Németh György

Egy-kiszolgálóhelyes sorbanállási rendszer szimulációja

Vizsgálatunk célja egyetlen kiszolgálóhellyel és egy várakozó sorral rendelkező sorbanállási rendszer szimulációs modelljének kidolgozása, illetve a rendszer viselkedését szimuláló néhány kísérlet és tapasztalat bemutatása. A dolgozat nem új tudományos eredmény közlésére készült, de a vázolt rendszer könnyen áttekinthető matematikai és számítástechnikai modelljének megadásával módszertani útmutatást ad, és megkönnyíti az alkalmazást bárki számára. A modell alkalmas arra is, hogy bonyolultabb szimulációs rendszer részeként működjön.

1. A modell leírása

A szimuláció során feltételezzük, hogy
(1) az egységek egyenként érkeznek,
(2) a kiszolgálóhelynél nincs prioritás, tehát a rendszer azt szolgálja ki előbb, aki előbb érkezett,
(3) az érkezési időpontok közötti időintervallumok változása ugyanazt a valószínűségeloszlást (I. eloszlás) követi a szimuláció folyamán,
(4) az I. eloszlás az idő múlásával nem változik, azaz a rendszer stacionárius,
(5) a kiszolgálási idők meghatározott valószínűségeloszlást (II. eloszlás) követnek,
(6) a II. eloszlás is stacionárius, és
(7) az érkezési időpontok közötti időintervallumok, illetve a kiszolgálási idők eloszlása egymástól független.

A rendszerre jellemző két valószínűségeloszlást (beérkezések és kiszolgálások) véletlenszám-generátorokkal állítjuk elő. Monte Carlo technikát alkalmazunk egyenletes eloszlású véletlen számok, illetve a kívánt valószínűségeloszlásra illeszkedő sztochasztikus változók előállítására, majd e generátorokat úgy kapcsoljuk a modellhez, hogy a sztochasztikus folyamatot szimulálja. A véletlenszámgenerátorok szubrutinok formájában jelennek meg, és a szimuláció folyamán attól függően változtatjuk bemenő paramétereit, hogy a modell milyen speciális eloszlást követel meg. Ennek a partikuláris szimulációs technikának az alkalmazása a szimulált rendszert viszonylag könnyen érthetővé teszi.

2. A modell változói

A szimuláció során az alábbi mennyiségeket határozzuk meg:
(1) a teljes várakozási időt,
(2) a teljes tétlen időt,

(3) a várakozni kényszerülő egységek számát,
(4) a tétlen idő előfordulásainak számát,
(5) a generált belépési idők összegét,
(6) a teljes kiszolgálási időt,
(7) a rendszerben levő egységek számának gyakorisági- és valószínűségeloszlását (így az egyszerre várakozók maximális számát is)
(8) a várakozó egységek számának gyakorisági és valószínűségeloszlását,
(9) a várakozó egységek számának gyakorisági és valószínűségeloszlását nem üres várakozó sor esetén.

A fenti mennyiségek leírására az alábbi jelöléseket használjuk:

RAR: két egymást követő érkezés közti véletlen időintervallum,
RSR: véletlen kiszolgálási idő,
S: az egyidejűleg várakozó egységek maximális száma,
L_p: a szimulált egységek száma,
k: a rendszerben levő egységek száma,
n: a várakozó sorban levő egységek száma,
i: a rendszerbe érkezett egységek száma,
i_w: a várakozni kényszerülő egységek száma,
i_{dc}: tétlen idő előfordulásainak száma,
j: index annak jelölésére, hogy a kiszolgálóhely aktív-e, vagy sem ($j = 0$, ha tétlen; $j = 1$, ha aktív),
jn: index a teljes várakozási idő korrekt értékének meghatározására,
TC: a szimuláció során elért aktuális időpont,
TP: a következő érkezés előrejelzett időpontja,
TS: egy kiszolgálás befejezési időpontja,
WT: valamely egység várakozási ideje,
ID: tétlen idő,
T_k: az az időtartam, amelyet k egység tölt a rendszerben ($k = 0, 1, \ldots, S + 1$),
WA_n: az az időtartam, amelyet egyidejűleg n egység tölt várakozással,
TWT: a teljes várakozási idő,
TID: a teljes tétlen idő,
TST: a teljes kiszolgálási idő,
TAT: a szimuláció kezdetétől a legutóbb rendszerbe lépett egység belépésének időpontjáig eltelt idő,
$TATS$: a szimuláció teljes időtartama.

A felsorolt változók között néhány egyszerű kapcsolat adható meg, így:

$$TAT = \sum_i RAR$$

$$TST = \sum_i RSR$$

$$TWT = \sum_{i_w} WT$$

$$TID = \sum_{i_{dc}} ID$$

$$TATS = TAT + (TS^{\text{last}} - TE),$$

ahol TS^{las} a szimuláció befejezési, TE pedig az utolsó kiszolgálandó egység rendszerbe lépésének időpontja.

3. A matematikai modell

A fenti jelölésekkel a szimulált mennyiségeket a következő összefüggések alapján számíthatjuk.

Annak valószínűsége, hogy a rendszerben k számú egység tartózkodik

$$p_k = \frac{T_k}{TATS}, \quad k = 0, 1, \ldots, S + 1.$$

Annak valószínűsége, hogy n számú egység várakozik kiszolgálásra

$$q_n = \frac{WA_n}{TATS}, \quad n = 0, 1, \ldots, S.$$

Annak valószínűsége, hogy w számú egység várakozik egy nem üres sorban

$$r_w = \frac{WA_n}{SUM}, \quad w = 1, 2, \ldots, S$$

$$SUM = \sum_{w=1}^{S} WA_w.$$

Nyilvánvaló, hogy

$$\sum_{k=0}^{S+1} p_k = 1, \quad \sum_{n=0}^{S} q_n = 1, \quad \sum_{w=1}^{S} r_w = 1,$$

$$T_m = WA_{m-1}, \; p_m = q_{m-1}, \text{ ha } m > 0.$$

A modellt jellemző többi szimulált mennyiséget az alábbiak szerint adjuk meg:

a rendszerben levő egységek várható száma:

$$E(k) = \sum_{k=0}^{S+1} k \cdot p_k$$

a várakozó sor várható hossza:

$$E(n) = \sum_{n=0}^{S} nq_n,$$

nem üres várakozó sor várható hossza:

$$E(w|w > 0) = \sum_{w=1}^{S} wr_w,$$

egy érkező egység várható tartózkodási ideje a rendszerben:

$$E(T_{ws}) = \frac{TWT + TST}{L_p},$$

egy érkező egység várható várakozási ideje:

$$E(WT) = \frac{TWT}{L_p},$$

TP	$TP^{(1)}$	$TP^{(2)}$		$TP^{(3)}$	$TP^{(4)}$		$TP^{(5)}$	$TP^{(6)}$		$TP^{(7)}$			$TP^{(8)}$			$TP^{(9)}$		$TP^{(10)}$	
TS			$TS^{(1)}$			$TS^{(2)}$			$TS^{(3)}$		$TS^{(4)}$	$TS^{(5)}$		$TS^{(6)}$	$TS^{(7)}$	$TS^{(8)}$	$TS^{(9)}$		$TS^{(10}$
k	1	2	1	2	3	2	3	4	3	4	3	2	3	2	1	1	0	1	
n	0	1	0	1	2	1	2	3	2	3	2	1	2	1	0	0	0	0	
WA_n		WA_1		WA_1	WA_2	WA_1	WA_2	WA_3	WA_2	WA_3	WA_2	WA_1	WA_2	WA_1					
ID																	ID		
T_k	T_1	T_2	T_1	T_2	T_3	T_2	T_3	T_4	T_3	T_4	T_3	T_2	T_3	T_2	T_1	T_1	T_0	T_1	

1. ábra
Szimuláció $L_p = 10$ és szabadon választott valószínűségeloszlások esetén

egy várakozó egység várható várakozási ideje:

$$E(WT|WT > 0) = \frac{TWT}{i_w},$$

a várható tétlen idő:

$$E(ID) = \frac{TID}{L_p},$$

a szimuláció teljes időtartamának az a várható hányada, amíg a kiszolgálóhely tétlen:

$$E(f_{id}) = p_0,$$

a szimuláció teljes időtartamának az a várható hányada, amíg a kiszolgálóhely foglalt:

$$E(f_{occ}) = 1 - p_0$$

4. A modell folyamatábrája

A modell folyamatábrájának leírása előtt az 1. ábrán egy szimulációs folyamat néhány jellemző mennyiségének változásait és ezek pillanatnyi értékeit tüntettük fel. Itt TP^i az i-edik egység rendszerbe lépésének, TS^i pedig kiszolgálásának befejezési időpontját jelöli.

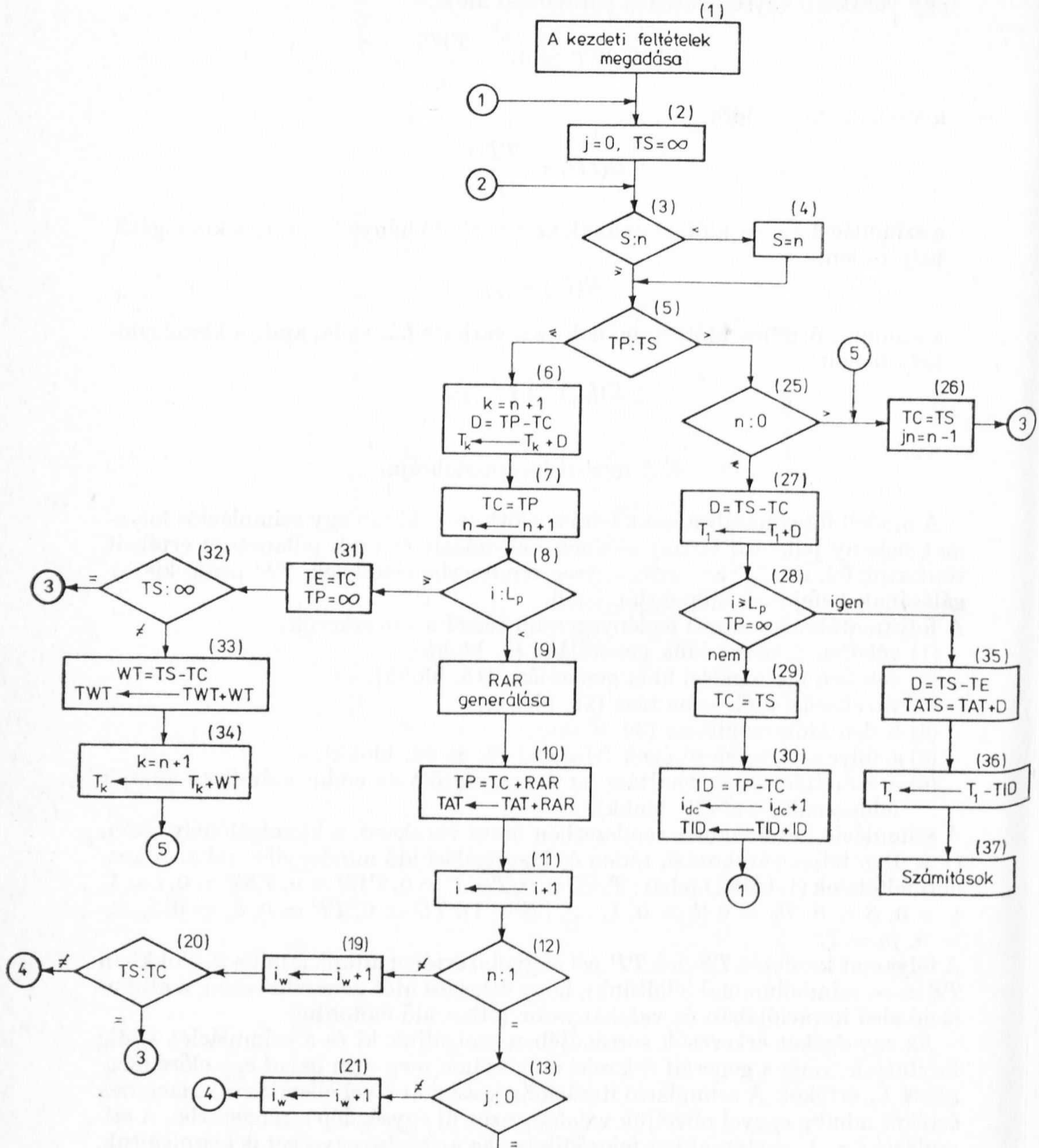
A folyamatábra (2. ábra) leglényegesebb részei a következők:
(1) véletlen érkezési idők generálása (9. blokk),
(2) véletlen kiszolgálási idők generálása (15. blokk),
(3) várakozási idők számítása (22. blokk),
(4) tétlen idők számítása (30. blokk),
(5) a folyamat befejezésének feltételei (8. és 28. blokk),
(6) a számítások végrehajtása az összegyűjtött és eddig számított adatok felhasználásával (37. blokk).

A szimuláció kezdetekor a rendszerben nincs várakozó, a kiszolgálóhely tétlen ($j = 0$), a teljes várakozási, tétlen és kiszolgálási idő mindegyike nulla. A kezdeti feltételek (1. blokk) tehát: $TAT = 0, TWT = 0, TID = 0, TST = 0, i = 1, n = 0, S = 0, T_k = 0$ $(k = 0, 1, \ldots, S + 1)$, $TC = 0, TP = 0, i_w = 0, i_{dc} = 0, jn = 1$.

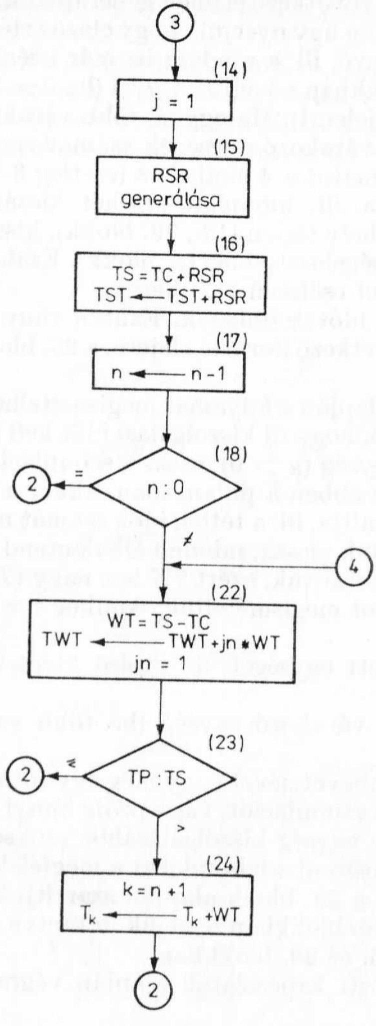
A folyamat kezdetén TS-nek TP-nél nagyobb értéket adunk (amit a 2. blokkban $TS = \infty$ szimbólummal jelöltünk), hogy érkezési időt generálhassunk a szimuláció első iterációjában és valahányszor tétlen idő előfordul.

Az egységeket érkezésük sorrendjében szolgáljuk ki és a szimulációt addig folytatjuk, amíg a generált érkezési idők száma meg nem halad egy előre megadott L_p értéket. A szimuláció iterációinak számát i-vel ellenőrizzük, melynek értékét mindig eggyel növeljük valahányszor új egység lép a rendszerbe. A szimuláció $i = L_p$ esetén akkor fejeződik be, ha az utolsó egységet is kiszolgáltuk.

Ha új egység érkezik a rendszerbe, a következő egység érkezési időpontját is szimuláljuk úgy, hogy az I. valószínűségeloszlásból mintát (TP) veszünk. Hasonlóan, ha valamely egység kiszolgálása megkezdődik, a kiszolgálás befejezési időpontját (TS) is meghatározzuk a II. valószínűségeloszlást szimuláló generátor segítségével. A folyamat bármely stádiumában a soronkövetkező esemény



2. ábra
(Folytatása a következő oldalon.)



bekövetkezését az fogja meghatározni, hogy az előbbi két sztochasztikus érték (TP, ill. TS) közül melyik a kisebb. Más szavakkal, a blokkdiagramot úgy állítottuk össze, hogy a következő érkezési időt csak akkor generáljuk, ha a teljes kiszolgálási és tétlen idők pillanatnyi értékeinek összege nagyobb vagy egyenlő mint TP aktuális értéke, azaz

$$TST + TID \geq TP.$$

Ha azonban $TST + TID < TP$, akkor kiszolgálási időt generálunk.

A szimuláció folyamán minden alkalommal kiszámítjuk a T_k értékeit (tehát azt az időt, amelyet k egység tölt a rendszerben) valahányszor új egység lép a rendszerbe, ill. megkezdődik vagy befejeződik egy kiszolgálás (6., 24., 27. és 34. blokk).

Mielőtt az érkezési idő következő értékét generálnánk TC felveszi TP aktuális értékét, TP új értékét pedig úgy nyerjük, hogy előző értékét megnöveljük RAR-rel. A várakozó sorban levő, ill. a rendszerbe már beérkezett egységek számát (n, ill. i) a 7., ill. a 11. blokkban növeljük 1-gyel (ha $i = 1$, RAR az első és második érkezés közötti időt jelenti). Ha egynél több várakozó egység van, vagy a kiszolgálóhely foglalt, a várakozó egységek számát eggyel növeljük (19. vagy 21. blokk), majd a folyamatot a 4 címkéhez (esetleg 3-hoz) vezéreljük, hogy a várakozási idők aktuális, ill. kumulált értékét kiszámíthassuk (22. blokk). Ha $n = 1$ és a kiszolgálóhely tétlen (12., 13. blokk), kiszolgálási időt (RSR) generálunk a II. valószínűségeloszlásból (15. blokk). Ezalatt a várakozó egységek aktuális száma (n) eggyel csökken (17. blokk).

Ha $n > 0$, várakozási időt számítunk. Ezután vagy közvetlenül, vagy közvetve visszatérünk a következő iteráció elejére, a 23. blokkban megadott döntés szerint.

Az 5. (döntési) blokk alapján a folyamat megismételhető, vagy a 25. blokkhoz vezérelhető. Nyilvánvaló, hogy új kiszolgálási időt kell generálnunk, ha a rendszerben van várakozó egység ($n > 0$), ezért a szimulációt a 3 címkénél folytatjuk. Ha nincs várakozás ebben a pillanatban, akkor a program az aktuális és kumulált tétlen időt számítja, ill. a tétlen idők számát növeli eggyel (30. blokk).

Ha az 1 címkéhez térünk vissza, minden alkalommal $TS < TP$, egy kivételével az összes érkezőt kiszolgáltuk, ezért TS-hez nagy (TP-nél nagyobb) értéket rendelünk és a folyamatot megismételjük. Amikor $i = L_p$, a rendszer két állapotban lehet:

(1) az összes beérkezett egységet, az utolsó kivételével, már kiszolgáltuk, vagy

(2) van legalább egy várakozó egység (ha több van, akkor kiszolgálás is folyik).

E két lehetőség figyelembevételével vagy még egy (utolsó) kiszolgálási időt generálunk és befejezzük a szimulációt, vagy pedig annyi kiszolgálási időt generálunk, amennyi az összes egység kiszolgálásához szükséges. A 31. és 32. blokk segítségével tudjuk biztosítani a feltételeket a megfelelő befejezéshez. A szimuláció teljes időtartamát a 35. blokk alapján számítjuk.

T_1 korrekt értékét a 36. blokkban nyerjük, tekintve, hogy a szimuláció során kétszer jelenik meg (a 6. és 30. blokkban).

Végül, a változók közti kapcsolatok alapján végrehajtjuk a számításokat (37. blokk).

5. A modell érvényességének vizsgálata

A modell érvényességét abban az esetben ellenőriztük, amikor az érkezések közti időintervallumok és a kiszolgálási idők adott várható értékekkel rendelkező exponenciális eloszlást követnek. Leggyakrabban ilyen eloszlásokkal találkozunk. A rendszerjellemzők analitikus értékei ebben az esetben könnyen hozzáférhetőek. Ismeretes ugyanis, hogy ha egy meghatározott idő alatt beérkezett egységek száma Poisson eloszlású, akkor az érkezések közötti időtartamok exponenciális eloszlást követnek. A t idő alatt bekövetkezett érkezések közötti időintervallumok eloszlásának sűrűségfüggvénye $\lambda e^{-\lambda t}$, ahol λ az átlagos érkezési ráta, és $1/\lambda$ két érkezés közti átlagos időtartam. Hasonló sűrűségfüggvénnyel jellemezhetjük a t idő alatti kiszolgálási időket is, amelyet a μ paraméterrel adunk meg. μ az átlagos kiszolgálási rátát, $1/\mu$ pedig az egy kiszolgálásra eső átlagos kiszolgálási időt jelenti.

A korábban bevezetett jelöléseket használva, $\lambda/\mu < 1$ esetén, a megfelelő elméleti értékek a következő egyenletekkel számíthatók [14, 131–133. o.], ahol az alsó *st* index a megfelelő statisztikáknak a tárgyalt sorbanállási rendszer stacionárius állapotában felvett elméleti értékeire utal:

$$E(k)_{st} = \frac{\lambda}{\mu - \lambda}$$

$$E(n)_{st} = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

$$E(w|w > 0)_{st} = \frac{\mu}{\mu - \lambda}$$

$$E(T_{ws})_{st} = \frac{1}{\mu - \lambda}$$

$$E(WT)_{st} = \frac{\lambda}{\mu(\mu - \lambda)}$$

1. táblázat

	$\psi = 0{,}22$			$\psi = 0{,}4$		
	Elméleti	Szimulált		Elméleti	Szimulált	
		$L_p = 200$	$L_p = 1200$		$L_p = 200$	$L_p = 1200$
$E(k)$	0,29	0,31	0,30	0,67	0,62	0,70
$E(n)$	0,06	0,06	0,07	0,27	0,23	0,30
$E(w\|w > 0)$	1,29	1,13	1,28	1,67	1,43	1,92
$E(T_{ws})$	2,86	2,48	2,52	2,22	1,69	2,08
$E(WT)$	0,64	0.20	0,27	0,89	0,43	0,72
$E(WT\|WT > 0)$	2,86	0,81	1,21	2,22	1,09	1,83
$E(f_{id})$	0,78	0,75	0,77	0,60	0,61	0,60
$E(f_{occ})$	0,22	0,25	0,23	0,40	0,39	0,40
Valószínűségek (p_n)						
n: 1	0,7778	0,7502	0,7680	0,6000	0,6125	0,6010
2	0,1729	0,1951	0,1790	0,2400	0,2235	0,2425
3	0,0384	0,0483	0,0418	0,0960	0,1102	0,0923
4	0,0085	0,0057	0,0082	0,0384	0,0404	0,0327
5	0,0019	0,0007	0,0025	0,0154	0,0105	0,0145
6	0,0004	—	0,0004	0,0061	0,0029	0,0067
7	0,0001	—	0,0001	0,0024	—	0,0037
8				0,0010	—	0,0008
9				0,0004	—	0,0019
10				0,0002	—	0,0018
11				0,0001	—	0,0003
12						0,0011
13						0,0007

$$E(ET|WT > 0)_{st} = \frac{1}{\mu - \lambda}$$

$$E(f_{occ})_{st} = \frac{\lambda}{\mu}$$

$$E(f_{id})_{st} = 1 - \frac{\lambda}{\mu},$$

és az a valószínűség, hogy k számú egység van a rendszerben:

$$P_{k,st} = \left(1 - \frac{\lambda}{\mu}\right) \cdot \left(\frac{\lambda}{\mu}\right)^k, \quad k \geq 0.$$

A λ és μ paraméterek értékeit változtatva a szimuláció útján nyert és az analitikusan számított jellemző mennyiségeket összehasonlítottuk, majd a szimulá-

2. táblázat

	$\varphi = 0{,}47$			$\varphi = 0{,}65$		
	Elméleti	Szimulált		Elméleti	Szimulált	
		$L_p = 200$	$L_p = 1200$		$L_p = 200$	$L_p = 1200$
$E(k)$	0,88	0,80	0,95	1,83	1,93	1,84
$E(n)$	0,41	0,35	0,49	1,18	1,28	1,20
$E(w\|w > 0)$	1,88	1,56	2,21	2,83	2,87	2,95
$E(T_{ws})$	1,25	0,96	1,24	4,35	4,56	4,22
$E(WT)$	0,58	0,33	0,56	2,81	2,94	2,64
$E(WT\|WT > 0)$	1,25	0,71	1,20	4,35	4,77	4,10
$E(f_{id})$	0,53	0,55	0,54	0,35	0,36	0,36
$E(f_{occ})$	0,47	0,45	0,46	0,65	0,64	0,64
Valószínűségek (p_n)						
n: 1	0,5333	0,5481	0,5345	0,3539	0,3550	0,3650
2	0,2489	0,2299	0,2462	0,2286	0,1982	0,2269
3	0,1161	0,1296	0,1110	0,1477	0,1286	0,1536
4	0,0542	0,0676	0,0509	0,0955	0,1073	0,0881
5	0,0253	0,0192	0,0237	0,0617	0,0574	0,0563
6	0,0118	0,0045	0,0143	0,0399	0,0535	0,0316
7	0,0055	0,0011	0,0064	0,0258	0,0560	0,0238
8	0,0026	—	0,0010	0,0166	0,0339	0,0131
9	0,0012	—	0,0033	0,0108	0,0101	0,0122
10	0,0006	—	0,0025	0,0070	—	0,0059
11	0,0003	—	0,0024	0,0045	—	0,0071
12	0,0001	—	0,0010	0,0029	—	0,0052
13	0,0001	—	0,0023	0,0019	—	0,0067
14			0,0005	0,0012	—	0,0023
15				0,0008	—	0,0008
16				0,0005	—	0,0005
17				0,0003	—	0,0009
18				0,0002	—	—
19				0,0001	—	—
20				0,0001	—	—

ciót megismételtük a véletlenszám generátorok különböző kezdeti értékei mellett. Ezután a modellt kipróbáltuk L_p eltérő (200 és 1200) értékei esetén is.

Az iterációk számának (L_p) növelésével az eredmények egyre jobban közelítik az analitikus értékeket, és a modell a paraméterek minden értékénél jó eredményeket szolgáltat. Ezek közül néhányat az 1. és 2. táblázatban közöltünk.

6. Összefoglalás

A modell felépítése során először a szimulációhoz szükséges változókat adtuk meg és a matematikai modellt definiáltuk, amelyet blokk diagram formájában állítottunk elő. A blokk diagram szubrutinként realizálva alkalmas a sorbanállási rendszert leíró sztochasztikus mennyiségek meghatározására — a szimuláció eszközeivel. A közölt részletes blokk diagram alapján a modell könnyen programozható és a program elvileg bármilyen számítógépen futtatható.

A cikkben kitértünk arra, hogy a modell jellemzői bizonyos feltételek esetén analitikus módszerekkel is meghatározhatók, azonban a szimuláció módszere sokkal általánosabb, nem csak speciális feltételek mellett alkalmazható. Ez a módszer kiterjeszthető olyan esetekre, amikor

— a szimulációval nyert és az analitikus értékeket össze akarjuk hasonlítani,
— a bemenő adatok nem illeszkednek semmiféle elméleti valószínűségeloszlásra,
— bonyolult, analitikus módszerekkel le nem írható sorbanállási modelleket kívánunk létrehozni és jellemezni.

7. Kiegészítések

Egyenletes eloszlású pszeudo-véletlen számok előállítására a *Multiplikatív Kongruencia Módszert* alkalmaztuk az

$$x_{i+1} \equiv ax_i \pmod m$$

kongruencia felhasználásával, ahol x_i, a, m és x_{i+1} pozitív egészek és m nagyobb mint x_i, a és x_{i+1}. A fenti kongruenciából nyert sorozatot az x_i/m transzformációval a $[0,1)$ intervallumra képeztük le. Maximális periódust akkor kapunk, ha

(1) $m = 2^\alpha$;
(2) az x_0 kezdő érték m-hez relatív prim;
(3) a az m-hez relatív prim;
(4) $a \equiv \pm 3 \pmod 8$, azaz $a = 8t \pm 3$, ahol t pozitív egész;
(5) a-nak $2^{\alpha/2}$ közelében kell lennie. a ily módon választott értéke kielégíti a COVEYOU [4] — GREENBERGER [10] feltételt.

Ezen feltételek mellett elérhető maximális periódus $p = 2^{\alpha-2}$ (HULL és DOBELL [11]).

Egyrészt a fenti meggondolások miatt, másrészt mivel e dolgozat megírásához használt programokat 24-bites szavakkal rendelkező számítógépen futtattuk, a kezdeti értékeket az alábbiak szerint választottuk:

$m = 2^{23}$
$x_0 = 2^{23} - 1$ (vagy bármely páratlan szám)
$a = 2^{11} + 3 + 8t$ (ahol t egész és $0 \leq t \leq 64$).

m, x_0 és a ilyen értékei mellett a periódus hossza $p = 2^{21}$.

A fenti módszer, amelyet számos statisztikai teszttel is ellenőriztünk, mindig megbízható eredményeket szolgáltatott.

A módszer alkalmazásával létesített véletlenszám-generátorokat használva nyertünk sztochasztikus változókat *inverz transzformációs technika* [18] felhasználásával. Ha ugyanis egy statisztikai populáció $F(x)$ eloszlásfüggvénye explicit formában létezik, ezzel a technikával valószínűségi változók generálhatók a populációból, ha ismételten használjuk az $x_i = F^{-1}(u_i)$ relációt, ahol u_i egyenletes eloszlású pszeudo-véletlen számokat jelent a $[0, 1)$ intervallumon. *Exponenciális* vagy *Poisson-eloszlás* esetén

$$x_i = -\left(\frac{1}{\lambda}\right)\log_e u_i$$

vagy

$$\prod_{i=0}^{x} u_i \geq e^{-\lambda} > \prod_{i=0}^{x+1} u_i \quad (x > 0, \text{ egész})$$

és így u_i minden értékéhez x_i és x egyetlen értéke tartozik, melyeknek sűrűségfüggvénye

$$\lambda e^{-\lambda x} \text{ vagy } \lambda^x e^{-\lambda}/x\,!$$

(Beérkezett: 1984. május 15-én.)

IRODALOM

1. Ahrens, J. H.—Dieter, U.: Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing* 12 (1974) 223—246.
2. Ashour, S.—Jha, R. D.: Numerical transient-state solutions of queueing systems. *Simulation* 21 No. 4., Oct. 1973.
3. Békéssy, A.: Remarks on beta distributed random numbers. *MTA Matematikai Kutató Intézet Közleményei* 9 (1964).
4. Coveyou, R. R.: Serial correlation in the generation of pseudorandom numbers. *Journ. of Assoc. for Comp. Mach.* 7 (1960).
5. Deák I.: *Véletlenszám generátorok* Tanulmány az ELTE TTK Valószínűségszámítási Tanszék részére (1977).
6. Deák I.: *Monte Carlo módszerek a többdimenziós térben elhelyezkedő halmazok valószínűségének meghatározására normális eloszlás esetén* Kandidátusi értekezés, MTA TMB 1980.
7. Deák I.: *Egyenletes eloszlású véletlen számok generálása* MTA SZTAKI Working Paper MO/29, 1981.
8. Deák, I.: An economical method for random number generation and a normal generator. *Computing* 27/1981 113—121. o.
9. Dieter, U.: Pseudo-random numbers: The exact distribution of pairs. *Mathematics of Computation* 25 No. 116., Oct. 1971.
10. Greenberger, M.: Notes on a new pseudo-random number generator. *J. Assoc. Comp. Mach.* 8 (1961) 163—167.
11. Hull, T. E.—Dobell, A. R.: Mixed random number generators for binary machines. *J. Assoc. for Comp. Mach.* 11 (1964) 31—40.
12. Iglehart, D. L.: Simulating stable stochastic systems VII: Selecting the best system. *TIMS Studies in the Management Sciences* 7 (1977) 37—49., North-Holland Publ. Co.
13. Iglehart, D. L.—Shedler, G. S.: *Regenerative simulation of response time in networks of queues.* Springer-Verlag, 1980.
14. Kaufmann, A.: *Methods and models of operations research.* Englewood Cliffs, N. J., Prentice-Hall, 1963.
15. Lehmer, D. H.: Mathematical methods in large-scale computing units. *Annals. Computer Labor.*, Harward Univ. XXVI (1951).

16. MACLAREN, M. D.—MARSAGLIA, G.: Uniform random number generators. *Journal of the ACM* XII. No. 1. 1965.
17. NAYLOR, T. H.: *Computer simulation experiments with models of economic systems.* J. Wiley, 1971.
18. NAYLOR, T. H.—BÁLINTFY, J. L.—BURDICK, D. S.—CHU, K.: *Computer simulation techniques.* J. Wiley, 1966.
19. NÉMETH GY.: Tömegkiszolgálás szimulációja sorbakapcsolt kiszolgálóhelyek esetén *Alkalm. Mat. Lapok* 5 (1979) 103—122.
20. PANICO, J. A.: *Queueing theory.* Prentice-Hall, Inc., 1969.
21. TOCHER, K. D.: *The art of simulation.* D. Van Nostrand Co. Princeton, N. J., 1967.

SIMULATION OF A QUEUING SYSTEM WITH ONE SERVICE STATION

The model is aimed at elaborating the simulation of a queuing system with a single service station and one waiting line and, respectively, presenting some experiments simulating the behaviour of the system and connected experiences. The paper has been made not with the aim of publishing any new scientific achievement, however, with submitting a mathematical and computational model of the outlined system easy to survey methodological guidelines are provided and application is made easier.

In the course of simulation it was supposed that units arrive one by one and the "first come first served" principle is enforced. Changes in intervals between arrival times follow the same probability distribution (distribution I) in the course of simulation, service times also follow a determined probability distribution (distribution II), furthermore, both distributions are stationary in time and independent of each other.

The two probability distributions (arrivals and services) characteristic of the system are obtained by random number generators. Firstly Monte-Carlo technique is applied for the generation of random numbers with even distribution and the stochastic variables fitting to the required probability distribution, respectively, then these generators are attached to the model in such a way that the stochastic process will be simulated. Random number generators appear in the form of sub-routines and input parameters are changed in the course of simulation depending on the special distribution required by the model. The application of this particular simulation technique makes the simulated system relatively easy to understand. On the basis of the submitted detailed block diagram the model is easily programmable and the programme may be run through any kind of computer in principle.

Characteristics of the model may in case of certain constraints be determined also with analytical methods, but the method of simulation is much more general and may be applied not only with special constraints. The method may be extended also to cases when we wish to compare analytical values with those obtained by simulation, input data do not fit to any theoretical probability distribution or we wish to establish and characterize complicated queuing models that cannot be described with analytical methods.

СИМУЛЯЦИЯ МОДЕЛИ ОЧЕРЕДНОСТИ С ОДНИМ МЕСТОМ ОБСЛУЖИВАНИЯ

Целью модели является симуляция системы с одним местом обслуживания и одним рядом ожидания, а также изложение опыта и нескольких экспериментов симуляции поведения системы. Статья не содержит новых научных результатов, а дает с помощью изложенной в ней вычислительной и математической модели системы методологические рекомендации, облегчая тем самым их применение.

В ходе симуляции предполагается, что единицы поступают по одной, и первыми те единицы обслуживаются, которые поступили первыми. Изменения временных интервалов между моментами поступления имеют одинаковое распределение вероятности (I. распределение) в процессе симуляции, время обслуживания имеет также определённое распределение вероятности, и оба распределения являются стационарными во времени и независимы друг от друга.

Две характерных для системы распределения вероятности (поступление и обслуживание) составляются с помощью генератора случайных чисел. Для составления стохастических переменных с нужным распределением вероятности и равномерно распределённых

случайных чисел применяется метод Монте-Карло, позже эти генераторы так подключены к модели, чтобы они симулировали стохастический процесс. Генераторы случайных чисел разработаны в виде подпрограмм и в процессе симуляции их входные параметры изменяются в зависимости от того, какое распределение требуется данной моделью. Применение такой техники симуляции позволяет относительно легко понять симулированную систему. С помощью данной подробной блокдиаграммы модель легко программируема, и программа может быть использована в принципе на любой вычислительной машине.

При наличии определённых условий свойства модели могут быть определены и аналитическими методами, однако метод симуляции более общий и может применяться не только при наличии специальных условий. Метод может быть распространен и на те случаи, когда нужно сравнить аналитические результаты с симуляционным, когда входные данные не имеют определённого теоретического распределения вероятности, а также когда нужно моделировать сложные модели очередности, которые нельзя описать с помощью аналитических методов.