



**Pannon Egyetem
Intézményi Tudományos Diákköri
Konferencia
2012**



**Logisztikai folyamatok vizsgálatát és optimalizálását
támogató módszertan**

Készítette: Ruppert Tamás

Témavezető: Dr. Abonyi János

Pannon Egyetem Mérnöki Kar, Folyamatmérnöki Intézeti Tanszék

Tartalom

TARTALMI ÖSSZEFOGLALÓ	1
1. BEVEZETÉS.....	2
2. KIDOLGOZOTT MÓDSZERTAN.....	7
3. AZ OPTIMALIZÁCIÓRA KIDOLGOZOTT KERETRENDSZER FELÉPÍTÉSE	9
3.1. A FOLYAMATMODELL ÉPÍTÉSÉNEK MÓDSZERTANA	11
3.2 A SIMUL8 FUNKCIONALITÁSÁT KITERJESZTŐ FEJLESZTÉSEK	13
3.3. MONTE-CARLO SZIMULÁCIÓ	17
3.4. OPTIMALIZÁCIÓT LEHETŐVÉ TEVŐ FEJLESZTÉSEK	20
4. SZIMULÁCIÓS KERETRENDSZER ALKALMAZÁSA	23
4.1 A BEVÉTELEZÉSI FOLYAMAT ISMERTETÉSE	23
4.2 A MODELL KIALAKÍTÁSA	24
4.3 INTERFÉSZ BEMENETI PARAMÉTEREI	25
4.4 FOLYAMATIDŐK.....	28
4.5 ERŐFORRÁSOK	31
4.6 SZIMULÁCIÓS VIZSGÁLATOK.....	32
4.7. ÉRZÉKENYSÉGVIZSGÁLAT MONTE-CARLO SZIMULÁCIÓVAL	34
4.8 OPTIMALIZÁLÁS	37
5. ÖSSZEGZÉS.....	41
KÖSZÖNETNYILVÁNÍTÁS	42
IRODALOMJEGYZÉK	43
MELLÉKLETEK	45
M1 SIMUL8 MODELLEK PARAMÉTEREZÉSE	45
M2 VISUAL LOGIC, ÉS MS EXCEL VBA KÓDOK.....	52

Tartalmi összefoglaló

A logisztikai tevékenységek hatékonysága az üzleti, logisztikai és technológiai folyamatok optimalizálásával javítható. A fejlesztési tevékenység hatékonysága jelentősen függ az alkalmazott módszertanok és eszközök megfelelőségétől, korszerűségétől. A modell alapú elemzési és optimalizálási technikák alkalmazási lehetőségeit aktuális, gazdasági szempontból jelentős kutatási területnek számítanak. A vizsgált szimulációs eszközök alkalmazásában rejlő lehetőségeket igazoló probléma egy raktár betárolási munkafolyamatainak fejlesztése. E probléma megoldásához a logisztikai rendszerek kapacitásainak, teljesítményének és költségének jellemzésére alkalmas szimulációs keretrendszert dolgoztam ki.

A módszertan által alkalmazott folyamatmodell paraméterei a kapcsolódó folyamatok felmérésén alapulnak. A felmérések során meghatározásra kerülnek az egyes logisztikai folyamatok jellemzői és a vállalatirányítási rendszerből kinyerhető, az adott logisztikai folyamat szempontjából lényeges adatok.

A folyamatmodellt SIMUL8 szimulációs programcsomagban implementáltam. A folyamatjellemzőket az általam kidolgozott MS Excel interfészen keresztül megadva számítógépes szimuláció segítségével válaszolhatunk a vizsgált logisztikai folyamat átszervezésére vonatkozó „Mi lenne ha...” típusú kérdésekre.

A logisztikai folyamatok sztochasztikus jellegűek. A tanulmányban bemutatom, hogy e tekintetben az elemzést Monte-Carlo szimulációval célszerű végrehajtani. A Monte Carlo szimuláción alapuló vizsgálatokhoz adatelemzési és optimalizációs technikákat illesztettem. Az eredményezett eszköztár alkalmas a vizsgált folyamatok fejlesztésére. Az átszervezésre vonatkozó javaslatok többsége paraméteroptimalizációs feladatként értelmezhető. A paraméterek optimumának megtalálására gradiens süllyedésezés módszert implementáltam.

A kidolgozott módszertan alkalmazhatóságát, az EON Gazdasági Szolgáltató Kft. raktározási logisztikai folyamatainak elemzése kapcsán illusztráltam.

1. Bevezetés

A logisztikai folyamatok fejlesztésének célja, hogy – a versenyképesség növelése érdekében – a fogyasztók által igényelt termékek és szolgáltatások előállításán és disztribúcióján javítsunk. Ez a gyakorlatban azt jelenti, hogy a fejlesztésnek javulást kell előidéznie azon mérőszámokban, melyek alapján a termékek és szolgáltatások értékelhetők. A logisztikai szolgáltatások terén általában két versenyprioritás rendelkezik alapvető fontossággal, a költséghatékonyság és a megrendelések teljesítésének pontossága és átfutási ideje. A folyamatok átszervezésével bizonyos esetekben elérhető, hogy e mérőszámok értéke párhuzamosan javuljon, az esetek többségében azonban nincs ekkora fejlesztési tartalék a meglévő műveletekben, így a két cél csak egymás rovására javítható: az átfutási idő csökkentéséhez újabb erőforrások (alkalmazottak, berendezések, új technológia, nagyobb készletszint) bevonása szükséges, a költségcsökkentés pedig általában a kapacitás csökkenésével, így az átfutási idő növelésével jár. Látszik, hogy összetett, soktényezős, ellentmondó célokat tartalmazó optimalizációs feladatként fogalmazhatók meg a logisztikai rendszerek fejlesztéséhez kötődő feladatok. Ezért nagyon kritikus, hogy megfelelő módszertannal, eszközzel támogassuk ezt a fejlesztést. Javasolt megoldás: folyamat modell alapú fejlesztés szükséges, mely megkívánja, a logisztikai folyamatok és tevékenységek modelljének alkotását, a generált modellek szimulációs eszközök által támogatott formába való transzformációját, a folyamatról rendelkezésre álló információk gyűjtését, rendszerzését, megfelelő kontextusba helyezését és természetesen a szimulációs eredmények feldolgozását, interpretálását.

A cél, ehhez egy felhasználóbarát eszköz és egy megfelelő módszertan kidolgozása volt. Természetesen e módszertannak kapcsolódnia kell az eddig kidolgozott és egyre szélesebb körben alkalmazott folyamatmodell és mutatószámrendszer alapú fejlesztési technikákhoz.

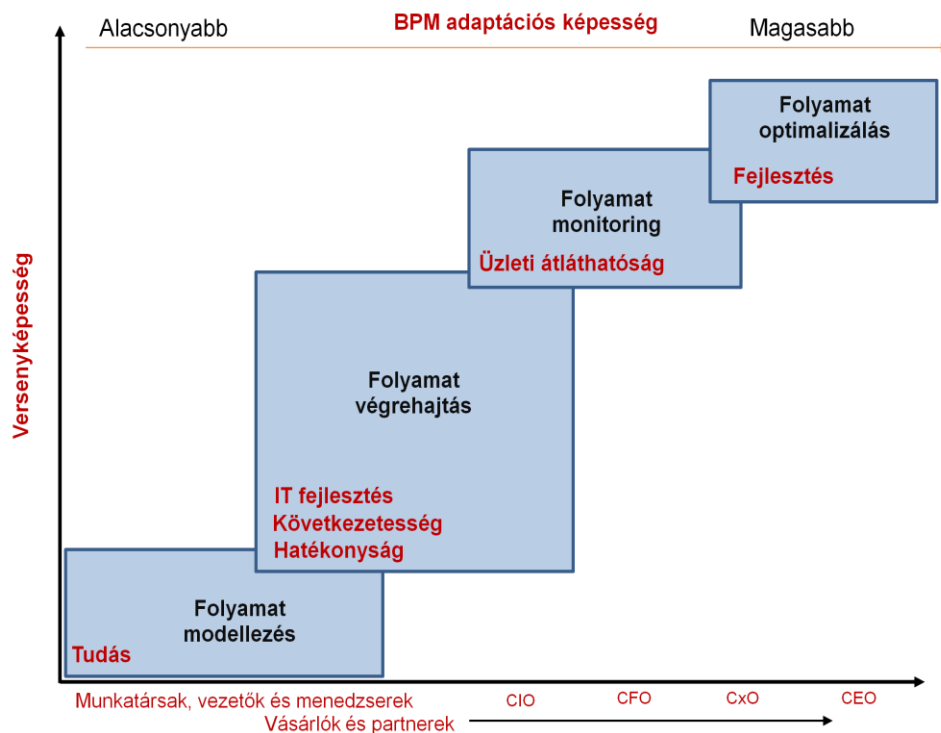
Ahhoz, hogy egy vállalat megőrizze versenyelőnyét folyamatosan fejlesztenie kell folyamatait, fel kell tárnia az esetleges változtatások lehetőségeit, a fejlesztések hatását, a kapcsolatot a folyamatokat jellemző mutatószámok között. A fejlesztési ötletek bevezetése, kipróbálása a való életben költséges művelet. A „Mi lenne, ha...” kérdések megválaszolása nagy és összetett folyamatmodellek esetén költséges és kockázatos, ezért az igazán érett, korszerű, versenyképes szervezetek erre a célra folyamat-szimulációs eszközöket alkalmaznak. A szimulációs vizsgálattal potenciális fejlesztési javaslatokat minősíthetünk, mielőtt bármilyen költséges változtatást hajtanánk végre a valós üzleti környezeten.

A folyamatok teljesítményének mérését, elemzését biztosító folyamatcontrolling rendszer fejlesztésének elengedhetetlen eszköze a folyamatok átláthatóságának biztosítása, melynek alapja a megfelelő folyamatmodell. Ahogy az 1. ábra is mutatja, az üzleti folyamatok formális leírása, és a folyamatmodellek alkalmazása lehetővé teszi a

- könnyebb kommunikációt
(pl. üzleti területen dolgozó és alkalmazásfejlesztő között),
- szoftvermegoldás fejlesztését,
- alkalmazottak betanítását,
- optimális végrehajtás kialakítását.

A folyamatok feltérképezéséhez szükséges információk egy része kinyerhető a rendelkezésre álló szervezeti dokumentumokból:

- korábbi folyamatleírások;
- minőségbiztosítási rendszer dokumentációja;
- egyéb szabályzatok.



1. ábra: Folyamatmodellek szerepe

Az ábrán jól látható, hogy a vállalat versenyképessége miként növekszik a folyamatok elemzése, optimalizálása következtében

A folyamat-szimuláció készítéséhez szükségünk van a technikai és elméleti tudáson túl fejlesztői környezetre, könnyen értelmezhető „megrendelői” felületre, ahol a folyamatmodell, a folyamat-szimuláció lehetőséget teremt a tevékenység alapú költségelemzésre. E tekintetben Whicker és munkatársai [1], valamint Comeilla és társai [2] mutatnak be kutatási eredményeket.

A folyamatok pontos leírása nem minden esetben triviális feladat. A megrendelővel folyamatosan egyeztetni kell folyamatmodelljeinket, ehhez pedig egy mindenki számára elérhető, könnyen kezelhető és átlátható modell leíróra van szükségünk. Ezek alapján a kutatás egyik célkitűzése egy olyan módszertan kidolgozása volt, amely alkalmas folyamat-szimulációkban alkalmazott folyamatmodellek fejlesztésére, továbbá egy olyan szimulációs keretrendszer kidolgozása, mely a vállalati információs rendszerrel kompatibilis. Az általam kidolgozott módszertant a második fejezetben, a kifejlesztett keretrendszert a harmadik fejezetben mutatom be, ahol a Bizagi Process Modeler rendszert ajánlom a BPM eszközök széles eszköztárából és ismertetem, hogy miként lehet e modellt felhasználni az általam javasolt SIMUL8 szimulációs keretrendszerben.

A folyamat-szimulátorok készítése során a cél, az előzőleg kialakított és a megrendelő által validált modellek szimulációja. A "Mi lenne, ha..." típusú kérdések megválaszolása is feladat, azaz a valós, illetve feltárt folyamat modelljének szimulációs vizsgálatával potenciális fejlesztési javaslatokat tesztelhetünk, mielőtt bármilyen költséges változtatást hajtanánk végre a valós üzleti környezeten. A raktározási folyamatok szimulációjára a SIMUL8 [3] folyamatorientált szimulációs programcsomagot alkalmaztam, mely alkalmas a feltárt és az SAP-ban rögzített adatok alapján a folyamatok részletes elemzésére. A SIMUL8 egy komplett folyamat-szimulációs elemző eszköz menedzserek, mérnökök, folyamattervezők számára, mely az alábbi tevékenységeket támogatja:

- Kockázatmentes ötletgyártás

"Mi lenne, ha..." scenario készítés, mielőtt időt és pénzt fektetnénk be.

- Problémafeltárás

Gyártósor telepítése vagy logisztikai módszer bevezetése előtt már a tervezés fázisában felfedi a rendszer rosszul működő, gyenge pontjait.

- Vizuális modellezés és elemzés

Könnyen és gyorsan, grafikusán tud modellt építeni. A standard beállításokon kívül saját igény szerint rugalmasan lehet paraméterezni. Dinamikusan és vizuálisan mutatja az eredményeket, a változtatások hatásait.

- **Költségelemzés:**

A gyártási, logisztikai folyamatokban történt változtatások hatásainak költség- és jövedelmezőség elemzése.

- **Optimalizálás:**

Automatikus eljárást nyújt az optimum megtalálásához.

Az egyre erősödő versenykörnyezetben, sok bizonytalan vagy véletlenszerű elem fedezhető fel az ellátási láncban, mint vevői igények, átfutási idők, termelésingadozás. A sztochasztikus modellek ezeket a bizonytalan és véletlenszerű elemeket veszik figyelembe. Egyik meghatározó munka, amely az ellátási láncok sztochasztikus természetét veszi figyelembe Midler [4] nevéhez fűződik. Egy olyan dinamikus programozási modellt fejlesztett ki, ami az optimális irányítási technikák, szállítási módok, az árucikk áramok és a vásárlók és beszállítók közötti utak optimális kombinációjának egy több periódusú tervezési horizonton történő meghatározására is alkalmas lehet. Tapiero és Soliman [5] optimális irányítási technikákat használtak a többfajta áruszállítás, és a több régiós termelés és készletezés tervezési problémák megoldására az időbeli bizonytalan kereslet függvényében. Lee és Billington [6] megkísérelte integrálni a marketing, a gyártás és az elosztás folyamatainak anyagáramait egy sztochasztikus program megalkotásával. A modelljük célja az volt, hogy meghatározza az anyagrendelési politikát és a vevői szolgáltatási szintet az egyes termékekre. Hasonlóképpen Lee és Feitzinger [7] valamint Swaminathan és Tayur [8] által bemutatott sztochasztikus modellek határozták meg az utánpótlási stratégiákat. Különösen Swaminathan és Tayur [8] modellje látszott megoldani a „vaníliás-doboz” néven emlegetett problémát, ahol a félkész termékek vaníliás dobozokban lettek elraktározva majd innen kerültek ki végső termékeként miután a vásárló ténylegesen megrendelte őket az ellátási lánc egy későbbi eleme számára. Az általuk kidolgozott modell figyelembe vette a véletlenszerű vásárlói igényeket is. Időközben számos kísérletet tettek a kínálat és kereslet egyensúlyának a számszerűsítésére. E próbálkozások közé tartozik Fisher, Hammond, Obermeyer és Raman [9] által kifejlesztett sztochasztikus program. Ennek célja a már említett egyensúlyhiány miatt az alultermelés és a túltermelés költségeinek minimalizálása volt. Hasonlóképpen, Lee [10] vizsgálta az ostor effektust és meghatározta, hogy a jelenség akkor jelentkezik, ha jelen van a rendelési változatosság által eltorzított vásárlói kereslet és a kereslet és kínálat közt egyensúlyhiány. Metters [11] követte Lee [10] kutatásait, és kifejlesztett egy dinamikus programozási modellt, melynek célja a várható termelési

költségek, készleten tartási költségek, és a túlzott kereslet okozta problémák minimalizálása abban az esetben, ha a termelés a kapacitási korlátoknak megfelelő.

Az előzőekben láttuk, hogy a véletlenszerűség fontos tényező a logisztikai folyamatok vizsgálatában, tehát a folyamat sztochasztikus jelleget figyelembe vevő elemzésre van szükség. Az elemzésre Monte-Carlo szimulációt javaslok. A kutatás legfontosabb eredménye, hogy teljesen egyedülálló módon a széles körben alkalmazott SIMUL8 szimulációs eszköz alkalmazási lehetőségeit kiterjesztettem, ugyanis az általam kidolgozott MS Excel interfész lehetőséget teremt többszörös Monte-Carlo vizsgálatokra. A sok párhuzamos sztochasztikus szimuláció eredményét természetesen ki kell értékelni, mely szintén eszközigenyes. Erre a célra érzékenyég vizsgálatot támogató eszközt fejlesztettünk ki, mely a Monte-Carlo vizsgálatok globális érzékenységi mátrixa.

Ezzel az eszközzel vázlatos képet lehet kapni a döntési változók és a folyamatot jellemző mutatószámok kapcsolatát illetően, mely információ támogathatja a folyamat paramétereinek hangolását. A SIMUL8-en alapuló szimulációs vizsgálat azon képessége, hogy a döntési változók függvényében lehetővé válik a fejlesztési célokat definiáló célfüggvény kiértékelése felveti az automatizált paraméteroptimalizáció lehetőségét. Erre a célra szintén egy a keretrendszerbe illeszthető eszközt dolgoztam ki. A paraméteroptimalizálásra a gradiens süllyedés módszerét választottam és a SIMUL8-el való összekapcsolást MS Excel VBA kódban valósítottam meg.

Eddigi eredményeink publikálásra kerültek [12] [13], illetve több konferencián is prezentáltuk őket. [14] [15] [16]

A dolgozatban vázolt esettanulmány egy összetett kutatás-fejlesztési projekthez kapcsolódott. A dolgozat ezen munka konkrét eredményeit nem tartalmazza, leginkább az általam végzett fejlesztési tevékenység elméleti eredményeit és azok alkalmazhatóságát taglalja. A projektben elvégzett saját munkám a folyamatmodellek kidolgozása, implementálása a folyamat-szimulációs eszközbe, a MS Excel interfész és a SIMUL8 összekapcsolása, és a SIMUL8 paraméterezésének és figyelemmel kísérésének kibővítése, továbbá a SIMUL8 modell fejlesztése arra a szintre, hogy azon Monte-Carlo szimulációt alkalmazhassunk. Végezetül pedig megvalósítottam azt az algoritmust, mellyel gradiens süllyedéssel paraméteroptimalizálást illeszthetünk a SIMUL8 modellre.

Dolgozatom második fejezetében bemutatom a kidolgozott módszertan összefoglaló leírását.

2. Kidolgozott módszertan

A hatékony folyamatelemzés érdekében összetett módszertant dolgoztam ki, melynek segítségével az elemzési eljárás felépíthető a folyamatok felmérésétől az optimalizálási javaslatokig. A cél az volt, hogy a lehető legáltalánosabban alkalmazható módszertant fejlesszek ki. A folyamatelemzési eljárás során 4 lépésre bontottam a fejlesztési lépéseket (2. ábra). A folyamatok felmérése során rálátást nyerhetünk a korábbi leírások esetleges hiányosságaira, eltéréseire. A felméréseket az egyes folyamatlépésekre bontva végeztük, ahol teljes munkanapokat mértünk fel és ezek alapján tényleges képet kaphattunk a munkafolyamatokról. A felmérések adatai alapján a korábbi folyamatleírások kiegészíthetők, javíthatók. Az így elkészített folyamatleírások alapján megkezdhető a folyamatmodellek felépítése. A folyamatmodell fejlesztése során fontos szempont, hogy az elkészített modell felhasználható legyen a folyamat-szimuláció fejlesztése során, továbbá a modellt a fejlesztés során folyamatosan egyeztetni tudjuk a vállalattal, így ennek könnyen áttekinthetőnek és értelmezhetőnek kell lennie.



2. ábra: A módszertan lépései

A fejlesztést 4 lépésre bontottam, így felépítve a módszertant a kezdeti lépésektől a fejlesztési javaslatokat lehetővé tevő módszerig

Az elkészített és véglegesített folyamatmodelleket a jól megválasztott környezeteknek köszönhetően már implementálhatjuk a folyamat-szimulációs szoftvercsomagba. A szimulációhoz interfészt fejlesztettem melynek segítségével integrálhatóvá vált a vállalatirányítási rendszerrel, továbbá a paraméterezés és futtatás a bonyolult szimulációs szoftver felülete helyett egy letisztultabb, felhasználóbarát platformot biztosít. Az interfész segítségével kibővíthető a szimulációs vizsgálatok monitoringja is. Ezen funkciók segítségével és további fejlesztéseknek köszönhetően a folyamatunkon különböző vizsgálatokat végezhetünk az egyszeri vizsgálatokon túl Monte-Carlo szimulációt is vizsgálhatunk, melynek eredményei szintén az interfészen kerülnek rögzítésre. Ezen eredményeket importálva egy elkészített MATLAB programba elkészíthetjük az érzékenységi mátrixot, mely az egyes paraméterek egymástól való függéseit vizsgálja.

Az érzékenységi mátrix, és a kibővített monitoring kiértékelése után javaslatokat tehetünk fejlesztésekre. Az interfész segítségével ezeket a javaslatok költségek nélkül kipróbálhatóak, és az újabb vizsgálatok után összehasonlíthatóak.

Dolgozatom harmadik fejezetében bemutatom a kidolgozott keretrendszer elméleti hátterét, a felhasznált szoftverek működését, beállítási lehetőségeit, továbbá részletezem az SAP vállalatirányítási rendszer integrálását az elkészített szimulációval.

A negyedik fejezetben a kidolgozott módszertant és keretrendszert egy az EON Gazdasági Szolgáltató Kft. raktározási logisztikai folyamatain végzett esettanulmányon keresztül mutatom be. A dolgozatban a raktározás bevételezési folyamatát részletezem.

3. Az optimalizációra kidolgozott keretrendszer felépítése

A keretrendszer kidolgozásának célja, egy olyan logisztikai folyamatokat elemző szimuláció létrehozása, melynek az alábbi szempontoknak kellett megfelelnie:

- Könnyű paraméterezhetőség
- SAP vállalatirányítási rendszerrel integrált
- Részletes kiértékelés – Monte Carlo szimuláció
- Paraméteroptimalizálás

A folyamat-szimulációk elkészítéséhez a SIMUL8 folyamatorientált szimulációs programcsomagot használtam, azonban a SIMUL8 szimulációk kezelése és értelmezése nem triviális feladat, a szimulációk megtekintésére, futtatására csak a szoftver megléte esetén van lehetőség. A szimuláció felépítése, és működésének megértése már kellő hozzáértést igényel. A folyamatmodell felépítése kevésbé átlátható, a megrendelő számára nem a leghasználatóbb folyamatrepresentálási eszköz, így a SIMUL8 a folyamatmodellek folyamatosan egyeztetett felépítéséhez nem volt megfelelő felület, ezért a modell leírások kezdeti fázisában a Bizagi Process Modeler szoftvert használtam, mely kitűnő folyamatmodellező eszköz, ugyanis lehetővé teszi, hogy a folyamatmodelleket olyan formátumban exportáljuk ki, hogy a modellek interaktív megjelenítéséhez a mindenki számára elérhető internetböngészők is alkalmazhatóak legyenek. További előnye a programnak, hogy képes az elkészített modelleket írásban is dokumentálni. A szoftver használatával a modellek tervezése és áttekintése nem igényel költséges szoftveres támogatást, és technikai hozzáértést sem a szervezet részéről.

A tervezett szimulációs keretrendszer felépítése összetett feladat, a fenti szempontokat figyelembe véve a rendszer kialakításának a lépései az alábbiak voltak:

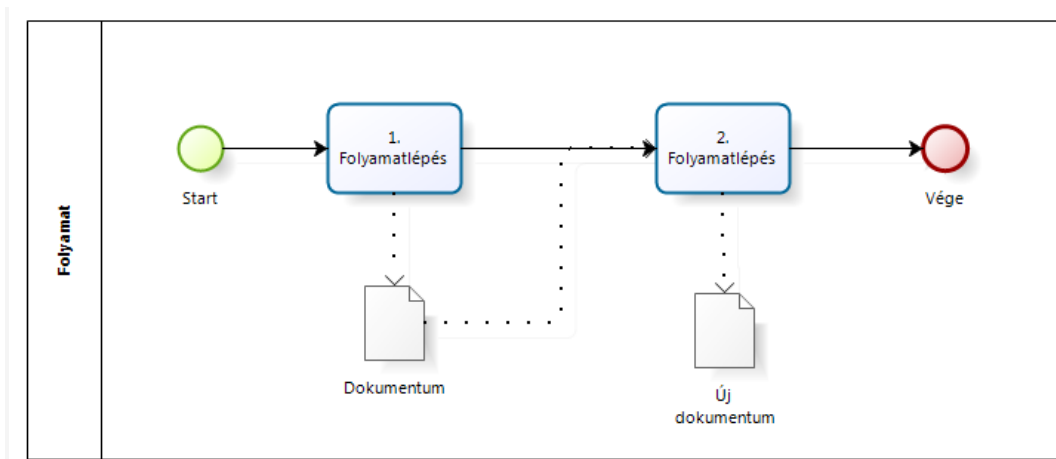
- Bizagi folyamatmodellek elkészítése:
A folyamatmodellek elkészítése céljából a Bizagi Process Modeler-t használtam, mellyel meg tudtam alkotni a közös felületet a szervezettel.
- Bizagi folyamatmodell exportálása Microsoft Visio-ba:
A Bizagi-ban elkészített folyamatmodelleket Visio fájlformátumban is ki tudjuk exportálni, ezzel is előkészítve a szimulációs modellezést.
- Visio fájl betöltése a SIMUL8 programba és a modell strukturális letisztázása:
A Visio formátumban exportált Bizagi fájlokat be tudjuk tölteni a SIMUL8 programba, így a folyamat-szimuláció felépítésének ideje lerövidül.

- Excel interfész megalkotása:
A folyamat-szimuláció paraméterezése érdekében MS Excel munkalapokat fejlesztettem, ezáltal egy olyan környezetet létrehozva, mely mindenki számára ismerős, így elérve a könnyebb használatot.
- A tevékenységek paraméterezése:
Folyamatidők, erőforrások rendelése minden egyes folyamatlépéshez, a felmérések és a megkapott dokumentumok kiértékelése alapján.
- Excel interfész és SIMUL8 összekapcsolása:
A létrehozott MS Excel munkalapok és a kifejlesztett SIMUL8 modellt össze kell kapcsolni, így kiküszöbölve a SIMUL8 bonyolultabb kezelését, paraméterezését.
- Külső adatok betöltése (pl.: SAP táblák adatai):
Az SAP vállalatirányítási rendszer és a folyamat-szimuláció az Excel interfészen keresztül kapcsolódik össze. A megadott SAP táblákból a kialakított interfész segítségével tudjuk kinyerni a szükséges paramétereket.
- Jelentések, mutatók generálása.
Az így megalkotott folyamatmodelleket Monte Carlo szimulációval is kiértékeljük, de további mutatókat is generáltam.
- A paraméterek optimalizálása
A paraméterek hangolására a célfüggvény ismeretében tehetünk javaslatokat. Esetünkben a célfüggvényünket a SIMUL8 értékeli ki. Ezen függvény optimumának megtalálásához a paraméterek finomhangolására van szükség, ehhez gradiens módszert használtam.

3.1. A folyamatmodell építésének módszertana

A Bizagi jelentős nemzetközi referenciákkal rendelkezik különböző méretű BPM projektek és eltérő iparágak viszonylatában is [17]. A BPMG véleménye alapján a Bizagi egyike azoknak a szállítóknak, akik kiemelkedő megoldást kínálnak az üzleti folyamatok menedzseléséhez. A Sodan elemzésére hivatkozva pedig elmondható, hogy ez a leginkább üzleti központú BPM rendszer, amit valaha vizsgáltak [17].

A Bizagi Process Modeler segítségével a szervezet folyamatait hatékony módon tudjuk definiálni és megalkotni a folyamatmodelleket. A Bizagi-val egyértelműen azonosíthatóak a folyamatok, a hozzá tartozó dokumentumokkal és a folyamatokért felelős személyekkel együtt. Így nem csak a folyamatok kapcsolódását tudjuk ábrázolni, de azt is, hogy egyes dokumentumok, mely folyamatlépéseknél vannak jelen és hol módosulnak (3. ábra). Az egyes folyamatokhoz kommenteket írhatunk, melyek kijelölésnél megjelennek, így könnyítve meg az átláthatóságot, továbbá itt jelenik meg a folyamatlépésért felelős beosztottak listája is (4. ábra). A könnyebb átláthatóság kedvéért a folyamatlépéseket tagolhatjuk, csoportosíthatjuk is. A Bizagi Process Modeler támogatja az elkészített folyamatmodellek htm formátumba való exportálását is (5. ábra).



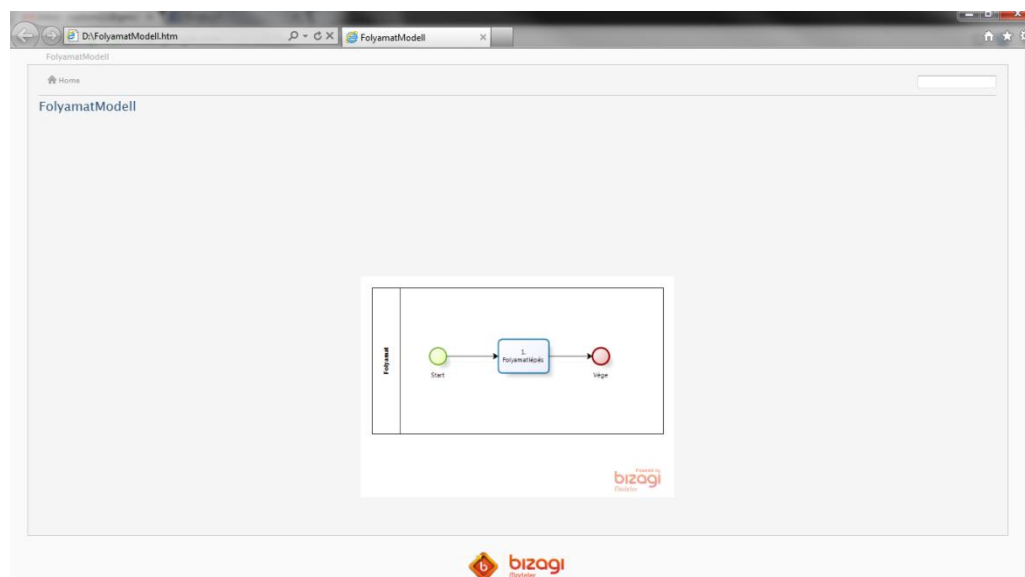
3. ábra: Bizagi folyamatmodell.

Áttekinthető módon ábrázolja az egyes folyamatlépéseket, és a hozzájuk tartozó dokumentumokat, továbbá jól ábrázolható a dokumentumok megváltozásának lépései is.



4. ábra: Folyamatlépések részletei.

Az egyes folyamatlépésekre kattintva megjelenik a leírása és a hozzárendelt erőforrások listája is



5. ábra: Bizagi htm felülete

A Bizagi Process Modeler segítségével lehetőségünk van a mindenki által ismert internetes böngészőkkel is megnyitható formátumba kiexportálni az elkészített folyamatmodellt

A létrehozott folyamatmodellek nem csupán vizualizációra alkalmasak, hanem megalapozzák a szimulációra és optimalizációra alkalmazandó modellek készítését, ugyanis a Bizagi programmal generált modellek VISIO fájlformátumon keresztül betölthetők a SIMUL8 folyamat-szimuláció céljára alkalmazott programcsomagba, jelentősen lerövidítve a SIMUL8 modellek felépítéséhez és validálásához szükséges időt.

3.2 A SIMUL8 funkcionalitását kiterjesztő fejlesztések

A raktári folyamatok paraméterezéséhez és megfelelő vizsgálatához a SIMUL8 lehetőségei nem minden esetben bizonyultak elegendőnek, így egy célirányos fejlesztést, egy paraméterezést és elemzést támogató keretrendszert kellett kidolgoznom. A bővített funkciók az alábbi öt pontra oszthatóak fel:

- SIMUL8 modellek egy külső könnyebben kezelhető interfészen való paraméterezése
- SAP táblákból kinyert adatok importálása a szimulációba
- Folyamatidők függése a bemeneti sorozatoktól
- Erőforrások számának beállítására létrehozott erőforrás mátrix
- Részletesebb monitoring, és Monte-Carlo szimuláció

SIMUL8 modellek külső táblából való paraméterezése

A kutatás egyik fő problémája az volt, hogy a szimuláció paraméterezése egy mindenki által jól ismert interfészen keresztül történjen. Mivel a programcsomag paraméterezése alapesetben csak a SIMUL8 keretein belül lehetséges (M1 melléklet), így ez a szervezet számára nem volt járható út, hiszen a program kezelése megfelelő szakmai tudást igényel, melynek elsajátítása költséges. Megismerve a felhasználói igényeket egy olyan interfészt határoztam meg, amelynek segítségével a felhasználó nem közvetlenül a SIMUL8 programmal találkozik. Ennek érdekében a szimulációs programcsomagot adaptálnom kellett az MS Excel programmal (6. ábra), melynek segítségével a paraméterezés már csak alap informatikai tudást igényel és az elhelyezett nyomógommbal egy kattintással elindítható a szimuláció.

	1	2	3	4
1	Anyagbevételezés			
2	Munkafolyamat	Időigény, perc		
3	SZE		1	
4	SZÁEL		3	
5	SZSAP		5	
6	ÁEL		7	
7	WMR		1	
8	BET		5	
9				
10	Diszpécserek száma:		3	
11				
12				
13				
14				
15				
16				
17				
18				
19				

Buttons: Szimuláció indítása, SIMUL8 bezárása

6. ábra: Könnyű paraméterezhetőség

Az általam kidolgozott Excel munkalapon belül, meghatározhatjuk a szimuláció paramétereit, és az elhelyezett nyomógombbal elindíthatjuk a szimulációt.

SAP táblákból kinyert adatok importálása a szimulációba

A dolgozatomban vizsgált központi raktárak egy SAP vállalatirányítási rendszerhez kapcsolódnak. A vállalatirányítási rendszer raktározási modulja biztosítja a raktározási folyamat során létrejövő események, tranzakciók rögzítését és a készletek nyilvántartását. A raktározási folyamatok szimulációját tekintve az SAP log fájlok fontos információkat tartalmaznak, amelyek kiegészítik a helyszíni felmérések eredményeit és hozzájárulnak a valós folyamatok paramétereinek pontosabb meghatározásához. Abból a célból, hogy az SAP log fájlok adatait integrálhassam a SIMUL8 környezetbe az MS Excel interfészt használtam, amely az SAP táblák szükséges adatait importálni tudja, és amelyen keresztül a SIMUL8 program meghívható és paraméterezhető. A SIMUL8 és az interfész összekapcsolásának ezen részének megoldása, az adatok átmásolása az adott SIMUL8 mátrixba, ennek dinamikus módon kell történnie. Ennek érdekében az Excel munkafüzetben egy számláló cellát hoztam létre, melynek átadásával a szimulációban megírt programkód felismeri, hogy pontosan mennyi adatsort kell átmásolnia, így ezeket az adatokat már kezelni tudjuk.

Folyamatidők függése a bemeneti sorozatoktól

A folyamatlépések idejét alapértelmezésben a SIMUL8 programcsomagban beállíthatjuk egy konstans értékre, illetve az értékre illeszthetünk szórásokat (M1 melléklet). Azonban ahogy azt a 4. fejezetben is bemutatom, az általunk vizsgált raktári folyamatoknál a folyamatlépések erősen függenek a bemeneti adatoktól, így ez a dinamikus paraméterezés komoly fejlesztési feladatnak bizonyult. Ennek megoldására a folyamatlépések futási idejét egy mátrixból állítjuk be, amely mátrix adatait a bemeneti sorozatból generálom és az adott folyamatlépés a mátrix adott elemét fogja megkapni, attól függően, hogy éppen melyik bemeneti egységet fogja feldolgozni. A mátrix elemeit a bemeneti adattábla adott értékének (szállítólevélszám, tételszám vagy csomagolási egység) és a felmérés során az adott folyamatlépésre megállapított egységnyi feldolgozási idő szorzata fogja megadni, ennek a műveleti sornak a beállítását a SIMUL8 Visual Logic kódjaiban (M2 melléklet) fejlesztettem ki. A kidolgozott interfész segítségével a felhasználónak csak a bemeneti adatsort kell legenerálnia az SAP adattáblákból és a felméréskor megállapított folyamatlépések idejét betöltenie, vagy akár új értékeket is megadhat.

Erőforrások számának beállítására létrehozott erőforrás mátrix

Az erőforrások számának beállítására a SIMUL8 konstans értékek megadását teszi lehetővé (M1 melléklet), a folyamatok teljes körű vizsgálatához viszont szükségünk van arra, hogy az erőforrások száma időben változó legyen. Erre a célra egy úgynevezett erőforrás mátrixot (7. ábra) hoztam létre, melynek segítségével a rendelkezésre álló erőforrások számát óránkénti eloszlásban állíthatjuk be.

	1	2	3	4	5	6
1	Erőforrások	Hétfő	Kedd	Szerda	Csütörtök	Péntek
2	7:00	3	4	4	4	4
3	8:00	3	3	3	3	3
4	9:00	2	2	2	2	2
5	10:00	5	5	5	5	5
6	11:00	4	4	3	4	4
7	12:00	2	2	2	2	2
8	13:00	1	1	1	1	1
9	14:00	3	3	3	3	3
10						

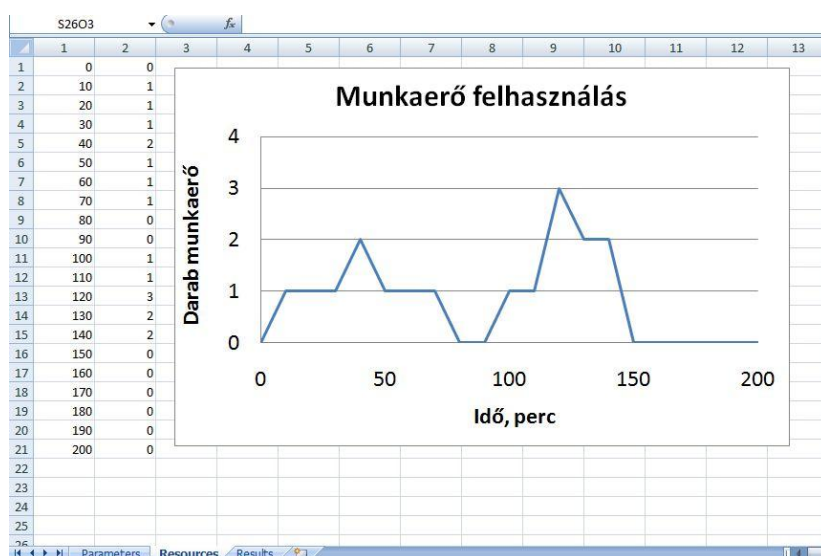
7. ábra: Erőforrás mátrix

Az egyes erőforrástípusok mennyiségének beállítására egy erőforrás mátrixot hoztam létre, melyben óránkénti elosztásban adhatjuk meg az értékeket.

Részletesebb monitoring, és Monte-Carlo szimuláció

A SIMUL8 KPI értékein túl további mutatószámokra is szükségem volt, ezeket a mutatókat a SIMUL8 Visual Logic kódjai (M2 melléklet) segítségével valósítottam meg. A vizsgálat után a folyamat-szimuláció eredményei megjelennek a képernyőn átlátható grafikus ábrázolásban is (8. ábra).

A szimulációs keretrendszernek képesnek kell lennie a Monte-Carlo szimulációra, így meg kellett oldanom, hogy ugyanazzal, de egy adott szórással ellátott bemenettel a szimuláció akár több százszor is kiértékelje a modellt és az összes eredményt értelmezhető, áttekinthető formában megjelenítse.



8. ábra: Átlátható monitoring

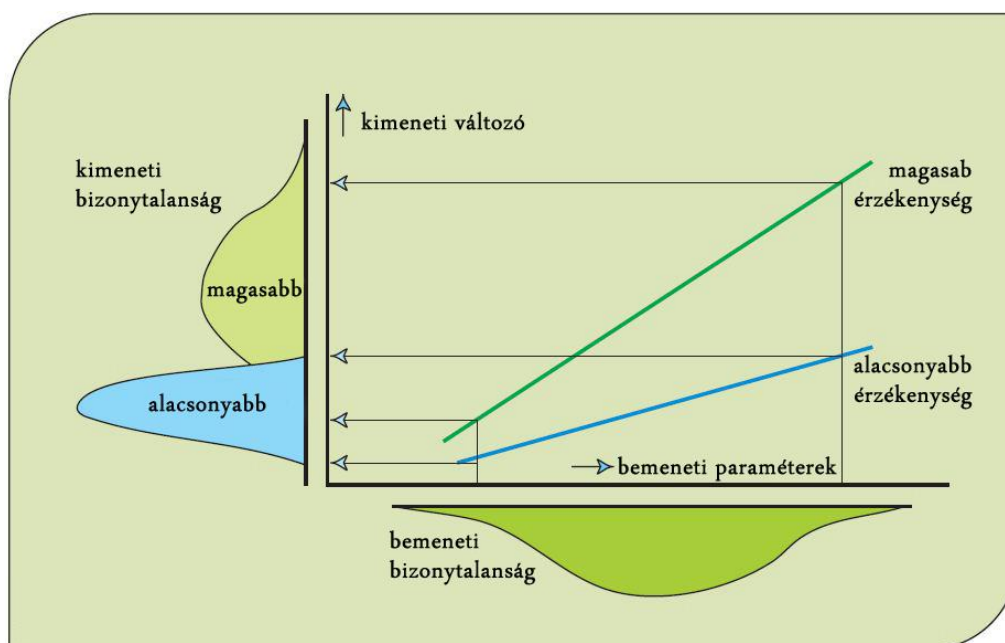
Olyan megoldást dolgoztam ki, amelyben a szimuláció átadja a számítások eredményét, így az Excel diagramok lehetőséget adnak az átlátható monitoringra

3.3. Monte-Carlo szimuláció

Ha a szimuláció során véletlenül választott pontokat, vagy mennyiségeket használunk, akkor Monte-Carlo (vagy véletlen) szimulációról beszélünk. A sztochasztikus modellek a bizonytalan és véletlenszerű elemeket veszik figyelembe, hogy számolni tudjunk ezzel a bizonytalansággal a kiértékelést úgy végezzük, hogy a SIMUL8 modellt sokszor meghívjuk. A többszöri szimulációs futtatás eredményeként látjuk a különböző paraméterek közötti kapcsolatokat (9. ábra).

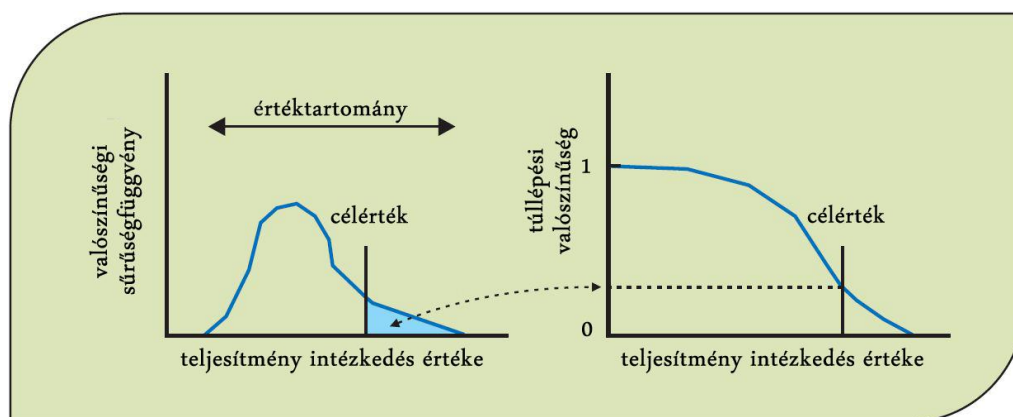
A Monte-Carlo szimulációs módszertan alapvetően a véletlenen alapul. A véletlen segítségével rengeteg bonyolult matematikai, fizikai problémára adhatunk közelítő megoldást, vagy használatával megfelelően kis hibával végezhetünk el összetett rendszerek vizsgálatára alkalmas szimulációs kísérleteket. Az egyik leggyakoribb, egyszersmind legegyszerűbb probléma, melyet könnyen megoldhatunk a véletlen segítségével, a π közelítő számítása. Illusztrációként ezt röviden ismertetem.

Köztudott, hogy egy a oldalú négyzet és a bele rajzolt a átmérőjű kör területének aránya éppen $\pi/4$. Vegyük a következő módszert: Dobjunk az így lerajzolt alakzatokra rizsszemeket (összesen N -szer), és számoljuk meg, hogy ezek közül mennyi esik a körön belül. Jelöljük ezek számát N_k -val. Ekkor az N_k/N arány megfelelően nagyszámú kísérlet esetén megegyezik a kör és a négyzet területének arányával. Így π értékét megadhatjuk a $\pi \approx 4N_k/N$ képlettel. Természetesen minél több kísérletet végzünk, a számítás annál pontosabb eredményt ad.



9. ábra [18]: Monte-Carlo szimuláció elvi felépítése

A teljesítmény mértékének az eloszlása meghatározza a lehetséges értékek skáláját és azt, hogy mennyire valószínű, hogy egy specifikus érték fog kiemelkedni. A 10. ábrán a bal oldalon a kisatírozott terület a sűrűségnek azt a lehetőségét reprezentálja, hogy a cél-érték kiemelkedő lesz.



10. ábra [18]: Cél-érték kiemelkedésének valószínűsége

A Monte-Carlo szimulációban a kialakított keretrendszerrel akár több száz szimulációs vizsgálat alapján számolhatjuk az eredményt. A szimulációkban a számolt és a sztochasztikus értékek alapján határozhatjuk meg a szimulációs paramétereket. Mivel a raktározási folyamatoknál a legbizonytalanabb tényező az ember, így a folyamatlépések idejére illeszthetünk egy meghatározott egyenletes eloszlást.

A Monte-Carlo szimulációt támogató interfész a MonteCarlogen (11. ábra) munkalapon érhető el. Ezen a munkalapon is egy nyomógommbal indítható el a szimuláció, miután beállítottuk, hogy összesen mennyi szimulációt szeretnénk egymás után kiértékelni. A munkalap tartalmazza az egyes folyamatok paramétereinek minimális és maximális értékeit, amelyek között minden kiértékelésnél a paraméterek véletlenszerűen generálásra kerülnek az oldalhoz rendelt VBA makró kód által. A generált folyamat paraméterek a Bevételezés munkalapra kerülnek átmásolásra, majd ezután következik a SIMUL8 meghívása a generált paraméterekkel.

A szimulációk eredményeit a Monte-Carlo munkalapra írja ki (12. ábra) a nyomógombhoz rendelt VBA makró, majd kiértékelődik a következő szimuláció. Minden vizsgálat után a munkalap következő oszlopába kerülnek kiíratásra az adott szimuláció eredményei. Az eredmények egyrészt a SIMUL8 eseményekhez és a folyamatlépésekhez rendelt Visual Logic kódok által számolt értékek, mint pl. az átlagos folyamatidők, másrészt a SIMUL8 által készített Results Summary lista sorai. A munkalap adatainak végső feldolgozásához egy MATLAB környezetben készült programot használtam.

	A	B	C	D	E	F
1	Indítás	Simul8 bezárása				
2						
3	Iterációk száma:	100				
4	Prioritás	min	max			
5	Szállítólevél ellenőrzése	40	40			
6	SAP ellenőrzése	80	80			
7	Áru átvétel	100	100			
8	Bekönyvelés	100	100			
9	Az anyag tárhelyre rakása	60	60			
10	TR létrehozása	100	100			
11	Anyag összekészítés	100	100			
12	Kikönyvelés	50	50			
13	Anyag átadása	50	50			
14	Párhuzamosság	min	max			
15	Szállítólevél ellenőrzése	3	3			
16	SAP ellenőrzése	3	3			
17	Áru átvétel	3	3			
18	Bekönyvelés	2	2			
19	Az anyag tárhelyre rakása	3	3			
20	TR létrehozása	2	2			
21	Anyag összekészítés	8	8			
22	Kikönyvelés	2	2			
23	Anyag átadása	8	8			
24	Folyamatidők, átlag	min	max			
25	Szállítólevél ellenőrzése	0.1	0.4			
26	SAP ellenőrzése/Tétel szám	0	4			

11. ábra: MonteCarlogen munkalap

Monte-Carlo szimuláció futtatására alkalmas felület. Itt az iterációk számának megadása után a beírt mennyiségnek megfelelően futtathatjuk a szimulációt.

	A	B	C
1	Szállítólevél ellenőrzése	2.19541	2.12417
2	SAP ellenőrzése	0.79631	0.93885
3	Áru átvétel	7.68095	8.51359
4	Bekönyvelés	2.52341	2.73947
5	Az anyag tárhelyre rakása	1.89872	1.93908
6	TR és elozmények	0.24015	0.20361
7	Anyag összekészítés	2.35272	2.49951
8	Kikönyvelés	0.16292	0.27262
9	Anyag átadása	8.10071	7.58048
10	Anyag átadása: Change Over %	0	0
11	Anyag átadása: Off Shift %	0	0
12	Anyag átadása: Resource Starved %	0	0
13	Anyag összekészítés: Waiting %	0	0
14	Anyag összekészítés: Working %	0	0
15	Anyag összekészítés: Blocked %	0	0
16	Anyag összekészítés: Stopped %	0	0
17	Anyag összekészítés: Number Completed Jobs	441	426
18	Anyag összekészítés: Minimum use	0	0
19	Anyag összekészítés: Average use	1.431249976	1.450000048

12. ábra: MonteCarlo munkalap

MonteCarlo fülön jelennek meg a sokszoros vizsgálat eredményei egymás melletti oszlopban, ennek a táblának az adatai alapján generáljuk az érzékenységi mátrixot.

3.4. Optimalizációt lehetővé tevő fejlesztések

Ahhoz, hogy egy szervezet mutatóit maximalizálni tudjuk, az egyik lehetséges megoldás lehet, az hogy a paramétereit úgy választjuk meg, hogy azok a beérkező munkaegységeket olyan ütembe tudják feldolgozni, ahogy a kereslet azt megköveteli. Ennek érdekében hangolhatjuk a folyamatunk egyes lépéseinek a lefutási idejét, vagy akár a munkadarabok beérkezésének gyakoriságát. Ezen paramétereket egy adott célfüggvény szerint kell optimalnunk, azonban a célfüggvény, vagyis esetünkben a költségfüggvény felírása sem triviális feladat, hiszen minden egyes folyamatlépés költségét és idejét definiálni kell.

A paraméterek optimalálásához szükséges lépéseket az alábbi 4 pontba soroltam fel:

1. Célfüggvény felírása
2. Modell struktúrája
3. Kereső eljárás megválasztása
4. SIMUL8-el való implementálás

Célfüggvény

Egy folyamat költségfüggvényének felírásánál figyelembe kell vennünk az egységek bekerülési költségét, az egyes folyamatlépések költségét egy egységre számítva, továbbá a tárolási költségeket egység/perc dimenzióban.

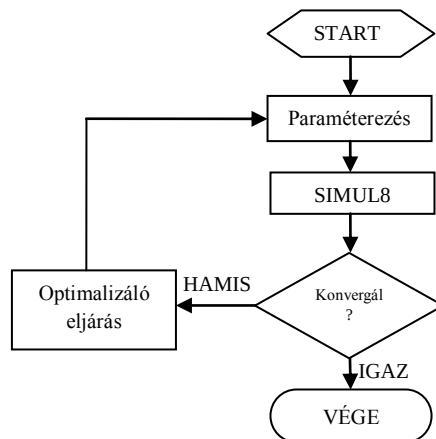
Az egyes folyamatmodellek esetén a költségfüggvény analitikus meghatározása nem triviális feladat, a nehézségét a tárolók költsége jelenti. Azonban a SIMUL8 a megadott paraméterekkel meghatározza a profitunkat, tehát ebben az esetben a költségfüggvényünket maga a szimuláció fogja kiértékelni. Ahhoz, hogy paraméterezni tudjuk a szimulációt meg kell határoznunk a fentebb említett objektumok költségeit. A költségek felírására az 1. táblázat ad példát.

Folyamatlépés	Belépési pont	Lépés 1 tárolója	Lépés 1	Lépés 2 tárolója	Lépés 2	Felhasználó tárolója	Felhasználó	Kilépési pont
								
Folyamatlépés átfutási ideje	10 perc/egység	-	Optimálandó	-	Optimálandó	-	10 perc/egység	-
Folyamatlépés költsége/egység	100Ft/egység	1Ft/egység	2Ft/egység	1 Ft/egység	3 Ft/egység	10 Ft/egység	-	-125 Ft/egység
Folyamatlépés költsége/nap	-	-	1000 Ft/nap	-	1000 Ft/nap	-	-	-

1. táblázat – Példa egy folyamat költségfüggvényére

Optimalizációs struktúra

Az optimalizáció struktúráját a 13. ábra mutatja. Az eljárás elindul egy alapértelmezett paraméterezési beállítással, ezután a SIMUL8 kiértékeli a célfüggvényünket, amennyiben a beállított paraméterek az optimális értéket adták eredményül a program leáll, ha a feltétel nem teljesül, az optimalizációs eljárás beállítja az új paraméterezési értékeket.



13. ábra: Modell struktúrája

Minden egyes szimulációs futtatás után az optimalizációs eljárás megvizsgálja a kiértékelt célfüggvényt, ez után eldönti, hogy megfelel-e a követelményeknek, vagy a paramétereket tovább kell optimalálni

Kereső eljárás

A probléma megoldására a leginkább elterjedt gradiens süllyedés módszert választottam. A gradiens süllyedés módszerének lényege, hogy a kereső eljárás a kezdeti paraméterekkel kiértékelt célfüggvény deriváltja alapján úgy választja meg a következő paraméter sort, hogy az a függvény minimum pontja felé tartson. Az eljárás egészen addig tart, amíg a két utolsó függvény értéke nem lesz kisebb egy adott értéknél. A gradiens módszert az alábbi formulával írhatjuk fel:

$$x^{k+1} = x^k - \alpha \frac{dy(x^k)}{dx} \quad (1)$$

ahol α a tanulási tényező egy nulla és egy közötti konstans, x^{k+1} a bemeneti paraméter következő futtatási értéke, x^k optimum változó vektor és x eleme R^n , ahol n az optimalizálandó változók száma. A $\frac{dy(x^k)}{dx}$ a célfüggvényünk derivált értéke.

SIMUL8 illesztése

Ahhoz, hogy a gradiens módszert alkalmazhassuk a SIMUL8 modellben egy közös beállítási környezetre van szükségünk, ez a fentebb leírt MS Excel interfész. A MS Excel rendelkezik beépített gradiens számítással, de esetünkben nincs analitikus gradiens, mivel a költségfüggvényünket a SIMUL8 értékeli ki, ezért nem volt használható. Ennek következtében egy saját gradiens algoritmust kellett megvalósítanom Visual Basic-ben, mely megvárja, míg a SIMUL8 kiértékeli a modellünket, és azután számolja az új bemeneti paramétereket, majd az új értékekkel ismételtlen lefuttatja a szimulációt, egészen addig, amíg el nem érjük az optimumot.

Mivel a költségfüggvényünket a SIMUL8 értékeli ki, így a deriváltat az alábbi módszerrel közelíthető:

$$x^{k+1} = x^k - \alpha \frac{SIM8(x^k) - SIM8(x^k - dx)}{dx} \quad (2)$$

A $SIM8(x^k)$ jelölés a szimuláció x^k értékkel való meghívása. A $\frac{SIM8(x^k) - SIM8(x^k - dx)}{dx}$ képlet a célfüggvényünk deriváltját közelíti.

Dolgozatom negyedik fejezetében egy esettanulmányon keresztül alkalmazom a fent bemutatott módszertant. A gradiens süllyedés eljárást a 4.8 fejezetben részletezem.

4. Szimulációs keretrendszer alkalmazása

Dolgozatomban a kialakított keretrendszert az EON Gazdasági Szolgáltató Kft. raktározási logisztikai folyamatainak a bevételezési fázisára alkalmazva mutatom be. A szervezet lehetőséget biztosított számunkra, hogy raktáraiban helyszíni felméréseket végezzünk, továbbá rendelkezésünkre bocsátották a szükséges SAP adattáblákat. Helyszíni felméréseket három raktárban, összesen három-három munkanapon végeztünk.

A bevételezési folyamat két nagy egységre osztható. Az első az áru beérkezése a raktárba. Ekkor a szállítóleveleket, a megrendelést, az áruk minőségét és mennyiségét is ellenőrizni kell. Ha megtörtént az átvételezés az áru egy átmeneti raktárhelyre kerül. A második fázis az áruk polcra rakása, ekkor minden áru kap egy raktári sorszámot, ez alapján tudják a raktári dolgozók a megfelelő helyre elhelyezni azokat. Az esettanulmányban bemutatott vizsgálatok és kiértékelések demonstratív jellegűek, illusztrációk, mivel a szervezet felé bizonyos mutatószámok titoktartás kötelezettek.

A szervezet számára a teljes raktári folyamatot modelleztük és elemeztük, a dolgozatomban a bevételezési fázist részletezem.

4.1 A bevételezési folyamat ismertetése

A bevételezési folyamat összesen öt munkafolyamatot tartalmaz (14. ábra). A modell kettőtípusú erőforrást használ, a beérkezési adatokat SAP táblákból nyertük ki, a folyamatidőket és a további paramétereket a helyszíni felmérések alapján állítottuk össze.

A folyamatok az alábbiak:

1. Szállítólevél ellenőrzése
2. SAP ellenőrzése
3. Áru átvétel
4. Bekönyvelés
5. Az anyag tárhelyre rakása

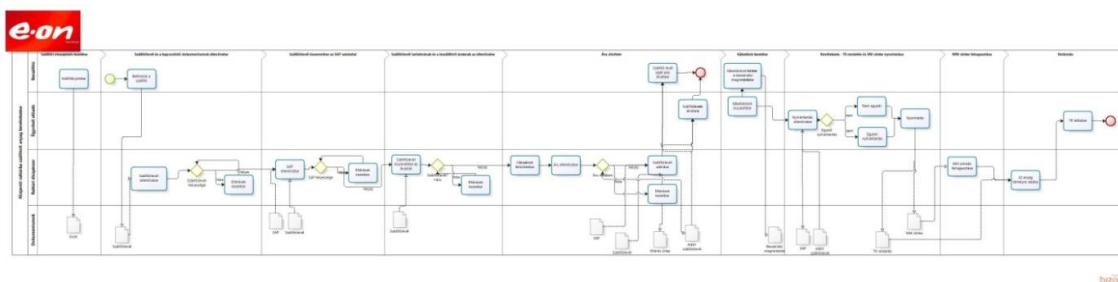


14. ábra: Bevételezési modell

Az ábrán látható az öt folyamatlépés kapcsolata, a konténerek elhelyezkedése, illetve a két emberi erőforrás

4.2 A modell kialakítása

A bevételezési folyamat modelljét először Bizagi Process Modeler szoftverrel készítettem el. A Bizagi kitűnő eszköz volt erre a feladatra, hiszen segítségével az elkészített modellt a mindenki által használt internetes böngészőkkel is meg tudjuk nyitni. Ennek köszönhetően a folyamatmodelleket a vezetőséggel egyszerűen le tudtam tisztázni, így megalkotva a végleges folyamatmodellt (15. ábra). A modellt úgy alkottam meg, hogy egyértelműen azonosíthatóak legyenek a folyamatban résztvevők feladatai, hatáskörei, lehessen látni az időrendiséget, és a párhuzamos feladatokat. A dokumentumok kezelését is ábrázoltam, így látható, mely lépéseknél milyen dokumentumok jönnek létre, vagy kerülnek átadásra. Ezáltal egy könnyen kezelhető és értelmezhető vizuális folyamatleírást készítettem el. A programmal szöveges dokumentumokat is generálhatunk, melyek részletezik a modell felépítését. Ezután a Bizagi-ból kiexportáltam a már véglegesített folyamatmodelleket visio formátumba, melyet már a SIMUL8 is kezelni tud, ezzel is előkészítve a folyamat-szimulációs modellt.



15. ábra: Bizagi modell

A raktári folyamatok bevételezési fázisának kész folyamatmodellje, melyet többek között folyamatleírásként is kiexportálhatunk

4.3 Interfész bemeneti paraméterei

A könnyű és átlátható paraméterezhetőség érdekében MS Excel interfészt fejlesztettem ki, ennek segítségével a szimuláció egyszerűen paraméterezhető. A paraméterezést a SIMUL8 Visual Logic (M2 melléklet) és az MS Excel VBA makró programkódok segítségével valósítottam meg. Az eseményvezérelt kódok leírását az M2 melléklet tartalmazza részletesen.

Az Excel makro gomb elhelyezésével hoztam létre azt az algoritmust, ami elindítja a szimulációt meghatározott futási idővel és adott iterációs számmal. A futási idő a bemenetként SAP-ből importált napok számából alakul ki, a napok számát szoroztam 480-al, mert egy munkanapot 8 órában határoztam meg, így adódik a szimulációs idő percben megadva

Az Excel interfészben control elemekhez, nyomógombokhoz rendeltem a VBA makró kódokat, amelyek feladata a szimulációs paraméterek generálása és a SIMUL8 szimuláció elindítása.

Konfigurálás munkalap

Erről az oldalról a „Szimuláció indítása” nyomógomb használatával indítható el egy egyszeri szimulációs vizsgálat.

	A	B	C	D	E	F	G
1	Prioritás					Párhuzamosság	
2	Szállítólevél ellenőrzése	100	TR létrehozása	100		Szállítólevél ellenőrzése	1
3	SAP ellenőrzése	80	Anyag összekészítés	80		SAP ellenőrzése	1
4	Áru átvétel	100	Kikönyvelés	50		Áru átvétel	1
5	Bekönyvelés	50	Anyag átadása	100		Bekönyvelés	1
6	Az anyag tárhelyre rakása	30				Az anyag tárhelyre rakása	1
7							
8	Simul8 szimulációs fájl	EON_raktarozas_v4.3_v2					
9							
10	Default paraméterek betöltése						
11							
12							
13	Szimuláció indítása		Mennyi időt akarunk szimulálni? (napokban megadva)	5			
14							
15	Simul8 bezárása						
16							

16. ábra: Konfigurálás

Erről a munkalapról indíthatjuk az egyszeres vizsgálatot, ezen az oldalon állítható be a szimulációs fájl neve, a szimulációs idő, a párhuzamosságok és a prioritásokat

Ezen a munkalapon állítható be a futtatni kívánt SIMUL8 fájl neve (melynek az Excel interfésszel egy mappában kell lennie) és itt adható meg a szimulációs időtartam is (napokban). Továbbá szintén itt paraméterezhetők az egyes folyamatok prioritása és párhuzamossága.

A prioritási értékek beállításai határozzák meg, hogy a folyamatlépések között, milyen fontossági sorrendet alkalmazunk, a felhasználható erőforrások tekintetében. A párhuzamossággal pedig adott folyamatlépés többszörözhető meg, így párhuzamosan több munkaegység is feldolgozható azonos folyamatlépéssel, ha megfelelő számú erőforrás áll rendelkezésre.

Bevételezés munkalap

Itt definiálhatóak a szimuláció bemeneti adatai, a folyamatidők, a folyamatidő szórások, és az erőforrások száma is. A felmérési eredmények betöltése nyomógommbal visszaállíthatóak a default (valós) értékek.

	A	B	C	D	E	F	G	H	I	J
1	ID	Beérkezési idő	Tételek száma	Csomagolási egységek száma	Default paraméterek betöltése				Folyamatidők	Szórás
2	1	8:22	1	29				Szállítólevél ellenőrzése	2	0.2
3	2	8:33	1	1				SAP ellenőrzése/Tétel szám	1	0.1
4	3	9:10	1	1				Áru átvétel/Tétel szám	10	1
5	4	9:54	2	2				Bekönyvelés/Tétel szám	3	0.3
6	5	10:09	4	7	Betárolásért felelős diszpécser számának eloszlása:			Az anyag tárhelyre rakása/Csomagolási egység	3	0.3
7	6	10:13	1	7						
8	7	10:24	3	14				Hétfő	Kedd	Szerda
9	8	10:40	2	7				Csütörtök	Péntek	
10	9	11:34	1	2						
11	10	11:45	1	2	7:00	3	3	3	3	3
12	11	11:52	1	1	8:00	3	3	3	3	3
13	12	13:48	1	1	9:00	3	3	3	3	3
14	13	13:56	1	11	10:00	3	3	3	3	3
15	14	14:13	1	2	11:00	3	3	3	3	3
16	15	7:34	2	2	12:00	3	3	3	3	3
					13:00	3	3	3	3	3
					14:00	3	3	3	3	3

17. ábra: Bevételezés munkalap

Ezen az oldalon tölthetjük be a bevételezési folyamat bemeneteit, meg kell határozunk az erőforrások számát és a folyamatlépések idejét

Monte-Carlo munkalap

A Monte-Carlo szimulációt támogató interfész a Monte-Carlo munkalapon érhető el. Ezen a munkalapon is egy nyomógommbal indítható el a szimuláció, miután meghatároztuk az iterációk számát. A munkalap tartalmazza az egyes folyamatok paramétereinek minimális és maximális értékeit, amelyek között minden vizsgálatnál a paraméterek véletlenszerűen generálásra kerülnek az oldalhoz rendelt VBA makró kód által. A generált folyamat paraméterek a Bevételezés munkalapra kerülnek átmásolásra majd ezután következik a SIMUL8 meghívása a generált paraméterekkel.

4.4 Folyamatidők

A szimulációs vizsgálatok szempontjából lényeges az egyes részfolyamatok folyamatidejének minél pontosabb ismerete. A folyamatidők meghatározására a helyszíni felmérések adatait használtam fel. A gyakorlati tapasztalatok és a felmérés eredménye azt mutatta, hogy a részfolyamatok időszükséglete a beérkező szállítólevél és a vevői rendelés tételeinek számától, a transzport rendelvények egyes tételeiben a csomagok számától és az egy kiszállításhoz tartozó transzport rendelvények számától függ, ill. vannak olyan részfolyamatok, amelyek függetlennek tekinthetők a fentiektől. A 2. táblázat összegzi a folyamatidők függését e tekintetben.

Az egyes lépések fajlagos folyamatidejének meghatározására helyszíni felméréseket végeztünk, majd ezeket az adatokat kiértékeljük. A felméréseket a 18. ábrán látható táblázatokba rögzítettük. A helyszíni felmérések során többnyire olyan tevékenységek idejét mértük, amelyek több tételre, transzport rendelvényre vagy csomagra vonatkoztak egyidejűleg. A fajlagos folyamatidőket a mért értékekből az SAP log táblák felhasználásával határoztuk meg. A helyszíni mérések által kapott átlagos fajlagos folyamatidőket és folyamatidő szórásokat a 3. táblázat tartalmazza.

Folyamat neve	Folyamatidő függése
Szállítólevél ellenőrzése (SZE)	fix ideig
Szállítólevél tételeinek SAP ellenőrzése (SZSAP)	A tételszámtól függ
Áru átvétele (ÁEL)	A tételszámtól függ
Bekönyvelés (BTR)	A tételszámtól függ
Áru tárhelyre rakása (BET)	A csomagok számától függ

2. táblázat - Folyamatidők függősége

2	3	4	5	6	7
A felmérés dátuma:			A felmérést végezte:		
Esemény (lehetőleg kóddal)	Kezdet (óra:perc)	Vég (óra:perc)	Dokumentum, tranzakció (azonosítóval)	Megjegyzés	Személy (kód)

18. ábra: Felmérési táblázat

A folyamatok felmérésére létrehozott táblázat, segítségével a raktárakban a teljes 8 órás munkanapot fel tudtuk mérni.

Folyamat neve	Átlagos fajlagos folyamatidő, perc	Átlagos fajlagos folyamatidő szórása, perc
Szállítólevél ellenőrzése (SZE)	0.5	0.1
Szállítólevél tételeinek SAP ellenőrzése (SZSAP) / tétel	1.5	1.0
Áru átvétele (ÁEL) / tétel	2.0	3.0
Bekönyvelés (BTR) / tétel	1.0	0.7
Áru tárhelyre rakása (BET) / csomag	1.	1.5

3. táblázat - Fajlagos folyamatidők és szórások

Mivel a szimuláció egyes bemeneti eseményei több munkaegységből is állhatnak, pl. egy szállítólevél több tétellel rendelkezhet, a szimulációs futtatásoknál az adott munkaegységre vonatkozó folyamatidőket is ennek megfelelően kell beállítani. A folyamatidőket az Excel táblából generált SIMUL8 tábla példáját felhasználva (19. ábra) az alábbi ismertetett algoritmus alapján állítottuk be.

	A	B	C	D	E	F	G	H	I	J	K
1	Beérkezési idő?		Szállítólevél db száma	Tételek száma	Csomagolási egységek száma	Eltérések				Folyamatid?k	Eloszlás
2	7:15	1	1	4	Szállítólevél ellenőrzése	100			Szállítólevél ellenőrzése	20	10
3	7:30	1	1	16	SAP ellenőrzése	100			SAP ellenőrzés	5	2.5
4	8:15	1	1	18	Szállítólevél összevetése az árukkal	100			Szállítólevél oszt	5	2.5
5	8:25	1	1	3	Áru átvételezése	100			Áru átvételezés	15	7
6	11:05	1	1	8	Mennyi időt akarunk simulálni? (napokban megadva)	5			Az anyag tárhé	20	10
7	12:05	1	1	10							
8	7:19	1	1	6	Betárolásért felel?s diszpécserok számának eloszlása:	Hét?	Kedd	Szerda	Csütörtök	Péntek	
9	10:35	1	1	12	7:00	4	4	4	4	4	4
10	7:05	1	1	20	8:00	4	4	4	4	4	4
11	10:55	1	1	10	9:00	4	4	4	4	4	4
12	14:35	1	1	10	10:00	4	4	4	4	4	4
13	7:36	1	1	6	11:00	4	4	4	4	4	4
14	8:19	1	1	8	12:00	4	4	4	4	4	4
15	10:25	1	1	2	13:00	4	4	4	4	4	4
16	8:25	1	1	7	14:00	4	4	4	4	4	4
17	13:15	1	1	11	Költségek	capitol	usage/min	usage/unit	Napok		
18					Szállítólevél ellenőrzése	2	0	4		1	
19					SAP ellenőrzése	2	0	5		2	
20					Szállítólevél összevetése az árukkal	2	0	6		3	
21					Áru átvételezése	2	0	7		4	
22					Az anyag tárhelyre rakása	2	0	8		5	
23					Iterációk száma:	2					

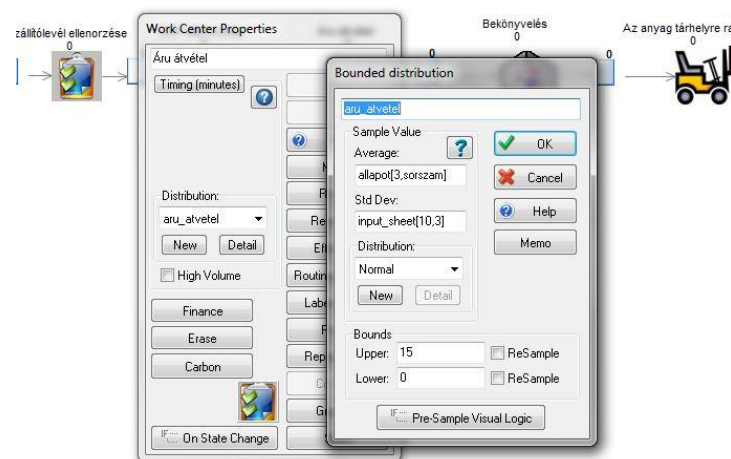
19. ábra: Excel interfészből generált SIMUL8 tábla

Ahhoz, hogy a SIMUL8 a megadott paraméterekkel tudjon dolgozni, az Excel munkalapot át kell konvertálnunk SIMUL8 táblázattá.

Folyamatidők számítás:

$$\text{FOLYAMATIDŐK} [x. \text{oszlop}, i. \text{sor}] = \text{INPUT} [11.\text{oszlop}, x. \text{sor}] * \text{INPUT} [y.\text{oszlop}, i. \text{sor}] \quad (3)$$

Ahol x az adott folyamatlépés, y pedig megadja, hogy az adott folyamatidő melyik mennyiségtől függ (Tételek száma, Szállítólevél db száma). Az i változó lépteti a végig az algoritmust a bemeneti egységeken. A folyamatidőket beállító Visual Logic kód az M2 mellékletben található. Mivel az i értékét léptetni kell, így ezt a kódot az első tárolóban hoztam létre, ahol az $i+1$ utasítást akkor hajtja végre a program, amikor beérkezik egy újabb egység így mindig az aktuális értéket számolja ki. Ezt a léptetést minden folyamatlépés előtt végre kell hajtani, hiszen a folyamatidejét a *Folyamatidők* táblából olvassuk ki és adjuk át a folyamatoknak (20. ábra), így mindegyik lépésnél az aktuálisan odaérkező adat folyamatidejét fogjuk beállítani. Ugyanitt adjuk át az eloszlás értékét is, melynek a beállítását végző algoritmus.



20. ábra: Folyamatidők beállítása

A folyamatidőket a bemeneti paraméterek alapján számolja az algoritmus, ennek eredményét egy *allapot* táblázatba menti, majd a folyamatoknak a táblázat megfelelő paramétereit adjuk át. A szórás ráillesztésénél alsó és felső határokat határoztam meg minden egyes folyamatlépésnél.

4.5 Erőforrások

A raktári folyamatok szimulációja során definiálni kell a folyamatlépések prioritását a felhasználandó munkaerő tekintetében. A folyamatban találhatók olyan folyamatlépések, amelyet az azonos munkakörben dolgozó alkalmazottak hajtanak végre. A SIMUL8 lehetőséget nyújt prioritások folyamatlépésekhez való rendelésére a munkaerő felhasználását tekintve, azaz megadhatjuk, hogy ha nincs elegendő munkaerő az összes folyamatlépés végrehajtásához, akkor melyik folyamatlépés végrehajtását preferáljuk. Az elvégzett szimulációs vizsgálatokban a prioritásokat többnyire a folyamatlépések folyamatideje alapján határoztam meg. Nagyobb prioritást rendeltem a nagyobb folyamatidővel rendelkező folyamatokhoz biztosítva ezáltal, hogy az időigényesebb folyamatok végrehajtása a lehető legkorábban elkezdődhessenek amint a szükséges munkaerő rendelkezésre áll.

A valós folyamatidők mellett a szimulációs vizsgálatok legfőbb paraméterei a raktári alkalmazottak száma. A felépített SIMUL8 modellben kettő típusú munkaerőt definiáltam, amely megfelel a jelenlegi gyakorlatnak és illeszkedik a fő raktári folyamatokhoz. Ezek a betárolásért felelős raktári diszpécser, és a betárolásért felelős ügyviteli előadó. A raktári folyamatok fejlesztése és a munkaerő elosztás különböző stratégiáinak szimuláció vizsgálata megköveteli az alkalmazott munkaerő pontos létszámának ismeretét.

Betárolásért felelős diszpécserek számának eloszlása:	Hétfő	Kedd	Szerda	Csütörtök	Péntek
7:00	3	2	2	1	2
8:00	2	4	4	2	3
9:00	4	3	3	4	1
10:00	5	6	5	3	1
11:00	2	1	1	1	1
12:00	1	2	2	2	3
13:00	3	3	3	3	2
14:00	2	1	3	1	1

21. ábra: Erőforrások megadása

Az erőforrások számának minél részletesebb meghatározása érdekében, az öt munkanapot óránkénti leosztásban alkottam meg, így akár az erőforrások rendelkezésre állása óráról-órára is változhat.

A szimuláció erőforrásainak beállítására, egy az Excel munkalapon keresztül paraméterezhető erőforrás mátrixot alakítottam ki. Az erőforrásokat 8 órás munkanapra leosztva adhatjuk meg, óránkénti elosztásban (21. ábra). Az erőforrás mátrix segítségével könnyedén vizsgálhatjuk, hogyan változnak az átfutási idők az erőforrások függvényében.

Az erőforrások beolvasását az Excel interfészből a Time Check Logic (M2 melléklet) eseményhez készített felhasználó programkód segítségével állítjuk be. A Time Chek Logic meghatározott időnként fut le a szimuláció alatt, esetünkben minden percben és a táblázat adatainak megfelelően a szimuláció minden 60. percében frissül a felhasznált erőforrás.

4.6 Szimulációs vizsgálatok

Az Excel táblákból két féleképpen futtathatjuk a szimulációt. A konfigurálás munkalapról közvetlenül a makró gomb segítségével elindíthatunk egy egyszeri futtatást. A szimulációs futtatásokról a LOG (22. ábra) munkalapon numerikus formában az Aktivitás-diagram (23. ábra) munkalapon pedig grafikus formában kaphatunk eredményeket.

A LOG munkalap első 13 oszlopából kiolvasható a folyamatlépések és erőforrások percenkénti felhasználása. A további szekciókban a munkaegységek egyes folyamatlépésekben eltöltött feldolgozási ideje kerül rögzítésre.

Az aktivitás-diagramokon a LOG munkalap adataiból generált diagramok találhatók, ezekről az ábrákról leolvashatóak a folyamatlépések, illetve az erőforrások aktuális felhasználása (fél napokban).

	Q	R	S	T	U	V
1		Szállítólevél ellenőrzése	SAP ellenőrzése	Áru átvétel	Bekönyvelés	Az anyag tárhelyre rakása
2	1	2.02549	0.87812	10.28328	2.91	15.00
3	2	2.16445	0.93153	9.1256	2.46	2.65
4	3	2.18775	1.00379	8.79433	2.81	2.79
5	4	2.26937	1.85614	15	5.67	5.36
6	5	2.19683	3.72192	15	11.62	15.00
7	6	2.18658	0.89706	10.85529	2.30	15.00
8	7	1.93333	2.76447	15	8.38	15.00
9	8	1.71013	1.83683	15	5.69	15.00
10	9	2.15287	0.82813	10.97123	3.22	5.43
11	10	1.63207	0.91051	9.08564	2.93	5.28
12	11	1.93727	1.04131	10.00596	3.07	2.61
13	12	2.24237	0.8775	8.7923	3.03	2.28
14	13	2.34244	0.78409	10.35058	2.87	15.00
15	14	1.82704	1.0023	10.59291	2.54	5.17
16	15	2.0802	1.97421	15	5.24	5.62
17	16	2.30029	0.89279	11.13975	2.63	10.34
18	17	1.99958	2.78343	15	8.21	15.00

22. ábra: Log munkalap

A LOG munkalapról leolvashatók a folyamatlépések ideje egységekre lebontva, továbbá a folyamatlépések és az erőforrások percenként kihasználtsága



23. ábra: Aktivitás-diagram munkalap

Aktivitás-diagram munkalapon a LOG adatai találhatóak grafikus ábrázolásban

A 24. ábrán látható felső diagram a folyamatlépések párhuzamos feldolgozásának a monitoringja. Láthatjuk, hogy adott bemenet mellett a teljes átfutáshoz egy folyamatlépésnek összesen mennyi munkaegységet kell párhuzamosan feldolgoznia

4.7. Érzékenységvizsgálat Monte-Carlo szimulációval

Az elkészített szimulációs keretrendszert a 24. ábrán látható példabemenettel Monte-Carlo szimulációs vizsgálatot végeztem. A szimuláció értékelésénél a bemeneti paraméterek száma 7, ebben a példában az öt folyamatlépésnek az idejére illesztettünk szórást, a maradék két bemenet a két erőforrások kihasználtsága.

Beérkezési idő	Tételek száma	Csomagolási egységek száma	Default paraméterek betöltése	Folyamatidők			Szórás	
8:22	1	29		Szállítólevél ellenőrzése	2	0,2		
8:33	1	1		SAP ellenőrzése/Tétel szám	1	0,1		
9:10	1	1		Áru átvétel/Tétel szám	10	1		
9:54	2	2		Bekönyvelés/Tétel szám	3	0,3		
10:09	4	7	Betárolásért felelős diszpécserok számának eloszlása:	Az anyag tárhelyre rakása/Csomagolási egység	3	0,3		
10:13	1	7						
10:24	3	14						
10:40	2	7						
11:34	1	2						
11:45	1	2						
11:52	1	1						
13:48	1	1						
13:56	1	11						
14:13	1	2						
7:34	2	2						
				Hétfő	Kedd	Szerda	Csütörtök	Péntek
				7:00	3	3	3	3
				8:00	3	3	3	3
				9:00	3	3	3	3
				10:00	3	3	3	3
				11:00	3	3	3	3
				12:00	3	3	3	3
				13:00	3	3	3	3
				14:00	3	3	3	3

24. ábra: A szimuláció példabemenetei

A táblázatban megadott paraméterekkel futtattam a szimulációt összesen százszor. Ezt az iterációk számánál adhatjuk meg.

A Monte-Carlo szimulációban 100 szimulációs futtatás alapján számoltam az eredményeket. A szimulációkban az Excel makrók által számított értékek és a sztochasztikus értékek alapján határoztam meg a szimulációs modell paramétereit.

A folyamatlépések átlagos idejét és szórását a 4. táblázat minimális és maximális értékeit felhasználva egyenletes eloszlás értéként generáltuk. A folyamatlépések prioritása és párhuzamossága minden szimulációs futtatásnál konstans volt (5. táblázat), az erőforrások száma pedig minden futtatásnál egy maximum és egy minimum érték között változott (6. táblázat).

Folyamatlépés	Átlag		Normál eloszlás	
	min	max	min	max
Szállítólevél ellenőrzése	0.1	0.4	0.0375	0.0625
SAP ellenőrzése	0	4.0	0.6375	1.0625
Áru átvétel	0	11.4	2.25	3.75
Bekönyvelés	0	2.64	0.42	0.7
Anyagok tárhelyre rakása	0	5.75	1.1475	1.9125

4. táblázat - Monte-Carlo szimuláció folyamatidő és
folyamatidő szórás paraméterei

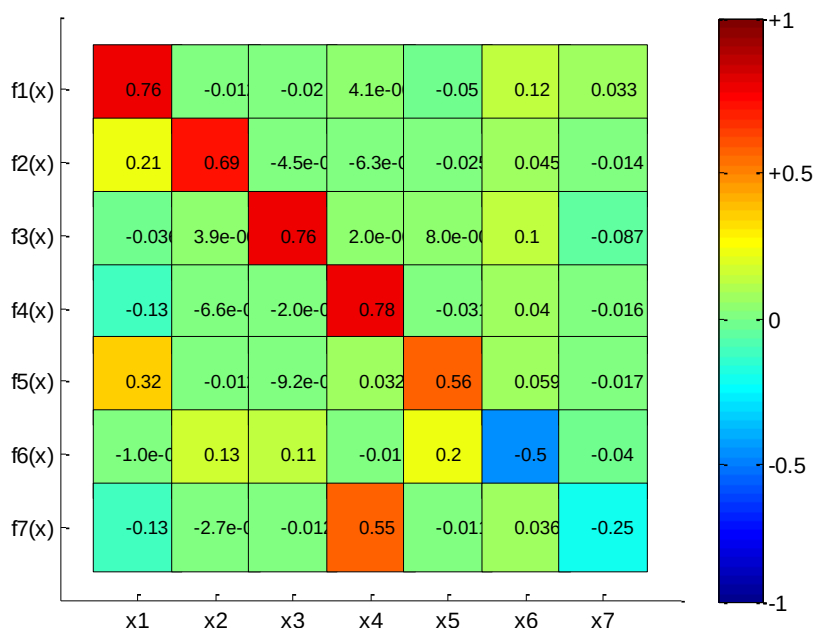
Folyamatlépés	Prioritás	Párhuzamosság
	40	3
Szállítólevél ellenőrzése	80	5
SAP ellenőrzése	100	5
Áru átvétel	100	3
Bekönyvelés	60	4
Anyagok tárhelyre rakása		

5. táblázat - Folyamatlépések prioritása és párhuzamossága

Resource	minimum	maximum
Raktári munkás	2	5
Ügyviteli előadó	2	4

6. táblázat - Az erőforrások minimum és maximum száma

A futtatások eredményeit az Excel interfész MonteCarlo munkalapján kerülnek összegyűjtésre, majd az eredményeket egy MATLAB környezetben készült programmal elemezhetjük. A 25. ábra mutat példát az eredményekből generált érzékenységi mátrixra.



25. ábra: Érzékenység mátrix

A $df(x)/dx$ gradiensek normált-értékeit mutatja. Az $x1 \dots x5$ bemenetek az 5 folyamatlépés átlagos folyamatideje. A számok fejezik ki, hogy a folyamatidők változása mekkora szerepet vállal az egyes folyamatok átlagos kihasználtságában.

A Monte-Carlo szimulációs eredményekre regressziós görbét illesztettünk és ez alapján kiszámítottuk a választott bemeneti és a kimeneti változók esetén a $df_i(x)/dx_i$ normalizált gradienseket, ahol $f_i(x)$ jelöli az i -dik kimeneti változót és x_i az i -dik bemeneti változót. Az $x_1, x_2, \dots, x_7$ bemeneti változók sorrendben a folyamatlépések, mint a szállítólevél ellenőrzése, a szállítólevél SAP ellenőrzése, az áru átvétele, az áru bekönyvelése és az anyagok tárhelyre rakása, generált folyamatideje valamint a betárolásért felelős raktári diszpécserek és ügyviteli előadók maximális alkalmazási száma a szimuláció ideje alatt. A vizsgált $f_1(x), f_2(x), \dots, f_7(x)$ kimeneti változók sorrendben az előbbiekben felsorolt öt betárolási folyamatlépés átlagos hasznátsága és a betárolásért felelős raktári diszpécserek kihasználtsági százaléka. Egy folyamatlépés átlagos hasznátsága szorosan kapcsolódik az adott folyamat párhuzamosságához és azt mutatja meg, hogy átlagosan hány ilyen párhuzamos folyamatlépésben folyt a munkaegységek feldolgozása a szimuláció teljes idejét tekintve. A munkaerő kihasználtsági szintje azt mutatja, hogy a szimuláció idejének hány százalékában végez tevékenységet.

Az érzékenységi mátrix átlójában mutatkozó magasabb abszolút értékek azt mutatják, hogy a folyamatidő természetesen hatással van a folyamatlépés átlagos használatára és a munkaerő maximális alkalmazási száma pedig jelentősen befolyásolja a kihasználtsági szintjét. Minél nagyobb az adott munkaerő típus maximális alkalmazási száma a szimuláció ideje alatt annál magasabb a kihasználtsági szintje. A 25. ábrából továbbá látható, hogy a bekönyvelés folyamatidejének növelése erős, pozitív hatása van a betárolásért felelős ügyviteli előadók kihasználtsági szintjére, mivel ugyanakkora időintervallumban a megnövekedett folyamatidőnek köszönhetően az összes tevékenységi idejük növekszik.

Az érzékenységi mátrix alkalmazása kiindulási pontot biztosíthat olyan további vizsgálatoknak, amelynek célja, hogy megtaláljuk a költséghatékony folyamat és erőforrás elosztást valamint, hogy jelentősen csökkentsük a raktározási folyamat teljes idejét.

4.8 Optimalizálás

Az optimalizáló eljárást egy leegyszerűsített modellre alkalmaztam. A modell (26. ábra) három folyamatlépésből áll, és mind a három folyamatlépéshez tartozik egy-egy tároló.



26. ábra: A bevételezési folyamat egyszerűsített modellje

Három folyamatlépésben határoztam meg a folyamatot, az optimalizációs eljárást majd az Áru átvétel és a Bekönyvelés lépésekre alkalmazom

Folyamatmodell költségfüggvénye

A célfüggvény felírásánál figyelembe vettem az átlagosan egy hét (5 munkanap) alatt beérkező csomagolási egységek számát, mely értéket az SAP táblák adatai alapján számoltam. A folyamatlépések költségének megállapításához demonstratív értékeket vettem alapul.

Folyamatlépés	Belépési pont	Áru átvétel tárolója	Áru átvételezése	Bekönyvelés tárolója	Bekönyvelés	Tárhelyre rakás tárolója	Az anyag tárhelyre rakása	Kilépési pont
Folyamatlépés átfutási ideje	0,9 perc/egység	-	Optimálendő	-	Optimálendő	-	0,9 perc/egység	-
Folyamatlépés költsége/egység	5.000Ft/egység	100Ft/egység/perc	90Ft/egység	1 Ft/egység/perc	20 Ft/egység	100 Ft/egység/perc	150 Ft/egység	-6.250 Ft/egység
Folyamatlépés költsége/nap	-	-	30.000 Ft/nap	-	20.000 Ft/nap	-	50.000 Ft/nap	-

7. táblázat - Bevételezési folyamat költségei

A 7. táblázatban nem szereplő fix heti (5 munkanap) költségei:

- Emberi erőforrások költségei: **250.000 Ft.**
- Egyéb költségek (bérleti, karbantartási díjak): **900.000 Ft.**

A költségfüggvényt az alábbi egyenlettel írtam fel:

$$Profit = \sum_{i=0}^n (1,25BK_i - ((90 + 20 + 150)CSE_i + 0,05 BK_i)) - Ek - Egyéb \quad (4)$$

Ahol:

BK - Beruházási költség

CSE – Csomagolási egység

Ek – Emberi erőforrás költségei

$Egyéb$ – Egyéb költségek

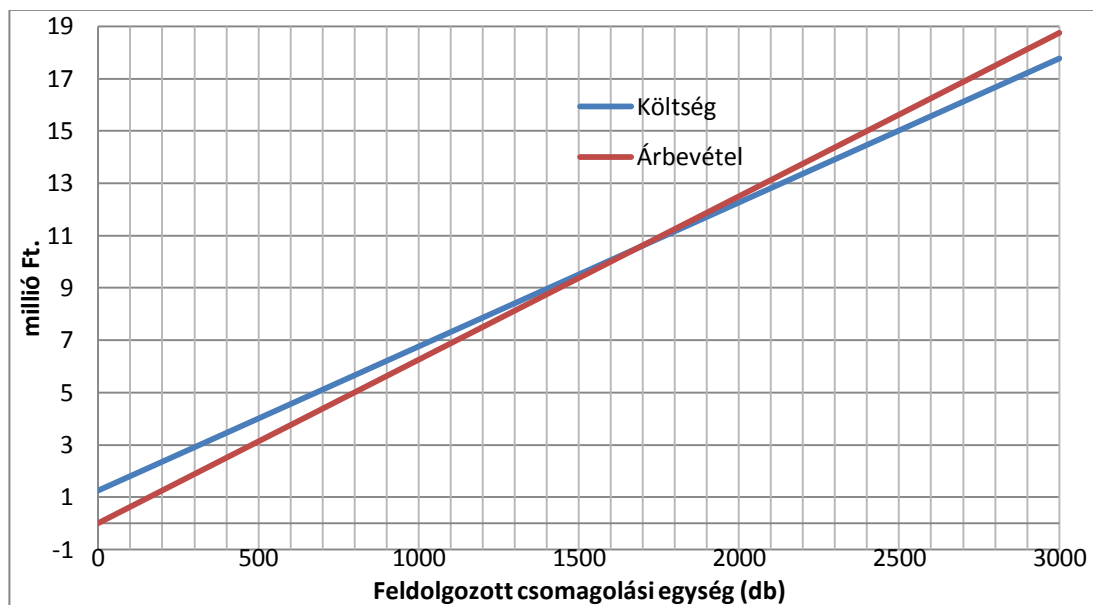
n - Az 5 munkanap alatt beérkezett csomagolási egységek száma

$i > 0$ egész

A példában jellemzett vállalat 25%-os haszonnal értékesíti a termékeket. A profit felírásánál az eladási árat az $1,25BK$ határozza meg, a CSE együtthatója az egyes folyamatlépések költsége csomagolási egységekre lebontva.

Profit grafikus ábrázolása a feldolgozott csomagolási egységek függvényében:

Átlagosan a bevételezési folyamatnál 2800 darab csomagolási egység érkezik be egy héten (5 munkanap). A diagramról (27. ábra) leolvasható, hogy a folyamat 1800db/hét csomagolási egység feldolgozása esetén lesz nyereséges.



27. ábra: Bevételezési folyamat költségfüggvénye

A diagram a feldolgozott csomagolási egységek függvényében ábrázolja a feldolgozási költséget és az eladási árat. A két görbe közötti terület a profit mértékét adja meg.

SIMUL8 és a gradiens módszer

Annak érdekében, hogy a SIMUL8 által kiértékelt célfüggvény gradiens süllyedés eljárással optimalizálható legyen, a kiértékelt függvény deriváltjának értékét közelítenem kellett, mivel az analitikus deriválás nem megoldható. A kétváltozós gradiens az alábbi formulával írható fel:

$$x_1^{k+1} = x_1^k - \alpha \frac{\text{sim8}(x_1^k, x_2^k) - \text{sim8}(x_1^k - dx_1, x_2^k)}{dx_1} \quad (5)$$

$$x_2^{k+1} = x_2^k - \alpha \frac{\text{sim8}(x_1^k, x_2^k) - \text{sim8}(x_1^k, x_2^k - dx_2)}{dx_2} \quad (6)$$

Egy iterációhoz összesen három SIMUL8 kiértékelés szükséges. Elsőként az x_1^{k+1} értékének meghatározásához meg kell hívni a szimulációt az $x_1^k - dx_1$ és az x_2^k értékekkel, majd hasonlóképpen az x_2^{k+1} érték meghatározásánál is. Ezután ki kell értékelni a célfüggvényt az új paraméterekkel, tehát vizsgálnunk kell a szimulációt az x_1^k és az x_2^k paraméterekkel. A módszert MS Excel VBA kódban valósítottam meg (M2 melléklet).

Az optimalizációs algoritmus kezelőablakát a 28. ábra mutatja. Az első három oszlopban az eljárás futási eredményét láthatjuk minden iterációban. A felhasználó a kezdeti paraméterek feltöltése után a nyomógomb segítségével indíthatja a szimulációt.

	A	B	C	D	E	F	G	H	I
1	x1 es x1-dx1	x2 es x2-dx2	y(x1,x2)	y'(x1,x2)	alpha	0.01	0	SIM8	
2	1.5	1.2	5.91664		megállási krit.	0.001		xi	xi-dxi
3	1.35	1.2	4.864295		dx1	0.15		1.5	1.35
4	1.5	1.05	5.91038		dx2	0.15		1.2	1.05
5	1.429843667	1.199582667	5.45887						
6	1.279843667	1.199582667	4.289735		Start				
7	1.429843667	1.049582667	5.452765						
8	1.351901333	1.199175667	4.882035						
9	1.201901333	1.199175667	3.61444						
10	1.351901333	1.049175667	4.875755						
11	1.267395	1.198757	4.177805						
12	1.117395	1.198757	2.897435						
13	1.267395	1.048757	4.17758						
14	1.182037	1.198742	3.458505						
15	1.032037	1.198742	2.068005						
16	1.182037	1.048742	3.36989						
17	1.089337	1.192834333	2.625905						
18	0.939337	1.192834333	0.984735						
19	1.089337	1.042834333	2.34821						

28. ábra: Gradiens munkalap

Az ábrán látható, hogy az algoritmus futtatásához összesen 6 paramétert kell megadnunk, ezek közül *alpha* a tanulási ráta, *dx1* és *dx2* a deriválás paramétere, a *megállási kritérium* a kereső pontosságát határozza meg, a *SIM8* alatt pedig a kezdeti értékeket kell definiálnunk

A fentebb bemutatott illusztratív SIMUL8 modellre ráillesztettem az algoritmust, a lefuttatás után az eljárás mind a két folyamatlépésre a 0.9 perc folyamatlépési időt hozta ki eredményül. Ebben a leegyszerűsített esetben látható, hogy az optimális eredmény pontosan ez, hiszen ahhoz, hogy a drága tárolási költségeket elkerüljük, a legjobb megoldás, ha mindegyik folyamatlépésünk ugyanazzal a lefutási idővel bír.

5. Összegzés

A kutatómunkámban logisztikai rendszerek kapacitásainak, teljesítményének és költségének jellemzésére használt mutatószámok összefüggéseit és bizonytalanságát is kezelő logisztikai kontrolling rendszer elméleti hátterének kidolgozásával kapcsolatos eredményeket foglaltam össze. A felvázolt keretrendszer kialakítása megköveteli a folyamatok részletes elemzését és felmérését. A modell alkalmazását példaként az EON Gazdasági Szolgáltató Kft. raktározási logisztikai folyamatainak optimalizálási feladatainak vizsgálatában mutattam be.

A kidolgozott módszertan hatékonyan integrálja a különböző szimulációs és vizualizációs eszközöket, mint pl. a SIMUL8 programcsomag. Ezen eszközök alkalmazásával a folyamat szimuláció és vizualizáció mellett lehetőség nyílik a folyamatok optimalizációjára is. A felmérések alapján megállapított folyamatidők statisztikai jellemzésén alapuló szimulációs eszközök segítségével, a Monte-Carlo szimulációval támogatott optimalizáció hatékonyan végrehajtható. A SIMUL8-en alapuló szimulációs vizsgálat azon képessége, hogy a döntési változók függvényében lehetővé válik a fejlesztési célokat definiáló célfüggvény kiértékelése felveti az automatizált paraméteroptimalizáció lehetőségét, ezért a kialakított keretrendszer segítségével gradiens süllyedés eljárással paraméteroptimalizálásra alkalmas eszközt fejlesztettem ki. Eredményeinket az elmúlt időszakban sikeresen publikáltuk konferenciákon [15] [16] [17] illetve kiadványokban.[13][14]

A kidolgozott módszertan és keretrendszer fejlesztésében az én feladatomban volt a SIMUL8 szimulációs programcsomag és az SAP adattáblák közötti adaptálás megteremtése, a könnyebb és funkcióiban kibővített SIMUL8 paraméterezési lehetőségek fejlesztése, a szimulációs modell kialakítása Monte-Carlo szimulációra alkalmasra, automatikus optimalizációs eljárás fejlesztése és integrálása a szimulációs programcsomaggal.

Köszönetnyilvánítás

A publikáció a TAMOP-4.2.1/B-09/1/KONV-2010-0003 projekt társfinanszírozásával készült, az ismertetett eredmények az E.ON Gazdasági Szolgáltató Kft által támogatott kutatás keretében készültek.

Továbbá szeretném megköszönni Dr. Ulbert Zsoltnak a MS Excel interfész és a szimulációs modellek elkészítésében való segítségét, Gyozsán Zoltánnak a projekt szakmai támogatását, és Varga Tamásnak a Monte-Carlo szimuláció eredmények kiértékelésében nyújtott segítségéért.

Végezetül szeretném megköszönni Dr. Abonyi Jánosnak a fáradhatatlan és ösztönző munkáját!

Irodalomjegyzék

- [1] Whicker, L., Bernon, M., Templar, S., Mena C., Understanding the relationships between time and cost to improve supply chain performance, *Int. J. Production Economics* 121, 641–650 (2009)
- [2] Comellia, M., Feniesa, P., Tcherneva N., A combined financial and physical flows evaluation for logistic process and tactical production planning: Application in a company supply chain, *Int. J. Production Economics*, 112, 77–95 (2008)
- [3] A Simtech Logisztika Kft. adatai szerint (simtech.hu) (2012.03.24.)
- [4] Midler, J. L. A stochastic multiperiod multimode transportation model. *Transportation Science*, 3, 8-29. (1969)
- [5] Tapiero, C. S., és Soliman, M. A. Multi-commodities transportation schedules over time. *Networks*, 2, 311-327. (1972)
- [6] Lee, H. L., és Billington, C. Material management in decentralized supply chains. *Operations Research*, 41 (5), 835-847. (1993)
- [7] Lee, H. L. és Feitzinger, E. Product configuration and postponement for supply chain efficiency. Fourth Industrial Engineering Research Conference Proceedings *Institute of Industrial Engineers*. (pp. 43-48). (1995)
- [8] Swaminathan, J. M., és Tayur, S. R. Stochastic programming models for managing product variety. In S. Tayur, R. Ganeshan és M. Magazine, Quantitative models for supply chain management *Boston, MA: Kluwer Academic Publishers*. (pp. 585-622). (1999)
- [9] Fisher, M., Hammond, J., Obermeyer, W., és Raman, A. Configuring a supply chain to reduce the cost of demand uncertainty. *Production and Operations Management*, 6 (3), 211-225. (1997)
- [10] Lee, H. L., Padmanabhan, V., és Whang, S. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4), 546-558. (1997)
- [11] Metters, R. Quantifying the bullwhip effect in supply chains. *Journal of Operations Management*, 15, 89-100. (1997)
- [12] Tamás Ruppert, Gábor Belvárdi, Zoltán Gyózsán, János Abonyi, Zsolt Ulbert, Logisztikai tevékenységet támogató folyamatmodellek fejlesztése és Monte-Carlo szimulációval történő kialakítása, *Műszaki Kémiai Napok* 2012, 47-53. (2012)

- [13] Zsolt Ulbert, Tamás Ruppert, Gábor Belvárdi, Zoltán Gyózsán, János Abonyi, Interactive Balanced Scorecard based on Simulation of Production and Services Systems, *Factory Automation 2012*, 88-91. (2012)
- [14] Műszaki Kémiai Napok 2012, Veszprém (2012. április 24-26.)
- [15] Proceedings of Factory Automation 2012, Veszprém (2012. május 21-22.)
- [16] WASET 2012, Amsterdam nemzetközi konferencia, (2012. július 26-27.)
- [17] A Poliphon adatai szerint (poliphon.hu)-(2012.04.02.)
- [18] Daniel P. Loucks, Eelco van Beek, Water Resources Systems Planning and Management: *An Introduction to Methods, Models and Applications*, 9, 256 – 265, (2005)

Mellékletek

M1 SIMUL8 modellek paraméterezése

A SIMUL8 szimulációs programcsomag széleskörű paraméterezhetőséget biztosít, mely paramétereket a kifejlesztett MS Excel interfész segítségével egy ismerős és felhasználóbarát környezetből adhatunk meg. A SIMUL8 környezetben létrehozott modellek gyakorlati paraméterezése a következő 7 főbb csoportra osztható, amely csoportok részben megfelelnek a SIMUL8 modellek grafikus építőelemeinek:

- Bemenet paraméterezése (Work entry point)
- Folyamatlépések paraméterezése (Work center)
- Tárolók paraméterezése (Queue)
- Erőforrások paraméterezése (Resource)
- Költségek
- Szimulációs idő paraméterezése
- Kimutatások, monitoring (KPI) paraméterezése

Bemenet paraméterezése:

A Work Entry Point elem a modell bemenetét, kezdő pontját jelképezi. A bemenet adatait, többféle formátumban is megadhatjuk:

- Manuálisan beállítva a SIMUL8 programon belül
- Közvetlenül egy csv fájlból

A bemenet eseménysorozat megadása az utolsó esetben az M1.1. ábra szerint történik. A táblának tartalmaznia kell a beérkezési időt és az adott időpillanatban beérkező munkaegység darabszámát is. A bemeneti időt megadhatjuk a munkakezdet óta eltelt idővel, vagy óra:perc formátumban is.

	A	B	C	D	E
1	Time	db			
2	30	3			
3	80	5			
4	120	7			
5	216	12			
6	478	9			
7	7:13	1			
8	8:45	6			
9	12:59	8			
10	14:25	13			

M1.1. ábra: Bemenet eseménysorozat mintafájl.

A SIMUL8 bemeneti adatait két oszlopban kell megadni. Az első oszlop a beérkezési idő, a második pedig a beérkező munkaegység darabszáma. A bemeneti időket kétféleképpen adhatjuk meg: vagy a munkanap kezdetétől eltelt idővel, vagy pedig óra:perc formátumban.

Folyamatlépések paraméterezése:

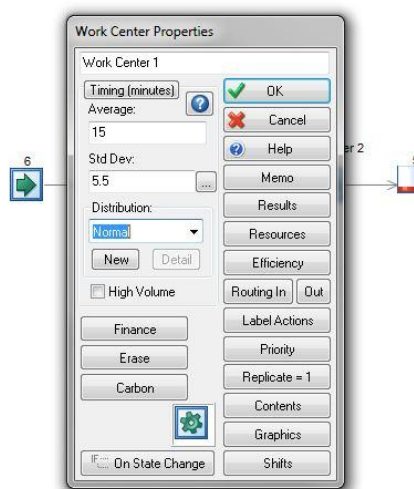
Minden folyamatlépés külön-külön paraméterezhető (M1.2. ábra), a főbb paraméterezhetőségek az alábbiak:

- Folyamatidő
Mindegyik folyamatlépésnek beállíthatjuk, hogy egy bemeneti egység feldolgozása mennyi időt vesz igénybe;
- A folyamatidők szórása
Mivel legtöbbször a valóságban a munkaegységek feldolgozási ideje nem konstans, a folyamatlépéshez egy eloszlástípust és szórást adhatunk meg. (M1.3. ábra);
- Erőforrások
A folyamatlépésekhez hozzárendelhetünk erőforrásokat, ezek lehetnek emberi és gépi erőforrások is (M1.4. ábra);
- Útvonal
Folyamat elágazásnál lehetőség van a munkaegységek százalékos eloszlású továbbítására (M1.5. ábra)
- Párhuzamos futtatás
Egy folyamat párhuzamosan többször is futhat az erőforrások rendelkezésre állásának függvényében. Lényegében az adott folyamatlépés megtöbbszörözését jelenti.
- Folyamatlépés költsége (M1.6. ábra)



M1.2. ábra: Folyamatlépések paramétrezhetősége

Egy folyamat, feldolgozó egység paramétereinek megadására szolgáló dialógus



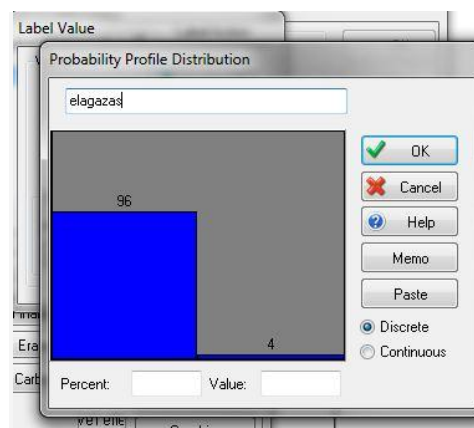
M1.3 ábra: Folyamatidő szórása

A lépések folyamatidejéhez szórásokat rendelhetünk, de beállíthatjuk fix, vagy átlagos értékre is.



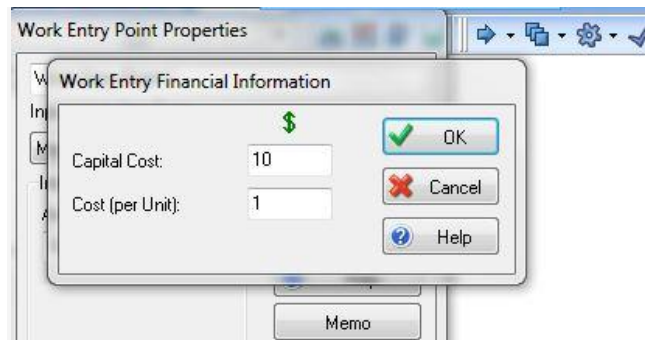
M1.4. ábra: Erőforrások

Különböző munkakörű erőforrásokat rendelhetünk folyamatlépéseinkhez, de beállíthatjuk az eszközigényeit is.



M1.5. ábra: Elágazás eloszlása

Az egyes folyamatlépések több irányba is továbbadhatják a munkadarabot, azt hogy melyik irányba küldi tovább százalékosan vagy címkézéssel is meghatározható.

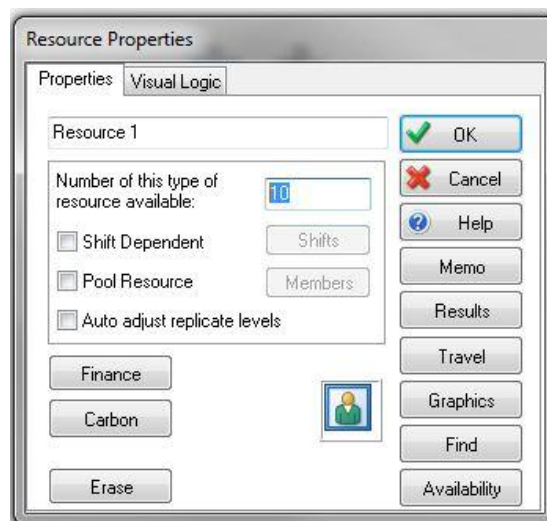


M1.6. ábra: Költségek

A folyamatoknál beállíthatjuk a munkavégzéshez szükséges anyagi forrást és a nyereséget is.

Erőforrások paraméterezése:

A személyi erőforrások beállításánál egy fix értéket állíthatunk be mindegyik erőforrásunknál (M1.7 ábra).



M1.7. ábra: Erőforrás fix érték

A rendelkezésre álló erőforrások számát egy fix értékre állíthatjuk be

További beállítási lehetőség az erőforrások paraméterezése kapcsán, hogy megadhatjuk, hogy egyes erőforrások egy munkanap során milyen százalékban elérhetőek.

Költségek:

A költségek beállítása a SIMUL8-ben általánosítva minden objektumnál két részre osztható, az egyik a tényleges munka utáni költség, a másik pedig a bekerülési költség, amely független az elvégzett munkától, ez az úgynevezett „Capital cost”, mely minden objektumnál megadható (M1.8. ábra). A tényleges munka költségeinek beállításait 5 csoportba oszthatjuk:

- A bemenet költségei

A bemeneti pontnál megadhatjuk, hogy mennyi a költsége a belépő egységeknek (Cost per Unit)

- Tartályok költségei

A tárolók esetén beállíthatjuk, hogy mennyi egy egységnek a költsége percenként ((Cost per item) per minute)

- Folyamatlépések költségei

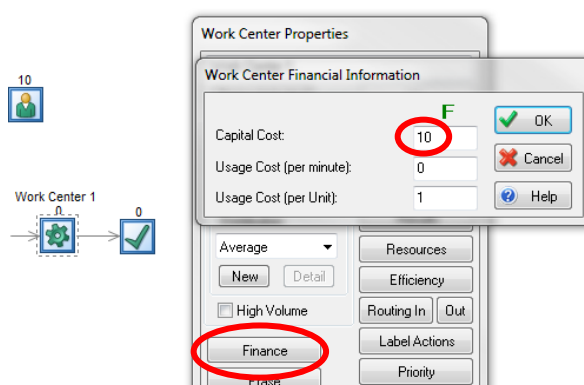
A folyamatlépések költségeinél két opcióból is választhatunk, megadhatjuk a költségét percenként, tehát az egységek költsége attól függ mennyi ideig tart a feldolgozása (Usage cost (per minute)), de megadhatjuk egységenként költséget (Usage Cost per Unit) is.

- Befejezés költségei

A kilépési pontnál paraméterezhetjük, hogy az elkészült egységeket milyen áron értékesítjük. (Revenue per Unit)

- Erőforrások költségei

Az erőforrásoknál megadhatjuk a költségét az erőforrások által ténylegesen ledolgozott idő és feldolgozott egység függvényében ((Cost per unit) per minute)



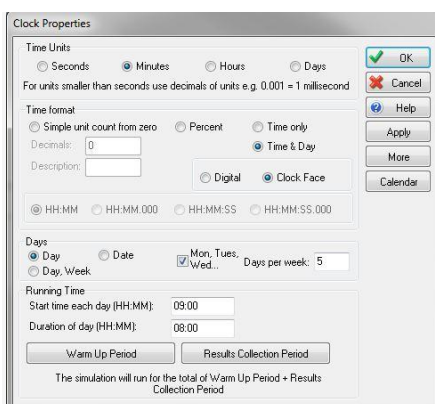
M1.8. ábra: Bekerülési költség beállítása

A „Capitol Cost” beállítása minden objektumnál azonos, a beállítási panelon a „Finance” kapcsolóval érhető el

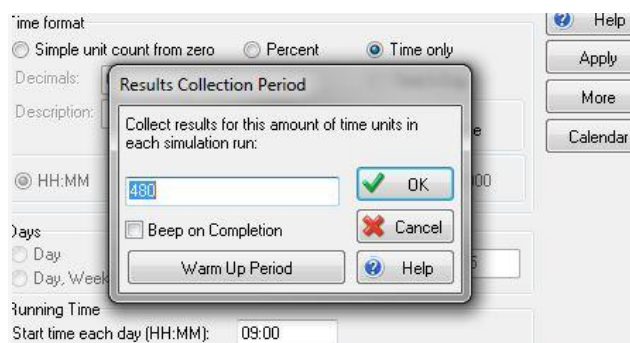
A szimuláció időtartama:

A szimuláció idejének beállításánál az alábbi paramétereket megadására van lehetőségünk (M1.9. ábra):

- A folyamatidők dimenziója: másodperc/perc/óra/nap
- A munkanapok száma egy hétre nézve
- A munkanap kezdeti ideje
- A munkanap hossza
- Szimuláció időtartama (M1.10. ábra)

**M1.9. ábra: Futási idő**

Ezen a felületen állíthatjuk be a heti munkanapok számát, a munkanap hosszát, és a szimulációs modellnél használt időegységet is.

**M1.10. ábra: A vizsgálni kívánt időtartam**

Ezen a felületen lehet beállítani, hogy milyen időintervallumot szeretnénk futtatni.

Monitoring (Kulcsteljesítmény mutatók - KPI)

A SIMUL8 az összes fontos információt képes monitoringozni, ezeknek a listáját az ún. KPI-ben állíthatjuk be (M1.11. ábra).

SIMUL8 Results Summary

Results

Detail

Help

OK

from most recent run only.

Result

Click "Multiple Runs" button to get ranges.

Terminator	Average Time in System	0.00
Áru átvételezése	Waiting %	0.00
	Working %	0.00
	Blocked %	0.00
	Stopped %	0.00
	Number Completed Jobs	15.00
	Minimum use	0.00
	Average use	0.89
	Maximum use	3.00
	Current Contents	0.00
	Change Over %	0.00
	Off Shift %	0.00
Resource Starved %	0.00	
Az anyag tárhelyre rakása	Waiting %	0.00
	Working %	0.00
	Blocked %	0.00
	Stopped %	0.00
	Number Completed Jobs	15.00
Minimum use	0.00	

M1.11. ábra: Mutatók monitoringja

A SIMUL8 Results Summary lehetőséget ad arra, hogy beállíthassuk mely mutatókat szeretnénk kinyerni.

M2 Visual Logic, és MS Excel VBA kódok

SIMUL8 Visual Logic kódok

```

Start Run Logic
-- Warning. This code can run at any simulation time when the user click the RUN button
-- Clear Sheet Area input_sheet[1,1] , 50 , 5000
-- Clear Sheet Area input_sheet[2,1,1] , 50 , 5000
-- Clear Sheet Area allapot[1,1] , 50 , 5000
-- Clear Sheet Area allapot[2,1,1] , 50 , 5000
-- Clear Sheet Area atlag[1,1] , 50 , 5000
-- Clear Sheet Area Zatlalag[1,1] , 50 , 5000
-- Clear Sheet Area prioritas[1,1] , 50 , 5000
-- Clear Sheet Area parhuzamos[1,1] , 50 , 5000
-- Clear Sheet Area output[1,1] , 50 , 5000
-- Clear Sheet Area ugyvitel[1,1] , 50 , 5000
-- Clear Sheet Area kitaros[1,1] , 50 , 5000
-- Clear Sheet Area betar_ugyvitel[1,1] , 50 , 5000
-- Clear Sheet Area diszpecser[1,1] , 50 , 5000
-- Set in EXCEL output[1,1] , "[raktarozas.xls]LOG" , 1 , 2 , 50 , 5000
-- Get from EXCEL sum_input , "[raktarozas.xls]bevetelezes" , 11 , 2 , 1 , 1
-- Get from EXCEL sum_input2 , "[raktarozas.xls]kitar" , 13 , 2 , 1 , 1
-- IF sum_input < 27
-- SET input_row = 26
-- IF sum_input2 < 27
-- SET input_row2 = 26
-- IF sum_input > 26
-- SET input_row = sum_input+1
-- IF sum_input2 > 26
-- SET input_row2 = sum_input2+1
-- Get from EXCEL input_sheet[1,1] , "[raktarozas.xls]bevetelezes" , 2 , 2 , 1 , input_row
-- LOOP 1 >>> i>>> sum_input
-- SET input_sheet[2,i] = 1
-- Get from EXCEL input_sheet[3,1] , "[raktarozas.xls]bevetelezes" , 3 , 2 , 9 , input_row
-- Get from EXCEL input_sheet[2,1,1] , "[raktarozas.xls]kitar" , 2 , 2 , 1 , input_row2
-- Get from EXCEL input_sheet[3,1] , "[raktarozas.xls]bevetelezes" , 3 , 2 , 9 , input_row
-- Get from EXCEL input_sheet[2,1,1] , "[raktarozas.xls]kitar" , 2 , 2 , 1 , input_row2
-- LOOP 1 >>> i>>> sum_input2
-- SET input_sheet[2,i] = 1
-- Get from EXCEL input_sheet[2,3,1] , "[raktarozas.xls]kitar" , 3 , 2 , 10 , input_row2
-- Get from EXCEL prioritas[1,1] , "[raktarozas.xls]Konfiguralas" , 1 , 1 , 4 , 21
-- Get from EXCEL parhuzamos[1,1] , "[raktarozas.xls]Konfiguralas" , 6 , 1 , 4 , 6
-- Get from EXCEL diszpecser[1,1] , "[raktarozas.xls]bevetelezes" , 6 , 9 , 5 , 8
-- Get from EXCEL betar_ugyvitel[1,1] , "[raktarozas.xls]bevetelezes" , 6 , 19 , 5 , 8
-- Get from EXCEL kitaros[1,1] , "[raktarozas.xls]kitar" , 8 , 9 , 5 , 8
-- Get from EXCEL ugyvitel[1,1] , "[raktarozas.xls]kitar" , 8 , 19 , 5 , 8
-- SET Szállítólevél ellenorzése.Priority = prioritas[2,2]
-- SET SAP ellenorzése.Priority = prioritas[2,3]
-- SET Áru átvétel.Priority = prioritas[2,4]
-- SET Bekönyvelés.Priority = prioritas[2,5]
-- SET Az anyag tárhelyre rakása.Priority = prioritas[2,6]
-- SET TR és elozmények.Priority = prioritas[4,2]
-- SET Anyag összekészítés.Priority = prioritas[4,3]
-- SET Kikönyvelés.Priority = prioritas[4,4]
-- SET Anyag átadása.Priority = prioritas[4,5]
-- SET Szállítólevél ellenorzése.Replication = parhuzamos[2,2]
-- SET SAP ellenorzése.Replication = parhuzamos[2,3]
-- SET Áru átvétel.Replication = parhuzamos[2,4]
-- SET Bekönyvelés.Replication = parhuzamos[2,5]
-- SET Az anyag tárhelyre rakása.Replication = parhuzamos[2,6]
-- SET TR és elozmények.Replication = parhuzamos[4,2]
-- SET Anyag összekészítés.Replication = parhuzamos[4,3]
-- SET Kikönyvelés.Replication = parhuzamos[4,4]
-- SET Anyag átadása.Replication = parhuzamos[4,5]
-- SET i2 = 0

```

```

-- LOOP 1 >>> i >>> sum_input2
--
-- SET jel = 0
--
-- LOOP 1 >>> i1 >>> i2
--
-- IF input_sheet2[3,i] = input_sheet2[25,i1]
--
-- SET jel = 1
-- SET input_sheet2[27,i1] = input_sheet2[27,i1]+1
-- Break
--
-- IF jel = 0
--
-- SET i2 = i2+1
-- SET input_sheet2[25,i2] = input_sheet2[3,i]
-- SET input_sheet2[26,i2] = input_sheet2[4,i]
-- SET osszeszamol[2,i2] = input_sheet2[3,i]
-- SET input_sheet2[27,i2] = 1
--
-- LOOP 1 >>> i >>> i2
-- SET osszeszamol[1,i] = 0
-- SET Work Center 10.Fixed Type Label Value = i2+1

```

Queue for Szállítólevél ellenőrzése On Entry Logic

```

-- SET row_ = row_+1
-- SET állapot[3,row_+[maradekok[2,1]]] = input_sheet[10,2]*input_sheet[3,row_+1]
-- SET állapot[4,row_+[maradekok[2,1]]] = input_sheet[10,3]*input_sheet[4,row_+1]
-- SET állapot[5,row_+[maradekok[2,1]]] = input_sheet[10,4]*input_sheet[4,row_+1]
-- SET állapot[6,row_+[maradekok[2,1]]] = input_sheet[10,5]*input_sheet[4,row_+1]
-- SET állapot[7,row_+[maradekok[2,1]]] = input_sheet[10,6]*input_sheet[4,row_+1]

```

Time Check Logic

```

--
-- IF temp = 60
-- SET temp = 0
--
-- IF temp = 0
-- SET Betár_ügyviteli.Max Available = betar_ugyvite[diszp_column,diszp_row]
-- SET Betárolásért felelos diszpécser.Max Available = diszpecser[diszp_column,diszp_row]
-- SET Kiszállítási diszpécser.Max Available = kitaros[diszp_column,diszp_row]
-- SET Ügyviteli eloadó.Max Available = ugyvitel[diszp_column,diszp_row]
-- SET diszp_row = diszp_row+1
--
-- IF diszp_row = 9
-- SET diszp_row = 1
-- SET diszp_column = diszp_column+1
--
-- IF diszp_column = 6
-- SET diszp_column = 1
--
-- SET temp = temp+1
--
-- SET output[1,out_row] = Simulation Time
-- SET output[2,out_row] = Szállítólevél ellenőrzése.Count Contents
-- SET output[3,out_row] = SAP ellenőrzése.Count Contents
-- SET output[4,out_row] = Áru átvétel.Count Contents
-- SET output[5,out_row] = Bekönyvelés.Count Contents
-- SET output[6,out_row] = Az anyag tárhelyre rakása.Count Contents
-- SET output[7,out_row] = TR és elozmények.Count Contents
-- SET output[8,out_row] = Anyag összekészítés.Count Contents
-- SET output[9,out_row] = Kikönyvelés.Count Contents
-- SET output[10,out_row] = Anyag átadása.Count Contents

```

```
-- Get Result  output[12,out_row] , Current Run , Betárolásért felelos diszpécser: Current Use
-- Get Result  output[13,out_row] , Current Run , Betár-ügyviteli: Current Use
-- Get Result  output[14,out_row] , Current Run , Kiszállítási diszpécser: Current Use
-- Get Result  output[15,out_row] , Current Run , Ügyviteli eloadó: Current Use

-- SET output[16,out_row] = Simulation Time/480

-- SET out_row = out_row+1
```

Szállítólevél ellenorzése On Exit Logic

```
-- SET 1sor2 = 1sor2+1
-- SET atlag1 = atlag1+Szállítólevél ellenorzése.Operation Time
-- SET atlag[2,sorszam] = sorszam
-- SET atlag[3,sorszam] = Szállítólevél ellenorzése.Operation Time
```

End Run Logic

```
-- IF 1sor2 <> 0
-- SET atlag[1,1] = atlag1/1sor2

-- IF 2sor2 <> 0
-- SET atlag[1,2] = atlag2/2sor2

-- IF 3sor2 <> 0
-- SET atlag[1,3] = atlag3/3sor2

-- IF 4sor2 <> 0
-- SET atlag[1,4] = atlag4/4sor2

-- IF 5sor2 <> 0
-- SET atlag[1,5] = atlag5/5sor2

-- IF 21sor2 <> 0
-- SET 2atlag[1,1] = atlag1_2/21sor2

-- IF 22sor2 <> 0
-- SET 2atlag[1,2] = atlag2_2/22sor2

-- IF 23sor2 <> 0
-- SET 2atlag[1,3] = atlag3_2/23sor2

-- IF 24sor2 <> 0
-- SET 2atlag[1,4] = atlag4_2/24sor2

-- Set in EXCEL  Szállítólevél ellenorzése.Name , "[raktarozas.xls]MonteCarlo" , 1 , 1 , 1 , 1
-- Set in EXCEL  SAP ellenorzése.Name , "[raktarozas.xls]MonteCarlo" , 1 , 2 , 1 , 1
-- Set in EXCEL  Áru átvétel.Name , "[raktarozas.xls]MonteCarlo" , 1 , 3 , 1 , 1
-- Set in EXCEL  Bekönyvelés.Name , "[raktarozas.xls]MonteCarlo" , 1 , 4 , 1 , 1
-- Set in EXCEL  Az anyag tárhelyre rakása.Name , "[raktarozas.xls]MonteCarlo" , 1 , 5 , 1 , 1
-- Set in EXCEL  TR és elozmények.Name , "[raktarozas.xls]MonteCarlo" , 1 , 6 , 1 , 1
-- Set in EXCEL  Anyag összekészítés.Name , "[raktarozas.xls]MonteCarlo" , 1 , 7 , 1 , 1
-- Set in EXCEL  Kikönyvelés.Name , "[raktarozas.xls]MonteCarlo" , 1 , 8 , 1 , 1
-- Set in EXCEL  Anyag átadása.Name , "[raktarozas.xls]MonteCarlo" , 1 , 9 , 1 , 1

-- SET mc1 = input_sheet[11,2]
-- Set in EXCEL  atlag[1,1] , "[raktarozas.xls]MonteCarlo" , mc1 , 1 , 1 , 5
-- Set in EXCEL  2atlag[1,1] , "[raktarozas.xls]MonteCarlo" , mc1 , 6 , 1 , 4

-- Set in EXCEL  output[1,1] , "[raktarozas.xls]LOG" , 1 , 2 , 15 , out_row
-- Set in EXCEL  atlag[2,1] , "[raktarozas.xls]LOG" , 17 , 2 , 6 , 1sor2
-- Set in EXCEL  2atlag[2,1] , "[raktarozas.xls]LOG" , 24 , 2 , 5 , 21sor2

-- Set in EXCEL  output[16,1] , "[raktarozas.xls]LOG" , 31 , 2 , 1 , out_row
```

MS Excel VBA gradiens eljárás programkódja

```
Private WithEvents simulation As SIMUL8.S8Simulation
```

```
Private x1 As Double
```

```
Private x2 As Double
```

```
Private n As Integer
```

```
Private k As Integer
```

```
Private i As Integer
```

```
Private Sub CommandButton1_Click()
```

```
'SIMUL8 fájl megnyitása, es egyszeri lefuttatása
```

```
    n = 0
```

```
    k = 1
```

```
    Dim path As String
```

```
    path = ActiveWorkbook.path
```

```
    path = path + "\koltseg_opt_2.3.S8"
```

```
    Set simulation = GetObject("", "SIMUL8.S8Simulation")
```

```
    Worksheets("Munka1").Cells(1, 10).Value = Worksheets("Munka1").Cells(3, 8).Value
```

```
    Worksheets("Munka1").Cells(2, 1).Value = Worksheets("Munka1").Cells(3, 8).Value
```

```
    Worksheets("Munka1").Cells(2, 10).Value = Worksheets("Munka1").Cells(4, 8).Value
```

```
    Worksheets("Munka1").Cells(2, 2).Value = Worksheets("Munka1").Cells(4, 8).Value
```

```
    simulation.Open path
```

```
    simulation.RunSim 480 * Worksheets("Munka1").Cells(2, 11).Value
```

```
End Sub
```

```
Private Sub simulation_S8SimulationEndRun()
```

```
    n = n + 1
```

```
'A program ki írja a SIMUL8 által kiértékelt célfüggvényt
```

```
    Worksheets("Munka1").Cells(n + 1, 3).Value = simulation.Results(simulation.ResultsCount).Value * (-1)
```

```
' x1 es x2 k+1-ik értékének számítása
```

```
    If (k = 3) Then
```

```
        x1 = Worksheets("Munka1").Cells(n - 1, 1) - Worksheets("Munka1").Cells(1, 6) *
```

```
((Worksheets("Munka1").Cells(n - 1, 3) - Worksheets("Munka1").Cells(n, 3)) /
```

```
Worksheets("Munka1").Cells(3, 6))
```

```
        x2 = Worksheets("Munka1").Cells(n - 1, 2) - Worksheets("Munka1").Cells(1, 6) *
```

```
((Worksheets("Munka1").Cells(n - 1, 3) - Worksheets("Munka1").Cells(n + 1, 3)) /
```

```
Worksheets("Munka1").Cells(4, 6))
```

```
        Worksheets("Munka1").Cells(n + 2, 1).Value = x1
```

```
        Worksheets("Munka1").Cells(n + 2, 2).Value = x2
```

```
        Worksheets("Munka1").Cells(1, 10).Value = x1
```

```
        Worksheets("Munka1").Cells(2, 10).Value = x2
```

```
        k = 1
```

```
'Ez a feltétel lesz a megállási kritérium
```

```
    If Worksheets("Munka1").Cells(2, 6).Value > Abs(Worksheets("Munka1").Cells(n + 2, 1).Value -  
Worksheets("Munka1").Cells(n - 1, 1).Value) Then
```

```
        i = n + 1
```

```
    End If
```

```
'x2-dx2 paraméter számítása
```

```
ElseIf (k = 2) Then
    Worksheets("Munka1").Cells(1, 10).Value = Worksheets("Munka1").Cells(n, 1).Value
    Worksheets("Munka1").Cells(2, 10).Value = Worksheets("Munka1").Cells(n, 2).Value -
Worksheets("Munka1").Cells(4, 6).Value
    Worksheets("Munka1").Cells(n + 2, 1).Value = Worksheets("Munka1").Cells(n, 1).Value
    Worksheets("Munka1").Cells(n + 2, 2).Value = Worksheets("Munka1").Cells(n, 2).Value -
Worksheets("Munka1").Cells(4, 6).Value
    k = k + 1
```

'x1-dx lparaméter számítása

```
ElseIf (k = 1) Then
    Worksheets("Munka1").Cells(1, 10).Value = Worksheets("Munka1").Cells(n + 1, 1).Value -
Worksheets("Munka1").Cells(3, 6).Value
    Worksheets("Munka1").Cells(2, 10).Value = Worksheets("Munka1").Cells(n + 1, 2).Value
    Worksheets("Munka1").Cells(n + 2, 1).Value = Worksheets("Munka1").Cells(n + 1, 1).Value -
Worksheets("Munka1").Cells(3, 6).Value
    Worksheets("Munka1").Cells(n + 2, 2).Value = Worksheets("Munka1").Cells(n + 1, 2).Value
    k = k + 1
End If
```

' SIMUL8 meghívása a fent beállított aktuális adatokkal

```
If n > i Then
    simulation.RunSim 480 * Worksheets("Munka1").Cells(2, 11).Value
End If
```

End Sub