Scientific Python Development Environment - Documentation. URL: https://docs.spyder-ide.org/

Winston, W. L. (2011). Operációkutatás - Módszerek és alkalmazások. *Typotex Kiadó.*

Závoti, J. (2010). Matematika III. 4., A valószínűségi változó és jellemzői 4.2 A valószínűségi változó fogalma. *Nyugat-magyarországi Egyetem.* URL: https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027_MA3-4/ch01s02.html