

**INSTITUTUL DE MATEMATICĂ ȘI INFORMATICĂ AL A.Ș.M.
UNIVERSITATEA ACADEMIEI DE ȘTIINȚE A MOLDOVEI**

Cu titlu de manuscris

C.Z.U: 519.872

MITEV LILIA

**MODELE POLLING CU PRIORITĂȚI, VACANȚE
SEMI-MARKOVIENE ȘI SERVIRE EXHAUSTIVĂ**

**112.03 - CIBERNETICĂ MATEMATICĂ ȘI CERCETĂRI
OPERAȚIONALE**

Teză de doctor în științe matematice

Conducător științific:

Mișcoi Gheorghe,
academician al A.Ș.M.,
doctor habilitat în științe fizico-
matematice, profesor universitar

Consultant științific:

Attahiru Sule Alfa, doctor, profesor,
Universitatea Manitoba, Canada

Autorul:

CHIȘINĂU, 2017

© Mitev Lilia, 2017

CUPRINS

ADNOTĂRI	5
LISTA ABREVIERILOR.....	8
INTRODUCERE	10
1. ANALIZA SITUAȚIEI ÎN DOMENIUL MODELELOR POLLING	18
1.1. Evoluția istorică în studierea modelelor Polling și a modelelor generalizate cu priorități .	18
1.2. Aplicații ale sistemelor de așteptare în diverse domenii	24
1.3. Reguli de servire a sistemelor de așteptare	31
1.4. Tipuri de modele de așteptare	35
1.5. Caracteristici de performanță ale sistemelor de așteptare	41
1.6. Concluzii la capitolul 1	43
2. METODE ȘI CARACTERISTICI DE PERFORMANȚĂ ÎN EVOLUȚIA MODELELOR POLLING CU SERVIRE EXHAUSTIVĂ ȘI VACANȚE	46
2.1. Metode analitice: metoda funcțiilor generatoare și metoda catastrofelor	46
2.2. Metode numerice: metoda aproximațiilor succesive și metoda modelărilor numerice.....	49
2.3. Repartiția virtuală și staționară a lungimii șirului de așteptare pentru modelul Polling cu servire exhaustivă.....	52
2.4. Algoritmi și modelări numerice ale repartiției lungimii șirului de așteptare	61
2.5. Concluzii la capitolul 2	69
3. CARACTERISTICI DE PERFORMANȚĂ ALE MODELELOR GENERALIZATE DE AȘTEPTARE CU PRIORITĂȚI	70
3.1. Concepte generale ale modelelor generalizate cu priorități	70
3.2. Concepte de strategii în stare liberă	74
3.3. Repartiția perioadelor de ocupare și caracteristici auxiliare ale modelelor Polling.....	79
3.4. Disciplina DD. Repartiția perioadei de ocupare și a caracteristicilor auxiliare	81
3.5 Repartiția lungimii șirului de așteptare pentru prioritatea DD.....	91
3.6. Algoritmi și modelări numerice de calcul ale perioadei de ocupare și a caracteristicilor	

auxiliare pentru Disciplina DD	94
3.7. Concluzii la capitolul 3	106
CONCLUZII GENERALE ȘI RECOMANDĂRI	107
BIBLIOGRAFIE	109
ANEXE	117
ANEXA 1. Listingul programelor pentru determinarea repartiției lungimii șirului de așteptare pentru modelul Polling cu servire exhaustivă	117
ANEXA 2. Listingul programelor pentru determinarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemul generalizat cu prioritatea DD	126
DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII	141
CURRICULUM VITAE	142

ADNOTARE
la teza de doctor a dnei Mitev Lilia
“Modele Polling cu priorități, vacanțe semi-Markoviene și
servire exhaustivă”

Teza este înaintată pentru obținerea titlului de doctor în științe matematice la specialitatea 112.03 - Cibernetică matematică și cercetări operaționale. Ea a fost elaborată la Universitatea Academiei de Științe a Moldovei și Institutul de Matematică și Informatică al Academiei de Științe a Moldovei, Chișinău, în anul 2016.

Structura tezei. Teza este scrisă în limba română și conține următoarele compartimente: introducere, trei capitole, concluzii generale și recomandări, bibliografie ce constă din 103 titluri și 2 anexe. Lucrarea conține 108 pagini de text de bază. Rezultatele obținute sunt publicate în 30 lucrări științifice.

Cuvintele-cheie: model Polling, sisteme generalizate cu priorități, transformata Laplace-Stieltjes, lungimea medie virtuală, perioada de ocupare, prioritatea DD.

Domeniul de studiu al tezei: Teoria Așteptării.

Scopul și obiectivele lucrării. Lucrarea are ca scop extinderea rezultatelor cunoscute din domeniul Teoriei Așteptării, elaborarea a noi tehnici și algoritmi numerici de determinare a unor caracteristici de performanță mai optime pentru modelele de așteptare Polling cu vacanțe semi-Markoviene și pentru cele cu prioritatea DD.

Pentru atingerea scopului propus s-au realizat următoarele obiective:

- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției lungimii virtuale a șirului de așteptare pentru sistemele Polling cu întârzieri semi-Markoviene;
- formalizarea caracteristicilor probabiliste de performanță pentru sistemele generalizate de așteptare cu prioritatea DD;
- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele Polling cu prioritatea DD;
- implementarea algoritmilor elaborați în limbaje de programare în vederea estimării parametrilor funcțiilor de repartiție ce intervin în optimizarea caracteristicilor de performanță ale modelelor de așteptare Polling.

Noutatea și originalitatea științifică: constă în elaborarea metodelor și algoritmilor numerici pentru determinarea unor caracteristici numerice de performanță pentru sistemele Polling, cât și pentru cele cu prioritatea DD. Astfel se poate stabili eficiența/performanța sistemului în dependență de legile de repartiție și de prioritate, strategia sistemului în stare liberă.

Problema științifică importantă soluționată: rezidă în determinarea unor valori mai optime ale caracteristicilor probabiliste pentru modelele Polling, rezultatele obținute atât în urma analizei modelelor de așteptare, cât și a funcțiilor de repartiție, legilor de prioritate, schemelor de servire și orientare, fapt care permite stabilirea staționarității și eficienței sistemului de așteptare.

Semnificația teoretică. Rezultatele obținute în teză pot fi utilizate pentru studiul altor sisteme reale, unde au loc fenomene de așteptare și determinarea altor caracteristici numerice..

Valoarea aplicativă a lucrării. Rezultatele prezentate permit aplicarea în diverse sfere, unde apar fenomene similare celor studiate, cum ar fi sistemele informatice, de telecomunicații, economice etc., care pot fi modelate matematic cu ajutorul modelelor studiate în teză.

Implementarea rezultatelor științifice. Algoritmii elaborați au fost implementați în limbajele de programare C++ și Kotlin.

АННОТАЦИЯ
на диссертацию Митев Лилии
“Модели Поллинг с приоритетами, полу-Марковскими задержками и
неограниченным обслуживанием”

Диссертация представлена на соискание учёной степени доктора математических наук по специальности 112.03 - Математическая Кибернетика и Операционные Исследования и выполнена в Университете Академии Наук Молдовы и Институте Математики и Информатики Академии Наук Молдовы, Кишинёв, 2016.

Структура работы. Диссертация написана на румынском языке и содержит: введение, три главы, выводы, библиографию, состоящую из 103 наименований, и два приложения. Основной текст диссертации написан на 108 страницах. Полученные результаты опубликованы в 30 научных работах.

Ключевые слова: модель Поллинг, обобщенные приоритетные системы, преобразование Лапласа-Стилтьеса, средняя виртуальная длина, период занятости, приоритет DD.

Область исследования: Теория массового обслуживания.

Цель исследования. Диссертация призвана расширить известные результаты из Теории массового обслуживания, а также разработку новых методов и численных алгоритмов для определения оптимальных характеристик для моделей ожидания Поллинг с полу-Марковскими задержками и для тех с приоритетом DD.

Для достижения намеченной цели были достигнуты следующие задачи:

- разработка и применение численных алгоритмов для моделирования распределения виртуальной длины очереди для моделей Поллинг с полу-Марковскими задержками;
- формализация вероятностных характеристик обобщенных систем для дисциплины с приоритетом DD;
- разработка и применение численных алгоритмов для моделирования распределения периода занятости и вспомогательных характеристик для системы Поллинг с приоритетом DD;
- внедрение разработанных алгоритмов в языках программирования для оценки параметров функций распределения, которые присутствуют в оптимизации вероятностных характеристик в моделях ожидания Поллинг.

Научная новизна и оригинальность: заключается в разработке методов и численных алгоритмов для определения численных характеристик для системы Поллинг, с полу-Марковскими задержками и с приоритетом DD. Это позволяет определить эффективность/ производительность системы ожидания в зависимости от законов распределения, законами приоритета и стратегии системы в свободном состоянии.

Решение важной научной проблемы: заключается в определении оптимальных значений вероятностных характеристик для моделей Поллинг, получение новых результатов исходящие из анализа моделей ожидания, а также функций распределения, приоритетные законы, схем обслуживания и ориентации, что позволяет установить стационарность и эффективность систем.

Теоретическая ценность. Полученные результаты могут быть использованы для изучения систем с ожиданием и для определения других численных характеристик.

Практическая ценность. Результаты могут быть применены в сферах, где происходят аналогичные явления, как компьютерные системы, телекоммуникации и т.п., которые могут быть математически смоделированы с использованием изученных моделей.

Внедрение результатов. Разработанные алгоритмы были реализованы на языках программирования C++ и Kotlin.

ANNOTATION
for PhD thesis by Mitev Lilia
“Polling models with priorities, semi-Markov vacations
and exhaustive service”

This thesis is submitted to obtain a doctoral degree in mathematics, at the specialty 112.03 - Mathematical cybernetics and operational research. It was elaborated at University of the Academy of Sciences of Moldova and Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova, in Chisinau, 2016.

Thesis structure. The thesis is written in Romanian and contains the following sections: introduction, three chapters, conclusions, bibliography consisting of 103 titles and two annexes. The text of thesis comprises 108 pages. The results of thesis are published in 30 scientific papers.

Keywords: Polling model, generalized queueing models with priorities, Laplace-Stieltjes transform, virtual queue length, busy period, DD priority.

The field of study of the thesis: Queueing theory.

The aim of research. The paper aims to extend the known results from Queueing theory, the development of new techniques and numerical algorithms for determining optimal performance characteristics for Polling models with semi-Markov vacations and for those with DD priority.

To achieve the intended goal has been achieved the following objectives:

- elaboration and application of numerical algorithms for modeling the distribution of virtual queue length of the Polling systems with semi-Markov vacations;
- formalization of probabilistic performance characteristics for generalized queueing systems with DD priority;
- development and application of numerical algorithms for modeling the distribution of busy period and of auxiliary characteristics for the Polling systems with DD priority;
- implementation of developed algorithms in programming languages in order to estimate parameters of distribution functions that occur in optimization of performance characteristics of the Polling queueing models.

Scientific novelty and originality: consists in the elaboration of methods and numerical algorithms for determination of numerical performance characteristics for Polling systems, and for those with DD priority. Thus it can determine the efficiency / performance of the queueing system depending on distribution laws, priority laws, strategy of the system in the free state.

The important scientific solved problem: resides in the determination of more optimal values of probabilistic characteristics for Polling models, the obtained new results both from analysis queueing models and distribution functions, priority laws, schemes of service and orientation, which allows the establishment stationarity and efficiency of the queueing system.

The theoretical significance. The obtained results from the thesis can be used to study some real systems where queueing phenomena occur and assessment of other numeric characteristics.

Applicative value of the thesis. The presented results permit application in various spheres, where queueing phenomena occur, similar to studied, such as computer systems, telecommunications, economic, etc., which can be modeled using studied mathematical models.

The implementation of the scientific results. The developed algorithms were implemented in programming languages C++ and Kotlin.

LISTA ABREVIERILOR

ISP - furnizorul de servicii Internet;

CoS - clasă a servirii;

QoS - calitate a servirii;

MAC - Control al Accesului la Mediu;

LAN - Rețea Locală;

EPON - Rețea Optică Pasivă Ethernet;

OLT - Terminal de Linie Optică;

ONU - Unitatea de Rețele Optice;

TCP - Protocol de Control al Transportului;

CDMA - Acces Multiplu cu Divizare a Codului;

FWLAN - Rețea locală fără fir bazată pe Feribot;

MANET - Rețea mobilă Adhoc;

FIFO - primul venit, primul ieșit;

LIFO - ultimul intrat, primul ieșit;

DD - disciplina Discretionary;

TLS - transformata Laplace-Stieltjes;

$\text{Exp}(\lambda)$ - funcția de repartiție exponențială cu parametrul λ ;

$\text{Erl}(\lambda, k)$ - funcția de repartiție Erlang cu parametrul λ de ordinul k ;

$U(a, b)$ - funcția de repartiție uniformă pe segmentul $[a, b]$;

$\text{Erl}(\lambda, k)$ - funcția de repartiție Erlang de parametri λ și k ;

$N(a, \sigma^2)$ - funcția de repartiție normală de parametri a și σ^2 ;

- Pentru sistemele de așteptare Polling cu întârzieri semi-Markoviene și servire exhaustivă:

λ_k - parametrul fluxului Poisson al cerințelor de clasă k ;

$B_k(x)$ - funcția de repartiție a timpului de servire al cerințelor de clasă k ;

$C_k(x)$ - funcția de repartiție a timpului de orientare către șirul de cerințe de clasă k ;

$\Pi_k^\delta(x)$ - funcția de repartiție a k -perioadei de ocupare;

$L_k(t)$ - funcția de repartiție a valorii medii virtuale a lungimii șirului de așteptare pentru utilizatorul k ;

$\beta_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $B_k(x)$;

$c_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $C_k(x)$;

$\pi_k^\delta(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $\Pi_k^\delta(x)$;

$l_k(s)$ - transformata Laplace a funcției de repartiție $L_k(t)$;

- Pentru sistemele de așteptare generalizate cu priorități:

a_k - parametrul fluxului Poisson al cerințelor de clasă k ;

σ_k - parametrul fluxului sumă al cerințelor de clasă k ($\sigma_k = a_1 + \dots + a_k$);

$N_k(x)$ - funcția de repartiție a k -ciclului de orientare;

$H_k(x)$ - funcția de repartiție a k -ciclului de servire;

$\Pi_k(x)$ - funcția de repartiție a k -perioadei de ocupare;

$\Pi_{kk}(x)$ - funcția de repartiție a kk -perioadei de ocupare;

$\overline{\Pi}_k(x)$ - funcția de repartiție a $\overline{\Pi}_k$ - perioadei de ocupare;

$\Pi(x)$ - funcția de repartiție a perioadei de ocupare a sistemului;

$\nu_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $N_k(x)$;

$h_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $H_k(x)$;

$\pi_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $\Pi_k(x)$;

$\pi_{kk}(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $\Pi_{kk}(x)$;

$\overline{\pi}_k(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $\overline{\Pi}_k(x)$;

$\pi(s)$ - transformata Laplace-Stieltjes a funcției de repartiție $\Pi(x)$.

INTRODUCERE

Actualitatea și importanța problemei abordate. Teoria matematică a fenomenelor de așteptare, cunoscută ca Teoria Așteptării, este un compartiment al matematicii moderne ce ține de teoria probabilităților și cercetări operaționale. Acest domeniu a apărut din necesitățile practicii și pe parcursul dezvoltării lui a jucat un rol important în soluționarea unui larg spectru de probleme aplicative. Printre aceste probleme se consideră organizarea rațională și eficientă a centrelor telefonice, a clinicilor, magazinelor, întreprinderilor de prelucrare, de stocare a materialelor, a centrelor de apel public, etc. Odată cu apariția și dezvoltarea rapidă a diverselor rețele, modelele matematice din domeniul teoriei așteptării continuă să joace un rol important în modelarea, proiectarea și analiza funcționării rețelelor contemporane. Mai mult decât atât, dezvoltarea vertiginoasă a rețelelor contemporane, apariția unor noi tehnologii de rețea, cum ar fi, de exemplu, tehnologiile înzestrate cu metodologiile QoS (Quality of Service) și CoS (Class of Service) înaintază noi cerințe asupra elaborării și studierii a noi modele matematice, mai flexibile și mai adecvate proceselor reale. În acest aspect, anume în elaborarea unor noi modele matematice, cercetarea lor, obținerea caracteristicilor de performanță ale acestor modele, elaborarea algoritmilor numerici pentru modelarea matematică și elaborarea recomandărilor de aplicare a acestor rezultate la soluționarea problemelor reale este axată atenția cercetătorilor din domeniu.

Modelele Polling cu vacanțe semi-Markoviene și servire exhaustivă reprezintă modele matematice, modele care joacă un rol important în analiza, modelarea, proiectarea și optimizarea rețelelor contemporane. Aceste modele sunt larg răspândite, în special, în analiza funcționării rețelelor de bandă largă fără fir Wi-Fi și Wi-Max cu un control centralizat. Creșterea impunătoare a numărului de rețele și servicii electronice, evoluția splendidă a rețelelor cu fir și fără fir, toate aceste realizări sunt marcate printr-un schimb continuu de tehnologii de rețea orientate spre noi posibilități de integrare a datelor, voce și transport a informației video. Acest schimb și progres continuu de tehnologii înaintază noi cerințe și probleme științifice, printre care remarcăm și dezvoltarea de noi modele matematice, capabile să descrie în mod mai adecvat procesele complexe care apar în probleme reale. În domeniul teoriei sistemelor de așteptare o atenție deosebită este acordată dezvoltării unor algoritmi numerici pentru determinarea caracteristicilor probabilistice de bază ale sistemelor de așteptare investigate.

Sistemele de așteptare cu priorități constituie o clasă largă ale sistemelor de așteptare unde cerințele (cererile, mesajele, clienții, etc.) ce intră în sistem sunt distinse după importanța lor. Astfel de sisteme reprezintă modele adecvate ale multor aspecte ale vieții de zi cu zi, atunci

când o servire preferențială se acordă pentru anumite tipuri de cerințe. În sistemele de timp real, pierderile de timp numite vacanțe, pentru comutare (orientare, trecere, schimb) între clasele de prioritate sunt inevitabile. De aceea, este necesar ca aceste pierderi să fie analizate. Dacă în trecut, în timpul epocii clasice de dezvoltare a sistemelor de așteptare, se accepta omiterea proceselor de comutare, atunci în zilele noastre, acest lucru nu mai este posibil. Există două motive majore care confirmă această afirmație. Primul motiv este evocat de cerințele insistente ale practicii contemporane. Astfel, spre exemplu, apariția și evoluția diverselor rețele de comunicare, inclusiv rețele fără fir, care funcționează în regim de timp real și unde factorul stochastic este permanent prezent, gestionarea fluxurilor informaționale, diversificarea surselor de trafic, etc., - toate aceste fenomene conduc la crearea și studiul unor noi modele matematice și ingineresti a proceselor din aceste rețele. Aspectul noilor servicii de rețea și a noilor tehnologii, cum ar fi standardul IEEE 802.11 cu diferite modificări, care acceptă metodologia QoS, ne convinge de necesitatea de dezvoltare a modelelor matematice corespunzătoare cu aplicarea ulterioară a acestora modele într-o serie largă de probleme înaintate de practica contemporană. În al doilea rând, elaborarea și studierea modelelor cu priorități și timp nenul de comutare, menționate mai sus ca modele generalizate, prezintă un interes deosebit din punct de vedere teoretic - fundamental. Într-adevăr, datorită faptului că modelele generalizate care utilizează comutarea reprezintă generalizări (și anume, prin introducerea comutării) a modelelor clasice de prioritate, ne putem aștepta ca rezultatele analitice pentru astfel de sisteme generalizate să conțină, ca cazuri particulare, rezultatele corespunzătoare pentru sistemele clasice.

Regula generală de servire în sistemele de așteptare cu priorități este după cum urmează: cerințele care sunt în sistem și au o prioritate mai mare trebuie să fie servite înaintea acelor care au o prioritate mai mică. Cu toate acestea, în astfel de sisteme modul de comportare al serverului, în esență, le poate diversifica. În plus, există un număr considerabil de sisteme în care serverul are nevoie de ceva timp pentru a trece de la servirea unui tip de cerințe la alt tip de cerințe. Toate acestea oferă o mare varietate de sisteme de așteptare cu priorități. Conform acestor fenomene, descrierea și clasificarea sistemelor de așteptare cu priorități și timp de schimb al stărilor, numite modele generalizate, sunt prezentate în teză.

Vom menționa că pe parcursul evoluției teoriei așteptării, începând cu N. Jaiswal (autorul primei monografii în domeniul sistemelor cu priorități, publicate în 1973) și continuând cu cercetătorii din Republica Moldova Gh. Mișcoi, E. Guțuleac, A. Bejan, O. Benderschi, I. Damian, D. Bejenari, au fost formulate și soluționate diverse probleme importante, inclusiv din domeniul modelelor generalizate. Au fost stabilite unele concepte, principii, tehnici și algoritmi, de ordin general, conform cărora s-au continuat cercetările în această direcție.

În această ordine de idei, vom menționa un alt argument important al actualității și importanței problemei abordate, care reprezintă seria voluminoasă de lucrări științifice din acest domeniu atât din țară cât și peste hotare. Sistematizarea și generalizarea rezultatelor teoretice obținute în domeniul studierii sistemelor Polling până în anul 1985, sunt redată lucrarea lui H. Takagi [72]. Dezvoltarea rezultatelor teoretice în această direcție, publicate înainte de anul 1995, sunt expuse în monografia S. Borst [23], iar lucrări publicate în perioada anilor 1996-2006, sunt prezentate în lucrarea [92]. Generalizarea și sistematizarea modelelor și metodelor pentru studierea sistemelor stohastice cu sondaj ciclic (sistemul Polling) și utilizarea lor pentru proiectarea rețelelor fără fir de bandă largă este dedicată monografia [93]. Analiza modelelor cu priorități cu schimb nenul de tip semi-Markov al șirurilor de așteptare cu priorități, denumite modele generalizate cu priorități, este expusă în monografia [100]. În această lucrare sunt prezentate noi discipline de prioritate, sunt dezvoltate noi metode de analiză și elaborați algoritmi numerici de calcul al caracteristicilor sistemelor generalizate. În lucrările G. P. Klimov și G. K. Mișcoi [99], M. I. Volkovinskii și A. N. Kabalevskii [95] este studiat un caz special al sistemelor Polling, sisteme de așteptare cu priorități, cu flux de intrare Poisson și timp de orientare de o formă specială. Aceste monografii extind rezultatele obținute în acea perioadă și sunt prezentate în lucrarea B. V. Gnedenko ș.a. [96]. În lucrarea S. Alfa [13], sunt prezentate modelele de așteptare, cu privire la modul de utilizare a instrumentelor matematice disponibile la analiza problemelor asociate cu modelele de așteptare.

Printre lucrările științifice remarcabile și bine-cunoscute din acest domeniu sunt și lucrări ale cercetătorilor din Republica Moldova. Astfel, vom menționa Gh. Mișcoi și A. Bejan în [1], [65], [101], au propus un algoritm îmbunătățit de soluționare a ecuației clasice Kendall, care poate fi utilizat în algoritmi multidimensionali. Gh. Mișcoi și O. Benderschi [4], [16] au elaborat noi metode, tehnici și algoritmi numerici de evaluare a caracteristicilor numerice pentru modele generalizate cu priorități. Gh. Mișcoi și D. Bejenari [2], [6], [62], au elaborat metode matriceale și algoritmi numerici de determinare a perioadei de ocupare pentru modelele de așteptare de tip Polling cu întârzieri semi-Markoviene și pentru modelele generalizate de așteptare cu priorități. În această ordine de idei, se încadrează și studiile efectuate de I. Damian, axate pe analiza rețelelor stohastice semi-markoviene [39]. În continuare vom menționa unele aspecte de aplicare a rezultatelor teoretice, și anume, de studiul unor caracteristici ale sistemelor de așteptare în port și modelarea numerică în scopul eficientizării activității portului, efectuate de către A. Costea și I.R. Țicu [5], [7].

Scopul și obiectivele tezei. Lucrarea are ca scop extinderea rezultatelor cunoscute din domeniul Teoriei Așteptării, elaborarea a noi tehnici și algoritmi numerici de determinare a unor

caracteristici de performanță mai optime pentru modelele de așteptare Polling cu vacanțe semi-Markoviene și pentru cele cu prioritatea DD.

Pentru atingerea scopului menționat s-au realizat următoarele obiective:

- studierea sistemelor de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă;
- analiza metodelor moderne de cercetare eficiente în obținerea a noi rezultate în Teoria Așteptării;
- cercetarea sistemelor generalizate de așteptare cu priorități și studierea aparatului matematic pentru aceste sisteme;
- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției lungimii virtuale a șirului de așteptare pentru sistemele Polling cu întârzieri semi-Markoviene;
- studierea și formalizarea caracteristicilor probabiliste de performanță pentru sistemele generalizate de așteptare cu prioritatea DD;
- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele Polling cu prioritatea DD;
- implementarea algoritmilor elaborați în limbaje de programare în vederea estimării parametrilor funcțiilor de repartiție ce intervin în optimizarea caracteristicilor de performanță ale modelelor de așteptare Polling.

Suportul metodologic al cercetărilor este bazat pe noțiuni din teoria probabilităților și statistică, teoria așteptării, metode ale teoriei proceselor aleatoare, ș.a. Algoritmii numerici elaborați sunt implementați în limbajele de programare C++ și Kotlin.

Noutatea și originalitatea științifică: constă în elaborarea metodelor și algoritmilor numerici pentru determinarea unor caracteristici numerice de performanță pentru sistemele Polling, cât și pentru cele cu prioritatea DD. Astfel se poate stabili eficiența/permanența sistemului de așteptare în dependență de legile de repartiție, legile de prioritate, strategia sistemului în stare liberă.

Problema științifică importantă soluționată: rezidă în determinarea unor valori mai optime ale caracteristicilor probabiliste pentru modelele Polling, rezultatele obținute atât în urma analizei modelelor de așteptare, cât și a funcțiilor de repartiție, legilor de prioritate, schemelor de servire și orientare, fapt care permite stabilirea staționarității și eficienței sistemului de așteptare.

Importanța teoretică și valoarea aplicativă a lucrării. Modelele matematice ale teoriei așteptării joacă un rol important în analiza, modelarea, proiectarea și optimizarea diferitor procese reale, în care apar fenomene de așteptare, cum ar fi rețelele contemporane, centrele de apel public, sistemele de transport, procesele de producție, etc. O atenție deosebită este acordată

studierii sistemelor de tip Polling cu întârzieri semi-Markoviene sau, cum mai sunt cunoscute, sisteme de așteptare cu un singur server (nod, stație), deoarece aceste sisteme oferă posibilități foarte bune pentru studierea sistemelor de așteptare complexe cu mai multe servere. După cum sa menționat, modelele Polling joacă un rol-cheie în analiza și proiectarea rețelelor regionale fără fir de bandă largă. Pe de altă parte, modelele generalizate cu priorități pot fi văzute ca o clasă specială a acestor modele. Astfel de sisteme reprezintă modele adecvate ale multor aspecte ale vieții de zi cu zi, atunci când o servire preferențială se acordă pentru anumite tipuri de cerințe.

Rezultatele obținute în teză, atât pentru sistemele de așteptare Polling cu vacanțe semi-Markoviene cât și pentru sistemele generalizate de așteptare cu priorități, pot fi utilizate ca suport pentru continuarea studiului și analizei științifice în această direcție, astfel se pot determina și alte caracteristici de performanță ale sistemelor de așteptare. După cum am menționat, o particularitate majoră a studierii acestor sisteme constă în caracterul extensibil și aplicabil a rezultatelor în diferite sfere de activitate, unde apar fenomene similare, spre exemplu, procesele de transport, comunicație, producție, etc.

Aprobarea rezultatelor. Rezultatele fundamentale și aplicative ale tezei au fost prezentate și aprobate:

I. la conferințe și seminare științifice naționale și internaționale:

1. *International Conference on Information Technologies, Systems and Networks ITNS - 2010*, Queue length modeling for exhaustive Polling systems, Chișinău, February 25-26, 2010;
2. *Conferința Științifică Internațională "Modelare Matematică, Optimizare și Tehnologii Informaționale"*, Modelarea perioadei de ocupare și a repartiției șirului de așteptare pentru sisteme Polling cu servire exhaustivă, Chișinău, 24-26 martie, 2010;
3. *International Conference "The 18th Conference on Applied and Industrial Mathematics CAIM 2010"*, Method of "catastrophes" and its application to analyze generalized queueing models, Iași, România, October 14-17, 2010;
4. *Conferința Științifică Internațională „Probleme și perspective de dezvoltare a potențialului economic și managerial al Republicii Moldova în condițiile de criză"*, Exemple software pentru unele modele matematice în fenomenele de așteptare, Chișinău, 21 aprilie 2011;
5. *International Conference "The 7-th Congress of Romanian Mathematicians"*, An analog of the Pollaczek-Khintchin transform equation, Brașov, România, June 29-July 5, 2011;
6. *International Conference "Mathematics & Information Technologies: Research and Education – MITRE-2011"*, Method of catastrophes and numerical problems in queueing

- theory, Chişinău, 22-25 august, 2011;
7. *International Conference “Mathematics & Information Technologies: Research and Education – MITRE-2011”*, Econometrical models and structures in the queueing phenomena, Chişinău, 22-25 august, 2011;
 8. *International Conference “The 19th Conference on Applied and Industrial Mathematics CAIM 2011”*, Numerical algorithms regarding Polling systems with exhaustive service, Iaşi, România, September 22-25, 2011;
 9. *Conferinţa Ştiinţifică Internaţională “Modelare Matematică, Optimizare şi Tehnologii Informaţionale”*, Caracteristice de performanţă în evoluţia modelelor de aşteptare, Chişinău, 19-23 martie, 2012;
 10. *International Conference “The 20th Conference on Applied and Industrial Mathematics CAIM 2012”*, Numerical results for probability of states with PH distribution for Polling models, Chişinău, August 22-25, 2012;
 11. *Conferinţă Ştiinţifică Internaţională „Strategii de dezvoltare socio-economică a societăţii în condiţiile globalizării”*, Algoritmi numerici cu aproximaţii succesive în soluţionarea caracteristicilor modelelor exhaustive Polling, Chişinău, 15-17 octombrie 2012;
 12. *Conferinţa Ştiinţifico-Practică Internaţională „Politici economice şi financiare pentru o dezvoltare competitivă”*, Metode analitice şi numerice în analiza sistemelor Polling, Chişinău, 12 aprilie 2013;
 13. *International Conference “The 37th Annual Congress of the American Romanian Academy of Arts and Sciences (ARA)”*, Numerical algorithm regarding symmetric discrete polling system, Chişinău, June 04-09, 2013;
 14. *International Conference “Mathematics & Information Technologies: Research and Education – MITRE-2013”*, On priority discipline DD with semi-Markov switching, Chişinău, 18-22 august, 2013;
 15. *International Conference “The 21st Conference on Applied and Industrial Mathematics CAIM 2013”*, Kendall and Pollaczek-Khintchin equations for queueing models with semi-Markov switching, Bucureşti, România, September 19-22, 2013;
 16. *The 17-th International Conference on “Distributed Computer and Communication Networks (DCCN-2013)”: Control, Computation, Communications*, Some questions of numerical modeling of priority discipline DD with semi-Markov switching, Moscow, Russia, October 07-10, 2013;
 17. *Conferinţă Ştiinţifică ”UNIVERSITAS EUROPAEA XXI: Ştiinţa Universitară în contextul*

Integrării Europene”, Modele Polling: rezultate analitice și aplicații, Chișinău, 18 octombrie 2013;

18. *Conferința Științifică Internațională a doctoranzilor „Tendințe contemporane ale dezvoltării științei: viziuni ale tinerilor cercetători”*, Valoarea medie a perioadei de ocupare pentru modelul polling cu vacanțe semi-Markoviene, Chișinău, 10 martie 2014;
19. *The International Conference “Mathematics Days in Sofia” – MDS 2014*, Some probabilistic characteristics for queueing systems with two priority classes, semi-Markov orientation and strategy in the free state, Sofia, Bulgaria, July 07-10, 2014;
20. *The Third Conference of Mathematical Society of the Republic of Moldova dedicated to the 50th anniversary of the foundation of Institute of Mathematics and Computer Science “IMCS-50”*, Performance characteristics for semi-Markov polling models with exhaustive service, Chișinău, 19-23 august, 2014;
21. *The Third Conference of Mathematical Society of the Republic of Moldova dedicated to the 50th anniversary of the foundation of Institute of Mathematics and Computer Science “IMCS-50”*, Auxiliary busy periods for $M_2|G_2|1$ system with PH distribution and strategy “reset-to-zero”, Chișinău, 19-23 august, 2014;
22. *Conferința Științifică Internațională a doctoranzilor „Tendințe contemporane ale dezvoltării științei: viziuni ale tinerilor cercetători”*, Repartiția perioadei de ocupare și caracteristici auxiliare pentru modelul cu prioritatea DD, Chișinău, 10 martie 2015;
23. *Seminarul științific în cadrul laboratorului de Modelare Matematică, Institutul de Matematică și Informatică al AȘM*, Sisteme exhaustive Polling cu vacanțe semi-Markoviene și priorități, Chișinău, 26 mai 2016.

II. în cadrul a 4 proiecte științifice de cercetare naționale și internaționale:

1. A.Ș.M.-STCU, 13.820.08.06 STCU.F/5854 (2013-2014);
2. Tineri Cercetători, 13.819.18.05A (2013-2014);
3. A.Ș.M.- Belarusia, 10.820.06BF (2010-2011);
4. A.Ș.M.- Germania 09.820.08.01/GF (2009).

Publicații la tema tezei. Rezultatele de bază ale tezei au fost publicate în 30 lucrări științifice, inclusiv 7 lucrări în reviste științifice recenzate, dintre care 4 articole în reviste de categoria B. Din numărul total de lucrări 4 sunt publicate fără coautori. Rezultatele științifice obținute au fost publicate în 22 lucrări în culegeri internaționale și 1 ciclu didactic de lucrări de laborator.

Sumarul compartimentelor tezei. Lucrarea este alcătuită din Adnotări (în limbile română, rusă și engleză), Introducere, trei capitole, Concluzii generale și recomandări, Bibliografie ce conține 103 titluri, 2 anexe și CV-ul autorului.

În **Primul capitol** se efectuează o prezentare generală a evoluției cercetărilor științifice cu privire la domeniul teoriei așteptării, și anume a studierii modelelor Polling, cât și a sistemelor generalizate cu priorități. Se formulează scopul și obiectivele tezei, direcțiile de cercetare și metodologiile utilizate. De asemenea, sunt prezentate diverse modele și metode în studiul sistemului Polling și a dezvoltării lor. Pentru relevarea importanței acestor modele, se prezintă domenii de aplicare ale sistemelor Polling, care își găsesc aplicații pe scară largă în diverse sfere de activitate, cum ar fi sistemele informatice, de transport, telecomunicații, etc. Pentru înțelegerea funcționării unui sistem de așteptare sunt definite unele caracteristici probabiliste de performanță pentru modelul studiat.

În **Capitolul al doilea** sunt prezentate și descrise unele metode analitice și numerice de cercetare care servesc pentru obținerea a noi rezultate în Teoria Așteptării, astfel sunt analizate metodele și procedee bazate pe aparatul funcțiilor generatoare, transformatelor Laplace și Laplace-Stieltjes. De asemenea, se definește aparatul matematic al modelului de așteptare Polling cu vacanțe semi-Markoviene și sunt studiate rezultate de bază cu privire la unele caracteristici probabilistice de performanță pentru acest model. Sunt prezentați algoritmi numerici ce servesc pentru determinarea repartiției lungimii virtuale a șirului de așteptare, pentru diferite legi de repartiție, însoțiți de exemple.

Capitolul al treilea este dedicat sistemelor generalizate de așteptare cu priorități, în special se pune accent pe disciplina de prioritate DD. În acest scop, se prezintă unele noțiuni generale, clasificări, rezultate analitice esențiale pentru sistemele generalizate de așteptare cu priorități. De asemenea, sunt descrise rezultate cu privire la unele caracteristici probabilistice pentru modelele generalizate de așteptare pentru prioritatea DD. Sunt prezentați algoritmi numerici pentru determinarea perioadei de ocupare și a caracteristicilor auxiliare, pentru diferite funcții de repartiție pentru disciplina DD, însoțiți de exemple.

În compartimentul “**Concluzii generale și recomandări**” sunt prezentate concluziile generale asupra rezultatelor obținute în cadrul tezei, sunt expuse impactul și valoarea elaborărilor propuse în dezvoltarea direcției date. Sunt prezentate recomandările autorului în formă de sugestii privind cercetările de perspectivă.

1. ANALIZA SITUAȚIEI ÎN DOMENIUL MODELELOR POLLING

În acest capitol se prezintă o analiză și se efectuează o generalizare a evoluției lucrărilor științifice cu privire la domeniul teoriei așteptării, și anume a studierii modelelor Polling, care reprezintă o clasă de modele de așteptare cu sondaj ciclic. De asemenea, sunt prezentate diverse modele și metode în studiul sistemului Polling și a evoluției lor. Pentru relevarea importanței acestor modele, se prezintă domenii de aplicare ale sistemelor Polling, care își găsesc aplicații pe scară largă în diverse sfere de activitate, cum ar fi sistemele informatice, de transport, telecomunicații, etc. Pentru înțelegerea funcționării unui sistem de așteptare sunt definite unele caracteristici probabiliste de performanță din domeniul teoriei așteptării.

1.1. Evoluția istorică în studierea modelelor Polling și a modelelor generalizate cu priorități

Sistematizarea și generalizarea rezultatelor teoretice obținute în domeniul studierii modelelor Polling până în anul 1985, sunt expuse în monografia H. Takagi [72]. În continuare, dezvoltarea rezultatelor teoretice în această direcție, publicate înainte de anul 1995, sunt reflectate în monografia S. Borst [23], iar lucrări publicate în perioada anilor 1996-2006, sunt prezentate în lucrarea [92]. Generalizarea și sistematizarea modelelor și metodelor pentru studierea sistemelor stohastice cu sondaj ciclic (sistemul Polling) și utilizarea lor pentru proiectarea rețelelor fără fir de bandă largă este dedicată monografia [93]. În această lucrare, de asemenea, sunt precăutate modele noi, care descriu funcționarea rețelelor fără fir de bandă largă care rulează sub protocoalele Wi-Fi și Wi-MAX cu un mecanism de control centralizat. Analiza modelelor cu priorități cu schimb nenul de tip semi-Markov al șirurilor de așteptare cu priorități, denumite modele generalizate cu priorități, este expusă în monografia [100]. În această lucrare sunt prezentate discipline noi de prioritate, apărute ca urmare a formalizării pierderii timpului la comutare, sunt dezvoltate noi metode de analiză și elaborați algoritmi numerici de calcul al caracteristicilor ale sistemelor generalizate. Este arătată conexiunea și continuitatea rezultatelor pentru sistemele clasice și sistemele generalizate.

În lucrările G. P. Klimov și G. K. Mișcoi [99], M. I. Volkovinskii și A. N. Kabalevskii [95] este studiat un caz special al sistemelor Polling: sisteme de așteptare cu priorități cu flux de intrare Poisson și timp de orientare de o formă specială. Aceste monografii extind rezultatele obținute în acea perioadă și care sunt prezentate în lucrarea B. V. Gnedenko ș.a. [96]. Însă în lucrarea [11] este studiat un model de așteptare de tip-vacanță precum și un model Polling cu multiple șiruri și un server cu o caracteristică specială a unui nou proces.

Într-o lucrare de Attahiru S. Alfa [13], este o încercare de a prezenta modelele de

așteptare, în mod accesibil cititorului, cu privire la modul de utilizare a instrumentelor matematice disponibile pentru a-l ajuta în analiza problemelor asociate cu modelele de așteptare. În acest scop, conținutul acestei lucrări este, în general, prezentat într-o formă care face ca aplicațiile să fie foarte ușor de văzut și, de asemenea, pentru ca noile evoluții și rezultate din domeniu să fie mai accesibile pentru utilizatorul pragmatic al modelelor de așteptare. În monografia de Attahiru S. Alfa [14] sunt descrise bazele teoretice pentru modelarea șirurilor de așteptare în timp discret, precum și procedurile de bază pentru dezvoltarea de modele de așteptare în timp discret. Există un accesnt pe aplicații în sistemele moderne de telecomunicații.

Este bine cunoscut faptul că șirurile de așteptare cu un singur server dau unele perspective foarte bune chiar și șirurilor de așteptare complexe cu mai multe servere dacă sunt corect approximate. Deci, șirurile de așteptare cu un singur server sunt foarte importante în domeniul Teoriei Așteptării și sunt foarte des întâlnite. Din acest motiv, această lucrare se concentrează pe șirurile de așteptare cu un singur server. Rețelele de așteptare cu timp discret sunt descrise în monografia lui H. Daduna [31].

Un domeniu important în care Teoria Așteptării se aplică foarte des este domeniul telecomunicațiilor. Sistemele de telecomunicații sunt analizate astăzi în timp discret [3], deoarece se bazează în general pe tehnologia discretă; timpul este divizat pe intervale și sistemul s-a transferat de la tehnologia analogică (cu timp continuu) la cea discretă. Prin urmare, timpul discret al modelelor de așteptare [66] necesită o atenție specială în domeniul de așteptare și al telecomunicațiilor.

Scopul principal al studierii sistemelor Polling îl constituie determinarea caracteristicilor probabiliste ale sistemului, cum ar fi: perioada de ocupare, lungimea șirului de așteptare, volumul de lucru, etc. Dar nu întotdeauna formulele analitice pot fi utilizate direct pentru a determina aceste caracteristici, astfel o importanță majoră se acordă elaborării unor noi metode numerice cât și algoritmilor realizați în baza acestor metode. Pentru analiza sistemelor Polling sunt propuse mai multe metode, în continuare vor fi descrise succint unele din ele.

Metoda mediilor. Una dintre metode, propusă în literatura de specialitate, pentru cercetarea sistemelor Polling este metoda mediilor. Această metodă este pe larg descrisă în lucrarea [89] și este destinată pentru calcularea lungimilor medii a șirurilor de așteptare din sisteme într-un moment de timp arbitrar, pentru care pot fi obținute duratele medii a vizitării șirurilor, în particular pentru sistemul cu sondaj ciclic $M/G/1$ și servire exhaustivă sau închisă. Pe baza timpului mediu de vizită a șirului de așteptare și a valorii medii rămase se calculează numărul mediu de cerințe în șirurile de așteptare ale sistemului ca soluție a sistemului de ecuații liniare. Vom menționa că metoda mediilor poate fi extinsă pentru următoarele sisteme Polling:

sisteme cu flux Poisson grupate, sisteme cu sondaj periodic, sisteme cu timp discret, de asemenea aplicarea acestei metodei la analiza aproximativă a altor modele Polling (a se vedea [78, 80, 83]).

Metoda mediilor este aplicată, în lucrarea [80], pentru calcularea aproximativă a timpului mediu de așteptare din sistem cu servire limitată a șirurilor de așteptare. Ideea de bază a aproximărilor constă în aceea de a descompune sistemul inițial de N -șiruri de așteptare în N -sisteme de așteptare cu un singur șir, stări de repaus a serverului și k -discipline de servire limitate. Și întrucât, cel mai probabil că după perioada lungă (scurtă) de servire urmează perioada lungă (scurtă) dintre vizitele șirurilor de așteptare, astfel se presupune că lungimea perioadelor dintre vizitele șirurilor corelează cu numărul de cerințe deservite în timpul vizitei precedente a șirului de așteptare. Analiza este dedicată determinării primelor două momente a perioadei dintre vizitele șirului în condiția că l -cerințe au fost servite din acest șir în timpul perioadei precedente de servire, de asemenea cu ajutorul acestei analize poate fi determinată repartiția perioadei dintre vizitele șirurilor de așteptare.

În lucrarea [88], metoda mediilor este propusă pentru analiza sistemelor Polling cu servire exhaustivă sau închisă pentru diferite discipline de servire a cerințelor din cadrul șirului: în ordinea sosirii cerințelor (ca caz particular, precăutat în [89]), în ordinea inversă cu întreruperea servirii sau fără întreruperi, disciplina de servire a primei cerințe cu timpul minim de servire rămas, disciplina descompunerii a serverului. În [78] această metodă se aplică pentru analiza aproximativă a sistemelor ce funcționează în condițiile unor încărcări mari.

Metoda mediilor, este descrisă în lucrarea [83], la analiza sistemelor Polling cu sondaj dinamic adaptiv. Sondajul adaptiv presupune că serverul omite acele șiruri de așteptare care nu aveau cerințe în momentul sondajului din ciclul precedent. Dacă toate șirurile sistemului trebuie să fie omise, atunci serverul se va afla în stare de repaus, după care va începe sondajul tuturor șirurilor în ordinea lor. Analiza este bazată pe calcularea aproximativă a probabilității că șirul va fi omis în ciclu, urmată de aplicarea metodei mediilor pentru calcularea timpilor medii de așteptare.

Metoda „catastrofelor”. O metodă de cercetare a sistemelor Polling, cu o bogată istorie de succes în obținerea unor noi rezultate în Teoria Așteptării, este metoda „catastrofelor” sau metoda introducerii unui eveniment aleatoriu suplimentar [8, 98]. Esența metodei „catastrofelor” constă în faptul că introducând un eveniment suplimentar („catastrofă”) se reușește de atribuit un sens probabilist clar transformatelor Laplace și Laplace-Stieltjes, după ce se precaută evoluția sistemului de așteptare și se determină aceste probabilități, aceasta ne permite să evităm anumite structuri complicate [40].

Metoda funcțiilor generatoare (de colorare, marcare) [8, 98] este o metodă eficientă în

cercetarea problemelor Teoriei Așteptării. Ideea principală a acestei metode constă în atribuirea funcției generatoare un anumit sens probabilistic și aceasta se obține prin procedura de colorare (marcare, vopsire) a cerințelor de intrare în sistemul de servire. Astfel, structura matematică abstractă definită ca funcție generatoare, datorită sensului ei probabilistic devine mai convenabilă în problemele aplicative. Cu ajutorul acestei metode, deseori este posibil de obținut expresii analitice pentru funcția generatoare reieșind din sensul ei probabilist și aceasta este posibil fără a se cunoaște funcția de repartiție a variabilei aleatoare. Tot așa cum se întâmplă și cu valorile numerice ale variabilei aleatoare, valoarea medie, dispersia etc.

Metoda proceselor ramificate. O altă metodă pentru cercetarea sistemelor Polling bazată pe teoria proceselor ramificate este propusă în lucrarea [77]. Metoda dată se aplică la sistemele ce funcționează în condițiile încărcăturilor mari și permite obținerea expresiilor aproximative pentru transformatele Laplace-Stieltjes de repartiție a lungimilor șirurilor și a timpurilor de așteptare pentru clase largi ale sistemelor Polling, comportamentul cărora poate fi descris de procesele ramificate.

Analiza sistemelor Polling utilizând teoria de descompunere a proceselor semi-regenerate este detaliat descrisă în lucrările [62, 102]. Sondajul șirurilor de așteptare care se consideră periodic, adică, este făcut conform tabelului sondajului dat. Este determinată transformata Laplace a funcției generatoare a numărului de cerințe în șirurile de așteptare a sistemului pentru diferite discipline de servire (exhaustivă, închisă și limitată).

Condiții de existența unui regim staționar. În lucrarea [68] se ia în considerare sistemul cu sondaj periodic al șirurilor de așteptare cu fluxurile de intrare de tip BMAP (flux Markov grupat al cerințelor). Pentru sistem sunt obținute condițiile necesare și suficiente pentru existența regimului staționar, precum și este indicat ordinea sondajului șirurilor de așteptare, pentru care sistemul nu are regimul de lucru staționar.

În continuare se va prezenta succint o sinteză a modelelor sistemelor Polling studiate în literatura științifică.

Sisteme cu două șiruri de așteptare. În lucrarea [21] este descris sistemul cu două șiruri de așteptare dintre care în unul sosesc două fluxuri de cerințe cu prioritate. Disciplina de servire - exhaustivă, închisă sau global-închisă (în care sunt deservite doar acele cerințe care erau în șirul de așteptare în momentul începerii ciclului, iar cerințele rămase trebuie să aștepte ciclul următor). Pentru acest model sunt obținute repartiția lungimii ciclului, repartiția numărului de cerințe din șiruri în momentele de sondaj al șirurilor de așteptare, de asemenea este făcută o analiză cu privire la timpul de așteptare. În lucrarea [84] se presupune că timpurile de conectare a serverului la șirul de așteptare și timpurile de servire a cerințelor din cadrul șirului sunt corelate și sunt

descrise timpurile de ședere a unui lanț Markov în stările lor. Este considerată, de asemenea, și o a doua variantă, în care timpii de conectare și de servire sunt definite de repartiția bidimensională Laplace.

Sisteme cu un singur dispozitiv de servire (server). În lucrarea [54] este studiat sistemul Polling cu parametri variabili. Se presupune că repartiția timpului de servire a cerințelor din șir și parametrul fluxului de intrare se schimbă de fiecare dată când serverul părăsește șirul de așteptare. În [25] se prezintă o analiză a sistemului Polling cu servire grupată (simultan sunt servite toate cerințele aflate în șirul de așteptare, la momentul sondajului) utilizând metoda funcțiilor generatoare și este obținută repartiția timpului de așteptare.

Analiza modelului Polling fluid este prezentată în lucrarea [30]. Sistemul Polling este format din N sisteme de așteptare fluide și un dispozitiv de servire (server). Disciplina de servire poate fi exhaustivă, închisă și global-inchisă. Ordinea de servire a șirurilor de așteptare - ciclică sau aleatoare. Este obținută repartiția Laplace-Stieltjes a nivelului de fluiditate din șirurile sistemului în momentul sondajului de către server a șirurilor de așteptare și în orice moment de timp. În plus, procedura de determinare a ordinii probabiliste optime de sondaj a șirurilor de așteptare este descrisă.

Sistemul Polling cu cerințe pozitive și negative este examinat în [70], pentru care cu ajutorul metodei funcțiilor generatoare sa obținut repartiția numărului de cerințe în șirurile de așteptare și repartiția timpului de așteptare în termenii transformatei Laplace-Stieltjes. În lucrarea [71] se analizează sistemul Polling cu două tipuri de refuzuri: refuzuri în servirea cerinței și refuzuri în servirea șirului de așteptare. Refuzul în primul caz se produce în timpul servirii cerinței, și ca rezultat aceasta părăsește sistemul, iar serverul trece la servirea următoarei cerințe din șirul de așteptare. În cazul refuzului de-al doilea tip, serverul imediat întrerupe servirea cerinței, părăsește șirul și trece la următorul șir de așteptare.

În lucrarea [91] se studiază sistemul cu disciplină exhaustivă cu prag de servire a șirurilor de așteptare. Șirul poate fi servit, dacă lungimea sa depășește un prag stabilit. Dacă lungimea tuturor șirurilor de așteptare este insuficientă pentru a începe servirea, atunci serverul încetează vizita șirurilor și o reîncepe în momentul când careva dintre șiruri va acumula numărul necesar de cerințe. Sistemul cu diferite discipline de servire a cerințelor, cum ar fi [88], este considerat în lucrarea [27], pentru care a fost determinată repartiția lungimii ciclului și a timpului de staționare a cerințelor din orice șir de așteptare al sistemului.

Sisteme cu multiple dispozitive de servire (servere). În [94] este analizat modelul Polling ce descrie funcționarea sistemelor cu sondaj ciclic (Polling) în rețele de viteză înaltă fără fir – MESH. Servirea ciclică a șirurilor de așteptare este realizată de două servere. O parte din șiruri

sunt disponibile pentru sondajul ambelor servere; fiecare dintre celelalte șiruri este determinat după serverul "său" în ciclul de servire. Pentru studiul unui astfel de sistem este aplicată metoda mediilor de analiză.

Sistemul Polling cu un număr infinit de servere este analizat în lucrarea [85]. Șirurile sistemului sunt intervievate ciclic de un grup cu un număr infinit de servere. Serverele servesc șirurile de așteptare într-un timp aleator, după expirarea căruia serverele părăsesc șirurile, iar cerințele, servirea cărora a fost întreruptă, se deservesc din nou la următoarea vizită a serverului la șirul de așteptare.

În urma analizei literaturii de specialitate, îndeosebi a publicațiilor recente, putem observa că majoritatea caracteristicilor probabiliste de performanță ale modelelor Polling și sistemelor generalizate cu priorități sunt obținute în termeni de ecuații funcționale, funcții generatoare, transformate Laplace și Laplace-Stieltjes, etc. Astfel, pentru determinarea caracteristicilor acestor sisteme, cum și în scop de modelare a evoluției lor pentru analiza problemelor practice necesită elaborarea diverselor metode, algoritmi numerici și modelări. Scopul acestor metode numerice cum și a algoritmilor numerici elaborați în baza lor constă în soluționarea acelor structuri matematice care nu dispun de soluție analitică exactă.

Progresele tehnologice din diverse domenii ale practicii contemporane duc la formularea și studierea a noi modele Polling cu caracteristici care nu sunt studiate în cadrul modelelor Polling clasice, pentru care sunt obținute și cunoscute astăzi careva rezultatele, aceste premise duc și vor duce la o varietate de provocări științifice în următorii anii. Printre lucrările științifice remarcabile și bine-cunoscute din acest domeniu sunt și lucrări ale cercetătorilor din Republica Moldova. Astfel, vom menționa Gh. Mișcoi și A. Bejan în [1], [65], [101], au propus un algoritm îmbunătățit de soluționare a ecuației clasice Kendall, care poate fi utilizat în algoritmi multidimensionali. Gh. Mișcoi și O. Benderschi [4], [16] au elaborat noi metode, tehnici și algoritmi numerici de evaluare a coeficientului de trafic și a altor caracteristici numerice pentru modele generalizate cu priorități. Gh. Mișcoi și D. Bejenari [2], [6], [62], au elaborat metode matriceale și algoritmi numerici de determinare a perioadei de ocupare pentru modelele de așteptare de tip Polling cu întârzieri semi-Markoviene și pentru modelele generalizate de așteptare cu priorități. În această succesiune se încadrează armonios și cercetările efectuate de I. Damian, axate pe analiza rețelelor stohastice semi-markoviene [39]. În continuare ne vom referi la unele aspecte de aplicare a rezultatelor teoretice în portul maritim Constanța, România, care sunt efectuate de către A. Costea și I. Țicu [5], [7]. Anumite caracteristici portuare ale modelelor de așteptare pot fi studiate în procesul de modelare, modelarea numerică făcându-se pe baza caracteristicilor analitice obținute anterior, dar completate cu date reale colectate din port.

Scopul final al acestor modelări constă în eficientizarea activității portului.

1.2. Aplicații ale sistemelor de așteptare în diverse domenii

Sistemele Polling își găsesc aplicații pe scară largă în sistemele de sănătate publică, în transportul aerian și feroviar, precum și în sistemele comunicaționale. Din acest motiv, studierea acestora, care datează încă de pe la sfârșitul anilor 1950, reprezintă un factor destul de important pentru economia țării noastre.

Modelul Polling este un sistem cu multiple șiruri de așteptare cu un singur server care vizitează șirurile conform tabelului Polling și servește clienții din aceste șiruri. În plus, modelele Polling sunt aplicabile în situațiile în care mai mulți utilizatori concurează pentru accesul la o resursă comună, care este disponibilă la un moment dat, cum ar fi sistemele de comunicații, sistemele de circulație și transport, sistemele de producție, etc.

Sistemele de așteptare prezintă interes, în general, pentru două categorii de persoane: 1) utilizatorii de sisteme (clienți) și 2) furnizorii de servicii. Clienții doresc să utilizeze sistemul și să minimizeze timpul de aflare în sistem, cheltuielile pentru servicii, timpul de așteptare a începerii servirii, întârzierile, etc. Pe de altă parte, furnizorul de servicii dorește să reducă la minimum costul de furnizare a serviciului pentru clienții, în același timp asigurându-se că clienții sunt „rezonabil” satisfăcuți. De exemplu, furnizorii de servicii nu doresc să atribuiască prea multe servere, ei nu doresc să furnizeze o dimensiune destul de mare spațiului liber de așteptare (buffer), care este adesea neocupat, etc. – și ei fac acest lucru fără a ști exact volumul de lucru care urmează să fie transmis de către clienții pentru că această componentă este stohastică. Uneori, regula folosită pentru a opera cu sistemul este un factor major în atingerea obiectivelor corespunzătoare. De exemplu, doar prin schimbarea regulii de operare de la primul venit-primul servit la un sistem de prioritate va schimba modul în care sistemul este perceput de către clienți. Deci, apare necesitatea de a înțelege cum să operăm cu un sistem de așteptare realizând în același timp toate aceste obiective conflictuale, cauzate atât de sosirea aleatoare a entităților cât și de durata aleatoare a sosirilor. Apare întrebarea dacă în genere se poate de studiat sistemele reale în care acest fenomen dublu aleatoriu are loc. Răspunsul este afirmativ și este bine cunoscut. Procedeu de cercetare constă din două etape: prima etapă constă din elaborarea modelului matematic al sistemului real; a doua etapă constă în cercetarea modelului elaborat, stabilirea anumitor legități, determinarea anumitor caracteristici, etc.

Modelele Polling își găsesc o gamă largă de aplicații în domeniul sistemelor informatice de comunicare, unde resursele (de exemplu, lățimea de bandă, capacitatea procesorului) sunt divizate între diferiți utilizatori. Lucrări, în care sunt menționate aplicații ale modelelor Polling

până la începutul anilor 1990, ne putem referi la sondaje lui Grillo [38], Levy și Sidi [52], Takagi [72] și Weststrate [87]. De asemenea, în lucrarea [81] se prezintă conceptul rețele mesh fără fir metropolitane și locale, care operează în bandă milimetrică. Pentru generalizare, principalele aplicații menționate în aceste lucrări sunt descrise mai jos și completate cu aplicații ale modelelor Polling în domenii mai recente ale sistemelor de comunicații.

Sisteme informatice cu timp partajat (Time-sharing computer systems). Aplicațiile clasice ale modelelor Polling reprezintă sistemele informatice cu timp partajat [47], acestea constau dintr-un număr de terminale conectate prin intermediul liniilor multi-drop la un calculator central. Transferul de date de la terminale la calculator și invers este controlat prin intermediul unui sistem Polling în care calculatorul interoghează terminalele, solicitând datele acestora, un terminal la un moment de timp dat. În astfel de aplicații ale modelelor Polling, serverul reprezintă calculatorul central, șirurile de așteptare reprezintă terminale și clienții reprezintă datele.

În rețelele de comunicație, diferite terminale concurează pentru accesul la o cotă parte a mediului. Dacă mai multe terminale transmit sau primesc date de la mediu în același timp, pot să apară coliziuni (ciocniri) de pachete și problemele de interferență. Din aceste considerente, multe protocoale de tipul control al accesului la mediu (MAC) au fost propuse pentru diferite tehnologii de rețea, în majoritatea cazurilor acestea conduc la modelele Polling.

Rețele Token-ring (Token-ring networks). Bux [28] folosește modelele Polling pentru a studia performanța schemelor de trecere a jetonului (token) în rețelele locale (LANs), unde un jeton reprezintă dreptul de transmisie și acesta este transmis în rândul diferitor utilizatorilor. În astfel de cazuri, schema de trecere a jetonului este de obicei configurată într-o topologie de tip inel sau magistrală. O rețea Token-ring poate fi caracterizată ca un set de stații conectate la un mediu comun de transmisie într-o topologie inel. Toate mesajele călătoresc pe o rută fixă de la o stație la altă stație în jurul unei bucle. Un jeton poate să se afle în două stări: ocupat sau liber. O stație cu date pentru transmitere citește jetonul liber și îl modifică în stare ocupată înainte de a-l retransmite. Jetonul ocupat este mai apoi încorporat ca parte al antetului datelor transmise de pe inel de către stație. Astfel, alte stații de pe inel pot citi antetul, remarca jetonul ocupat și să se abțină de la transmitere. Când jetonul revine la stația care l-a modificat în stare ocupată și stația a decis de a transfera dreptul de transmitere la o altă stație, aceasta modifică starea jetonului în stare liberă. Rețeaua Token-ring permite transmiterea pachetelor într-un mod fără conflicte. Cu toate acestea, transmiterea pachetelor poate eșua încă din cauza unor erori și denaturări însăși de pe inel. De obicei, aceste erori sunt rare și așa numitul Selective-Repeat ARQ (SR-ARQ) poate fi folosit pentru a înlătura aceste erori. Într-o astfel de schemă, o stație care primește un mesaj

eronat transmite o confirmare negativă către stația de emisie pentru a indica că mesajul trebuie să fie retransmis. Pentru a analiza performanța schemelor SR-ARQ, Levy și Sidi [52] utilizează un model Polling, în care fiecare stație este reprezentată de două șiruri de așteptare, unul pentru mesajele care trebuie să fie trimise și unul pentru confirmările negative care trebuie să fie trimise înapoi, atunci când sunt primite mesaje eronate. Altman și Kofman [12] propun o soluție pentru a face față proceselor de sosire corelate, neregulate și explozive în rețelele Token-ring. Astfel, ei utilizează un model Polling cu așa-numitul trafic de tip Cruz, care filtrează fluxurile de sosire cu ajutorul găleților găurite (leaky buckets).

Rețele Token-bus (Token-bus networks). Rețeaua Token-bus constă dintr-un set de stații conectate între ele într-o topologie magistrală (bus topology). Ideea acestei tehnici este de a combina caracteristici atractive ale topologiei magistrală cu cele ale unui protocol de acces la mediu fără conflicte. Într-o rețea Token-bus, un inel logic este format din moment ce jetonul este transmis. Din moment ce topologia magistrală nu impune nici o ordonare secvențială a stațiilor, inelul logic este definit printr-o secvență de adrese ale stației. Conceptual, trecerea jetonul prin inele și magistrale este foarte similară, și același tip de model de performanță poate fi utilizat pentru a descrie aceste tehnici. Diferența dintre un Token-ring și un Token-bus, din punct de vedere al modelării, este că în rețeaua Token-ring serverul vizitează șirurile de așteptare într-un mod ciclic, în timp ce în modelul Token-bus serverul se mișcă de-a lungul șirurilor de așteptare într-o manieră periodică non-ciclică, care poate fi modelată cu ajutorul tabelului Polling. Un exemplu de o rețea Token-bus este prezentat în lucrarea lui Manfield [55], în care o rețea de comunicație reprezintă mediul de transmisie între un procesor central și un set de procesoare periferice. Aceste sisteme funcționează astfel: procesorul central interoghează fiecare dintre procesoarele periferice, la rândul lor. Ulterior, acesta primește un răspuns care indică dacă procesorul periferic are vreun pachet pentru a fi trimis. Dacă are, atunci pachetul din capătul aceluși șir de așteptare este trimis către procesorul central. După interogarea unui șir de așteptare de intrare periferic, procesorul central verifică propriul său șir de așteptare privitor la pachetele de ieșire, și dacă șirul de așteptare nu este liber, acesta preia controlul magistralei. Avantajul acestui sistem, constă în aceea că se asigură ca mesajele de ieșire să fie trimise rapid și se evită blocare mesajului în procesorul central. Această schemă Polling este numită stea Polling, în literatura de specialitate.

Rețele Slotted-ring (Slotted-ring networks). O altă clasă de protocoale de comunicație pentru rețele cu o topologie inel este așa-numita rețea Slotted-ring. Într-o rețea Slotted-ring, una sau mai multe sloturi circulă de-a lungul stațiilor. Dacă există un pachet pregătit pentru transmitere la o stație și un slot liber vine, atunci pachetul este pus în slot împreună cu adresa

stației de destinație. Acel slot este mai apoi examinat de către fiecare din celelalte stații, la rândul său, până când stația de destinație îl recunoaște și copie conținutul său. Există două posibilități pentru a elibera slotul: fie că este golit de stația sursă (aceasta se numește Cambridge Ring Protocol) sau de stația de destinație (aceasta se numește Orwell Ring Protocol). Vom face referință la lucrarea lui Bux [28], în care este descris un model Polling cu eliberarea sursei și la lucrarea lui Van Arem [74] care se referă la un model Polling a unui protocol Slotted-ring, cu eliberarea destinației.

Rețele Interfață de Date cu Fibră Distribuită (Fibre Distributed Data Interface networks). Rețeaua Interfață de Date cu Fibră Distribuită este un protocol de trecere a jetonului pentru LANs cu o topologie inel, în care accesul la inel pentru transmitere este controlat prin intermediul unui așa-numit protocol jeton cronometrat (timed-token), adică în care timpul de transmisie este limitat pentru fiecare stație. Acest tip de modele conduc la formularea modelelor Polling cu politici de servire cu timp limitat [69].

Rețele Magistrale Duble cu Șiruri de Așteptare Distribuite (Distributed Queue Dual Bus networks - DQDBN). Protocolul DQDB este un protocol de acces multiplu pentru rețelele de comunicație, acesta constă din două magistrale unidirecționale purtătoare de informații în direcții opuse. Stațiile sunt distribuite de-a lungul celor două magistrale și au capacitatea de a transmite/primi informații la/de la cele două magistrale. Protocolul DQDB este destinat pentru a integra traficul de date, voce și video într-o rețea unică de comunicație. Modelul Polling este utilizat pentru studierea mecanismului de acces la mediu ale unei singure stații în rețeaua DQDB, în lucrarea lui Bisdikian [18].

Scheme cu acces aleator (Random access schemes). Vizavi de protocoale cu acces multiplu programate sunt protocoale cu acces multiplu aleator, unde o entitate cu un mesaj îl va transmite indiferent de ciocnirile potențiale. Schema cu acces ALOHA este un exemplu de schema cu acces aleator. De îndată ce un pachet ajunge la o stație acesta este transmis. Atunci când transmisia eșuează, de exemplu, deoarece alte pachete au fost transmise în același timp, cauzând o coliziune, pachetul este programat pentru transmitere după o perioadă de timp aleatoare. O alternativă pentru această schemă este așa-numita schemă cu acces ALOHA rezervată. În această schemă, unei stații i se acordă dreptul exclusiv de a transmite, fără a fi perturbat de către orice altă stație, pentru o anumită perioadă de timp. Atunci când o stație de emisie nu își mai rezervă canalul, unele sau toate stațiile din sistem încep lupta, în scopul de a profita de canal. Durata perioadei de dispută este aleatorie și următoarea stație, care va profita de canal este, de asemenea, aleatoare. Acest tip de protocoale, în mod natural, conduc la formularea modelelor Polling cu dirijare aleatoare [24, 46]. O comparație detaliată a diferitor scheme cu

acces multiplu, inclusiv a schemelor cu acces aleator, cum ar fi ALOHA și CSMA, este dat de Kleinrock [45].

Rețele optice (Optical networks - ONs). Modelele Polling, de asemenea, își găsesc aplicații în domeniul Rețelelor Optice Pasive Ethernet (EPONs), unde pachetele de la diferite Unități de Rețele Optice împărtășesc capacitatea canalului în direcția amonte. Un EPON este o rețea punct-la-multipunct în direcția aval și o rețea multipunct-la-punct în direcția amonte. OLT (Optical Line Terminal) se află în oficiul local, conectând rețeaua de acces la Internet. OLT-ul alocă lățimea de bandă pentru Unitățile de Rețele Optice (ONUs), care sunt situate la sediul clientului, oferind interfețe între OLT și rețeaua utilizatorului final pentru a trimite trafic de voce, video și date. Într-un EPON procesul de transmitere a datelor în aval de la OLT la ONUs este transmis în pachete de lungime variabilă conform protocolului 802.3 [49]. Cu toate acestea, capacitatea ponderii ONUs în direcția amonte, precum și diferite sondaje, bazate pe scheme de alocare de lățime de bandă poate fi implementată. Scheme simple de acces multiplu prin divizarea timpului (TDMA) bazate pe alocare fixă a unui slot de timp suferă din lipsă de multiplexare statistică, făcând ineficientă utilizarea lățimii de bandă disponibile, ceea ce mărește necesitatea de scheme de Alocare dinamică de Lățime de Bandă (DBA). Un sistem dinamic, care reduce dimensiunea unui timp-slot atunci când nu există date pentru a fi transmise, ar permite excesul de lățime de bandă pentru a fi utilizat de către alte ONUs. Kramer și alții [48, 50] propun o schemă Polling intercalată bazată pe OLT similară cu hub-polling pentru a susține alocarea de lățime de bandă dinamică. Pentru a evita monopolizarea utilizării lățimii de bandă a ONUs cu volume mari de date, ei propun o schemă DBA intercalată cu o limită a dimensiunii ferestrei de transmisie maxime.

Bluetooth. Bluetooth este un standard de tehnologie fără fir, utilizat pentru schimbul de date între dispozitive mobile, cum ar fi telefoanele mobile, laptop-uri și căști. Aceste dispozitive formează rețele mici, cunoscute ca Rețelele Personale fără Fir (WPANs). De bază, topologia rețelei Bluetooth se numește picorețea, aceasta este compusă dintr-un dispozitiv master și până la șapte dispozitive slave. Miorandi și alții [56] au observat că structura unei picorețe ce constă din N slave-uri, poate fi modelată, în mod adecvat, cu ajutorul unui sistem Polling compus din $2N$ șiruri de așteptare. Un șir de așteptare este folosit pentru fiecare legătură de comunicare master-slave și un șir de așteptare suplimentar este necesar pentru fiecare legătură slave-master. Aproximări ale întârzierilor medii sunt determinate pentru disciplinele Pure Round-Robin (corespunzător la servirea limitată-1), dependentă și exhaustivă [3]. Zussman și alții [90] au analizat același model, derivând rezultate exacte cu privire la întârzierile pachetelor.

Subsisteme I/O (I/O subsystems). Modelele Polling apar în contextul subsistemelor I/O

de servere de fișiere sau în sistemele de management ale bazelor de date. Serverele de fișiere sunt preconizate să se ocupe de volume mari de cerințe de informații-recuperate într-un interval de timp rezonabil. După ce informațiile solicitate (fie statice sau dinamice) au fost adunate, informația este pregătită pentru transmisia către client prin rețeaua de comunicare, care conectează clientul la server. Pentru acest scop, informația este plasată într-un subsistem I/O, unde acesta este gata pentru a fi transmisă prin intermediul rețelei către client. Acest lucru poate fi realizat, de exemplu, prin intermediul Protocolului de Control al Transportului (TCP), punând în aplicare un mecanism de control bazat pe fereastră. Subsistemul I/O, în general, va constata dintr-un număr de buffere (locuri de așteptare) TCP paralele. Drenarea diferitor buffere este gestionată de un controller I/O, care controlează accesul conținutului buffer-ului la rețea. În acest scop, controlerul I/O "vizitează" buffer-ele într-o anumită ordine, pentru a verifica dacă un buffer are informații ca să fie drenate, și în caz afirmativ, indiferent dacă fereastra TCP corespunzătoare este deschisă, permite transmiterea segmentelor de date ce se află în buffer prin rețea către client. Modelele de performanță ale serverelor Web, inclusiv cu buffere de controlat I/O, au fost analizate în [41, 76]. Acest tip de control I/O duce la formularea modelelor Polling în care disciplina de servire reprezintă dinamica complicată de control al ferestrei TCP. În [29], Czerniak și alții analizează sistemele TCP cu ajutorul modelelor Polling pentru a descrie diferite metode de transmisie.

Rețele de telefonie mobilă (Mobile networks). Modelele Polling pot fi, de asemenea, găsite în domeniul rețelelor de telefonie mobilă, unde diferiți utilizatori concurează pentru accesul la resursele radio limitate de partajate. În astfel de medii, stația de bază este, de obicei, responsabilă de atribuirea de sloturi de timp pentru diferiți utilizatorilor, într-un anumit mod. În acest context, serverul reprezintă dreptul de transmisie și clienții reprezintă pachetele de date care trebuie transmise. Exemple tipice de mecanisme Polling apar în contextul Accesului Multiplu cu Divizare a Codului (CDMA), bazat pe Accesul de Pachete de Viteză Mare (HSPA), unde operatorul stației de bază acordă acces la mediu într-o bază per-timeslot. Există diferite mecanisme de programare pentru a decide care dintre terminale au acces la mediu pentru durata unui slot de timp unic. O implementare comună este o programare simplă Round-Robin (RR), în care accesul la mediu este transmis printre terminale, indiferent de calitatea semnalului [75]. Acest lucru duce imediat la modelele Polling cu politici de servire limitate și dirijare ciclică a servirii. O extensie simplă de programare RR este Round-Robin Ponderat (WRR), ceea ce duce la formularea modelelor Polling cu dirijare periodică a servirii. Pentru a spori eficiența accesului la mediu pentru rețelele HSPA au fost propuse mecanisme de programare cu canale deschise extrem de sofisticate, acordând slot-uri de timp la terminale bazate pe măsurări instantanee ale

Rapoartelor Semnalului la Zgomot (SINRs) la fiecare dintre terminale [20, 22]. Întâmplările inerente în condițiile canalului, și prin urmare, în ordinea în care stațiile beneficiază de acces la mediul, în mod natural, conduc la formularea modelelor Polling cu dirijare a serverului aleatoare sau Markoviană [24, 46].

LANs fără Fir bazate pe Feribot (Ferry-based Wireless LANs - FWLANs). Modelele Polling sunt aplicabile în contextul proiectării rutelor mesajelor de feribot (ferry) în LANs fără fir bazate pe feribot. În astfel de FWLANs, un număr de noduri izolate sunt dispersate peste unele zone geografice unde comunicarea între un nod și lumea exterioară, sau comunicarea între noduri, este posibilă printr-un mesaj feribot. Feribotul urmează un traseu ciclic prestabilit, colectând mesaje și livrând mesaje la noduri ori de câte ori este în imediata apropiere a nodului (ceea ce înseamnă că traseele nu trebuie să treacă prin toate punctele din spațiu). Intervalul de transmitere și recepție este flexibil: presupunând o putere de transmisie fixă, intervalul poate fi majorat la cost redus de transfer. Luând în considerare relația dintre interval și transfer, se poate proiecta un traseu ciclic optim al feribotului. Rezultate cu privire la FWLANs sunt prezentate în lucrarea lui Kavitha și Altman [43].

Rețele mobile adhoc (Mobile adhoc networks - MANETs). Modelele Polling apar, în mod natural, în modelarea rețelelor mobile ad-hoc, care constau din ambele terminale fără fir fixe și mobile. O caracteristică tipică a acestor tipuri de rețele, este că dispozitivele fără fir creează propria lor rețea fără fir într-un mod distribuțional. Utilizatorii telefoanelor mobile pot schimba locația și, prin urmare, modifică legăturile de comunicare în rețea. Exemple de MANETs sunt sistemele de monitorizare animală, programarea conferințelor de colaborare, rețelele de circulație pentru vehicule, sistemele file-sharing peer-to-peer și rețele de răspuns la dezastru. Spre deosebire de cele mai multe rețele fără fir clasice, MANETs permite comunicarea multi-hop. Legăturile de comunicare apar și se întrerup dinamic. Durata lor este aleatorie și independentă de volumul de trafic oferit sau transmis prin astfel de legături. Pentru a capta acest fenomen, De Haan [34] propune așa-numita politică de servire pure time-limited, în care serverul vizitează un șir de așteptare pentru o durată de timp aleatoare, independent de orice altceva în sistem; vom menționa că această politică nu este un lucru conservator. O prezentare generală a aplicațiilor modelelor Polling în MANETs este dată în lucrarea [33].

Rețelele bazate pe cipuri (Networks on chips - NOCs). Un alt domeniu de aplicare interesant al modelelor Polling este rețelele bazate pe cipuri, care au fost propuse ca o soluție pentru ineficiența cauzată de conexiunile magistralelor tradiționale [17, 32]. În NOCs, blocurile de proprietate intelectuală (un termen general pentru modulele on-chip) nu sunt conectate la o singură legătură partajată, dar la interfețele de rețea care implementează protocoale de

comunicație. Datele sunt transmise prin intermediul comutatoarelor care constau din porturi de intrare și de ieșire. Dacă mai multe porturi au date pentru același port de ieșire, doar un singur port de intrare poate transmite datele, iar comutatorul îl selectează pe care. Datele care nu sunt transmise imediat sunt stocate în locurile de așteptare și vor fi transmise mai târziu. Vom face referință la lucrarea [15], în care este făcută o prezentare generală de aplicabilitate ale modelelor Polling pentru rețelele bazate pe cipuri, și la lucrarea [79] în care este prezentat un algoritm numeric eficient.

Extinderea aplicabilității rezultatelor cu privire la modelele Polling, care sunt cunoscute în prezent, este limitată de mai mulți factori cum ar fi: studierea modelelor pentru anumite funcții de repartiție, cu toate că în marea majoritatea proceselor reale acestea implică diverse situații, astfel la modelarea cât mai adecvată a unui proces real sunt necesare o gamă mai largă de funcții de repartiție; necesitatea de noi metode de analiză teoretică a șirurilor de așteptare, astfel dezvoltarea unor metode flexibile și adecvate ar spori puternic aplicabilitatea rezultatelor teoretice ale modelelor Polling; ipoteza de a avea un singur server care servește multiple șiruri de așteptare este, de asemenea, o limitare importantă pentru multe aplicații reale. Așa cum s-a menționat mai sus, există numai câteva rezultate pentru modelele Polling care sunt compuse din mai multe servere.

În domeniul rețelelor mobile și fără fir, o perspectivă promițătoare constă în mărirea lărimii de bandă, astfel pentru a spori robustețea calității serviciului prestat utilizatorului, trebuie să facem uz de multiple antene, permițând utilizatorilor să folosească simultan mai multe rețele (de exemplu, MIMO). În astfel de medii, fluxurile de pachete pot fi împărțite pe mai multe rețele paralele, unde fiecare dintre acestea pot fi modelate ca un model Polling, în cazul în care traseul pachetului poate depinde de starea actuală a acestor rețele. Studierea și obținerea a noi rezultate pentru modelele Polling va continua să prezinte un domeniu provocator de cercetare pentru următorii anii.

1.3. Reguli de servire a sistemelor de așteptare

Sistemele Polling reprezintă un sistem de așteptare cu un singur nod în care există mai multe șiruri de așteptare și un server, care deservește fiecare șir de așteptare după o regulă stabilită.

Regula conform căreia serverul alege șirul de servire se numește ordinea de servire. Drept exemplu, pe marginea acestei reguli, poate servi sondajul ciclic al șirurilor, când serverul verifică șirurile de la început până la sfârșit și din nou revine la primul șir de așteptare, sau ordinea aleatoare, în care șirul de servire se alege aleatoriu ori conform anumitor priorități.

În dependență de numărul de șiruri de așteptare într-un sistem, sistemele Polling pot fi *discrete* (numărul locurilor de așteptare este finit sau numărabil) sau *continue* (numărul locurilor de așteptare este infinit). În ultimul caz se consideră că șirurile de așteptare sunt plasate pe un cerc sau pe un domeniu n -dimensional.

În continuare vor fi prezentați parametri după care se studiază un sistem de așteptare și în dependență de clasificarea acestora vor fi aduse exemple de diferite tipuri de sisteme Polling [10], [53].

Sistemele Polling discrete se caracterizează după următorii parametri: numărul de șiruri de așteptare, capacitatea lor (numărul locurilor de așteptare), numărul de servere, procesele de sosire și deservire a clienților, durata de conectare a serverului de la un șir de așteptare la altul, precum ordinea și disciplina de servire a șirului de așteptare. Vom presupune că toate șirurile de așteptare sunt numerotate de la 1 la N , unde $N \geq 2$ reprezintă numărul șirurilor de așteptare în sistem. Șirul de așteptare cu numărul $i = \overline{1, N}$ va fi notat cu Q_i .

Prin *ordinea de alegere (vizitare)* a șirurilor de așteptare se înțelege regula folosită de către server pentru alegerea următorului șir de așteptare. Ordinea de alegere poate fi atât *statică* cât și *dinamică*. Prin ordinea *statică*, se înțelege că alegerea șirurilor de așteptare rămâne invariabilă pe întregul parcurs de funcționare al sistemului. Prin ordinea *dinamică* - șirul de așteptare este ales pentru servire la anumite momente de luare a deciziilor, pe baza unor informații complete sau parțiale cu privire la starea sistemului.

Printre tipurile de *ordine statică*, se cunosc [3], [91]:

1. Ordine *ciclică*, unde serverul interoghează șirurile de așteptare în ordinea $Q_1, Q_2, \dots, Q_N, Q_1, Q_2, \dots, Q_N, \dots$. Aceste sisteme Polling sunt numite sisteme *ciclice*.
2. Ordine *periodică*, în care serverul interoghează șirurile de așteptare în ordinea $Q_{T(1)}, Q_{T(2)}, \dots, Q_{T(M)}, Q_{T(1)}, Q_{T(2)}, \dots, Q_{T(M)}, \dots$ care este caracterizată de așa numitul tabel Polling $(T(1), T(2), \dots, T(M))$ de lungime $M (M \geq N)$, $T(i) \in \{1, \dots, N\}$, $i = \overline{1, M}$. Se presupune că tabelul Polling cuprinde numerele tuturor șirurilor de așteptare din sistem.
3. Ordine *aleatoare*, unde șirul de așteptare Q_i , este luat pentru servire cu probabilitatea

$p_i, i = \overline{1, N}, \sum_{i=1}^N p_i = 1$. Este posibilă o altă variantă de alegere a șirului, de exemplu, după

interogarea şirului de aşteptare Q_i , serverul trece la şirul Q_j , cu probabilitatea

$$p_{ij}, i, j = \overline{1, N}, \sum_{j=1}^N p_{ij} = 1, i = \overline{1, N}.$$

4. Ordine de *prioritate*, apare în cazul în care sistemul are şiruri de aşteptare de diferite priorităţi, iar unele şiruri de aşteptare pot fi servite numai în cazul în care toate şirurile de prioritate mai mare nu au clienţi.

Cazuri speciale ale ordinei periodice Polling sunt reprezentate: prin tipul Polling *stea*, în care şirurile de aşteptare sunt servite în ordinea $(Q_1, Q_2, Q_1, Q_3, \dots, Q_1, Q_N)$, şi prin tipul Polling de tip *elevator*, în care şirurile de aşteptare sunt servite în ordinea $Q_1, Q_2, \dots, Q_{N-1}, Q_N, Q_N, Q_{N-1}, \dots, Q_2, Q_1$.

Perioadele de timp numite cicluri sunt menţionate în activitatea sistemului Polling ciclic sau periodic. Pentru sistemele Polling ciclice, prin ciclu se înţelege timpul necesar pentru server pentru a servi şirurile de aşteptare de la Q_1 la Q_N . Pentru sistemele Polling periodice, prin ciclu se înţelege timpul necesar pentru servirea şirurilor de la $Q_{T(1)}$ la $Q_{T(M)}$. În timpul funcţionării unor sisteme Polling, este specificat *ciclul Hamiltonian*, care reprezintă timpul în care serverul parcurge toate şirurile de aşteptare doar o singură dată.

Prin *disciplina de servire a şirului de aşteptare* se înţelege numărul de clienţi serviţi de către server într-un ciclu. În cadrul şirului de aşteptare, clienţii sunt serviţi în ordinea definită de *disciplina de servire a clientului*. Disciplinele de servire a şirului de aşteptare pot fi *deterministe* şi *aleatoare*.

Pentru disciplina *deterministă*, numărul maxim de clienţi serviţi de către server în cadrul unui ciclu a şirului de aşteptare este constant. Printre disciplinele de servire a şirurilor de aşteptare deterministe (fie Q_i), putem specifica următoarele discipline:

1. *exhaustivă*, în care serverul serveşte clienţii până când şirul de aşteptare devine liber.
2. *dependentă (închisă)*, în care serverul serveşte doar acei clienţi care se află în şirul de aşteptare la momentul vizitării. Dacă serverul deserveste doar acei clienţi care se aflau în şirul de aşteptare până la începutul ciclului, această disciplină se numeşte *disciplina global-dependentă*.
3. l_i - *limitată*, în care numărul de clienţi, care pot fi deserviţi de către server este limitat de l_i , $l_i \geq 1$.

4. l_i - decrementată, în care serverul deservește clienții aflați în șirul de așteptare până când lungimea șirului este decrementată (micșorată) cu l_i comparativ cu sondajul inițial, $l_i \geq 1$.
5. o disciplină în care timpul aflării serverului în șirul de așteptare este limitat.

Pentru disciplina aleatoare, numărul de clienți, care pot fi deserviți de către server în șirul de așteptare Q_i este definit de valoarea variabilei aleatoare discrete ξ_i cu legea de repartiție $\{a_j^i, j \geq 1\}$ care poate varia la fiecare ciclu al șirului de așteptare. Menționăm câteva discipline aleatoare:

1. Disciplina *binomială* dată de variabila aleatoare ξ_i , având legea de repartiție binomială cu parametrii X_i și p_i , unde X_i este numărul de clienți aflați în șirul de așteptare Q_i la momentul vizitării de către server și numărul p_i , $0 < p_i \leq 1$. Pentru această disciplină, $a_i^j = C_{X_i}^j p_i^j (1-p_i)^{X_i-j}$, $j = \overline{1, X_i}$, $a_i^j = 0$, pentru $j > X_i$.
2. Disciplina *Bernoulli*, în care primul client aflat în șirul de așteptare Q_i este servit cu probabilitatea 1 și fiecare client ulterior, cu o probabilitate dată p_i . Serverul părăsește șirul de așteptare cu probabilitatea $1-p_i$. Pentru această disciplină, $a_i^j = p^{j-1}$, $j \geq 1$.

Dacă toate șirurile de așteptare ale sistemului Polling au discipline de servire identice, deosebim un sistem Polling cu disciplina de servire dată (exhaustivă, l -limitată sau alta). În cazul în care disciplinele de servire ale șirurilor de așteptare diferă, atunci vom distinge un sistem Polling, cu o disciplină de servire *mixtă*.

Ordinea șirurilor de așteptare Polling și disciplinele lor de servire constituie *politica de servire* a sistemului Polling, adică, regula de alegere a clientului următor din șirul de așteptare conectat la server sau de la alt șir.

Printre sistemele Polling, deosebim sistemele cu *timp discret*, în care timpul este împărțit în intervale egale numite slot-uri și sisteme cu *timp continuu*.

Sistemul Polling se numește sistem *simetric* sau omogen dacă procesele care caracterizează șirurile sale de așteptare (procesele de sosire și servire pentru clienți, precum și cele care definesc durata de trecere a serverului dintre șiruri) sunt stohastic echivalente. În caz contrar, sistemul se numește sistem *non-simetric* sau neomogen. Dacă în sistemul Polling serverul nu are nevoie de timp pentru conectarea de la un șir la altul, putem afirma că acesta este un sistem cu timp de orientare nulă, în caz contrar, putem afirma că acesta este un sistem cu o orientare nenulă.

1.4. Tipuri de modele de așteptare

Este bine cunoscut faptul că șirurile de așteptare cu un singur server (nod) oferă unele perspective foarte bune chiar și șirurilor de așteptare complexe cu mai multe servere, dacă sunt corect approximate [13]. Astfel, șirurile de așteptare cu un singur server sunt foarte importante în domeniul Teoriei Așteptării și sunt foarte des întâlnite în activitatea practică.

Șirurile de așteptare cu un singur nod sunt cele mai mici unități care pot fi clasificate ca un sistem de așteptare. Alte tipuri de sisteme de așteptare sunt configurații a mai multor șiruri de așteptare cu un singur nod.

Un șir de așteptare cu un singur nod este un sistem de așteptare, în care un element (mesaj, cerință) ajunge să fie prelucrat într-o singură locație, cum ar fi la un server. După ce elementul a fost prelucrat, acesta nu merge la o altă locație pentru prelucrare ulterioară, sau mai degrabă neglijăm ceea ce se întâmplă cu acesta în alte locații după ce servirea în locația curentă este finalizată. Cu toate acestea, elementul poate reîntra în aceeași locație pentru prelucrare imediată după ce prelucrarea acestuia este finalizată. Dacă acesta merge la o altă locație pentru o altă prelucrare și revine mai târziu la prima locație pentru prelucrare, atunci vom considera această întoarcere ca o nouă sosire și o nouă prelucrare.

Deci, un sistem de așteptare cu un singur nod constă doar dintr-o singură locație de servire și servirea este furnizată de către același set de servere în acea locație, chiar dacă un element reîntră pentru o altă servire imediat după finalizarea deservirii lui.

Vom menționa, în continuare, unele exemple de șiruri de așteptare cu un singur nod în continuare. În Figura 1.1, în partea superioară este reprezentat un șir simplu cu un singur server, unde cercul reprezintă server-ul, săgeata care duce spre server indică sosirile elementelor și săgeata care pleacă de la server indică plecarea elementelor care au fost prelucrate în urma servirii.

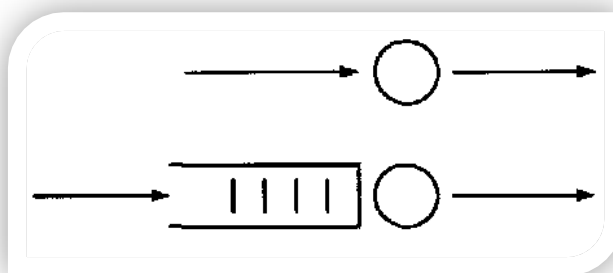


Fig. 1.1. Model de așteptare cu un șir de așteptare și un singur nod

În partea inferioară, a Figurii 1.1, avem un șir de așteptare cu un singur nod. Există doar

un singur server (reprezentat de cerc) în acest sistem. Elementele ajung în spațiul de așteptare (reprezentat de un dreptunghi deschis) și sunt servite conform ordinii stabilite. Atunci când servirea lor este finalizată, elementele vor părăsi sistemul.

Un exemplu aplicativ de sistem de așteptare, care poate fi reprezentat conform acestei scheme, este un restaurant fast-food cu o fereastră drive-in unde există doar un singur ghișeu de servire al clienților și un șir de vehicule în așteptare să fie deservite la această fereastră (Figura 1.2).



Fig. 1.2. Exemplu de model de așteptare cu un șir de așteptare și un singur nod

În Figura 1.3 avem reprezentat un șir de așteptare cu un singur nod cu multiple servere paralele. Un element, ce sosește pentru procesare, poate fi trimis la oricare dintre servere pentru servire.

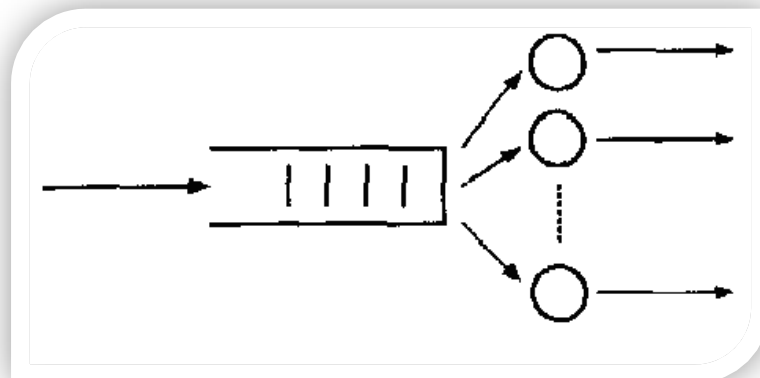


Fig. 1.3. Model de așteptare cu un șir de așteptare și multiple servere paralele

Acest tip de sistem de așteptare este foarte specific pentru următoarele exemple aplicative: șirului locurilor de așteptare a unui comutator (switch) în rețelele de comunicații,

înregistrarea ieșirilor într-un magazin alimentar, la o bancă unde observatori conlucrează cu clienți, o parcare comercială în care locurile de parcare reprezintă serverele și timpul de servire este durata de parcare (Figura 1.4), etc.

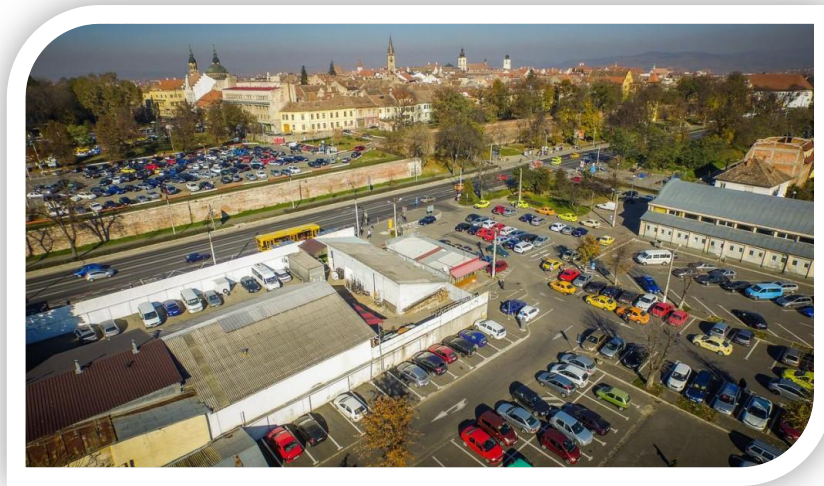


Fig. 1.4. Exempu de model de așteptare cu un șir de așteptare și multiple servere paralele

Un alt exemplu de sistem de așteptare este șirul cu un singur nod în care există mai multe locuri de așteptare pentru diferite șiruri de așteptare și un server care deservește fiecare șir de așteptare bazat pe o regulă stabilită (regulile în Vishnevski) [93]. Acesta se numește un sistem Polling și este reprezentat schematic în Figura 1.5.

Cu alte cuvinte, modelul Polling este un sistem cu multiple șiruri de așteptare cu un singur server care vizitează șirurile conform tabelului Polling și servește cerințele din aceste șiruri. În plus, modelele Polling sunt aplicabile în situațiile în care mai mulți utilizatori concurează pentru accesul la o resursă comună, care este disponibilă la un moment dat.

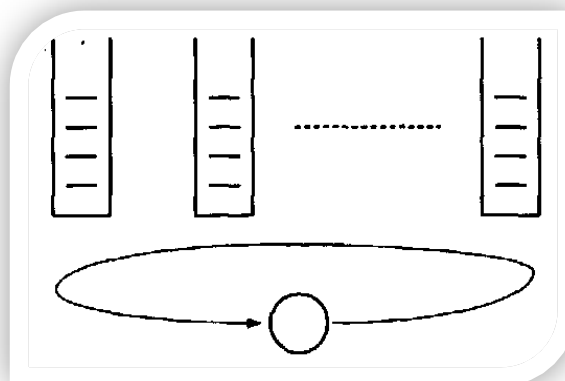


Fig. 1.5. Model de așteptare cu multiple șiruri de așteptare paralele și un server

Acest tip de sistem de așteptare este foarte specific sistemelor de telecomunicații, în controlul mediu de acces (MAC), unde din diferite surse sosesc elementele la un router și trebuie să fie prelucrate. Un alt exemplu pentru acest sistem poate fi considerat și o intersecție de trafic dirijată de semafoare - fiecare bandă din intersecție este un șir de așteptare și server-ul care reprezintă semaforul, aprinde culoarea verde în conformitate cu o regulă stabilită pentru fiecare bandă de circulație.



Fig. 1.6. Exempu de model de așteptare cu multiple șiruri de așteptare paralele și un server

În Figura 1.7 este reprezentat un șir de așteptare cu un singur nod în care un element, după servirea completă, poate reveni imediat la același sistem pentru o altă servire. Acesta este un șir de așteptare cu feedback.

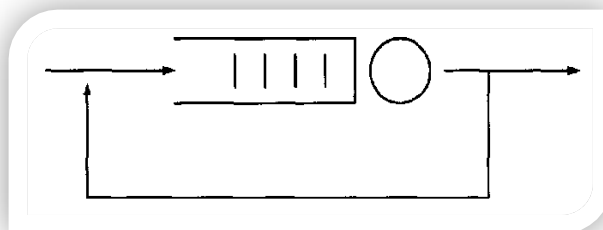


Fig. 1.7. Model de așteptare cu un server și Feedback

De exemplu, să considerăm un sistem de fabricare, în care un element după ce trece printr-un proces de fabricație este inspectat instantaneu și plasat înapoi în același șir de fabricație în cazul în care este considerat a fi defect și este necesară o prelucrare ulterioară.

Să considerăm mai multe șiruri de așteptare cu un singur nod în serie, fie N numărul

șirurilor de așteptare notate prin $Q_1, Q_2, \dots, Q_N$. Elemente care sosesc la șirul Q_1 sunt prelucrate și odată ce servirea lor este finalizată acestea pot reveni la șirul Q_1 pentru o altă servire, pot părăsi complet sistemul sau trece la șirul Q_2 pentru o altă servire suplimentară. La șirul Q_2 ajung elementele din șirul Q_1 care trebuiesc procesate în șirul Q_2 și unele elemente noi care s-au alăturat la șirul Q_2 dintr-o sursă externă. Toate aceste elemente care sosesc în șirul Q_2 sunt prelucrate la șirul Q_2 . Elementele ajunse trec prin șirurile în serie N în funcție de cerințele lor de servire.

De obicei, utilizăm termenul tandem pentru șirurile de așteptare în care elementele merg numai într-o direcție, adică $Q_1 \rightarrow Q_2 \rightarrow \dots \rightarrow Q_N$, și vom spune că $p_{i,j}$ este probabilitatea ca un element care a fost servit la nodul Q_i , trece la nodul Q_j , cu $p_{i,j} \geq 0, j = i, i+1$ și $p_{i,j} = 0, j < i, j < i+1$, ținând cont de faptul că $\sum_{i=1}^N p_{i,j} \leq 1$, din moment ce un element poate părăsi șirul după servirea la nodul Q_i cu probabilitatea $1 - \sum_{i=1}^N p_{i,j} \leq 1$, adică $1 - p_{i,i+1} \leq 1$. Un exemplu de sistem de așteptare tandem este ilustrat în Figura 1.8.

După cum se poate vedea, un sistem tandem de așteptare fizic poate fi descompus în N șiruri cu un singur nod. Chiar dacă este implicată analiza ca un set de șiruri cu un singur nod și se poate termina de asemenea, doar ca o aproximație.

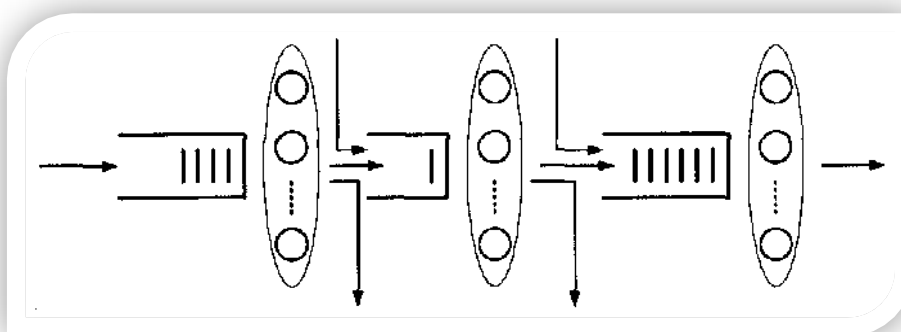


Fig. 1.8. Model de așteptare tandem

Un exemplu din viața reală a unui sistem tandem de așteptare poate servi un circuit virtual într-un sistem de comunicare. Un exemplu, cu care majoritatea dintre noi se ciocnește aproape de zi cu zi, este un traseu de călătorie. Să presupunem că am determinat un traseu de călătorie cu automobilul într-un oraș. Vom descrie acest traseu ca un traseu liniar și fiecare dintre intersecțiile de-a lungul drumului formează un set de șiruri cu un singur nod și toate împreună formează un

sistem tandem de așteptare.

În cele din urmă, vom analiza o rețea a sistemelor de așteptare, care este o clasă mai generală a sistemelor de așteptare [13]. Într-un astfel de sistem putem spune că aveam N noduri. Elementele pot ajunge la oricare dintre nodurile N , de exemplu, nodul Q_i pentru prelucrare. După prelucrare la nodul Q_i elementul poate părăsi sistemul sau de a trece la un alt nod Q_j pentru prelucrare sau chiar să părăsească sistemul în totalitate. Vom considera $p_{i,j} \geq 0, \forall(i, j)$, și $1 - \sum_{i=1}^N p_{i,j} \leq 1$ este probabilitatea ca o servire părăsește sistemul la nodul Q_i . În Figura 1.9 este ilustrată un exemplu de o rețea de șiruri de așteptare.

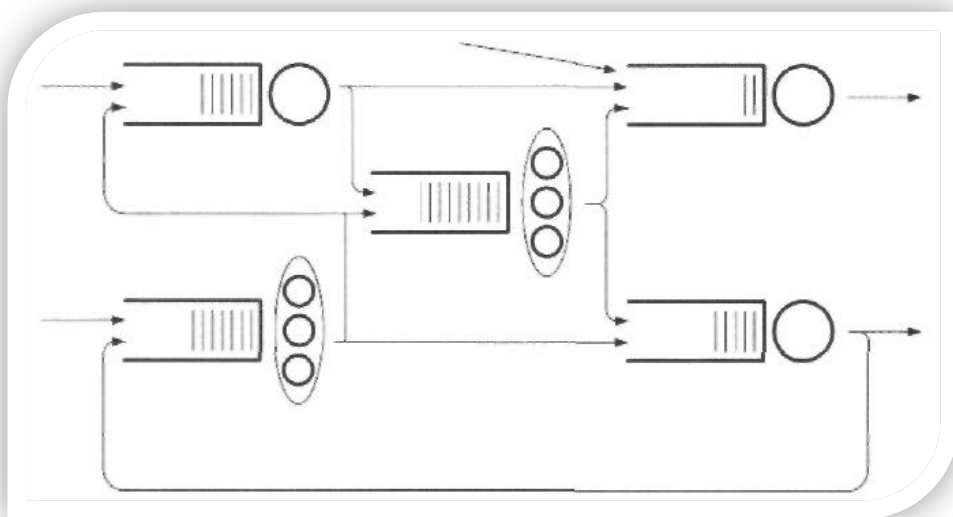


Fig. 1.9. Rețea de șiruri de așteptare

Nu există nici o restricție cu privire la care nod un element poate trece după servirea completă la un alt nod, spre deosebire de sistemele de așteptare tandem, care au restricții în definiția lor.

Cele mai frecvent întâlnite sisteme de așteptare în viața reală sunt, de obicei, rețelele. Exemplu aplicativ poate servi sistemul de comunicare de pachete cu comutare, rețelele de drumuri, etc.

Încă odată putem vedea că un sistem de rețea de așteptare este o simplă colecție de mai multe șiruri cu un singur nod. Uneori, o rețea de așteptare este descompusă în mai multe șiruri de așteptare cu un singur nod ca o aproximare în scopul analizei.



Fig. 1.10. Exemplu de rețea de șiruri de așteptare

1.5. Caracteristici de performanță ale sistemelor de așteptare

Sistemele de așteptare reprezintă orice tipuri de sisteme de servire unde cerințele trebuie să aștepte în rând pentru servire atunci când serverul nu este disponibil sau este ocupat cu servirea altor cerințe. Astfel de tipuri de sisteme sunt actuale și se întâlnesc în activitățile de zi cu zi a vieții noastre, de exemplu în restaurante fast-food, în sisteme de telecomunicații, în sisteme de transport, în sisteme informaționale, în sisteme de fabricație, etc. De exemplu, există cerințe care ar putea fi oameni sau obiecte, unde elementele (cererile) ar putea fi pachete de date, piese într-o uzină de producție, etc. De asemenea, într-un astfel de sistem sunt furnizori de servire care oferă facilități pentru elementele care urmează să fie servite.

Să luăm spre exemplu un sistem de așteptare - Sistemul Internet. Oamenii trimit mesaje și pachete informaționale prin acest sistem pentru a fi prelucrate și transmise către o destinație. Cineva poate considera pachetele ca elemente care trebuie să fie prelucrate pentru un client. Furnizorul de Servicii Internet (ISP) se asigură că clientul este deservit cu resurse pentru a obține pachetele procesate și trimise la destinația corectă fără întârziere și cu probabilitatea pierderii minimă. Clientul plătește pentru un serviciu și se așteaptă la o anumită calitate QoS (quality of service) a servirii, proporțional cu taxa achitată.

În lumea ideală, clienții ar dori procesarea imediată a pachetelor și ISP-ul ar dori să obțină venituri maxime posibile fără a suporta careva cheltuieli. Însă, în lumea reală resursele

necesare pentru deservirea pachetelor costă bani și ISP-ul trebuie să obțină profit, astfel că dorește să ofere cea mai eficientă sumă a resurselor, în timp ce clientul care plătește pentru serviciu setează ținta ei QoS care trebuie îndeplinită de către ISP. În proiectarea unui sistem de așteptare este necesar de a găsi configurații optime și reguli care vor optimiza profitul pentru ISP și vor îndeplini QoS a clienților.

În scopul de a face acest lucru, este necesar să înțelegem modul în care sistemul de așteptare lucrează în conformitate cu diferite configurații și reguli. Unele caracteristici de performanță ce prezintă interes pentru un operator de sistem de așteptare sunt următoarele [13]:

- *Lungimea șirului de așteptare (Queue Length)*: lungimea șirului de așteptare se referă la numărul de elemente, în acest caz pachete, care sunt în așteptare într-o oarecare locație sau loc de așteptare, pentru a fi prelucrate.

Această caracteristică adesea reprezintă un indiciu despre cât de calitativ este un sistem de așteptare. Cu cât este mai lungă lungimea șirului de așteptare cu atât mai rea este calitatea de servire din punctul de vedere al utilizatorului, cu toate că, nu întotdeauna aceasta este corect.

- *Probabilitatea de pierdere a cerinței (Loss Probability)*: dacă locul de așteptare în care elementele trebuie să aștepte este limitat, ceea ce se întâlnește foarte des în sistemele reale, atunci elementele care vor sosi după ce locul de așteptare este ocupat de alte elemente vor fi considerate pierdute și pot reveni la un moment de timp mai târziu.

În sistemele de pachete de date pierderea unui pachet poate fi foarte inacceptabilă și clienții sunt îngrijorați de această probabilitate a perioadei de așteptare. Cu cât mai mare este această valoare cu atât mai proastă este calitatea de servire a sistemului, din perspectiva clientului.

- *Timpul de așteptare (Waiting Times)*: timpul de așteptare sau întârzierea prevăzută de la unii clienți este durata de timp dintre sosirea unui element în sistem și până la începerea servirii acestuia.

Aceasta este caracteristica cea mai utilizată a calității sistemului de către clienții. Desigur, cu cât este mai mare această caracteristică cu atât este mai proastă este calitatea de servire a sistemului, din punctul de vedere al clientului.

- *Timpul de sistem (System Time)*: acesta este timpul de așteptare plus timpul de servire a cerinței. Acesta este perceput în același mod ca și timpul de așteptare a începutului

servirii cu excepția cazului când se ocupă cu un sistem preventiv unde unele elemente pot avea servirea uneori întreruptă.

- *Volumul de lucru (Work Load)*: volumul de lucru reprezintă timpul necesar pentru a procesa elementele de așteptare și este egal cu suma dintre timpul rămas de servire a elementului în servire și timpul de servire a tuturor elementelor de așteptare într-un sistem de lucru de conservare.

Într-un sistem de lucru de conservare servirea ce nu este completă este repetată și nici o lucrare nu este eliminată. Un sistem de așteptare devine liber și serverul devine inactiv (vacant) din moment ce volumul de muncă se reduce la zero.

- *Perioada de ocupare (Busy Period)*: perioada de ocupare reprezintă intervalul de timp care începe cu schimbul serverului către un nou șir de așteptare, după ce șirul precedent servit este liber, și sfârșește când șirul de așteptare respectiv devine liber.
- *Perioada inactivă/pasivă (Idle Period)*: aceasta este o măsură a timpului care se scurge din momentul în care ultimul element este procesat lăsând șirul de așteptare liber până în momentul în care servirea va începe din nou, de obicei, aceasta are loc după sosirea primului element.

Cu cât mai lungă este perioada pasivă cu atât resursele sunt mai risipitoare. Un furnizor de servicii dorește să păstreze perioada pasivă cât mai puțin posibil.

1.6. Concluzii la capitolul 1

În capitolul 1 se descrie succint evoluția cercetărilor cu privire la sistemele de așteptare, și anume a studierii modelelor Polling și a sistemelor de așteptare generalizate cu priorități. Scopul principal al studierii sistemelor de așteptare îl constituie determinarea caracteristicilor probabiliste ale sistemului, cum ar fi: perioada de ocupare, lungimea șirului de așteptare, volumul de lucru, timpul de așteptare, etc. Pentru îmbunătățirea funcționării unui sistem de așteptare, specialiștii din domeniu au ca sarcină să propună și elaboreze mai multe metode, tehnici și algoritmi numerici.

Astfel:

- s-a prezentat o analiză generală a domeniului de cercetare a tezei la momentul actual;
- s-a prezentat unele exemple de aplicare ale modelelor de așteptare în domeniul sistemelor informatice de comunicare;

- s-a studiat și s-a detaliat clasificarea modelelor de așteptare, după anumite reguli de servire și conform structurii lor;
- s-au analizat unele caracteristici probabilistice de performanță ale sistemelor de așteptare.

În cadrul realizării prezentei teze s-a propus de a atinge următorul **scop**: extinderea rezultatelor cunoscute din domeniul Teoriei Așteptării, elaborarea a noi tehnici și algoritmi numerici de determinare a unor caracteristici de performanță mai optime pentru modelele de așteptare Polling cu vacanțe semi-Markoviene și pentru cele cu prioritatea DD.

Pentru atingerea scopului propus s-au realizat următoarele obiective:

- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției lungimii virtuale a șirului de așteptare pentru sistemele Polling cu întârzieri semi-Markoviene;
- formalizarea caracteristicilor probabiliste de performanță pentru sistemele generalizate de așteptare cu prioritatea DD;
- elaborarea și aplicarea algoritmilor numerici pentru modelarea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele Polling cu prioritatea DD;
- implementarea algoritmilor elaborați în limbaje de programare în vederea estimării parametrilor funcțiilor de repartiție ce intervin în optimizarea caracteristicilor de performanță ale modelelor de așteptare Polling.

Suportul metodologic al cercetărilor este bazat pe noțiuni din teoria așteptării, teoria probabilităților și statistica matematică, metode ale teoriei proceselor aleatoare, ș.a. Algoritmii numerici elaborați sunt implementați în limbajele de programare C++ și Kotlin.

Problema de cercetare. Lucrarea este dedicată cercetării modelelor Polling cu priorități, vacanțe semi-Markoviene și servire exhaustivă. În continuare ne vom referi la două particularități ale modelelor studiate în teză, care sunt esențial noi și care atestă avantajele acestor modele față de cele cunoscute. Prima particularitate constă în determinarea repartiției lungimii medii virtuale a șirului de așteptare Polling cu servire exhaustivă pentru diferite funcții de repartiție. De asemenea, în cadrul acestor modele este considerat timpul de „vacanță” în procesul de servire. Acest timp, inevitabil și obiectiv există în sistemele reale, și este formalizat și precăutat în modelele din teză ca o variabilă aleatoare continuă cu funcția de repartiție arbitrar dată.

A două particularitate constă în studierea modelelor generalizate de așteptare cu disciplina DD. Astfel, se presupune că numărul de clase de prioritate este arbitrar, și în al doilea rând, se presupune că procesul de servire al trecerii de la o clasă de cerințe la alta necesită ceva timp pentru această trecere. Folosind rezultatele modelelor analizate, de exemplu, în practica de proiectare tehnică, aceasta are o importantă semnificativă, deoarece ea ne oferă posibilitatea de a

obține o evaluare mai exactă despre funcționarea sistemelor reale.

Cu toate acestea, aplicarea rezultatelor analizate implică dificultăți considerabile, principala dintre ele - complexitatea rezultatelor, "incompetența" lor de utilizare directă în aplicații. Într-adevăr, după cum poate fi observat, rezultatele obținute sunt exprimate, de regulă, în termenii de transformate Laplace sau Laplace-Stieltjes, deseori în termeni de sisteme de ecuații funcționale recurente. Pentru a determina, de exemplu, $\Pi_k(t)$, trebuie să rezolvăm unele ecuații funcționale, apoi să luăm inversa transformatei Laplace-Stieltjes. Chiar și pentru determinarea unei caracteristici astfel ca valoarea medie a perioadei de ocupare - nu este o sarcină atât de simplă, deoarece pentru determinarea ei este necesar de a calcula valoarea transformatei Laplace-Stieltjes a funcției $\pi_k(s)$ în unele puncte. Depășirea acestor dificultăți constă în elaborarea algoritmilor numerici și elaborarea soft-ului specializat.

Dirrecțiile de soluționare constau în:

- a) formalizarea matematică a timpului de trecere de la o clasă la alta, ca variabilă aleatoare de tip semi-Markov;
- b) extinderea modelelor clasice de la două clase de priorități la un număr arbitrar r de clase;
- c) obținerea caracteristicilor de performanță;
- d) elaborarea algoritmilor numerici, necesari pentru modelarea practică;
- e) implementarea algoritmilor în limbaje moderne de programare.

2. METODE ȘI CARACTERISTICI DE PERFORMANȚĂ ÎN EVOLUȚIA MODELELOR POLLING CU SERVIRE EXHAUSTIVĂ ȘI VACANȚE

În acest capitol se analizează unele metode și procedee bazate pe aparatul funcțiilor generatoare, transformatelor Laplace și Laplace-Stieltjes, sunt prezentate unele proprietăți și noțiuni, astfel sunt descrise unele metode analitice și numerice de cercetare cu o bogată istorie de succes în obținerea a noi rezultate în Teoria Așteptării [10]. De asemenea, sunt prezentate rezultate de bază cu privire la unele caracteristici probabilistice de performanță pentru modelele de așteptare de tip Polling cu vacanțe semi-Markoviene. Sunt propuși unii algoritmi numerici și exemple pentru determinarea repartiției lungimii șirului de așteptare Polling cu servire exhaustivă pentru diferite funcții de repartiție.

2.1. Metode analitice: metoda funcțiilor generatoare și metoda catastrofelor

Instrumentele analitice din Teoria Așteptării includ metodele: metoda funcțiilor generatoare (metoda de colorare) și metoda „catastrofelor”.

Metoda de colorare (marcare) [7, 98] este o metodă eficientă în cercetarea problemelor Teoriei Așteptării. Ideea principală a acestei metode constă în atribuirea funcției generatoare un anumit sens probabilistic și aceasta se obține prin procedura de colorare (marcare) a mesajelor de intrare în sistemul de servire. Astfel, structura matematică abstractă definită ca funcție generatoare, datorită sensului ei probabilistic devine mai comodă și mai pe înțeles în problemele aplicative. Mai mult de cât atât, datorită acestei metode, deseori este posibil de obținut expresii analitice pentru funcția generatoare reieșind din sensul ei probabilist. Și aceasta este posibil fără a se ști funcția de repartiție a variabilei aleatoare. Tot așa cum se întâmplă și cu valorile numerice ale variabilei aleatoare, valoarea medie, dispersia etc. Cele menționate prezintă un deosebit interes pentru problemele aplicative, oferindu-ne noi posibilități în cercetarea lor.

Dat fiind faptul că metoda respectivă implică noțiunea de funcție generatoare, vom da o scurtă definiția a acesteia și unele proprietăți.

Fie dată variabila aleatoare discretă X domeniul de valori a căreia este

$$\{0, 1, 2, \dots\} \text{ și } P_k = P\{X = k\}, \sum_{k=0}^{\infty} P_k = 1.$$

Funcția generatoare $P(z)$ a variabilei aleatoare X se definește astfel:

$$P(z) = Mz^X = \sum_{k=0}^{\infty} P_k z^k,$$

unde $|z| \leq 1$.

Proprietățile principale ale funcției generatoare sunt:

a) dacă $P(z)$ este funcție generatoare a variabilei aleatoare X atunci

$$P_k = P\{X=k\} = \frac{1}{k!} \frac{\partial^k}{\partial z^k} P(z) \Big|_{z=0} = \frac{1}{k!} P^{(k)}(z) \Big|_{z=0} \quad (2.1)$$

Această proprietate ne permite obținerea repartiției P_k , având dată expresia analitică pentru funcția generatoare.

b) dacă $P(z)$ este funcție generatoare a variabilei aleatoare X atunci,

$$M(X) = P'(1) \quad (2.2)$$

$$D'(X) = P''(1) + P'(1)(1 - P'(1)) \quad (2.3)$$

unde $M(X)$ și $D(X)$ reprezintă valoarea medie și dispersia variabilei aleatoare X respectiv,

$$M(X) = \sum_{k=0}^{\infty} k P_k \quad (2.4)$$

$$D(X) = M(X - M(X))^2$$

Astfel, având dată, spre exemplu, expresia analitică a funcției $P(z)$ ușor putem obține conform formulei (2.2) valoarea medie a variabilei aleatoare X . Într-adevăr

$$P'(z) = \sum_{k=0}^{\infty} k z^{k-1} P_k$$

și considerând în continuare $z = 1$, obținem

$$P'(1) = \sum_{k=0}^{\infty} k P_k,$$

ceea ce coincide cu (2.4).

c) Fie variabila aleatoare Y reprezintă suma variabilelor aleatoare X_1 și X_2 , $Y = X_1 + X_2$, iar prin $P(z)$, $P_1(z)$ și $P_2(z)$ sunt notate funcțiile generatoare ale variabilelor Y , X_1 și X_2 , respectiv.

Atunci are loc relația:

$$P(z) = P_1(z) \cdot P_2(z).$$

În continuare vom precăuta metoda de colorare. Fie că variabila aleatoare X reprezintă numărul de cerințe sosite în sistemul de așteptare în careva interval concret de timp.

Vom considera că fiecare din cerințele sosite se colorează în roșu cu probabilitatea z ($0 \leq z \leq 1$) sau în albastru cu probabilitatea $1-z$, independent de faptul cum au fost colorate restul cerințelor. Atunci $P_k z^k$ va reprezenta probabilitatea că k cerințe sosite în intervalul

precăutat sunt cerințe roșii, iar $\sum_{k=0}^{\infty} P_k z^k$ va fi nu altceva decât probabilitatea că în intervalul de

timp precăutat vor fi înregistrate doar cerințe roșii.

Deoarece $P(z) = \sum_{k=0}^{\infty} P_k z^k$, unde funcția generatoare $P(z)$ reprezintă probabilitatea că în

intervalul de timp precăutat în sistem au sosit doar cerințe roșii, sau cu alte cuvinte, probabilitatea că în intervalul precăutat în sistem nu a sosit nici o cerință albastră.

Astfel, pentru determinarea funcției generatoare $P(z)$ poate fi aplicat sensul ei probabilistic obținut prin introducerea unui eveniment aleatoriu suplimentar: colorarea (marcarea, vopsirea) cerințelor. În aceasta și constă metoda de colorare. Deoarece $P(z)$ este funcție analitică, aplicând principiul de extindere analitică putem afirma că concluziile obținute pentru $0 \leq z \leq 1$ sunt valabile și pentru domeniul $|z| \leq 1$.

O altă metodă de cercetare cu o bogată istorie de succes în obținerea a noi rezultate în Teoria Așteptării este metoda „catastrofelor”, sau, cu alte cuvinte, metoda introducerii unui eveniment aleatoriu suplimentar [5, 98]. Esența metodei „catastrofelor” constă în faptul că introducând un eveniment suplimentar („catastrofă”) se reușește de atribuit un sens probabilist clar transformatelor Laplace și Laplace-Stieltjes, după ce se precaută evoluția sistemului de așteptare și se determină aceste probabilități, aceasta ne permite să evităm anumite structuri complicate.

Vom nota prin A durata „vieții” a unui careva element, fie a unei siguranțe, iar prin $A(t)$ funcția sa de repartiție. Fie că independent de durata „vieții” se produc careva evenimente, pe care le vom numi „catastrofe” și care formează un flux Poisson cu parametrul $s > 0$. Atunci numărul

$$\alpha(s) = \int_0^{\infty} e^{-st} dA(t), \quad (2.5)$$

este probabilitatea că în durata „vieții” nu s-a produs evenimentul „catastrofă”. Într-adevăr, conform proprietăților fluxului Poisson, probabilitatea $P_n(t)$ că în intervalul $[0, t)$ vor fi înregistrate n mesaje a fluxului Poisson cu parametrul s , este

$$P_n(t) = \frac{(st)^n}{n!} e^{-st}.$$

Din ultima expresie, pentru $n=0$ avem $P_0(t) = e^{-st}$. Cu alte cuvinte e^{-st} este probabilitatea că în $[0, t)$ nu a fost înregistrat (nu s-a produs) nici un mesaj al fluxului Poisson de „catastrofe”. Pe de altă parte conform definiției integralei Stieltjes avem:

$$dA(t) = P\{A \in [t, t + dt)\}.$$

Astfel partea dreaptă a formulei (2.5) este probabilitatea că în timpul „vieții” elementului

nu s-a produs nici un eveniment al fluxului de „catastrofe”, sau, cu alte cuvinte, nu s-a produs nici o „catastrofă”. În aceasta și constă metoda „catastrofelor”. Datorită introducerii unui eveniment aleatoriu suplimentar „catastrofă”, care evident, nicicum nu influențează asupra duratei „vieții” elementului precăutat, se reușește de atribuit transformatei Laplace-Stieltjes un sens probabilistic bine determinat. Deoarece s a fost considerată o valoare arbitrară, concluzia este valabilă pentru orice $s > 0$.

2.2. Metode numerice: metoda aproximațiilor succesive și metoda modelărilor numerice

Rezultatele analitice pentru sistemele de așteptare deseori implică utilizarea transformatelor Laplace și Laplace-Stieltjes. Multe din aceste rezultate sunt obținute sub formă de sisteme de ecuații funcționale, ceea ce complică obținerea acestor transformate, astfel se recurge la metodele numerice pentru soluționarea acestei probleme [8].

Metoda numerică (algoritmul numeric) este o metodă de rezolvare a unei probleme practice utilizând un număr finit de operații aritmetice și logice (operațiile uzuale pe care le poate executa un procesor sau coprocesor matematic). În practică apar probleme concrete cu date de intrare cunoscute. De obicei, acestei probleme se asociază un model matematic, mai fin sau mai puțin fin. Soluționarea problemei matematice, în general, nu se poate rezolva manual printr-un număr finit de pași (operații), deci se caută să se rezolve problema printr-o metodă numerică. Algoritmul obținut se poate programa într-un limbaj de programare, iar rezultatele obținute la compilare se verifică practic. Aceste date de ieșire ar trebui să fie o aproximare reală pentru problema practică inițială. Schematic aceasta are următoarea reprezentare:

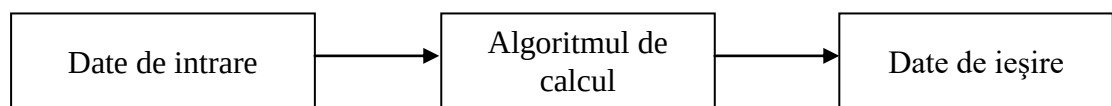


Fig. 2.1. Etapele modelării numerice

Metoda aproximațiilor succesive, după cum reiese și din denumirea ei, determină o soluție aproximativă a unei ecuații neliniare prin construirea unui șir de aproximații succesive.

Modelul exhaustiv Polling este format dintr-un singur server și r șiruri de așteptare. Mecanismul de servire este dat de tabelul Polling $f : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, r\}$, unde funcția f arată că la etapa $j, j = \overline{1, n}$, sunt servite cerințele, care se află în șirul de așteptare $k, k = \overline{1, r}$. Aceste cerințe, numite și cerințe de clasă k , sosesc după legea de repartiție Poisson cu parametrul λ_k . Timpul de servire pentru cerințele de clasă k este o variabilă aleatoare B_k cu

funcția de repartiție arbitrară $B_k(x) = P\{B_k < x\}$. Timpul de orientare de la un șir de așteptare către șirul de așteptare k este considerat o variabilă aleatoare C_k cu următoarea funcție de repartiție arbitrară $C_k(x) = P\{C_k < x\}$.

Vom nota prin k - perioada de ocupare intervalul de timp, care începe cu schimbul serverului către șirul de așteptare k și sfârșește când acest șir devine liber de cerințe de clasă k .

Fie că prin Π_k^δ este notată lungimea acestei k -perioade de ocupare și prin $\Pi_k^\delta(x) = P\{\Pi_k^\delta < x\}$ - funcția ei de repartiție.

Vom considera $\pi_k^\delta(s) = \int_0^\infty e^{-st} d\Pi_k^\delta(x)$ transformata Laplace-Stieltjes a funcției de repartiție a k -perioadei de ocupare.

Are loc următorul rezultat [60]:

Teorema 2.1 Transformata Laplace-Stieltjes a funcției de repartiție a k -perioadei de ocupare $\pi_k^\delta(s)$ se determină din ecuația funcțională

$$\pi_k^\delta(s) = c_k(s + \lambda_k - \lambda_k \pi_k(s)) \pi_k(s), \quad (2.6)$$

unde

$$\pi_k(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k(s)), \quad (2.7)$$

iar prin $c_k(s)$ și $\beta_k(s)$ sunt notate transformatele Laplace-Stieltjes ale funcțiilor de repartiție

$$C_k(x) \text{ și } B_k(x), \text{ respectiv, deci } c_k(s) = \int_0^\infty e^{-sx} dC_k(x) \text{ și } \beta_k(s) = \int_0^\infty e^{-sx} dB_k(x).$$

Prin $L_k(t)$ vom nota lungimea medie a șirului de cerințe în momentul t pentru utilizatorul k , iar prin $l_k(s)$ - transformata Laplace a funcției $L_k(t)$,

$$l_k(s) = \int_0^\infty e^{-st} L_k(t) dt,$$

atunci obținem

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^\delta(s)} \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^\delta(s))}{s^2(1 - \beta_k(s))} - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^\delta(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Vom observa, că dacă ecuația clasică (2.6) este formulată doar pentru regimul staționar, în cazul dat această ecuație (mai precis analogul ei) este obținută pentru sistemul Polling și pentru regimul virtual, ceea ce este deosebit de important deoarece cu ajutorul ei este descris

comportarea sistemului pentru orice timp t . Observăm, că pentru determinarea funcției $l_k(s)$ este necesar de soluționat ecuația funcțională (2.7).

Această metodă în contextul modelului Polling și a ecuației Kendall formulate mai sus, prevede realizarea următorului algoritm:

Date de intrare: $\{\lambda_k\}_{k=1}^r; \{b_k\}_{k=1}^r; \{\bar{c}_k\}_{k=1}^r; s; r; \varepsilon > 0$.

Date de ieșire: $k; \{\pi_k(s)\}_{k=1}^r; \{\pi_k^\delta(s)\}_{k=1}^r; \{l_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ și $C_k(x)$ sunt ambele funcții de repartiție Exponențiale.

$$\beta_k(s) = \frac{b_k}{s + b_k}; \quad c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

- b) Se calculează repartiția perioadei de ocupare $\pi_k^{\delta(n)}(s)$ conform relațiilor:

Pentru $n=0$, avem $\pi_k^{(0)}(s) = 0$.

$$\pi_k^{\delta(n)}(s) = c_k(s + \lambda_k - \lambda_k \pi_k^{(n)}(s)) \pi_k^{(n)}(s),$$

$$\pi_k^{(n)}(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k^{(n-1)}(s)).$$

- c) Se calculează repartiția lungimii șirului de așteptare $l_k(s)$, conform formulei:

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^{\delta(n)}(s)} \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^{\delta(n)}(s))}{s^2 (1 - \beta_k(s))} - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^{\delta(n)}(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Condiția de oprire: $|\pi_k^{(n)}(s) - \pi_k^{(n-1)}(s)| < \varepsilon$.

În cele ce urmează vom analiza metoda modelărilor numerice. În linii generale, conceptual A. Samarschii [103] a propus următoarea triadă:

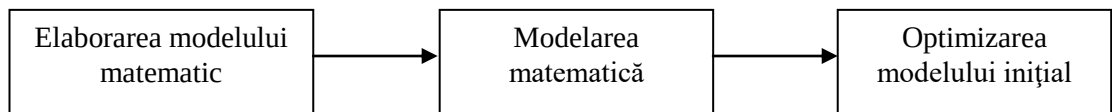


Fig. 2.2. Schema generală a conceptului Samarschi

Această metodă prevede:

- a) *elaborarea modelului matematic*, astfel modelul matematic constă din anumite formalizări matematice (descrieri în limbajul matematicii) a unor situații reale, soluționarea căreia este solicitată. Modelul matematic preia într-un limbaj generalizat cele mai evidențiate particularități ale evoluției situației reale.
- b) *modelarea matematică* poate fi analitică și numerică. Modelarea analitică a problemei precăutate prevede cercetarea analitică (deductivă) conform anumitor axiome, legături, exprimate prin ecuații de anumit tip și ordin (liniare, funcționale, integrale, sisteme de astfel de ecuații, etc.). Modelarea numerică apare, ca regulă, acolo unde aparatul analitic devine nesatisfăcător sau imposibil pentru soluționarea definitivă a problemei precăutate. Astfel, spre exemplu, se demonstrează că ecuația respectivă are soluție unică, însă această soluție nu poate fi obținută prin metodele analitice elaborate. Atunci se aplică anumite metode numerice în baza cărora se elaborează algoritmi numerici de calcul.
- c) *optimizarea modelului* inițial prevede o reconsiderare a modelului matematic anterior elaborat. Această reconsiderare este, de fapt, o precizare argumentată obținută în baza modelării. Ea permite „îmbunătățirea” modelului inițial, uneori în scopul descrierii mai adecvate a problemei reale, alteori în scopul detalizării sau evidențierii anumitor laturi ale modelului.

Triada menționată mai sus, în anumite probleme aplicative poate deveni un suport cheie deosebit de efectiv. Spre exemplu, compania americană General Electric a elaborat un sistem de testare a calității producerii numit „Six Sigma Quality”, implementarea căruia aduce acestei companii anual profit de bilioane dolari. Sistemul „Six Sigma Quality” se bazează pe un rezultat analitic din statistica matematică, cunoscut ca „regula celor $3\text{-}\sigma$ ”, și care permite să mențină calitatea produselor elaborate în anumite limite apriori specificate. Acest sistem invocă 4 etape: *Measure* $\Rightarrow$ *Analyze* $\Rightarrow$ *Improve* $\Rightarrow$ *Control*. În urma îndeplinirii riguroase a fiecărei etape, se obține un rezultat final impunător. De exemplu, la nivelul $\sigma = 6$, se garantează că dintr-un milion de piese produse, vor fi nu mai mult de 34 piese rebut. Acesta este un exemplu de aplicare cu succes a modelării matematice. Vom observa că etapele *Measure* și *Analyze* corespund în triada Samarschii cu blocul *Elaborarea modelului matematic*.

2.3. Repartiția virtuală și staționară a lungimii șirului de așteptare pentru modelul Polling cu servire exhaustivă

Una din caracteristicile importante, des folosite în analiza modelelor matematice ale fenomenelor de așteptare este repartiția șirului de așteptare, sau cu alte cuvinte repartiția

numărului de cerințe în șirul de așteptare. Funcția de repartiție pentru această variabilă discretă X , după cum se știe, se definește ca $P_n = P\{X = n\}$, $n = 0, 1, 2, \dots$.

În ultimul deceniu în domeniul de cercetare tot mai des se folosește repartiția virtuală a șirului de așteptare. Ce este repartiția virtuală? Mai sus s-a notat prin X numărul de cerințe în șirul de așteptare pentru regimul staționar de funcționare a sistemului, adică pentru un timp t care corespunde regimului staționar. Dar cum procedăm dacă este necesar de determinat repartiția șirului de așteptare pentru orice t ? Vom descrie succint această procedură.

În teoria clasică a așteptării repartiția șirului de așteptare se determină prin intermediul funcției generatoare $P(z) = \sum_n P_n z^n$, $0 \leq z \leq 1$. De exemplu, dacă se analizează sistemul clasic $M|G|1$ atunci poate fi demonstrat că regimul staționar va avea loc, dacă se îndeplinește inegalitatea $\lambda * \beta_1 < 1$, unde λ este parametrul fluxului Poisson de intrare și $\beta_1 = \int_0^\infty x dB(x)$ este valoarea medie a timpului de servire, iar $B(x) = P\{B < x\}$ este funcția de repartiție a timpului de servire a cerințelor. Pentru sistemul clasic menționat, în condiția că sistemul se află în regim staționar, funcția generatoare se determină conform expresiei pentru funcția generatoare care se numește ecuația Pollaczek-Khinchin, după numele matematicienilor Pollaczek și Khinchin [100], care au obținut de prima dată (independent unul față de altul) această ecuație, care în literatura engleză poartă denumirea de „transform equation” este nu altceva decât funcția generatoare a repartiției șirului de așteptare pentru regimul staționar al evoluției sistemului.

În cele ce urmează se vor prezenta repartițiile virtuale și staționare ale șirului de așteptare pentru modelul Polling cu servire exhaustivă și schimb nenul al stărilor de trecere de la o clasă de utilizatori la alta. Rezultatele analitice sunt obținute în termeni de transformate Laplace, Laplace-Stieltjes și funcții generatoare [61].

- **Analogul virtual al ecuației Pollaczek-Khintchin**

Vom nota prin $P_m(t)$ probabilitatea că în momentul de timp t în șirul utilizatorului k se află m – mesaje. Fie $P_k(z, t)$ funcția generatoare ale acestor probabilități.

$$P_k(z, t) = \sum_{k=1}^{\infty} P_k(t) z^k, \quad 0 \leq z \leq 1.$$

(2.8)

Vom colora (marca) la întâmplare mesajele din șirul utilizatorului k . Un mesaj arbitrar din șirul k se va colora în roșu cu probabilitatea z sau în albastru cu probabilitatea $1-z$. Vom observa

că relația (2.8) este probabilitatea că în timpul t în șirul k se află doar mesaje roșii.

Vom nota prin $p_k(z, s)$ transformata Laplace a funcției generatoare

$$p_k(z, s) = \int_0^{\infty} e^{-st} P_k(z, t) dt. \quad (2.9)$$

În continuare vom presupune că independent de funcționarea sistemului de așteptare se produc careva evenimente aleatoare, pe care le vom numi ”catastrofe”, și care formează un flux Poisson cu parametrul $s > 0$.

Vom înmulți ambele părți ale egalității (2.9) la s :

$$s \cdot p_k(z, s) = \int_0^{\infty} s e^{-st} P_k(z, t) dt. \quad (2.10)$$

Vom observa că partea dreaptă a relației (2.10) este probabilitatea că prima ”catastrofă” s-a produs în momentul de timp t , când în șirul k se aflau doar mesaje roșii. Într-adevăr, cum s-a menționat mai sus, funcția $P_k(z, t)$ este probabilitatea că în momentul de timp t în șirul k se aflau doar mesaje roșii, e^{-st} este probabilitatea că până la momentul de timp t nu s-a produs evenimentul ”catastrofă”, iar $s dt$ este probabilitatea că prima ”catastrofă” s-a produs în vecinătatea punctului t .

În acest mod, $s \cdot p_k(z, s)$ (transformata Laplace a funcției generatoare $p_k(z, s)$ înmulțită la s) obține următorul sens probabilist: $s p_k(z, s)$ este probabilitatea că prima ”catastrofă” s-a produs în momentul de timp t când în șirul k se aflau doar mesaje roșii.

Analogic, dacă vom nota prin $\Pi_k^{\delta}(z, t)$ și $B_k(z, t)$ funcțiile generatoare pentru o Π_k^{δ} - perioadă și o servire B_k luate aparte, vom obține că $s \pi_k^{\delta}(z, s)$ și $s \beta_k(z, s)$ sunt probabilitățile că prima ”catastrofă” s-a produs pe perioada Π_k^{δ} luată aparte și pe perioada B_k luată aparte, respectiv. Folosind sensul probabilist menționat mai sus, cum și unele noțiuni ale proceselor de regenerare, se demonstrează următoarea teoremă:

Teorema 2.2 Transformata Laplace a funcției generatoare $P_k(z, t)$ este

$$p_k(z, s) = \frac{1 + \lambda_k \pi_k^{\delta}(z, s)}{s + \lambda_k - \lambda_k \pi_k^{\delta}(s)}, \quad (2.11)$$

unde

$$\pi_k^{\delta}(z, s) = \frac{1 - c_k(s + \lambda_k - \lambda_k z)}{s + \lambda_k - \lambda_k z} + \frac{\beta_k(z, s)[z c_k(s + \lambda_k - \lambda_k z) - \pi_k^{\delta}(s)]}{z - \beta_k(s + \lambda_k - \lambda_k z)}, \quad (2.12)$$

$$\beta_k(z, s) = \frac{1 - \beta_k(s + \lambda_k - \lambda_k z)}{s + \lambda_k - \lambda_k z}. \quad (2.13)$$

Remarca 2.1 Vom observa că formulele (2.12) și (2.13) au și un sens independent. Adică, relația (2.12) nu este altceva decât repartiția (în termenii de transformată Laplace) a numărului de mesaje de clasa k pe o Π_k^δ perioadă de timp luată aparte, iar formula (2.13) - repartiția (în termenii de transformată Laplace) a numărului de mesaje de clasa k pe o perioadă de servire B_k luată aparte.

- **Ecuția Pollaczek-Khintchin pentru regimul Polling staționar**

Fie că are loc regimul staționar (steady state) și $\lambda_k \beta_{k1} < 1$, $\lambda_k c_{k1} < 1$, unde

$$\beta_{k1} = \int_0^\infty t dB_k(t), \quad c_{k1} = \int_0^\infty t dC_k(t),$$

iar $B_k(t) = P\{B_k < t\}$ este funcția de repartiție a duratei de servire a mesajelor de clasa k , și $C_k(t) = P\{C_k < t\}$ este funcția de repartiție a timpului de schimb către clasa de mesaje k . Folosind metodologia lanțurilor Markov poate fi demonstrat că, în acest caz, există $\lim_{t \rightarrow \infty} P_k(t)$ și $\lim_{t \rightarrow \infty} P_k(z, t)$.

Notăm prin $P_k(z) = \lim_{t \rightarrow \infty} P_k(z, t)$. Conform teoremei Tauber [96] avem

$$P_k(z) = \lim_{t \rightarrow \infty} P_k(z, t) = \lim_{s \downarrow 0} s p_k(z, s).$$

sau, conform Teoremei 2.2,

$$P_k(z) = \lim_{s \downarrow 0} \frac{s(1 + \lambda_k \pi_k^\delta(z, s))}{s + \lambda_k - \lambda_k \pi_k^\delta(s)}. \quad (2.14)$$

Pentru soluționarea nedeterminării de tip $\frac{0}{0}$, apărută în relația (2.14), folosim regula lui

L'Hospital. Astfel avem:

$$P_k(z) = \lim_{s \downarrow 0} \frac{1 + \lambda_k \pi_k^\delta(z, s)' + s \pi_k^{\delta'}(z, s)}{1 - \lambda_k \pi_k^{\delta'}(s)}$$

și definitiv obținem

Teorema 2.3 Dacă $\lambda_k \beta_{k1} < 1$ și $\lambda_k c_{k1} < 1$, atunci

$$P_k(z) = \frac{1 + \lambda_k \pi_k^\delta(z, 0)}{1 + \lambda_k \pi_{k1}^\delta}. \quad (2.15)$$

Funcția π_{k1}^δ este primul moment al perioadei de ocupare Π_k^δ , care a fost obținută ulterior [100], astfel avem:

$$\pi_{k1}^\delta = \frac{\beta_{k1}}{1 - \lambda_k \beta_{k1}} + \frac{c_{k1}}{1 - \lambda_k c_{k1}} \quad (2.16)$$

însă $\pi_k^\delta(z, 0)$ se determină din relația (2.12) pentru $s = 0$.

• Ecuația clasică Pollaczek-Khintchin

În acest paragraf vom arăta că formula (2.14) conține ca caz particular renumita formula clasică, cunoscută ca ecuația transformată Pollaczek-Khintchin. Vom precăuta cazul particular când timpul de schimb este egal cu zero, $C_k = 0$, iar numărul de clase de utilizatori este egal cu 1, $k=1$. În acest caz, obținem că π_{k1}^δ din relația (2.15) se exprimă astfel

$$\pi_{k1}^\delta = \pi_{11}^\delta = \frac{\beta_{11}}{1 - \lambda_1 \beta_{11}}. \quad (2.17)$$

Parametrul $\pi_1^\delta(z, 0)$ se determină din relația (2.12), respectiv pentru $C_k = 0$, $s = 0$ și $k = 1$, astfel obținem

$$\pi_1^\delta(z, 0) = \beta_1(z, 0) \frac{z - \pi_1^\delta(0)}{z - \beta_1(\lambda_1 - \lambda_1 z)} \quad (2.18)$$

Funcția $\beta_1(z, 0)$ se obține din (2.13) pentru $s = 0$ și $k = 1$, obținem

$$\beta_1(z, 0) = z \frac{1 - \beta_1(z - \lambda_1 z)}{\lambda_1 - \lambda_1 z} \quad (2.19)$$

Substituind expresia (2.19) în relația (2.18), după anumite transformări, obținem

$$\pi_1^\delta(z, 0) = - \frac{z(1 - \beta_1(z - \lambda_1 z))}{\lambda_1(z - \beta_1(\lambda_1 - \lambda_1 z))} \quad (2.20)$$

Expresiile (2.17) și (2.20) obținute pentru cazul particular studiat ale lui π_{k1}^δ și $\pi_k^\delta(z, 0)$ le înlocuim în relația (2.15) și după efectuarea transformărilor necesare, obținem

$$P_1(z) = \frac{\beta_1(\lambda_1 - \lambda_1 z)(z - 1)(1 - \lambda_1 \beta_{11})}{z - \beta_1(\lambda_1 - \lambda_1 z)} \quad (2.21)$$

Astfel, dacă trecem la notațiile clasice (cu alte cuvinte, dacă ometem indicele din formula (2.21)) obținem următorul rezultat:

Teorema 2.4

$$P(z) = \frac{\beta(\lambda - \lambda z)(z - 1)(1 - \lambda \beta_1)}{z - \beta(\lambda - \lambda z)} \quad (2.22)$$

Formula (2.22) este prezentată în majoritatea monografiilor și manualelor din domeniul Teorii Așteptării și este cunoscută ca ecuația transformată (transform equation) Pollaczek-Khintchin [73, 97]. Ea a fost obținută, pentru modelul clasic M|G|1 cu cerințe omogene, de fondatorii Teorii Așteptării - matematicienii Pollaczek în anul 1961, și respectiv Khintchin în anul 1963.

Remarca 2.2 Rezultatele menționate mai sus, din Teorema 2.2 și Teorema 2.3 pot fi considerate ca generalizarea acestei ecuației pentru regimul nestaționar (virtual) și staționar al sistemului Polling utilizat în rețele fără fir.

- **Repartiția virtuală a șirului de așteptare**

Repartiția virtuală implică repartiția perioadei de ocupare, care poate fi considerată și ca o caracteristică aparte.

Reamintim unele notații și rezultate referitor la modelul Polling cu servire exhaustivă și schimb nenul al stărilor de trecere de la o clasă de utilizatori la alta [61].

Mecanismul de servire este dat de tabelul Polling: $f: \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, r\}$, unde aplicația $f(j)=k$ denotă că la etapa j este servit utilizatorul cu numărul k . Considerăm că cerințele de la utilizatorul k sosesc conform fluxului Poisson cu intensitatea λ_k .

Funcția de repartiție a servirilor cerințelor al utilizatorului cu numărul k o vom nota prin $B_k(x) = P\{B_k < x\}$. Vom considera că timpul de schimb C_j depinde doar de indicele etapei către care se produce schimbul, adică $C_{f(j-1),j} = C_j$. Funcția de repartiție a perioadelor de schimb o vom nota prin $C_j(x) = P\{C_k < x\}$.

Prin k -perioadă de ocupare vom considera intervalul de timp ce începe cu schimbul sistemului către utilizatorul k și sfârșește când sistemul devine liber de cerințe de clasă k .

Fie că prin Π_k^δ este notată lungimea acestei k – perioadei de ocupare, iar prin $\Pi_k^\delta(x) = P\{\Pi_k^\delta < x\}$ – funcția ei de repartiție.

Fie, în continuare, că

$$\pi_k^\delta(s) = \int_0^\infty e^{-st} d\Pi_k^\delta(x)$$

este TLS a funcției de repartiție a k – perioadei.

Are loc următorul rezultat [94]:

Teorema 2.5 Funcția $\pi_k^\delta(s)$ se determină din ecuația

$$\pi_k^\delta(s) = c_k(s + \lambda_k - \lambda_k \pi_k(s)) \pi_k(s), \quad (2.23)$$

unde

$$\pi_k(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k(s)),$$

iar prin $c_k(s)$ și $\beta_k(s)$ sunt notate transformatele Laplace-Stieltjes ale funcțiilor de repartiție $C_k(x)$ și $B_k(x)$, astfel

$$c_k(s) = \int_0^\infty e^{-sx} dC_k(x), \quad \beta_k(s) = \int_0^\infty e^{-sx} dB_k(x).$$

Pentru determinarea lungimii nestăționare (virtuale) a șirului de așteptare a mesajelor poate fi utilizată Teorema 2.2. Într-adevăr, fie $L_k(t)$ valoarea medie virtuală a lungimii șirului de așteptare pentru utilizatorul k , iar $l_k(s)$ - transformata Laplace a funcției $L_k(t)$, atunci:

$$l_k(s) = \int_0^\infty e^{-st} L_k(t) dt.$$

Este ușor de demonstrat că are loc relația:

$$-l_k(s) \Big|_{s=0} = L_k(t).$$

Astfel, dacă aplicăm această relație din Teorema 2.2 (după efectuarea derivării după s , schimbarea semnelui și plasarea pentru $s=0$), obținem următoarea expresie analitică pentru $l_k(s)$:

Teorema 2.6 Transformata Laplace a funcției $L_k(t)$ este:

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^\delta(s)} \times \\ \times \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^\delta(s))}{s^2 (1 - \beta_k(s))} - \right. \\ \left. - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^\delta(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Remarca 2.3 Soluție analitică exactă pentru $l_k(s)$ nu există, deoarece conform expresiilor (2.11) și (2.12), pentru obținerea lui $l_k(s)$ este necesar de avut soluția pentru ecuația funcțională $\pi_k^\delta(s)$ din relația (2.23), care conform [100], nu dispune de soluție analitică exactă. Însă, $l_k(s)$ poate fi determinată numeric. În acest scop au fost elaborați algoritmi numerici care permit obținerea valorii numerice a lui $l_k(s)$ cu exactitatea cerută. Algoritmii elaborați se bazează pe soluționarea numerică a ecuației generalizate Pollaczek-Khintchin.

În continuare sunt prezentați algoritmi și modelări numerice pentru determinarea valorii lui $l_k(s)$ pentru diferite funcții de repartiție a timpului de servire a cerințelor, respectiv timpul de orientare. În continuare vom face o descriere succintă a unor funcții de repartiție care au fost

utilizate în cadrul algoritmilor numerici și al soft-urilor. Astfel:

- Repartiția exponențială, $\text{Exp}(\lambda)$. Variabila aleatoare continuă X este repartizată după o lege exponențială de parametrul $\lambda > 0$, notată $\text{Exp}(\lambda)$, unde densitatea sa este egală cu:

$$f(x) = \begin{cases} 0, & \text{dacă } x < 0, \\ \lambda e^{-\lambda x}, & \text{dacă } x \geq 0. \end{cases}$$

Funcția de repartiție este dată de relația:

$$F(x) = \int_{-\infty}^x f(t)dt = \int_{-\infty}^0 0dt + \int_0^x \lambda e^{-\lambda t} dt = -e^{-\lambda t} \Big|_0^x = 1 - e^{-\lambda x}.$$

Astfel,

$$f(x) = \begin{cases} 0, & \text{dacă } x < 0, \\ 1 - e^{-\lambda x}, & \text{dacă } x \geq 0. \end{cases}$$

Pentru care, valoarea medie este:

$$E(x) = \frac{1}{\lambda}.$$

Iar, transformata Laplace-Stieltjes (TLS) este dată de relația:

$$f(s) = \frac{\lambda}{s + \lambda}.$$

- Repartiție uniformă pe segmentul $[a, b]$, $U(a, b)$. O variabilă aleatoare X este uniform repartizată pe $[a, b]$, notată $U(a, b)$, unde densitatea sa de repartiție este egală cu:

$$f(x) = \begin{cases} \frac{1}{(b-a)}, & \text{dacă } x \in [a, b], \\ 0, & \text{dacă } x \notin [a, b]. \end{cases}$$

Funcția de repartiție este dată de relația:

$$F(x) = \int_{-\infty}^x f(t)dt = \int_a^x \frac{1}{b-a} dt = \frac{x-a}{b-a}, \text{ dacă } x \in [a, b].$$

Dacă $x < a$ sau $x > b$, atunci $f(x)=0$ și $F(x)=0$.

Astfel,

$$F(x) = \begin{cases} 0, & \text{dacă } x < a, \\ \frac{x-a}{b-a}, & \text{dacă } a \leq x \leq b, \\ 1, & \text{dacă } x > b. \end{cases}$$

Repartiția uniformă depinde de parametrii a și b . Pentru variabila aleatoare repartizată uniform X , valoarea medie este dată de expresia:

$$E(x) = \frac{a+b}{2}.$$

Iar, TLS este dată de relația:

$$f(s) = \frac{1}{b-a} \frac{1}{s} (e^{-sa} - e^{-sb}).$$

- Repartiția Erlang, $Erl(\lambda, k) = E_{k, \lambda}$. O variabila aleatoare continuă urmează legea de repartiție Erlang, notată $Erl(\lambda, k)$ are funcția de repartiție egală cu:

$$F(x) = \begin{cases} 0, & \text{dacă } x < 0, \\ \int_0^x \lambda \frac{(\lambda x)^{k-1}}{(k-1)!} e^{-\lambda x} dx, & \text{dacă } x \geq 0. \end{cases}$$

Pentru care, valoarea medie este:

$$E(x) = \frac{k}{\lambda}.$$

Iar, TLS este dată de relația:

$$f(s) = \left(\frac{\lambda}{\lambda + s} \right)^k.$$

- Repartiția normală, $N(a, \sigma^2)$. O variabilă aleatoare continuă urmează legea de repartiție normală de parametri a și σ^2 , notată $N(a, \sigma^2)$, dacă densitatea sa de repartiție este dată de relația:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}, \quad \sigma > 0, \quad a \in R, \text{ unde } x \in R.$$

Funcția de repartiție este dată de expresia:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-a)^2}{2\sigma^2}} dx, \quad -\infty < 0 < \infty.$$

Valoarea medie este:

$$E(x) = a, \quad V(x) = \sigma^2.$$

Iar, TLS este dată de relația:

$$f(s) = e^{\left(\frac{\sigma^2 s}{2} - sa\right)}.$$

Remarca 2.4 Repartiția normală standard $N(0,1)$.

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt, \quad -\infty < 0 < \infty.$$

Pentru care, valoarea medie este:

$$E(x) = a = 0, V(x) = \sigma^2 = 1.$$

Iar, TLS este dată de relația:

$$f(s) = e^{\frac{s}{2}}.$$

2.4. Algoritmi și modelări numerice ale repartiției lungimii șirului de așteptare

Dat fiind că rezultatele analitice au fost obținute în termeni de transformate Laplace, Laplace-Stieltjes și funcții generatoare, au fost elaborate metode și algoritmi numerici pentru determinarea repartițiilor caracteristicilor menționate, pentru diferite funcții clasice de repartiție al timpul de servire al mesajelor de clasă k și al timpul de orientare către șirul de așteptare k , pentru sistemele de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă.

Algoritmii elaborați au fost implementați în limbajul de programare C++, codul sursă a se vedea în Anexa 1.

Algoritmul 2.1

Date de intrare: $\{\lambda_k\}_{k=1}^r; \{b_k\}_{k=1}^r; \{\bar{c}_k\}_{k=1}^r; s; r; \varepsilon > 0$.

Date de ieșire: $k; \{\pi_k(s)\}_{k=1}^r; \{\pi_k^\delta(s)\}_{k=1}^r; \{l_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ și $C_k(x)$ sunt ambele funcții de repartiție exponențiale.

$$\beta_k(s) = \frac{b_k}{s + b_k}; \quad c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

- b) Se calculează repartiția perioadei de ocupare $\pi_k^{\delta(n)}(s)$ conform relațiilor:

Pentru $n=0$, avem $\pi_k^{(0)}(s) = 0$.

$$\pi_k^{\delta(n)}(s) = c_k(s + \lambda_k - \lambda_k \pi_k^{(n)}(s)) \pi_k^{(n)}(s),$$

$$\pi_k^{(n)}(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k^{(n-1)}(s)).$$

c) Se calculează repartiția lungimii șirului de așteptare $l_k(s)$, conform formulei:

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^{\delta(n)}(s)} \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^{\delta(n)}(s))}{s^2(1 - \beta_k(s))} - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^{\delta(n)}(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Condiția de oprire: $|\pi_k^{(n)}(s) - \pi_k^{(n-1)}(s)| < \varepsilon$.

Exemplul 2.1 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 9}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.4, 0.1, 0.5, 0.6, 0.4, 0.2, 0.3, 0.3, 0.2\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.4, 0.1, 0.2, 0.4, 0.1, 0.3, 0.5, 0.1, 0.2\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială $C_k(x) = 1 - e^{-c_k x}$, $x > 0$, cu parametrii $c_k = \{0.6, 0.2, 0.1, 0.2, 0.3, 0.1, 0.4, 0.4, 0.4\}$.

Tabelul 2.1. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.3$)

k	$\pi_k(s)$	$\pi_k^{\delta}(s)$	$l_k(s)$
1	0.498534	0.298945	3.865394
2	0.267857	0.113208	2.870085
3	0.258652	0.038566	3.017352
4	0.421053	0.112676	1.910954
5	0.156794	0.056180	5.328591
6	0.498069	0.124397	2.178971
7	0.611354	0.341255	1.637092
8	0.152873	0.064088	1.259054
9	0.313859	0.149952	0.915321

Exemplul 2.2 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.4, 0.3, 0.6, 0.7, 0.2, 0.1, 0.1, 0.6, 0.7, 0.9\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.1, 0.1, 0.1, 0.4, 0.1, 0.3, 0.2, 0.1, 0.2, 0.3\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială $C_k(x) = 1 - e^{-c_k x}$, $x > 0$, cu parametrii $c_k = \{0.2, 0.3, 0.5, 0.1, 0.4, 0.1, 0.3, 0.5, 0.2, 0.1\}$.

Tabelul 2.2. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.4$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.117218	0.024597	0.321892
2	0.131483	0.041065	0.349156
3	0.095928	0.033252	0.086835
4	0.312131	0.031801	0.049447
5	0.149219	0.061524	0.353467
6	0.394490	0.070367	0.044075
7	0.298438	0.116251	0.137342
8	0.095928	0.033252	0.086835
9	0.169275	0.028654	0.077389
10	0.213025	0.017630	0.192477

Exemplul 2.3 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.3, 0.2, 0.1, 0.3, 0.5, 0.2, 0.4, 0.2, 0.4, 0.5\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.3, 0.1, 0.1, 0.2, 0.2, 0.3, 0.1, 0.2, 0.5, 0.3\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială $C_k(x) = 1 - e^{-c_k x}$, $x > 0$, cu parametrii $c_k = \{0.4, 0.2, 0.1, 0.3, 0.5, 0.3, 0.1, 0.1, 0.3, 0.3\}$.

Rezultatele numerice ale caracteristicilor probabiliste pentru sistemul menționat sunt prezentate în Tabelul 2..

Tabelul 2.3. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.2$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.451415	0.236165	4.985419
2	0.219223	0.078835	4.179151
3	0.267949	0.071797	1.926889
4	0.333332	0.142857	5.356316
5	0.259687	0.121331	5.947426
6	0.499999	0.249999	3.811722
7	0.156930	0.024627	3.106591
8	0.381966	0.090170	2.635105
9	0.574607	0.257226	1.208105
10	0.367543	0.135088	4.348334

Algoritmul 2.2

Date de intrare: $\{\lambda_k\}_{k=1}^r; \{b_k\}_{k=1}^r; \{\bar{c}_k\}_{k=1}^r; \{p_k\}_{k=1}^r; s; r; \varepsilon > 0$.

Date de ieșire: $k; \{\pi_k(s)\}_{k=1}^r; \{\pi_k^{\delta}(s)\}_{k=1}^r; \{l_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ este o funcție de repartiție exponențială și $C_k(x)$ este o funcție de repartiție Erlang.

$$\beta_k(s) = \frac{b_k}{s + b_k}; \quad c_k(s) = \left(\frac{\bar{c}_k}{s + \bar{c}_k} \right)^{p_k}.$$

- b) Se calculează repartiția perioadei de ocupare $\pi_k^{\delta(n)}(s)$ conform relațiilor:

Pentru $n=0$, avem $\pi_k^{(0)}(s) = 0$.

$$\pi_k^{\delta(n)}(s) = c_k(s + \lambda_k - \lambda_k \pi_k^{(n)}(s)) \pi_k^{(n)}(s),$$

$$\pi_k^{(n)}(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k^{(n-1)}(s)).$$

- c) Se calculează repartiția lungimii șirului de așteptare $l_k(s)$, conform formulei:

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^{\delta(n)}(s)} \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^{\delta(n)}(s))}{s^2 (1 - \beta_k(s))} - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^{\delta(n)}(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Condiția de oprire: $|\pi_k^{(n)}(s) - \pi_k^{(n-1)}(s)| < \varepsilon$.

Exemplul 2.4 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.2, 0.4, 0.4, 0.1, 0.6, 0.9, 0.2, 0.4, 3, 0.6\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.2, 0.3, 0.1, 0.2, 0.7, 0.5, 0.3, 0.4, 0.1, 0.4\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este o variabilă

aleatoare cu funcția de repartiție Erlang $C_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x c_k \frac{(c_k x)^{p_k-1}}{(p_k-1)!} e^{-c_k x} dx, & x \geq 0 \end{cases}$, cu parametrii

$c_k = \{0.4, 0.3, 0.7, 0.2, 0.3, 0.6, 0.1, 0.2, 0.3, 0.2\}$ și $p_k = \{2, 3, 4, 2, 7, 9, 4, 3, 9, 4\}$.

Tabelul 2.4. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.2$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.381966	0.116718	3.552115
2	0.406930	0.027421	2.011821
3	0.156930	0.016080	3.428980
4	0.438447	0.084285	1.309998
5	0.620847	0.001259	0.074391
6	0.404566	0.000301	0.204650
7	0.500000	0.001953	0.214405
8	0.500000	0.018519	0.998200
9	0.0311879	1.939473e-12	2.693705
10	0.422650	0.002179	0.721259

Exemplul 2.5 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.4, 0.3, 0.6, 0.7, 0.2, 0.1, 0.1, 0.6, 0.7, 0.9\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.1, 0.1, 0.1, 0.4, 0.1, 0.3, 0.2, 0.1, 0.2, 0.3\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este o variabilă aleatoare cu funcția de repartiție Erlang $C_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x c_k \frac{(c_k x)^{p_k-1}}{(p_k-1)!} e^{-c_k x} dx, & x \geq 0 \end{cases}$, cu parametrii $c_k = \{0.2, 0.3, 0.5, 0.1, 0.4, 0.1, 0.3, 0.5, 0.2, 0.1\}$ și $p_k = \{2, 6, 8, 3, 4, 7, 5, 9, 15, 3\}$.

Tabelul 2.5. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.4$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.117218	0.005161	0.283897
2	0.131483	0.000122	0.167444
3	0.095928	1.999486e-05	0.426192
4	0.312132	0.000330	0.581023
5	0.149219	0.004312	0.118068
6	0.394449	2.2681e-06	0.027201
7	0.298438	0.002676	0.029745
8	0.095928	6.930903e-06	0.428771
9	0.169275	4.544618e-13	0.588000
10	0.213026	0.000121	0.819456

Exemplul 2.6 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de

așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{2, 5, 9, 24, 11, 3, 88, 7, 1, 17\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.3, 0.4, 0.3, 0.2, 0.1, 0.1, 0.6, 0.8, 0.1, 0.3\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este o variabilă aleatoare cu funcția de repartiție Erlang $C_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x c_k \frac{(c_k x)^{p_k-1}}{(p_k-1)!} e^{-c_k x} dx, & x \geq 0 \end{cases}$, cu parametrii $c_k = \{0.1, 0.3, 0.3, 0.3, 0.8, 0.5, 0.2, 0.2, 0.1, 0.1\}$ și $p_k = \{4, 7, 5, 22, 6, 7, 12, 3, 16, 22\}$.

Tabelul 2.6. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.5$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.116904	3.729325e-07	2.31388
2	0.072216	1.121748e-10	6.111828
3	0.031525	9.815786e-10	11.194942
4	0.008162	6.401593e-45	31.738775
5	0.008692	6.895743e-10	8.795697
6	0.028452	1.577843e-08	3.124018
7	0.006779	1.269516e-34	122.503824
8	0.105831	2.512041e-06	8.247283
9	0.065153	6.868969e-21	0.966667
10	0.017134	9.818321e-52	23.945714

Algoritmul 2.3

Date de intrare: $\{\lambda_k\}_{k=1}^r$; $\{b_k\}_{k=1}^r$; $\{\bar{c}_k\}_{k=1}^r$; $\{p_k\}_{k=1}^r$; s ; r ; $\varepsilon > 0$.

Date de ieșire: k ; $\{\pi_k(s)\}_{k=1}^r$; $\{\pi_k^\delta(s)\}_{k=1}^r$; $\{l_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ este o funcție de repartiție Erlang și $C_k(x)$ este o funcție de repartiție Normală.

$$\beta_k(s) = \left(\frac{b_k}{s + b_k} \right)^{p_k}; \quad c_k(s) = e^{\left(\frac{\sigma^2 s}{2} - s \bar{c}_k \right)}.$$

- b) Se calculează repartiția perioadei de ocupare $\pi_k^{\delta(n)}(s)$ conform relațiilor:

Pentru $n=0$, avem $\pi_k^{(0)}(s) = 0$.

$$\pi_k^{\delta(n)}(s) = c_k(s + \lambda_k - \lambda_k \pi_k^{(n)}(s)) \pi_k^{(n)}(s),$$

$$\pi_k^{(n)}(s) = \beta_k(s + \lambda_k - \lambda_k \pi_k^{(n-1)}(s)).$$

c) Se calculează repartiția lungimii șirului de așteptare $l_k(s)$, conform formulei:

$$l_k(s) = \frac{\lambda_k}{s + \lambda_k - \lambda_k \pi_k^{\delta(n)}(s)} \left\{ \frac{c_{k1} \lambda_k s + \lambda_k (1 - c_k(s))}{s^2} + \frac{[1 - \beta_k(s)(s - \lambda_k) + \lambda_k s \beta_{k1}](c_k(s) - \pi_k^{\delta(n)}(s))}{s^2 (1 - \beta_k(s))} - \frac{(1 + \lambda_k \beta_{k1})(c_k(s) - \pi_k^{\delta(n)}(s))}{s(1 - \beta_k(s))^2} \right\}.$$

Condiția de oprire: $|\pi_k^{(n)}(s) - \pi_k^{(n-1)}(s)| < \varepsilon$.

Exemplul 2.7 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.1, 0.8, 0.2, 0.2, 0.3, 0.9, 0.8, 0.4, 0.4, 0.5\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de

repartiție Erlang $B_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x b_k \frac{(b_k x)^{p_k-1}}{(p_k-1)!} e^{-b_k x} dx, & x \geq 0 \end{cases}$, cu parametrii $b_k = \{0.5, 0.2, 0.1, 0.1, 0.4,$

$0.8, 0.5, 0.4, 0.5, 0.8\}$ și $p_k = \{4, 3, 6, 5, 11, 7, 19, 23, 8, 5\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este o variabilă aleatoare cu funcția de repartiție Normală

$C_k(x) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-c_k)^2}{2\sigma^2}} dx$, $-\infty < 0 < \infty$, cu parametrii $c_k = \{0.2, 0.5, 0.7, 0.1, 0.5, 0.6, 0.7,$

$0.7, 0.6, 0.4\}$. Rezultatele numerice ale caracteristicilor probablistice pentru sistemul menționat sunt prezentate în Tabelul 2.7.

Tabelul 2.7. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.3$)

k	$\pi_k(s)$	$\pi_k^{\delta}(s)$	$l_k(s)$
1	0.012449	0.011867	1.874136
2	0.032032	0.019647	3.286044
3	0.041058	0.029387	2.298903
4	0.000129	0.000123	3.046041
5	0.000171	0.000127	3.399852
6	0.000458	0.000245	5.444821
7	1.609478e-08	8.550547e-09	4.908130
8	1.192093e-07	7.723716e-08	3.749759
9	0.002070	0.001380	3.846974
10	0.004151	0.003029	4.554868

Exemplul 2.8 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{0.4, 0.3, 0.6, 0.7, 0.2, 0.1, 0.1, 0.6, 0.7, 0.9\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de

$$\text{repartiție Erlang } B_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x b_k \frac{(b_k x)^{p_k-1}}{(p_k-1)!} e^{-b_k x} dx, & x \geq 0 \end{cases}, \text{ cu parametrii } b_k = \{0.1, 0.1, 0.1, 0.4, 0.1,$$

0.3, 0.2, 0.1, 0.2, 0.3\} și $p_k = \{2, 6, 8, 3, 4, 7, 5, 9, 15, 3\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este o variabilă aleatoare cu funcția de repartiție Normală

$$C_k(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-c_k)^2}{2\sigma^2}} dx, \quad -\infty < 0 < \infty, \text{ cu parametrii } c_k = \{0.2, 0.3, 0.5, 0.1, 0.4, 0.1, 0.3,$$

0.5, 0.2, 0.1\}.

Rezultatele numerice ale caracteristicilor probabile pentru sistemul menționat sunt prezentate în Tabelul 2.8.

Tabelul 2.8. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.4$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.041357	0.035498	1.194672
2	0.000730	0.000626	1.501022
3	0.000152	9.436252e-05	1.934324
4	0.000579	0.000545	2.497509
5	0.026142	0.020669	0.972084
6	3.572245e-06	3.617178e-06	0.764531
7	0.007450	0.007249	0.729838
8	5.081456e-05	3.097554e-05	1.951070
9	6.401176e-13	5.398243e-13	2.412227
10	0.000365	0.000322	2.729439

Exemplul 2.9 Se consideră un sistem de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă format din k șiruri de așteptare, $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare, conform fluxului de tip Poisson, cu parametrii $\lambda_k = \{3, 9, 5, 12, 23, 45, 7, 14, 10, 17\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție Erlang

$$B_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x b_k \frac{(b_k x)^{p_k-1}}{(p_k-1)!} e^{-b_k x} dx, & x \geq 0 \end{cases}, \text{ cu parametrii } b_k = \{0.4, 0.7, 0.2, 0.4, 0.5, 0.5, 0.1, 0.4, 0.2,$$

0.3\} și $p_k = \{3, 6, 8, 12, 22, 32, 2, 8, 5, 3\}$, iar timpul de orientare de la un șir de așteptare la șirul

de așteptare k este o variabilă aleatoare cu funcția de repartiție Normală

$$C_k(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-c_k)^2}{2\sigma^2}} dx, \quad -\infty < 0 < \infty, \text{ cu parametrii } c_k = \{0.3, 0.3, 0.2, 0.2, 0.5, 0.5, 0.1,$$

0.2, 0.6, 0.4\}.

Rezultatele numerice ale caracteristicilor probabile pentru sistemul menționat sunt prezentate în Tabelul 2.9.

Tabelul 2.9. Valorile numerice ale repartiției lungimii medii a șirului de așteptare ($s = 0.1$)

k	$\pi_k(s)$	$\pi_k^\delta(s)$	$l_k(s)$
1	0.000688	0.000289	244.951511
2	1056721e-09	1.427259e-10	257.863911
3	4.11181e-12	3.713086e-12	100.754215
4	3.415849e-22	3.226865e-23	96.023895
5	1.490308e-37	1.611891e-42	120.771071
6	1.906186e-63	7.560634e-73	100.174372
7	0.000193	0.000109	3.364830
8	1.464036e-15	2.761287e-15	135.578308
9	5.544181e-07	2.903514e-09	191.408058
10	1.194211e-05	2.775137e-07	699.389098

2.5. Concluzii la capitolul 2

Scopul principal al studierii modelelor de așteptare îl constituie determinarea caracteristicilor probabile de performanță ale sistemului. Dar nu întotdeauna formulele analitice pot fi utilizate direct pentru a determina aceste caracteristici, astfel o importanță majoră se acordă elaborării unor noi metode numerice cât și algoritmilor realizați în baza acestor metode.

În capitolul 2 s-au obținut următoarele rezultate:

- s-au analizat metodele analitice și cele numerice în cercetarea problemelor teoriei așteptării;
- s-au prezentat rezultate de bază cu privire la unele caracteristici probabilistice pentru modelele de așteptare de tip Polling cu servire exhaustivă;
- s-au elaborat algoritmi numerici pentru determinarea lungimii medii a șirului de cerințe într-un moment arbitrar de timp t pentru utilizatorul k ;
- s-au propus exemple numerice pentru calcularea repartiției lungimii medii a șirului de cerințe pentru modelele de așteptare Polling cu vacanțe semi-Markoviene.

3. CARACTERISTICI DE PERFORMANȚĂ ALE MODELELOR GENERALIZATE DE AȘTEPTARE CU PRIORITĂȚI

În acest capitol se prezintă unele noțiuni generale, clasificări, rezultate analitice cunoscute bazate pe aparatul funcțiilor generatoare și transformatelor Laplace și Laplace-Stieltjes pentru sistemele de așteptare Polling cu priorități. De asemenea, sunt prezentate rezultate cu privire la unele caracteristici probabilistice pentru modelele generalizate de așteptare pentru disciplina de prioritate DD (Discretionary Discipline). Sunt propuși unii algoritmi numerici și exemple pentru determinarea perioadei de ocupare și a perioadelor auxiliare pentru diferite funcții de repartiție și scheme de servire pentru prioritatea DD.

3.1. Concepte generale ale modelelor generalizate cu priorități

Sistemele de așteptare cu priorități constituie o clasă mare ale sistemelor de așteptare unde cerințele ce intră în sistem sunt distinse după importanța lor. Astfel de sisteme reprezintă modele adecvate ale multor aspecte ale vieții de zi cu zi, atunci când o servire preferențială se acordă pentru anumite tipuri de cerințe. Sistemele de așteptare cu priorități își găsesc, de asemenea, aplicații importante în modelarea și analiza sistemelor informatice și de comunicare.

Regula generală de servire în sistemele de așteptare cu priorități este după cum urmează: cerințele care sunt în sistem și au o prioritate mai mare trebuie să fie servite înaintea acelor care au o prioritate mai mică. Cu toate acestea, în astfel de sisteme modul de comportare al serverului, în esență, le poate diversifica. În plus, există un număr considerabil de sisteme în care serverul are nevoie de ceva timp pentru a trece de la servirea unui tip de cerințe la alt tip de cerințe. Toate acestea oferă o mare varietate de sisteme de așteptare cu priorități. Conform acestor fenomene, descrierea și clasificarea sistemelor de așteptare cu priorități și timp de schimb al stărilor, numite modele generalizate, este prezentată în continuare în linii generale.

Vom considera un sistem de așteptare cu un singur server și clase r clase de cerințe, acestea fiind notate cu *class 1*, *class 2*, ..., *class r*, fiecare având propriul flux de intrare și șir de așteptare [100]. Cerințele de o anumită clasă sunt servite conform uneia din următoarele două reguli, proprie fiecărui șir de așteptare:

- primul sosit – primul servit (FIFO);
- ultimul sosit – primul servit (LIFO).

Vom presupune că perioadele de timp dintre două sosiri consecutive ale cerințelor de clasa i sunt identic și independent repartizate, cu funcția de repartiție $A_i(t)$, $i=1, \dots, r$. Similar, vom presupune că timpul de servire a unei cerințe de clasa i reprezintă o variabilă aleatoare B_i cu funcția de repartiție $B_i(t)$, adică

$$B_i(t) = P(B_i < t), \quad i = 1, \dots, r.$$

Vom spune că cerințele de clasa i au o prioritate mai mare decât cerințele de clasa j , dacă $1 \leq i < j \leq r$. Astfel, cerințele cu cea mai mare prioritate sunt cerințele de clasa 1, în timp ce cerințele de clasa r au cea mai mică prioritate. Serverul oferă o preferință în servirea cerințelor cu cea mai înaltă prioritate în rândul celor prezentate în sistem.

Cu toate acestea, serverul are nevoie de ceva timp pentru a trece de la un șir de așteptare la altul. Acest timp este considerat o variabilă aleatoare și vom spune că C_{ij} este timpul de trecere de la servirea cerinței de clasa i la servirea cerinței de clasa j , $1 \leq i \leq r$, $1 \leq j \leq r$, $i \neq j$. În continuare, ne vom referi la C_{ij} ca ij - timp de trecere cu funcția de repartiție $C_{ij}(t)$.

Vom considera două discipline de servire – ambele sunt tradiționale în teoria sistemelor de așteptare cu priorități: disciplina absolută de servire și disciplina relativă de servire. Se presupune, în prima disciplină, că orice cerere de prioritate mai mare decât cea care este servită întrerupe procesul de servire și necesită trecerea imediată a serverului la clasa sa. În cea de a doua disciplină, cerința de un nivel de prioritate mai mică va primi o servire completă, după care serverul va continua trecerea, dacă este necesar. În ambele cazuri, după finalizarea servirii cerințelor dintr-o anumită clasă, serverul va fi pregătit pentru a trece la un șir de așteptare nenul corespunzător clasei de cel mai înalt nivel de prioritate prezent la acel moment în sistem.

▪ **Disciplina absolută de servire**

Vom considera diferite situații pentru cerințele a căror servire a fost întreruptă:

- 1) *politica preemptive resume* – cerința care a fost întreruptă este servită perioada de timp reziduu după revenirea serverului, adică timpul în care această cerință ar fi fost servită, dacă servirea sa nu a fost întreruptă, din momentul întreruperii.
- 2) *politicele repeat again (servire din nou)*:
 - *politica preemptive identical repeat (servirea identică din nou)* – cerința întreruptă va fi servită din nou după revenirea serverului. Timpul de servire va coincide cu timpul final de servire al acestei cerințe dacă servirea sa nu ar fi fost întreruptă.
 - *politica preemptive non-identical repeat (servirea neidentică din nou)* – exact ca și în cazul precedent, doar că timpul de servire este realizat din nou, deși repartizate în conformitate cu legea servirii respective, adică având funcția de repartiție $B_i(t)$ dacă cerința servită din nou este de clasa i .
 - *politica preemptive loss (cu pierderi)* – cerințele servirea cărora este întreruptă sunt "pierdute" și sunt eliminate din sistem.

▪ **Disciplina relativă de servire**

În cadrul acestei discipline se presupune că cerințele nu pot fi întrerupte imediat în procesul servirii. Însă, după finalizarea servirii fiecărei cerințe (mulțime de cerințe dintr-un șir de așteptare), serverul este pregătit pentru a trece la un șir nenul cu cele mai mari nivele de prioritate de cerințe, dacă sunt prezente în sistem, și care așteaptă să fie servite. Conform lucrării lui Gaver (1961), *disciplină relativă de servire* se mai numește și *disciplina postponable priority service*.

Disciplina *postponable priority service* poate fi de diverse tipuri, în dependență de regula după care se efectuează trecerea către cerințele cu prioritate mai mare, distingem:

- 1) *disciplina request postponable priority service* – după finalizarea servirii oricărei cerințe, serverul este pregătit să treacă la șirul de așteptare nenul cu cerințe de prioritate mai mare.
- 2) ▪ *disciplina exhaustive postponable priority service* – serverul va fi pregătit să treacă la șirul de așteptare nenul cu cerințe de prioritate mai mare, numai după finalizarea servirii tuturor cerințelor din șirul de așteptare pe care le deservea la momentul respectiv.
 - *disciplina gated postponable priority service* – exact ca și *disciplina exhaustive postponable*, diferența constă în faptul că serverul va servi numai acele cerințe care au sosit în sistem înaintea celei întrerupte.

▪ **Discipline de orientare (switching)**

Ar trebui să luăm în considerare faptul că unele cerințe ce intră în sistem pot găsi serverul trecând la cerințe de prioritate mai mică. Prin urmare, în mod analogic cu disciplinele procesului de servire, distingem următoarele discipline în procesul de orientare (comutare, trecere): *disciplina orientarea absolută (preemptive switching)*, *disciplina orientarea absolută cu stare neutră (preemptive neutral state switching)*, *disciplina orientarea relativă (non-preemptive switching)*, *disciplina orientarea relativă cu stare neutră (non-preemptive neutral state switching)*.

Orientare absolută (preemptive switching). În cadrul disciplinelor de *orientare absolută* și *absolută cu starea neutră* se presupune că orice ij - orientare poate fi întreruptă imediat de k -cerințe sosite, numai și numai dacă $k < j$, adică dacă în sistem a intrat o cerință cu o prioritate mai mare. După întreruperea se inițiază o nouă comutare la aceste cerințe.

Orientare relativă (non-preemptive switching). În cadrul disciplinelor *orientare relativă* și *orientare relativă cu stare neutră* se presupune că nici o trecere nu poate fi întreruptă de cerințe cu prioritate mai mare. Aceste discipline diferă numai prin existența unei stări intermediare de trecere – stare neutră, așa cum s-a prezentat mai sus.

Să considerăm *disciplina de orientare relativă cu stare neutră* și să presupunem că cerința de clasa k a sosit în sistem în timpul procesului de trecere către cerințele de clasa j , unde $k < j$, adică realizând ij - orientarea cu durata C_{ij} . Acest eveniment poate avea loc în unul din următoarele momente de timp: când are loc orientarea către starea neutră (de lungimea T_i) sau când are loc trecerea de la starea neutră (de lungimea S_j). Prin urmare vom considera următoarele două subdiscipline:

- *orientare normală* – trecerea la cerințele de clasa k se va face sau după finalizarea orientării la starea neutră de la cerințele de clasa i (și în acest caz durata orientării va fi S_k) sau după trecerea de la starea neutră la cerințele de clasa j (și atunci durata va fi C_{jk}).
- *orientare postponable* – trecerea la cerințele de clasa k întrerupte este posibilă numai după completarea ij - orientării (și durata va fi timpul C_{jk}).

▪ *Strategia serverului în stare liberă*

Vom trece la specificațiile regimurilor de comportare ale serverului în stare de vacanță [100]. Inițial, independent de regim, vom presupune că serverul are nevoie de ceva *timp de pregătire (încălzire)* pentru a efectua orientarea sau servirea primei cerințe sosite în sistemul liber, adică după *perioada vacantă*. Timpul de încălzire este o variabilă aleatoare W cu funcția de repartiție $W(t)$.

Deosebim următoarele moduri de comportare a serverului atunci când sistemul devine liber:

- *inițializarea la zero* (reset to zero) - după finalizarea servirii ultimei cerințe din sistem, serverul imediat se orientează la starea neutră. Dacă prima cerință care intră în sistemul liber este o cerință de prioritatea i , atunci serverul realizează orientarea de durata S_i . Evident că acest regim este bine definit pentru sistemele cu disciplina de orientare cu *stare neutră*;
- *se uită înainte* (look ahead) – serverul trece către șirul de așteptare cu cerințe de clasa 1 în momentul când sistemul devine liber;
- *așteaptă și privește* (wait and look) - serverul rămâne orientat către șirul de cerințe din care face parte ultima cerință servită;
- *așteaptă cea mai probabilă intrare* (wait for the most probable) - serverul se orientează către fluxul de cerințe unde intensitatea de apariție a cerinței este cea mai mare.

Cele expuse permit descrierea unei clase largi de sisteme de așteptare cu priorități. În

continuare vom prezenta unii identificatori folosiți pentru aceasta descriere.

- fluxul de intrare – repartiția perioadelor de timp dintre două sosiri consecutive (pentru fiecare flux);
- timpii de servire – repartiția timpului de servire (pentru fiecare flux);
- timpii de orientare – precizarea tipului orientării (stare neutră sau nu) și repartiția timpilor de orientare;
- ordinea de servire în cadrul unui șir (FIFO, LIFO);
- disciplina de orientare;
- disciplina de servire;
- comportamentul serverului în starea inactivă.

3.2. Concepte de strategii în stare liberă

În sistemele de așteptare clasice, sistemul se consideră în stare liberă, după ce acesta este liber de cerințe (mesaje, cereri). Modul de a fi într-o stare liberă este produs după realizarea perioadei de ocupare. De regulă, timpul ca sistemul să se afle în stare liberă este o variabilă aleatoare. În cazul în care o cerere vine în sistemul de așteptare, care este în stare liberă, această cerință va fi servită imediat. Cu începerea servirii, perioada de inactivitate a sistemului se încheie. Astfel, perioadele de ocupare (în cazul în care sunt servite cerințele din șirurile de așteptare) pentru sistemele clasice vor începe neapărat cu servirea cerinței care "a deschis" perioada de ocupare. În general, situația este diferită în cazul modelelor generalizate.

Ne vom opri și vom discuta mai multe detalii cu privire la descrierea sistemului generalizat. Sistemul de așteptare generalizate este un sistem de așteptare cu cerințe neomogene, care sunt reorganizate în funcție de categoriile de omogenitate. Un anumit grad de prioritate este atribuit acestor cerințe neomogene. De asemenea, comutarea procesului de servire de la o clasă de prioritate la alta nu este momentană. Cu alte cuvinte, comutarea între procesele de servire va necesita pierderi de timp. Acest timp, care este consumat la comutarea între clasele de priorități, este considerat o variabilă. Această variabilă este fixă și depinde de unii factori imprevizibili. Astfel, obiectiv vorbind, aceasta este o variabilă aleatoare [100].

Astfel, dacă, de exemplu, considerăm că pierderile de timp de la trecerea procesului de la clasa de prioritate i la clasa j sunt C_{ij} variabile aleatoare cu funcția de repartiție $C_{ij}(x)=P\{C_{ij}<x\}$, ij și ultima cerință servită în perioada de ocupare a fost o cerință de clasa i , atunci va apărea următoarea întrebare. Cum va începe următoarea perioadă de ocupare?

Într-adevăr, în cazul în care o cerință nouă sosită este de clasă i , este evident că va fi servită imediat, deoarece sistemul este deja pregătit pentru cerința de clasă i . Dacă o cerință nouă sosită este de o altă clasă de prioritate, să presupunem că este de clasă k , ($k < i$), atunci, este evident că, înainte ca sistemul să fie implicat în servirea acesteia, va fi necesar pentru a efectua comutarea $i \rightarrow k$ cu o variabilă aleatoare C_{ik} cu funcția de repartiție $C_{ik}(x) = P\{C_{ik} < x\}$. Astfel, în acest caz, perioada de ocupare va începe nu cu servirea cerinței sosite, dar cu orientarea (comutarea) sistemului de servire către cerința dată.

Strategii ale sistemului generalizat în stare liberă

În acest compartiment vom formula 3 strategii concrete. Formulările vor fi elaborate pentru 2 clase de priorități. Ulterior acestea vor fi extinse pentru cazul n -dimensional. Vom nota cu L_1 și L_2 fluxul de cerințe de clasă 1 și 2, respectiv.

Strategia 1. Când sistemul de așteptare este în stare liberă, după ce s-a eliberat de cerințe, sistemul face o resetare a reorientării. Resetarea este realizată momentan. Astfel, în momentul în care o cerință de o prioritate oarecare vine în sistemul de așteptare, care se află în stare liberă, pentru a începe procesul de servire, orientarea sistemului este necesară. Vom considera că timpul de orientare a sistemului din starea liberă în starea L_1 ($0 \rightarrow 1$) și din stare liberă în L_2 ($0 \rightarrow 2$) este, respectiv, egal cu timpul de orientare ($2 \rightarrow 1$) și ($1 \rightarrow 2$).

Strategia 2. În cazul în care sistemul de așteptare este în stare liberă și ultima cerință servită este de prioritatea 2, atunci orientarea ($2 \rightarrow 1$) va începe momentan. Astfel, în situația în care o cerință de prioritatea 2 ajunsă în sistemul aflat în stare liberă, atunci în primul rând orientarea ($1 \rightarrow 2$) va avea loc și apoi va începe deservirea acesteia. Dar, dacă cerința sosită este de prioritatea 1, atunci aceasta este servită imediat. S-ar putea întâmpla cazul când unele cerințe de prioritatea 2 vor veni în sistemul aflat în stare liberă în procesul de orientarea ($2 \rightarrow 1$) (care începe după eliberarea sistemului de cerințe). În acest caz, orientarea ($2 \rightarrow 1$) nu este întreruptă, dar când se termină, orientarea ($1 \rightarrow 2$) începe, și după aceasta procesul de servire al cerinței de prioritatea 2 va avea loc.

Strategia 3. După ce sistemul de așteptare devine în stare liberă, prepararea (orientarea) acestuia pentru servirea cerinței cu cea mai mare probabilitate de sosire în sistem va avea loc. Astfel, dacă p_i - probabilitatea de sosire a cerinței din fluxul L_i , $i=1,2$ și $p_2 > p_1$ a sistemului în stare liberă, atunci înainte de a începe următoarea perioadă de ocupare, sistemul va avea orientarea ($1 \rightarrow 2$).

Urmând lucrarea lui Gaver [37] vom numi strategia 1 - strategia "set to zero", strategia 2 - strategia "look ahead" și strategia 3 - strategia "wait for the most probable".

Extinderea legilor de prioritate clasice la sistemele generalizate

Ca și mai sus, vom considera 2 clase de priorități. Numim cerințele fluxului de intrare L_i – cerințe de prioritate i , $i=1,2$, și vom spune că cerințele fluxului de intrare L_1 au o prioritate mai mare decât cerințele fluxului de intrare L_2 . Vom considera că prioritatea cerințelor fluxului de intrare L_1 în ceea ce privește cerințele fluxului L_2 constă în următoarele: în primul rând, printre cerințele care așteaptă să fie servite, cerințele din L_1 vor fi servite înaintea cerințelor din L_2 ; în al doilea rând, în cazul în care o cerință din L_1 vine în momentul orientării sistemului pentru servire sau servirii cerinței fluxului L_2 , atunci următoarele situații pot avea loc:

- 1) Orientarea și servirea sunt întrerupte de cerința fluxului L_1 ;
- 2) Orientarea nu este întreruptă, dar servirea este întreruptă de cerința fluxului L_1 ;
- 3) Orientarea este întreruptă, iar servirea nu este întreruptă;
- 4) Nici orientarea, nici servirea nu sunt întrerupte de cerința fluxului L_1 .

În cazul 1) vom spune că cerințele fluxului L_1 au prioritate absolută, în cazul 2) – prioritate semi-absolută, în cazul 3) – prioritate relativă, în cazul 4) – prioritate semi-relativă.

După întrerupere, orientarea și servirea a cerinței fluxului L_1 începe imediat. În funcție de tipul de întrerupere (orientarea întreruptă) a cerințelor vom examina diferite orientări și discipline de servire. Vom descrie mai în detaliu aceste discipline de orientare și de servire în punctul următor.

Discipline de servire și orientare în modelele generalizate

Vom examina următoarele scheme de servire [63], [100]:

▪ Prioritatea absolută

Schema 1.1. - orientarea-din nou, cu timp nou de realizare; servirea-din nou, cu timp nou de realizare:

a) dacă cerința fluxului L_1 sosește în timpul orientării sistemului de la L_1 către L_2 , atunci orientarea ($1 \rightarrow 2$) este întreruptă și orientarea sistemului de la L_2 la L_1 ($2 \rightarrow 1$) imediat începe. Când sistemul este liber de cerințele fluxului L_1 , atunci orientarea întreruptă începe din nou (cu noi realizări de orientare a timpului).

b) dacă cerința fluxului L_1 vine în timpul de servire al cerințelor fluxului L_2 , atunci servirea se întrerupe, orientarea ($2 \rightarrow 1$) va începe imediat, și cerința va fi servită. Când sistemul este liber de cerințe ale fluxului L_1 , atunci orientarea ($1 \rightarrow 2$) începe. Când orientarea ($1 \rightarrow 2$) se încheie, atunci servirea cerinței întrerupte va începe din nou (cu noi realizări de servire a timpului).

Schema 1.2. - orientarea-din nou, cu timp nou de realizare; servirea-cerința se pierde:

a) aceeași situație ca și în a) schema 1.1.;

b) aceeași situație ca și în b) schema 1.1., dar cerințele întrerupte vor fi pierdute și eliminate din sistem.

Schema 1.3. - orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire:

a) aceeași situație ca și în a) schema 1.1.;

b) aceeași situație ca și în b) schema 1.1., însă servirea cerințelor întrerupte va fi reluată și completată după ce sistemul este din nou la șirul de așteptare corespunzător.

Schema 1.4. - orientarea-din nou, cu timp nou de realizare; servirea-din nou, dar cu aceeași durată de realizare:

a) aceeași situație ca și în a) schema 1.1.;

b) aceeași situație ca și în b) schema 1.1., deși cerința întreruptă va fi servită din nou și timpul de servire va fi o nouă realizare a unei variabile aleatoare corespunzător (servire identică din nou).

Schema 2.1. - orientarea-timpul rămas; servirea-din nou, cu timp nou de realizare:

a) aceeași situație ca și în a) schema 1.1., cu diferența că orientarea întreruptă ($1 \rightarrow 2$) va fi reluată;

b) aceeași situație ca și în b) schema 1.1.

Schema 2.2. - orientarea-timpul rămas; servirea-cerința se pierde:

a) aceeași situație ca și în a) schema 2.1.;

b) aceeași situație ca și în b) schema 1.2.

Schema 2.3. - orientarea-timpul rămas; servirea-continuă timpul rămas de servire:

a) aceeași situație ca și în a) schema 2.1.;

b) aceeași situație ca și în b) schema 1.3.

Schema 2.4. - orientarea-timpul rămas; servirea-din nou, însă cu aceeași durată de realizare:

a) aceeași situație ca și în a) schema 2.1.;

b) aceeași situație ca și în b) schema 1.4.

Schema 3.1. - orientarea-din nou, dar cu aceeași durată de realizare; servirea-din nou, cu timp nou de realizare:

a) aceeași situație ca și în a) schema 1.1., deși comutarea întreruptă ($1 \rightarrow 2$) va fi realizată din nou (comutare identică din nou);

b) aceeași situație ca și în b) schema 1.1.

Schema 3.2. - orientarea-timpul rămas; servirea-cerința se pierde:

a) aceeași situație ca și în a) schema 3.1.;

b) aceeași situație ca și în b) schema 1.2.

Schema 3.3. - orientarea-timpul rămas; servirea-continuă timpul rămas de servire:

a) aceeași situație ca și în a) schema 3.1.;

b) aceeași situație ca și în b) schema 1.3.

Schema 3.4. - orientarea-timpul rămas; servirea-din nou, dar cu aceeași durată de realizare:

a) aceeași situație ca și în a) schema 3.1.;

b) aceeași situație ca și în b) schema 1.4.

▪ **Prioritatea semi-absolută**

Schema 4.1. - orientarea-nu este întreruptă; servirea-din nou, cu timp nou de realizare:

a) Dacă în timpul comutării ($1 \rightarrow 2$) o cerință a fluxului L_1 sosește în sistem, atunci această comutare nu este întreruptă. După ce este finalizată o nouă comutare ($2 \rightarrow 1$) este începută și cerințele din L_1 vor fi servite. Când serverul este liber de cerințe de clasa 1 acesta se poate întoarce înapoi la cerințele de clasa 2, dacă altele nu sunt prezente în sistem.

b) aceeași situație ca și în b) schema 1.1.

Schema 4.2. - orientarea-nu este întreruptă; servirea-cerința se pierde:

a) aceeași situație ca și în a) schema 4.1.;

b) aceeași situație ca și în b) schema 1.2.

Schema 4.3. - orientarea-nu este întreruptă; servirea-continuă timpul rămas de servire:

a) aceeași situație ca și în a) schema 4.1.;

b) aceeași situație ca și în b) schema 1.3.

Schema 4.4. - orientarea-nu este întreruptă; servirea-din nou, dar cu aceeași durată de realizare:

a) aceeași situație ca și în a) schema 4.1.;

b) aceeași situație ca și în b) schema 1.4.

▪ **Prioritatea semi-relativă**

Schema 5.1. - orientarea-din nou, cu timp nou de realizare; servirea-nu este întreruptă:

a) aceeași situație ca și în a) schema 1.1.;

b) servirea cerinței fluxului L_2 nu este întreruptă dacă o cerință a fluxului L_1 sosește în sistem. Sistemul va continua cu comutarea ($2 \rightarrow 1$) la finalizarea de servire a unei cerințe de clasa 2.

Schema 5.2. - orientarea-timpul rămas; servirea-nu este întreruptă:

a) aceeași situație ca și în a) schema 2.1.;

b) aceeași situație ca și în b) schema 5.1.

Schema 5.3. - orientarea-din nou, dar cu aceeași durată de realizare; servirea-nu este întreruptă:

a) aceeași situație ca și în a) schema 3.1.;

b) aceeași situație ca și în b) schema 5.1.

▪ **Prioritatea relativă**

Schema 5.4. - orientarea-nu este întreruptă; servirea-nu este întreruptă:

a) aceeași situație ca și în a) schema 4.1.;

b) aceeași situație ca și în b) schema 5.1.

3.3. Repartiția perioadelor de ocupare și caracteristici auxiliare ale modelelor Polling

În această secțiune vom prezenta repartițiile perioadei de ocupare și a perioadelor auxiliare, pentru fiecare strategie în stare liberă și schemele de prioritate absolută menționate mai sus, pentru sistemul de așteptare Polling cu priorități. Toate repartițiile sunt prezentate în termeni de transformată Laplace-Stieltjes și sunt obținute prin metoda introducerii unui eveniment aleatoriu suplimentar (metoda catastrofelor).

În continuare vom defini perioada de ocupare și perioadele auxiliare pentru sistemul generalizat de așteptare [100], care sunt valabile pentru toate schemele:

Definiția 3.1 *k-ciclu de orientare* - această perioadă de timp începe cu orientarea serverului către servirea cerinței a_k și se finalizează imediat ce serverul este pregătit pentru servirea acestor cerințe.

Vom nota prin variabila aleatoare N_k durata k -ciclului de orientare și prin $N_k(x)$ - funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $v_k(s)$.

Definiția 3.2 *k-ciclu de servire* - această perioadă de timp începe cu momentul servirii cerinței a_k și se finalizează imediat ce sistemul este liber de aceste cerințe.

Vom nota prin variabila aleatoare H_k durata k -ciclului de servire și prin $H_k(x)$ - funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $h_k(s)$.

Definiția 3.3 *k-perioadă de ocupare* - această perioadă de timp începe cu momentul sosirii cerinței a_i , $i < k$, în sistemul de așteptare liber și se încheie imediat ce sistemul se eliberează de cerințe a_k .

Vom nota prin variabila aleatoare Π_k durata k -perioadei de ocupare și prin $\Pi_k(x)$ -

funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $\pi_k(s)$.

Definiția 3.4 *kk-perioadă de ocupare* - această perioadă de timp începe cu momentul sosirii cerinței a_k în sistemul liber și se finalizează de îndată ce sistemul devine liber de aceste cerințe.

Vom nota prin variabila aleatoare Π_{kk} - durata kk -perioadei de ocupare și prin $\Pi_{kk}(x)$ - funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $\pi_{kk}(s)$.

Definiția 3.5 $\bar{\Pi}_k$ -*perioadă de ocupare*- această perioadă de timp începe cu sosirea cerinței a_k în sistemul liber de cerinței a_i , $i \leq k$ și orientat către șirul de așteptare i , și se finalizează de îndată ce sistemul devine liber de cerințe a_k .

Vom nota prin variabila aleatoare $\bar{\Pi}_k$ durata $\bar{\Pi}_k$ - perioadei de ocupare și prin $\bar{\Pi}_k(x)$ funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $\bar{\pi}_k(s)$.

Definiția 3.6 *Perioadă de ocupare a sistemului* - această perioadă de timp începe cu momentul sosirii cerințelor în sistemul liber de așteptare și se încheie imediat ce sistemul devine liber de cerințe.

Vom nota prin variabila aleatoare Π - durata perioadei de ocupare a sistemului și prin $\Pi(x)$ - funcția ei de repartiție, iar transformata Laplace-Stieltjes o vom nota cu $\pi(s)$. Evident este că $\Pi(x) = \Pi_r(x)$.

Pentru sistemele de așteptare menționate mai sus au loc următoarele relații care ne exprimă transformatele Laplace-Stieltjes ale funcțiilor de repartiție a duratei ciclului de servire și a ciclului de orientare. Pentru fiecare dintre scheme, valorile funcțiilor $v_k(s)$ și $h_k(s)$ se determină din următoarele expresii [100]:

1. Valoarea k -ciclului de orientare $v_k(s)$

Pentru schemele 1.1-1.4

$$v_k(s) = c_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} \cdot [1 - c_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1}, \quad (3.1)$$

Pentru schemele 2.1-2.4

$$v_k(s) = c_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)]), \quad (3.2)$$

Pentru schemele 3.1-3.4

$$v_k(s) = c_k(s + \sigma_{k-1}) \int_0^{\infty} \frac{e^{-(s+\sigma_{k-1})\tau} dC_k(\tau)}{s + \sigma_{k-1} \{1 - \pi_{k-1}(s)[1 - e^{-(s+\sigma_{k-1})\tau}]\}}. \quad (3.3)$$

2. Valoarea k -ciclului de servire $h_k(s)$

Pentru schemele 1.1, 2.1 și 3.1

$$h_k(s) = \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \nu_k(s) \right\}^{-1}, \quad (3.4)$$

Pentru schemele 1.2, 2.2 și 3.2

$$h_k(s) = \beta_k(s + \sigma_{k-1}) + \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \nu_k(s), \quad (3.5)$$

Pentru schemele 1.3, 2.3 și 3.3

$$h_k(s) = \beta_k(s + \sigma_{k-1}) [1 - \pi_{k-1}(s) \nu_k(s)]; \quad (3.6)$$

Pentru schemele 1.4, 2.4 și 3.4

$$h_k(s) = (s + \sigma_{k-1}) \int_0^\infty \frac{e^{-(s + \sigma_{k-1})u} dB_k(u)}{s + \sigma_{k-1} \{1 - \pi_{k-1}(s) \nu_k(s) [1 - e^{-(s + \sigma_{k-1})u}]\}}. \quad (3.7)$$

Valorile $\nu_k(s)$, menționate în relațiile (3.4)-(3.7) pentru fiecare dintre scheme, se determină din expresiile (3.1)-(3.3), iar valoarea funcției $\pi_{k-1}(s)$, care apare în formulele de mai sus se determină recurent din următorul sistem descris de teorema ce urmează [100]:

Teorema 3.1 Pentru sistemul de așteptare menționat mai sus, transformata Laplace-Stieltjes a funcției de repartiție pentru perioada de ocupare $\pi(s) = \pi_r(s)$ se determină (pentru $k=r$) din următoarele relațiile recurente:

$$\begin{aligned} \pi_k(s) = & \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k) + \frac{\sigma_{k-1}}{\sigma_k} \{ \pi_{k-1}(s + a_k - a_k \bar{\pi}_k(s)) - \\ & - \pi_{k-1}(s + a_k) \} \nu_k(s + [a_k - a_k \bar{\pi}_k(s)]) + \frac{a_k}{\sigma_k} \pi_{kk}(s); \end{aligned} \quad (3.8)$$

$$\pi_{kk}(s) = \nu_k(s + a_k - a_k \bar{\pi}_k(s)) \bar{\pi}_k(s); \quad (3.9)$$

$$\bar{\pi}_k(s) = h_k(s + a_k - a_k \bar{\pi}_k(s)); \quad (3.10)$$

unde valorile funcțiilor $\nu_k(s)$, $h_k(s)$ se determină pentru fiecare dintre schemele menționate mai sus din relațiile (3.1)-(3.7) respectiv, pentru $s = s + a_k - a_k \bar{\pi}_k(s)$.

3.4. Disciplina DD. Repartiția perioadei de ocupare și a caracteristicilor auxiliare

Disciplina de prioritate DD (Discretionary Discipline) își are originea în monografia lui Jaiswal [42], unde este studiată problema de servire pentru un server cu două fluxuri de cerințe (cereri). Această disciplină este mai flexibilă decât disciplinele clasice de prioritate absolută și

relativă, care sunt caracterizate printr-un nivel ridicat de conservare. De exemplu, în cazul disciplinei de prioritate absolută, sosirea unei cerințe cu o prioritate mai mare întrerupe, în mod inevitabil servirea cerinței cu o prioritate mai mică, cu toate că servirea acestei cerințe este aproape finalizată. Și invers, pentru disciplina de prioritate relativă, o cerință cu o prioritate mai mare trebuie să aștepte încheierea servirii cerinței cu o prioritate mai mică, deși servirea acestei cerințe abia a început. Pentru două fluxuri de cerințe, disciplina DD, conform [42], este descrisă după cum urmează: dacă timpul de servire a unei cerințe este mai mică decât valoarea stabilită θ , atunci se va realiza prioritatea absolută, în caz contrar – relativă.

Printre lucrările dedicate acestui subiect, menționăm următoarele lucrări [19, 36, 57, 86, 100], în care disciplina DD este analizată în caz mai general. Și anume, în primul rând, se presupune că numărul de clase de prioritate este arbitrar, și în al doilea rând, se presupune că procesul de servire al trecerii de la o clasă de cerințe la alta necesită ceva timp pentru această trecere. Durata de trecere este o variabilă aleatoare cu o funcție de repartiție arbitrară. Această condiție este importantă pentru aplicații, deoarece ea permite de a modela și analiza obiectiv diferite perioade de așteptare, care au loc în sistemele reale. Folosind rezultatele modelelor analizate, de exemplu, în practica de proiectare tehnică, aceasta are o importantă semnificativă, deoarece ea ne oferă posibilitatea de a obține o evaluare mai exactă despre funcționarea sistemelor reale.

Cu toate acestea, aplicarea rezultatelor analizate implică dificultăți considerabile, principala dintre ele - complexitatea rezultatelor, "incompetența" lor de a le utiliza în aplicații. Într-adevăr, după cum poate fi observat din relațiile de mai jos, rezultatele obținute sunt exprimate, de regulă, în termenii de transformate Laplace sau Laplace-Stieltjes, deseori în termeni de ecuații funcționale recurente. Pentru a determina, de exemplu, $\Pi_k(t)$, trebuie să rezolvăm ecuațiile (3.16)-(3.19), apoi să luăm inversa transformatei Laplace-Stieltjes. Chiar și pentru determinarea unei caracteristici simple ca valoarea medie a perioadei de ocupare - nu este o sarcină atât de simplă, astfel că pentru determinarea ei este necesar de a calcula valoarea transformatei Laplace-Stieltjes a funcției $\pi_k(s)$ în unele puncte. Depășirea acestor dificultăți constă în elaborarea algoritmilor numerici și elaborarea soft-ului specializat.

Să considerăm un sistem de așteptare $M_r | G_r | 1 | \infty$ cu prioritatea DD: dacă timpul de servire al cerinței a_k este mai mică decât valoarea stabilită θ_k ($k = \overline{2, r}$), atunci cerința sosită cu prioritatea mai mare decât k (σ_{k-1} - cerință) va realiza prioritatea relativă, în caz contrar – absolută [64].

Duratele serviri cerințelor a_k sunt variabile aleatoare independente B_k cu funcțiile de repartiții $B_k(x)$, $(k = \overline{1, r})$, respectiv.

Trecerea are loc numai la întreruperea servirii și la revenirea de la servirea întreruptă. Dacă servirea cerinței a_j este întreruptă de sosirea cerinței a_i , $i < j$, atunci dintr-o dată începe trecerea către fluxul L_i ($\rightarrow i$). În cazul în care sistemul va fi liber de cerințe de prioritate mai mare ca j , trecerea către ($\rightarrow j$) va începe, și numai atunci serverul este pregătit să servească cerința întreruptă. Duratele trecerii ($\rightarrow i$) sunt variabile aleatoare C_i cu funcțiile de repartiții $C_i(x)$ ($i = \overline{1, r}$), respective. Variabilele aleatoare B_k și C_i sunt independente.

O trecere arbitrară ($\rightarrow k$) de asemenea, poate fi întreruptă de sosirea cerinței σ_{k-1} , dacă a sosit o cerință cu prioritate mai mare.

Să introducem următoarea clasificare a schemelor. Vom nota fiecare schemă prin doi indici IJ . Primul indice va arăta starea viitoare a trecerii întrerupte, al doilea - starea viitoare a servirii întrerupte.

Vom considera că pentru $I = 1$ trecerea întreruptă începe din nou, pentru $I = 2$ trecerea întreruptă continuă din momentul întreruperii și pentru $I = 3$ identic începe din nou. Pentru $J = 1$ cererea întreruptă este servită din nou și pentru $J = 2$ cererea întreruptă este servită din momentul întreruperii.

În unele situații practice, întreruperea acceptată la servirea cerinței a_i se permite numai prin "cuantumul" timpului de servire. De aceea, vom analiza și disciplina "simetrică" pentru disciplina menționată mai sus. Și anume: să se realizeze prioritatea relativă dacă timpul de servire este mai mic decât θ_k și prioritatea absolută – în caz contrar.

Vom nota prin $\Pi(x)$, $\bar{\Pi}_k(x)$, $\bar{\Pi}_{kk}(x)$, $H_k(x)$, $\bar{\Pi}_{kk}^{(n)}(x)$, $N_k(x)$, $\Pi_k(x)$, $\Pi_{kk}(x)$ funcția de repartiție a perioadei de ocupare, k - perioadă, kk - perioadă, k - ciclu de servire, kkn - perioadă, k - ciclu de trecere, Π_k - perioadă și Π_{kk} - perioadă (definiția acestora, vezi [100]). Să considerăm, de asemenea, $\sigma_k = a_1 + \dots + a_k$, unde a_k - parametrul fluxului Poisson de prioritatea k .

▪ Disciplina Absolută DD

Transformata Laplace-Stieltjes a funcției de repartiție a k - ciclului de servire este determinată din următoarele relații [100].

Lema 3.1 Pentru schemele $I.1$, $I = \overline{1, 3}$

$$\begin{aligned}
h_k(s) &= \int_0^{\theta_k} e^{-(s+\sigma_{k-1}[1-\pi_{k-1}(s)]\nu_k(s))x} dB_k(x) + \\
&+ e^{-(\sigma_{k-1}[\bar{\pi}_{k-1}(s)-\pi_{k-1}(s)]\nu_k(s))\theta_k} \int_{\theta_k}^{\infty} e^{-(s+\sigma_{k-1}[1-\bar{\pi}_{k-1}(s)]x)} dB_k(x),
\end{aligned} \tag{3.11}$$

Lema 3.2 Pentru schemele $I.2$, $I = \overline{1,3}$

$$\begin{aligned}
h_k(s) &= \left\{ \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} dB_k(x) + e^{-\sigma_{k-1}\bar{\pi}_{k-1}(s)\theta_k} \times \right. \\
&\times \int_0^{\infty} e^{-(s+\sigma_{k-1}[1-\pi_{k-1}(s)]x)} dB_k(x) \Big\} \times \\
&\times \left\{ 1 - \sigma_{k-1}\pi_{k-1}(s)\nu_k(s) \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} [1 - B_k(x)] dx \right\}^{-1},
\end{aligned} \tag{3.12}$$

unde $\nu_k(s)$ este exprimat:

pentru schemele $1.J$

$$\nu_k(s) = c_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - c_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1}, \tag{3.13}$$

pentru schemele $2.J$

$$\nu_k(s) = c_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)]), \tag{3.14}$$

pentru schemele $3.J$

$$\begin{aligned}
\nu_k(s) &= (s + \sigma_{k-1}) \int_0^{\infty} e^{-(s+\sigma_{k-1})\tau} \{s + \sigma_{k-1}[1 - \pi_{k-1}(s) \times \\
&\times 1 - e^{-(s+\sigma_{k-1})\tau}] \}^{-1} dC_k(\tau),
\end{aligned} \tag{3.15}$$

expresiile $\bar{\pi}_k(s)$ și $\pi_{k-1}(s)$, menționate mai sus, sunt determinate în mod unic din relațiile recurente ale Teoremei 3.2.

Teorema 3.2 Pentru toate schemele

$$\sigma_k \bar{\pi}_k(s) = \sigma_{k-1} \bar{\pi}_{k-1}(s + a_k - a_k \bar{\pi}_{kk}(s)) + a_k \pi_{kk}(s), \tag{3.16}$$

$$\bar{\pi}_{kk}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}(s)), \tag{3.17}$$

$$\begin{aligned}
\sigma_k \pi_k(s) &= \sigma_{k-1} \pi_{k-1}(s + a_k) + \sigma_{k-1} \{ \pi_{k-1}(s + a_k [1 - \bar{\pi}_{kk}(s)]) - \\
&- \pi_{k-1}(s + a_k) \} \nu_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) + a_k \pi_{kk}(s),
\end{aligned} \tag{3.18}$$

$$\pi_{kk}(s) = \nu_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) \bar{\pi}_{kk}(s), \tag{3.19}$$

unde $h_k(s + a_k - a_k \bar{\pi}_{kk}(s))$ și $\nu_k(s + a_k - a_k \bar{\pi}_{kk}(s))$ pentru fiecare dintre schemele $I.J$ sunt

determinate din relațiile (3.11)-(3.15) respectiv, pentru $s = s + a_k - a_k \bar{\pi}_{kk}(s)$.

Demonstrație. Vom demonstra relațiile (3.11) și (3.12). Mai întâi de toate, este posibil să arătăm, că dacă servirea cerinței de prioritatea k a durat mai mult decât valoarea θ_k , atunci funcția de repartiție a timpului de servire rămas are forma:

$$\frac{B_k(\theta_k + x) - B_k(\theta_k)}{1 - B_k(\theta_k)}.$$

În continuare vom introduce un eveniment suplimentar, cu alte cuvinte vom aplica metoda "catastrofelor" [100], și vom stabili valabilitatea expresiilor ce urmează, astfel conform structurii k ciclului de servire avem:

pentru schemele I.1

$$h_k(s) = \int_0^{\theta_k} e^{-(s+\sigma_{k-1}[1-\pi_{k-1}(s)\nu_k(s)])x} dB_k(x) + e^{-(\sigma_{k-1}[1-\pi_{k-1}(s)\nu_k(s)])\theta_k} [1 - B_k(\theta_k)] \times \\ \times \sum_{n \geq 0} \int_0^{\infty} e^{-sx} \frac{(\sigma_{k-1}x)^n}{n!} e^{-\sigma_{k-1}x} \frac{dB_k(\theta_k + x)}{1 - B_k(\theta_k)} [\bar{\pi}_{k-1}(s)]^n;$$

pentru schemele I.2

$$h_k(s) = \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} dB_k(x) + \int_0^{\theta_k} e^{-sx} [1 - B_k(x)] e^{-\sigma_{k-1}x} \sigma_{k-1} dx \times \\ \times \pi_{k-1}(s) \nu_k(s) h_k(s) + e^{-(s+\sigma_{k-1})\theta_k} [1 - B_k(\theta_k)] \times \\ \times \sum_{n \geq 0} \int_0^{\infty} e^{-sx} \frac{(\sigma_{k-1}x)^n}{n!} e^{-\sigma_{k-1}x} \frac{dB_k(\theta_k + x)}{1 - B_k(\theta_k)} [\bar{\pi}_{k-1}(s)]^n.$$

De unde, după sumare și integrare, vom primi expresiile (3.11) și (3.12). Relațiile (3.16) și (3.17) sunt demonstrate în [96]. Ele sunt veridice și pentru sistemul cu prioritatea DD.

Din relațiile menționate pot fi obținute caracteristici numerice ale servirii. Vom prezenta momentele inițiale ale k - ciclului de trecere, k - ciclului de servire, $\bar{\Pi}_k$ - perioadei, $\bar{\Pi}_{kk}$ - perioadei, Π_k - perioadei și Π_{kk} - perioadei, de exemplu, pentru schemele I.1. Să considerăm

$$\rho_k = \sum_{i=1}^k a_i b_i, \\ b_1 = \frac{\beta_{11} + c_{11}}{1 + a_1 c_{11}}, \\ b_i = \beta_{i1} + [\theta_i - \int_0^{\theta_i} B_i(x) dx] [\Phi_{i-1} \dots \Phi_2 (1 + a_1 c_{11}) q_i - 1],$$

$$\Phi_1 = 1, \Phi_i = 1 + \frac{\sigma_{i-1}\pi_{i-1}(a_i)}{\sigma_{i-1}}[q_i - 1], i = 2, \dots, r.$$

Termenii q_i menționați mai sus, sunt egali cu:

pentru schemele 1.1

$$q_i = \frac{1}{c_i(\sigma_{i-1})},$$

pentru schemele 1.2

$$q_i = 1 + \sigma_i c_{i1},$$

pentru schemele 1.3

$$q_i = \frac{1}{c_i(-\sigma_{i-1})}.$$

Dacă $\rho_k < 1$, atunci momentele inițiale ale k - ciclului de orientare, k - ciclului de servire, $\bar{\Pi}_{kk}$ - perioadei, $\bar{\Pi}_k$ - perioadei, Π_{kk} - perioadei și Π_k - perioadei sunt egale, respectiv:

$$\nu_{k1} = \frac{1}{\sigma_{k-1}}[q_k - 1] \frac{\Phi_{k-1} \cdots \Phi_2(1 + a_1 c_{11})}{1 - \rho_{k-1}},$$

$$h_{k1} = \frac{b_k}{1 - \rho_{k-1}},$$

$$\bar{\pi}_{kk1} = \frac{b_k}{1 - \rho_k},$$

$$\sigma_k \bar{\pi}_{k1} = \frac{\rho_k}{1 - \rho_k},$$

$$\pi_{kk1} = [b_k + \Phi_{k-1} \cdots \Phi_2 \frac{\rho_{k-1}}{\sigma_{k-1}}] \frac{1}{1 - \rho_k},$$

$$\sigma_k \pi_{k1} = \frac{\Phi_k \cdots \Phi_2(1 + a_1 c_{11}) + \rho_k - 1}{1 - \rho_k}.$$

Remarca 3.1 La formalizarea schemelor pentru disciplina Absolută DD schema cu "pierdere" a cerinței întrerupte nu a fost luată în considerare. Vom nota această schemă prin identificatorul I.3 și vom considera următoarea sa modificare: schema I.3₀ - cerința întreruptă este pierdută imediat, de îndată ce a existat o întrerupere în servirea sa; schema I.3₁ - cerința întreruptă "este pierdută" de îndată ce servirea tuturor cerințelor de prioritate mai mare decât cea întreruptă este finalizată; schema I.3₂ - "pierderea" cerinței apare atunci când serverul este pregătit să înceapă servirea următoarei cerințe de aceeași prioritate.

În continuare, lemele menționate ne descriu relațiile pentru repartiția k -ciclului de servire (3.20) - (3.22) pentru schemele I.3₀ - I. 3₂.

Lema 3.3 Pentru schemele I.3₀

$$h_k(s) = \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} dB_k(x) +$$

$$+ e^{-\sigma_{k-1}\bar{\pi}_{k-1}(s)\theta_k} \int_{\theta_k}^{\infty} e^{-(s+\sigma_{k-1}-\sigma_{k-1}\bar{\pi}_{k-1}(s))u} dB_k(u).$$

(3.20)

Lema 3.4 Pentru schemele I.3₁

$$h_k(s) = \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} dB_k(x) + \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} [1-B_k(x)] \sigma_{k-1} dx \pi_{k-1}(s) +$$

$$+ e^{-\sigma_{k-1}\bar{\pi}_{k-1}(s)\theta_k} \int_{\theta_k}^{\infty} e^{-(s+\sigma_{k-1}-\sigma_{k-1}\bar{\pi}_{k-1}(s))u} dB_k(u). \quad (3.21)$$

Lema 3.5 Pentru schemele I.3₂

$$h_k(s) = \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} dB_k(x) + \int_0^{\theta_k} e^{-(s+\sigma_{k-1})x} [1-B_k(x)] \sigma_{k-1} dx \pi_{k-1}(s) \nu_k(s) +$$

$$+ e^{-\sigma_{k-1}\bar{\pi}_{k-1}(s)\theta_k} \int_{\theta_k}^{\infty} e^{-(s+\sigma_{k-1}-\sigma_{k-1}\bar{\pi}_{k-1}(s))u} dB_k(u). \quad (3.22)$$

Demonstrația formulelor (3.20)-(3.22) este analogică demonstrației relațiilor (3.11)-(3.12). Teorema 3.2 este veridică și pentru schema cu "pierdere" a cerinței, trebuie doar de menționat că pentru schemele I.3₀ - I. 3₂ cu pierdere $h_k(s)$ se determină din relațiile (3.11)-(3.12).

▪ Cazuri particulare

Din rezultatele menționate mai sus urmează rezultatele pentru sistemele clasice de prioritate relativă și absolută. Într-adevăr, dacă în sistemul de servire cu prioritatea DD vom considera $C_j = 0$, $j = 1, \dots, r$, $\theta_k = 0$ atunci vom obține un sistem de așteptare $M_r | G_r | 1$ cu prioritatea relativă. Este cunoscut [96], că repartiția perioadei de ocupare este definită (pentru $k = r$) din relațiile:

$$h_k(s) = \beta_k(s + \sigma_{k-1} - \sigma_{k-1}\bar{\pi}_{k-1}(s)), \quad (3.23)$$

$$\bar{\pi}_{kk}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}(s)), \quad (3.24)$$

$$\sigma_k \bar{\pi}_k(s) = \sigma_{k-1} \bar{\pi}_{k-1}(s + a_k - a_k \bar{\pi}_{kk}(s)) + a_k \bar{\pi}_{kk}(s), \quad (3.25)$$

Aici notațiile noastre pentru $\bar{\pi}_{k-1}(s)$ corespund cu $\pi_{k-1}(s)$ conform [96]. Vom lua ca exemplu schema I.2 pentru explicații. Din lema 3.2, pentru $\theta_k = 0$, $C_j = 0$ rezultă $h_k(s) = \beta_k(s + \sigma_{k-1} - \sigma_{k-1} \bar{\pi}_{k-1}(s))$ care conicide cu (3.23). Relațiile (3.16) și (3.17) din Teorema 3.2 coincid cu (3.24) și (3.25), ca o demonstrație a teoremei, structura k -ciclului de servire a rămas anterior. Acum, să considerăm $C_j \neq 0$, $\theta_k = \infty$. Suntem în condițiile unui sistem $M_r | G_r | 1$ cu orientare și prioritate absolută [100].

Pe de altă parte, din lema 3.2 rezultă că

$$h_k(s) = \beta_k(s + \sigma_{k-1}) + \frac{\sigma_{k-1}[1 - \beta_k(s + \sigma_{k-1})]}{s + \sigma_{k-1}} \pi_{k-1}(s) \nu_k(s) h_k(s),$$

sau

$$h_k(s) = \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \nu_k(s) \right\}^{-1}.$$

Astfel, pentru $\theta_k = \infty$ și $C_j \neq 0$ se obțin rezultatele pentru sistemele cu prioritate absolută și orientare [100].

▪ Disciplina Relativă DD

Dacă timpul de servire a cerinței a_k - este mai mic decât valoarea stabilită θ_k , ($k = 2, \dots, r$), atunci cerința sosită de o prioritate mai mare decât k (cerința σ_{k-1}) primește prioritatea relativă, în caz contrar - absolută.

De asemenea, toate notațiile și definițiile menționate mai sus le vom păstra. Suplimentar vom lua în considerare:

$$\bar{\gamma}_k(s) = s + \sigma_k - \sigma_k \bar{\pi}_k(s), \quad \gamma_k(s) = s + \sigma_k - \sigma_k \pi_k(s) \nu_{k+1}(s),$$

$$R_k(s, \theta_k) = 1 - \sigma_{k-1} \pi_{k-1}(s) \nu_k(s) e^{\sigma_{k-1} \bar{\pi}_{k-1}(s) \theta_k} \int_{\theta_k}^{\infty} e^{-(s + \sigma_{k-1})x} [1 - B_k(x)] dx.$$

Lema 3.6 Pentru schemele I.1

$$h_k(s) = \left\{ \int_0^{\theta_k} e^{-\bar{\gamma}_{k-1}(s)x} dB_k(x) + e^{-\sigma_{k-1} \pi_{k-1}(s) \theta_k} \int_{\theta_k}^{\infty} e^{-(s + \sigma_{k-1})x} dB_k(x) \right\} / R_k(s, \theta_k). \quad (3.26)$$

Lema 3.7 Pentru schemele 1.2

$$h_k(s) = \int_0^{\theta_k} e^{-\bar{\gamma}_{k-1}(s)x} dB_k(x) + e^{-(\bar{\gamma}_{k-1}(s) - \gamma_{k-1}(s))\theta_k} \int_{\theta_k}^{\infty} e^{-\gamma_{k-1}(s)x} dB_k(x), \quad (3.27)$$

unde $\nu_k(s)$, în mod corespunzător, pentru fiecare dintre scheme este:

pentru schemele 1.J

$$\nu_k(s) = c_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - c_k(s + \sigma_{k-1})] \bar{\pi}_{k-1}(s) \right\}^{-1}, \quad (3.28)$$

pentru schemele 2.J

$$\nu_k(s) = c_k(\gamma_{k-1}(s)), \quad (3.29)$$

pentru schemele 3.J

$$\begin{aligned} \nu_k(s) &= (s + \sigma_{k-1}) \int_0^{\infty} e^{-(s + \sigma_{k-1})x} dC_k(x) \times \\ &\times (s + \sigma_{k-1} - \sigma_{k-1} [1 - e^{-(s + \sigma_{k-1})x}] \bar{\pi}_{k-1}(s)), \end{aligned} \quad (3.30)$$

unde $\bar{\pi}_{k-1}(s)$ și $\pi_{k-1}(s)$ menționați mai sus, sunt definiți în mod univoc din relațiile recurente ale Teoremei 2.

Teorema 3.3 Pentru toate schemele, transformata Laplace-Stieltjes a funcției de repartiție a perioadei de ocupare $\sigma\pi(s) = \sigma_r\pi_r(s)$ se definește din următoarele relații recurente:

$$\sigma_k \bar{\pi}_k(s) = \sigma_{k-1} \bar{\pi}_{k-1}(s + a_k - a_k \bar{\pi}_{kk}(s)) + a_k \pi_{kk}(s),$$

$$\bar{\pi}_{kk}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}(s)),$$

$$\sigma_k \pi_k(s) = \sigma_{k-1} \pi_{k-1}(s + a_k - a_k \bar{\pi}_{kk}(s)) + a_k \pi_{kk}(s),$$

$$\pi_{kk}(s) = \nu_k(s + a_k - a_k \bar{\pi}_{kk}(s)) \bar{\pi}_{kk}(s),$$

unde $h_k(s + a_k - a_k \bar{\pi}_{kk}(s))$ și $\nu_k(s + a_k - a_k \bar{\pi}_{kk}(s))$, pentru fiecare din scheme, sunt exprimate din relațiile (3.26)-(3.30), respectiv, pentru $s = s + a_k - a_k \bar{\pi}_{kk}(s)$.

Remarca 3.2 Din expresiile obținute se poate determina caracteristicile numerice ale servirii. Vom menționa momentele inițiale ale k - ciclului de orientare, k - ciclului de servire, $\bar{\Pi}_{kk}$ - perioadei, $\bar{\Pi}_k$ - perioadei, Π_{kk} - perioadei și Π_k - perioadei.

$$\text{Vom considera } \rho_k = \sum_{i=1}^k a_i b_i,$$

unde $b_1 = \beta_{11}$ și b_i , $i = 2, \dots, r$ sunt exprimate:

pentru schemele I.1

$$b_i = \{\theta_i[1 - B_i(\theta_i)] + \int_0^{\theta_i} x dB_i(x) + (a_1 c_{11} + q_i + \sum_{j=2}^{i-1} \frac{a_j}{\sigma_{j-1}} (q_j - 1) \times \\ \times (\frac{1 - B_i(\theta_i)}{\sigma_{i-1}} - \frac{e^{\sigma_{i-1}\theta_i}}{\sigma_{i-1}} \int_{\theta_i}^{\infty} e^{-\sigma_{i-1}x} dB_i(x)) \{B_i(\theta_i) + e^{\sigma_{i-1}\theta_i} \int_{\theta_i}^{\infty} e^{-\sigma_{i-1}x} dB_i(x)\}^{-1},$$

pentru schemele I.2

$$b_i = \theta_i[1 - B_i(\theta_i)] + \int_0^{\theta_i} x dB_i(x) + \\ + (a_1 c_{11} + q_i + \sum_{j=2}^{i-1} \frac{a_j}{\sigma_{j-1}} (q_j - 1) (\int_{\theta_i}^{\infty} x dB_i(x) - \theta_i[1 - B_i(\theta_i)]).$$

În aceste formule q_i sunt egali cu:

pentru schemele 1.J

$$q_i = \frac{1}{c_i(\sigma_{i-1})},$$

pentru schemele 2.J

$$q_i = 1 + \sigma_{i-1} c_{i1},$$

pentru schemele 3.J

$$q_i = \frac{1}{c_i(-\sigma_{i-1})}.$$

Dacă $\rho_k < 1$, atunci momentele inițiale a k - ciclului de orientare, k - ciclului de servire, $\bar{\Pi}_k$ - perioadei, $\bar{\Pi}_{kk}$ - perioadei, Π_{kk} - perioadei, Π_k - perioadei sunt egale, respectiv:

$$\nu_{k1} = \frac{\rho_k - 1}{\sigma_{k-1}(1 - \rho_{k-1})}, \\ h_{k1} = \frac{b_k}{1 - \rho_{k-1}}, \quad \bar{\pi}_{kk1} = \frac{b_k}{1 - \rho_k}, \\ \sigma_k \bar{\pi}_{k1} = \frac{\rho_k}{1 - \rho_k}, \quad \pi_{kk1} = (b_k + \frac{\rho_k - 1}{\sigma_{k-1}}) / \rho_{k-1}, \\ \sigma_k \pi_{k1} = (a_1 c_{11} + \sum_{i=2}^k \frac{a_i}{\sigma_{i-1}} (q_i - 1) + \rho_k) / (1 - \rho_k).$$

Demonstrația rezultatelor menționate se efectuează analog utilizând aceleași tehnici ca și în cazul precedent.

3.5 Repartiția lungimii șirului de așteptare pentru prioritatea DD

Vom nota prin $P_m(t)$ probabilitatea că la momentul t în șirul de așteptare sunt $m = (m_1, \dots, m_r)$ cerințe, unde m_i este numărul de cerințe de clasa de prioritate i . Vom considera

$P(z, t) = \sum P_m(t) z^m$, $z^m = z_1^{m_1}, \dots, z_r^{m_r}$, $0 \leq z_i \leq 1$, iar transformata Laplace este

$$p(z, s) = \int_0^\infty e^{-st} P(z, t) dt.$$

▪ Disciplina Absolută DD

Repartiția lungimii șirului de așteptare a unui k - ciclu de servire separat, în termeni de transformată Laplace, este dată de următoarea teoremă.

Lema 3.8 Pentru schemele I.1 [63]

$$\begin{aligned} h_k(z, s) &= z_k [1 + \sigma_{k-1} (\pi_{k-1}(z, s) + \pi_{k-1}(s + \omega_k) v_k(z, s))] \times \\ &\times \int_0^{\theta_k} [1 - B_k(x)] e^{-\eta_k(s, z)x} dx + z_k (1 + \sigma_{k-1} \bar{\pi}_{k-1}(z, s)) \times \\ &\times e^{-[\eta_k(s, z) - \xi_k(s, z)] \theta_k} \int_{\theta_k}^\infty [1 - B_k(x)] e^{-\xi_k(s, z)x} dx. \end{aligned}$$

Pentru schemele I.2

$$\begin{aligned} h_k(z, s) &= \{z_k [1 + \sigma_{k-1} (\pi_{k-1}(z, s) + \pi_{k-1}(s + \omega_k) v_k(z, s))] \times \\ &\times \int_0^{\theta_k} [1 - B_k(x)] e^{-(s + \omega_k + \sigma_{k-1})x} dx + z_k (1 + \sigma_{k-1} \bar{\pi}_{k-1}(z, s)) \times \\ &\times e^{-[\eta_k(s, z) - \xi_k(s, z)] \theta_k} \int_{\theta_k}^\infty [1 - B_k(x)] e^{-\xi_k(s, z)x} dx\} \times \\ &\times \{1 - \sigma_{k-1} \pi_{k-1}(s + \omega_k) \int_0^{\theta_k} [1 - B_k(x)] e^{-(s + \omega_k + \sigma_{k-1})x} dx\}^{-1}. \end{aligned}$$

unde

$$\xi(s, z) = s + \omega_k + \sigma_{k-1} [1 - \bar{\pi}_{k-1}(s + \omega_k)],$$

$$\eta_k(s, z) = s + \omega_k + \sigma_{k-1} [1 - \pi_{k-1}(s + \omega_k)];$$

și $v_k(z, s)$ pentru fiecare schemă respectiv, este egal cu:

pentru schemele 1.J:

$$v_k(z, s) = \frac{1 - c_k(s + \omega_k) + \sigma_{k-1} [1 + \sigma_{k-1} \pi_{k-1}(z, s)]}{s + \omega_k + \sigma_{k-1} [1 - c_k(s + \omega_k + \sigma_{k-1})] \pi_{k-1}(s + \omega_k)}, \quad (3.31)$$

pentru schemele 2.J:

$$v_k(z, s) = (1 + \sigma_{k-1}\pi_{k-1}(z, s)) \frac{1 - c_k(s + \omega_k + \sigma_{k-1}[1 - \pi_{k-1}(s + \omega_k)])}{s + \omega_k + \sigma_{k-1}[1 - \pi_{k-1}(s + \omega_k)]}, \quad (3.32)$$

pentru schemele 3.J:

$$v_k(z, s) = (1 + \sigma_{k-1}\pi_{k-1}(z, s)) \times \int_0^\infty \frac{\{1 - e^{-(s + \omega_k + \sigma_{k-1})x}\} dC_k(x)}{s + \omega_k + \sigma_{k-1} - \sigma_{k-1}\pi_{k-1}(s + \omega_k)v_k(s + \omega_k)[1 - e^{-(s + \omega_k + \sigma_{k-1})x}]}. \quad (3.33)$$

unde $v_k(s + \omega_k)$, $\pi_{k-1}(s + \omega_k)$ și $\bar{\pi}_{k-1}(s + \omega_k)$ sunt definite din relațiile (3.13)-(3.15) și (3.16)-(3.19) pentru $s = s + \omega_k$; însă expresiile $\pi_{k-1}(z, s)$, $\bar{\pi}_{k-1}(z, s)$ - din formulele (3.34)-(3.36), recurent.

Teorema 3.4 [100]

$$p(z, s) = \frac{1 + \sigma\pi(z, s)}{s + \sigma - \sigma\pi(s)}, \quad (3.34)$$

$$\sigma_k\pi_k(z, s) = \sigma_{k-1}\pi_{k-1}(z, s) + \gamma_{k-1}(s, z)v_k(z, s) + \frac{h_k(z, s)}{z_k - h_k(s + \omega_k)} \times \\ \times [\gamma_{k-1}(s, z)v_k(s + \omega_k)\sigma_{k-1}\pi_{k-1}(s + a_k) - \sigma_k\pi_k(s)], \quad (3.35)$$

$$\sigma_k\bar{\pi}_k(z, s) = \sigma_{k-1}\bar{\pi}_{k-1}(z, s) + h_k(z, s) \frac{\xi_k(s, z) - \xi_{k-1}(s, z)}{z_k - h_k(s + \omega_k)}, \quad (3.36)$$

unde $\gamma_{k-1}(s, z) = \sigma_{k-1}[\bar{\pi}_{k-1}(s + \omega_k) - \pi_{k-1}(s + a_k)] + a_k z_k$, și $\sigma\pi(s) = \sigma_r\pi_r(s)$ sunt definite din teorema 3.2.

Vom nota prin $P(z)$ funcția generatoare a repartiției comune a lungimii șirului de așteptare pentru regimul staționar. Dacă $\rho_r < 1$

$$P(z) = \frac{1 + \hat{\pi}(z)}{1 + \sigma\pi_1},$$

unde $\sigma\hat{\pi}(z) = \sigma_r\pi_r(z, 0)$, $\sigma\pi_1 = \sigma_r\pi_{r1}$.

▪ Disciplina Relativă DD

Vom considera

$$\omega_k = v_k = a_k(1 - z_k) + \dots + a_r(1 - z_r), \quad u_k = (a_1 z_1 + \dots + a_k z_k) / \sigma_k.$$

Lema 3.9 Pentru schemele I.1 [63]

$$\begin{aligned}
h_k(z, s) = & \{z_k \int_0^{\theta_k} e^{-(s+v_1)x} [1 - B_k(x)] dx + \\
& + [1 + \sigma_{k-1} \pi_{k-1}(z, s)] z_k e^{\sigma_{k-1} u_{k-1} \theta_k} \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_k)x} [1 - B_k(x)] dx + \\
& + \nu_k(z, s) \sigma_{k-1} \pi_{k-1}(s + v_k) z_k e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k} \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_k)x} [1 - B_k(x)] dx + \\
& + \frac{\pi_{k-1}(z, s)}{u_{k-1} - \pi_{k-1}(s + v_k)} \left[\int_0^{\theta_k} (e^{-(s+v_k)x} - e^{-\gamma_{k-1}(s)x}) dB_k(x) + (e^{\sigma_{k-1} u_{k-1} \theta_k} - \right. \\
& - e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k}) \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_{k-1})x} dB_k(x) + \sigma_{k-1} \pi_{k-1}(s + v_k) z_k (e^{\sigma_{k-1} u_{k-1} \theta_k} - \\
& \left. - e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k}) \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_{k-1})x} [1 - B_k(x)] dx \right] / R_k(s + v_k, \theta_k).
\end{aligned}$$

Pentru schemele I.2

$$\begin{aligned}
h_k(z, s) = & \{z_k \int_0^{\theta_k} e^{-(s+v_1)x} [1 - B_k(x)] dx + [1 + \sigma_{k-1} \pi_{k-1}(z, s)] z_k (e^{\sigma_{k-1} u_{k-1} \theta_k} - \\
& - e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k}) \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_{k-1})x} [1 - B_k(x)] dx + [1 + \sigma_{k-1} \pi_{k-1}(z, s) + \\
& + \sigma_{k-1} \pi_{k-1}(s + v_k) - \nu_k(z, s)] z_k (e^{-(\bar{\gamma}_{k-1}(s, z) - \gamma_{k-1}(s, z)) \theta_k} \int_{\theta_k}^{\infty} e^{-\gamma_{k-1}(s, z)x} [1 - B_k(x)] dx + \\
& + \frac{\bar{\pi}_{k-1}(z, s)}{u_{k-1} - \pi_{k-1}(s + v_k)} \left[\int_0^{\theta_k} (e^{-(s+v_k)x} - e^{-\bar{\gamma}_{k-1}(s, z)x}) dB_k(x) + \right. \\
& + (e^{\sigma_{k-1} u_{k-1} \theta_k} - e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k}) \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_{k-1})x} dB_k(x) + \\
& \left. + \sigma_{k-1} \pi_{k-1}(s + v_k) z_k (e^{\sigma_{k-1} u_{k-1} \theta_k} - e^{\sigma_{k-1} \bar{\pi}_{k-1}(s+v_k) \theta_k}) \int_{\theta_k}^{\infty} e^{-(s+v_k+\sigma_{k-1})x} [1 - B_k(x)] dx \right],
\end{aligned}$$

unde

$$\bar{\gamma}_{k-1}(s, z) = \bar{\gamma}_{k-1}(s + v_k), \quad \gamma_{k-1}(s, z) = \gamma_{k-1}(s + v_k); \quad \nu_k(z, s)$$

sunt definite din expresiile (3.31)-(3.33), iar relațiile $\pi_{k-1}(z, s)$ și $\bar{\pi}_{k-1}(z, s)$ sunt definite recurent din fomulele de mai jos menționate.

Teorema 3.5. [100]

$$p(z, s) = \frac{1 + \sigma\pi(z, s)}{s + \sigma - \sigma\pi(s)},$$

$$\sigma_k \bar{\pi}_k(z, s) = \sigma_{k-1} \bar{\pi}_{k-1}(z, s) + h_k(z, s) \frac{\bar{\gamma}_k(s, z) - \bar{\gamma}_{k-1}(s, z)}{z_k - h_k(s + v_k)},$$

$$\sigma_k \pi_k(z, s) = \sigma_{k-1} \pi_{k-1}(z, s) + h_k(z, s) [\sigma_{k-1} \pi_{k-1}(s + v_k) + \sigma_k \pi_k(s + v_{k+1}) + a_k z_k v_k(s + v_k)] \{z_k - h_k(s + v_k)\}^{-1} + a_k z_k v_k(z, s).$$

Remarca 3.3 Expresiile pentru schema "cu pierdere" a cerinței se obțin similar lemelor 3.8 și 3.9.

Momentul cheie pentru rezolvarea sistemului de ecuații recurente (3.16) - (3.19) este ecuația funcțională (3.17). În [100] se demonstrează că această ecuație poate fi văzută ca analogul n - dimensional (în sensul claselor de prioritate) a cunoscutei ecuației funcționale Kendall. Această ecuație nu are soluție analitică exactă, totuși, este, în mod eficient, rezolvată numeric.

Sistemul de ecuații recurente funcționale (3.16)-(3.19) este implicat în majoritatea caracteristicilor de performanță pentru toate schemele de prioritate DD, astfel ca lungimea șirului de așteptare, probabilitățile stărilor, etc. [63]. Algoritmii numerici pentru determinarea acestor caracteristici obligatoriu necesită soluționarea numerică și utilizarea soluției sistemului de ecuații menționat.

3.6. Algoritmi și modelări numerice de calcul ale perioadei de ocupare și a caracteristicilor auxiliare pentru Disciplina DD

În continuare se vor prezenta algoritmi numerici elaborați pentru soluționarea numerică a repartiției perioadei de ocupare și a caracteristicilor auxiliare de performanță pentru unele schemele cu prioritatea DD și diverse funcții de repartiții. Algoritmii elaborați au fost implementați în mediul de programare Kotlin, codul sursă este prezentat în Anexa 2.

Modelarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele de așteptare generalizate cu prioritatea DD, pentru schema 2.3 - orientarea-timpul rămas, servirea-continuă timpul rămas de servire, se face în baza formulelor:

$$h_k(s) = \beta_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)v_k(s)]);$$

$$v_k(s) = c_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)]).$$

Perioada de ocupare și caracteristicile auxiliare cu prioritatea DD, pentru schema 2.3, pot

fi evaluate cu ajutorul următorului algoritm:

Algoritm 3.1

Date de intrare: $\{a_k\}_{k=1}^r; \{b_k\}_{k=1}^r; \{c_k\}_{k=1}^r; s; r; \varepsilon > 0$.

Date de ieșire: $k; \{h_k(s)\}_{k=1}^r; \{v_k(s)\}_{k=1}^r; \{\pi_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ și $C_k(x)$ sunt funcții de repartiție exponențiale.

$$\beta_k(s) = \frac{b_k}{s + b_k}; \quad c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

- b) Se calculează repartiția perioadei de ocupare și caracteristicilor auxiliare, conform relațiilor:

- dacă $B_k < \theta_k$ ($k = \overline{2, r}$) atunci

$$\begin{aligned} \pi_k(s) &= \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k) + \frac{\sigma_{k-1}}{\sigma_k} \{ \pi_{k-1}(s + a_k [1 - \bar{\pi}_{kk}(s)]) - \\ &- \pi_{k-1}(s + a_k) \} v_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) + \frac{a_k}{\sigma_k} \pi_{kk}(s); \end{aligned}$$

$$\pi_{kk}(s) = v_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) \bar{\pi}_{kk}(s);$$

$$\bar{\pi}_{kk}^{(n)}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}^{(n-1)}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1} [1 - \pi_{k-1}(s) v_k(s)]);$$

$$v_k(s) = c_k(s + \sigma_{k-1} [1 - \pi_{k-1}(s)]);$$

$$|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| < \varepsilon;$$

- dacă $B_k \geq \theta_k$ ($k = \overline{2, r}$) atunci

$$\pi_k(s) = \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k (1 - \pi_{kk}(s))) + \frac{a_k}{\sigma_k} \pi_{kk}(s);$$

$$\pi_{kk}^n(s) = h_k(s + a_k - a_k \pi_{kk}^{n-1}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1};$$

$$|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| < \varepsilon;$$

Condiția de oprire: $k = r$.

Exemplul 3.1 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 2.3: orientarea-timpul rămas, servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametri $a_k = \{0.6, 0.1, 0.9, 0.2, 0.3, 0.7, 0.5, 0.1, 0.3, 0.9\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametri $b_k = \{0.2, 0.4, 0.8, 0.9, 0.2, 0.3, 0.1, 0.2, 0.6, 0.3\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametri $c_k = \{0.3, 0.1, 0.2, 0.7, 0.3, 0.4, 0.5, 0.1, 0.8, 0.9\}$. Valoarea stabilită $\theta_k = 0.4$, $s = 0.2$, $\varepsilon = 10^{-5}$.

Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.1.

Tabelul 3.1. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$v_k(s)$	$\pi_k(s)$
1	0.666666	0.333333	0.186139
2	0.723293	0.000000	0.506333
3	0.372252	0.000000	0.437902
4	0.091455	0.000000	0.312222
5	0.323268	0.355876	0.043644
6	0.333333	0.714285	0.060384
7	0.223194	0.129902	0.043979
8	0.432717	0.508384	0.074951
9	0.196031	0.000000	0.129207
10	0.291546	0.830957	0.062050

Exemplul 3.2 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 2.3: orientarea-timpul rămas, servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametri $a_k = \{2, 1, 4, 6, 3, 2, 3, 2, 1, 4\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametri $b_k = \{0.3, 0.9, 0.1, 0.3, 0.9, 0.4, 7, 0.8, 0.2, 0.1\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție

exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametri $c_k = \{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.4, 0.4, 0.3, 0.3\}$. Valoarea stabilită $\theta_k = 0.7$, $s = 0.5$, $\varepsilon = 10^{-5}$.

Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.2.

Tabelul 3.2. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.642857	0.615384	0.202807
2	0.049754	0.000000	0.043514
3	0.051747	0.098797	0.001321
4	0.072584	0.041717	0.001089
5	0.026188	0.000000	0.024315
6	0.011976	0.017870	3.148412e-4
7	0.615384	0.000000	0.295256
8	0.061352	0.000000	0.156268
9	0.027809	0.080035	0.001902
10	0.038463	0.026019	4.749227e-4

Modelarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele de așteptare generalizate cu prioritatea DD pentru schema 1.3 - orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, se face în baza relațiilor:

$$h_k(s) = \beta_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)\nu_k(s)]);$$

$$\nu_k(s) = c_k(s + \sigma_{k-1})\left\{1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}}[1 - c_k(s + \sigma_{k-1})\pi_{k-1}(s)]\right\}^{-1}.$$

Perioada de ocupare și caracteristicile auxiliare cu prioritatea DD pentru schema 1.3 pot fi evaluate cu ajutorul următorului algoritm:

Algoritm 3.2

Date de intrare: $\{a_k\}_{k=1}^r$; $\{b_k\}_{k=1}^r$; $\{c_k\}_{k=1}^r$; s ; r ; $\varepsilon > 0$.

Date de ieșire: k ; $\{h_k(s)\}_{k=1}^r$; $\{\nu_k(s)\}_{k=1}^r$; $\{\pi_k(s)\}_{k=1}^r$.

Descriere:

- Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ și $C_k(x)$ sunt funcții de repartiție exponențiale.

$$\beta_k(s) = \frac{b_k}{s + b_k};$$

$$c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

b) Se calculează repartiția perioadei de ocupare și caracteristicilor auxiliare, conform relațiilor:

- dacă $B_k < \theta_k$ ($k = \overline{2, r}$) atunci

$$\pi_k(s) = \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k) + \frac{\sigma_{k-1}}{\sigma_k} \{ \pi_{k-1}(s + a_k [1 - \bar{\pi}_{kk}(s)]) - \pi_{k-1}(s + a_k) \} v_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) + \frac{a_k}{\sigma_k} \pi_{kk}(s);$$

$$\pi_{kk}(s) = v_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) \bar{\pi}_{kk}(s);$$

$$\bar{\pi}_{kk}^{(n)}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}^{(n-1)}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1} [1 - \pi_{k-1}(s) v_k(s)]);$$

$$v_k(s) = c_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - c_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1},$$

$$|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| < \varepsilon;$$

- dacă $B_k \geq \theta_k$ ($k = \overline{2, r}$) atunci

$$\pi_k(s) = \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k (1 - \pi_{kk}(s))) + \frac{a_k}{\sigma_k} \pi_{kk}(s);$$

$$\pi_{kk}^n(s) = h_k(s + a_k - a_k \pi_{kk}^{n-1}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1};$$

$$|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| < \varepsilon;$$

Condiția de oprire: $k = r$.

Exemplul 3.3 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 1.3: orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{0.2, 0.5, 0.3, 0.1, 0.8, 0.4, 0.5, 0.6, 0.7, 0.8\}$. Timpul

de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.2, 0.1, 0.8, 0.2, 0.1, 0.1, 0.6, 0.3, 0.2, 0.1\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.2, 0.4, 0.9, 0.3, 0.1, 0.1, 0.2, 0.3, 0.2, 0.1\}$. Valoarea stabilită $\theta_k = 0.3$, $s = 0.1$, $\varepsilon = 10^{-5}$.

Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.3.

Tabelul 3.3. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.500000	0.800000	0.070265
2	0.580376	0.614390	0.290623
3	0.154268	0.000000	0.148270
4	0.091700	0.099534	0.002969
5	0.271605	0.053822	0.006230
6	0.857142	0.666666	0.319176
7	0.244746	0.000000	0.204944
8	0.115296	0.000000	0.092162
9	0.050040	0.050810	0.001310
10	0.228612	0.035938	0.005375

Exemplul 3.4 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 1.3: orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{0.3, 1, 23, 0.9, 8, 6, 11, 0.7, 0.3, 45\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.5, 0.3, 0.9, 2, 0.7, 0.5, 0.7, 0.1, 0.1, 0.2\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.23, 0.1, 0.9, 22, 15, 0.78, 23, 17, 0.34, 6\}$. Valoarea stabilită $\theta_k = 0.8$, $s = 0.3$, $\varepsilon = 10^{-5}$.

Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.4.

Tabelul 3.4. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$v_k(s)$	$\pi_k(s)$
1	0.500000	0.250000	0.018333
2	0.410502	0.412528	0.001366
3	0.073762	0.000000	0.071695
4	0.026125	0.000000	0.020437
5	0.213339	0.153409	0.028100
6	0.700000	0.987124	0.042304
7	0.008996	0.610749	0.005025
8	0.008272	0.028549	2.245865e-4
9	0.016180	0.335497	3.311862e-4
10	0.003478	0.108955	3.235874e-4

Modelarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele de așteptare generalizate cu prioritatea DD pentru schema 2.2 - orientarea-timpul rămas, servirea-cerința se pierde, este următoarea:

$$h_k(s) = \beta_k(s + \sigma_{k-1}) + \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) v_k(s),$$

$$v_k(s) = c_k(s + \sigma_{k-1} [1 - \pi_{k-1}(s)]).$$

Perioada de ocupare și caracteristicile auxiliare cu prioritatea DD, pentru schema 2.2, pot fi evaluate cu ajutorul următorului algoritm:

Algoritm 3.3

Date de intrare: $\{a_k\}_{k=1}^r$; $\{b_k\}_{k=1}^r$; $\{c_k\}_{k=1}^r$; s ; r ; $\varepsilon > 0$.

Date de ieșire: k ; $\{h_k(s)\}_{k=1}^r$; $\{v_k(s)\}_{k=1}^r$; $\{\pi_k(s)\}_{k=1}^r$.

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ și $C_k(x)$ sunt funcții de repartiție exponențiale.

$$\beta_k(s) = \frac{b_k}{s + b_k}; \quad c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

- b) Se calculează repartiția perioadei de ocupare și caracteristicilor auxiliare, conform relațiilor:

- dacă $B_k < \theta_k$ ($k = \overline{2, r}$) atunci

$$\begin{aligned}
\pi_k(s) &= \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k) + \frac{\sigma_{k-1}}{\sigma_k} \{ \pi_{k-1}(s + a_k [1 - \bar{\pi}_{kk}(s)]) - \\
&- \pi_{k-1}(s + a_k) \} \nu_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) + \frac{a_k}{\sigma_k} \pi_{kk}(s); \\
\pi_{kk}(s) &= \nu_k(s + a_k [1 - \bar{\pi}_{kk}(s)]) \bar{\pi}_{kk}(s); \\
\bar{\pi}_{kk}^{(n)}(s) &= h_k(s + a_k - a_k \bar{\pi}_{kk}^{(n-1)}(s)); \\
h_k(s) &= \beta_k(s + \sigma_{k-1}) + \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \nu_k(s), \\
\nu_k(s) &= c_k(s + \sigma_{k-1} [1 - \pi_{k-1}(s)]); \\
|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| &< \varepsilon;
\end{aligned}$$

- dacă $B_k \geq \theta_k$ ($k = \overline{2, r}$) atunci

$$\begin{aligned}
\pi_k(s) &= \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k (1 - \pi_{kk}(s))) + \frac{a_k}{\sigma_k} \pi_{kk}(s); \\
\pi_{kk}^n(s) &= h_k(s + a_k - a_k \pi_{kk}^{n-1}(s)); \\
h_k(s) &= \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1}; \\
|\bar{\pi}_{kk}^{(n)}(s) - \bar{\pi}_{kk}^{(n-1)}(s)| &< \varepsilon;
\end{aligned}$$

Condiția de oprire: $k = r$.

Exemplul 3.5 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 2.2: orientarea-timpul rămas, servirea-cerința se pierde, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{0.3, 0.2, 0.9, 0.7, 0.5, 0.45, 0.23, 0.9, 0.1, 0.27\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.2, 0.4, 0.1, 0.6, 0.3, 0.45, 0.23, 0.69, 0.22, 0.1\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.1, 0.3, 0.2, 0.6, 0.9, 0.02, 0.9, 0.34, 0.29, 0.1\}$. Valoarea stabilită $\theta_k = 0.7$, $s = 0.2$, $\varepsilon = 10^{-5}$. Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.5.

Tabelul 3.5. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.666666	0.600000	0.301514
2	0.244693	0.370578	0.017302
3	0.318984	0.318984	0.028790
4	0.137554	0.315991	0.021774
5	0.267979	0.222227	0.021797
6	0.534883	0.818181	0.295400
7	0.645449	0.484290	0.101413
8	0.156590	0.193034	0.082893
9	0.070025	0.070025	0.047477
10	0.027009	0.234875	0.029265

Exemplul 3.6 Se consideră un sistem Polling cu prioritatea DD pentru schema 2.2: orientarea-timpul rămas, servirea-cerința se pierde, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{3, 2, 8, 16, 22, 1, 7, 33, 56, 8\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție exponențială, $B_k(x) = 1 - e^{-b_k x}$, $x > 0$, cu parametrii $b_k = \{0.3, 0.1, 0.2, 0.56, 0.23, 0.45, 0.89, 0.12, 0.3, 0.44\}$ și timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.4, 3, 8, 0.1, 4, 0.6, 9, 11, 41, 0.7\}$. Valoarea stabilită $\theta_k = 0.5$, $s = 0.4$, $\varepsilon = 10^{-5}$. Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.6.

Tabelul 3.6. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare pentru prioritatea DD

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.200000	0.882352	0.023339
2	0.090795	0.772698	0.007211
3	0.051157	0.009589	6.198288e-4
4	0.008716	0.000000	0.006778
5	0.010107	0.312558	5.780597e-4
6	0.689922	0.957446	0.069039
7	0.055413	0.000000	0.006239
8	0.007986	0.503993	6.475204e-4
9	0.004548	0.007213	5.215633e-4
10	0.008551	0.008551	4.099195e-4

Modelarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele de așteptare generalizate cu prioritatea DD pentru schema 1.3- orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, este următoarea:

$$h_k(s) = \beta_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)\nu_k(s)]);$$

$$\nu_k(s) = c_k(s + \sigma_{k-1})\{1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}}[1 - c_k(s + \sigma_{k-1})]\pi_{k-1}(s)\}^{-1}.$$

Perioada de ocupare și caracteristicile auxiliare cu prioritatea DD, pentru schema 1.3, pot fi evaluate cu ajutorul următorului algoritm:

Algoritm 3.4

Date de intrare: $\{a_k\}_{k=1}^r; \{b_k\}_{k=1}^r; \{c_k\}_{k=1}^r; \{p_k\}_{k=1}^r; s; r; \varepsilon > 0.$

Date de ieșire: $k; \{h_k(s)\}_{k=1}^r; \{\nu_k(s)\}_{k=1}^r; \{\pi_k(s)\}_{k=1}^r.$

Descriere:

- a) Se determină transformatele Laplace-Stieltjes ale funcțiilor de repartiție $B_k(x)$ și $C_k(x)$, respectiv, unde $B_k(x)$ este o funcție de repartiție Erlang și $C_k(x)$ este o funcție de repartiție exponențială.

$$\beta_k(s) = \left(\frac{b_k}{s + b_k} \right)^{p_k};$$

$$c_k(s) = \frac{\bar{c}_k}{s + \bar{c}_k}.$$

- b) Se calculează repartiția perioadei de ocupare și caracteristicilor auxiliare, conform relațiilor:

- dacă $B_k < \theta_k$ ($k = \overline{2, r}$)

atunci

$$\begin{aligned} \pi_k(s) &= \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k) + \frac{\sigma_{k-1}}{\sigma_k} \{ \pi_{k-1}(s + a_k[1 - \bar{\pi}_{kk}(s)]) - \\ &- \pi_{k-1}(s + a_k) \} \nu_k(s + a_k[1 - \bar{\pi}_{kk}(s)]) + \frac{a_k}{\sigma_k} \pi_{kk}(s); \end{aligned}$$

$$\pi_{kk}(s) = \nu_k(s + a_k[1 - \bar{\pi}_{kk}(s)]) \bar{\pi}_{kk}(s);$$

$$\bar{\pi}_{kk}^{(n)}(s) = h_k(s + a_k - a_k \bar{\pi}_{kk}^{(n-1)}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1}[1 - \pi_{k-1}(s)\nu_k(s)]);$$

$$v_k(s) = c_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - c_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1},$$

$$\left| \overline{\pi}_{kk}^{(n)}(s) - \overline{\pi}_{kk}^{(n-1)}(s) \right| < \varepsilon;$$

- dacă $B_k \geq \theta_k$ ($k = \overline{2, r}$)

atunci

$$\pi_k(s) = \frac{\sigma_{k-1}}{\sigma_k} \pi_{k-1}(s + a_k(1 - \pi_{kk}(s))) + \frac{a_k}{\sigma_k} \pi_{kk}(s);$$

$$\pi_{kk}^n(s) = h_k(s + a_k - a_k \pi_{kk}^{n-1}(s));$$

$$h_k(s) = \beta_k(s + \sigma_{k-1}) \left\{ 1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}} [1 - \beta_k(s + \sigma_{k-1})] \pi_{k-1}(s) \right\}^{-1};$$

$$\left| \overline{\pi}_{kk}^{(n)}(s) - \overline{\pi}_{kk}^{(n-1)}(s) \right| < \varepsilon;$$

Condiția de oprire: $k = r$.

Exemplul 3.7 Se consideră un sistem de așteptare generalizat cu prioritatea DD pentru schema 1.3: orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{0.2, 0.1, 0.9, 0.6, 0.8, 0.2, 0.1, 0.3, 0.2, 0.9\}$. Timpul de servire al cerințelor de clasă k este

o variabilă aleatoare cu funcția de repartiție Erlang, $B_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x b_k \frac{(b_k x)^{p_k-1}}{(p_k-1)!} e^{-b_k x} dx, & x \geq 0 \end{cases}$, cu

parametrii $b_k = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.2, 0.3, 0.2, 0.1, 0.1\}$ și $p_k = \{0.5, 0.2, 8, 0.4, 4, 0.1, 0.3, 0.5, 0.9, 0.23\}$, iar timpul de orientare de la un șir de așteptare la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.4, 0.9, 0.6, 0.9, 0.1, 0.2, 0.2, 0.01, 0.9, 0.7\}$.

Valoarea stabilită $\theta_k = 0.4$, $s = 0.5$, $\varepsilon = 10^{-5}$.

Rezultatele numerice ale caracteristicilor de performanță pentru sistemul menționat sunt prezentate în Tabelul 3.7.

Tabelul 3.7. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.778370	0.642857	0.489282
2	1.918259e-4	0.521253	0.025329
3	0.537282	0.378999	0.075940
4	8.396915e-4	0.000000	0.135735
5	0.837261	0.000000	0.292063
6	0.522701	0.444444	0.140271
7	0.534522	0.019607	0.007274
8	0.138574	0.530092	0.025610
9	0.576717	0.414889	0.117760
10	0.127903	0.127903	0.058166

Exemplul 3.8 Se consideră un sistem Polling cu prioritatea DD pentru schema 1.3: orientarea-din nou, cu timp nou de realizare; servirea-continuă timpul rămas de servire, format din k șiruri de așteptare $k = \overline{1, 10}$. Cerințele sosesc în șirurile de așteptare conform fluxului de tip Poisson, cu parametrii $a_k = \{3, 1, 9, 23, 48, 3, 33, 41, 9, 14\}$. Timpul de servire al cerințelor de clasă k este o variabilă aleatoare cu funcția de repartiție Erlang,

$$B_k(x) = \begin{cases} 0, & x < 0 \\ \int_0^x b_k \frac{(b_k x)^{p_k-1}}{(p_k-1)!} e^{-b_k x} dx, & x \geq 0 \end{cases}, \text{ cu parametrii } b_k = \{0.5, 0.9, 0.3, 0.1, 0.1, 0.7, 0.4, 0.1,$$

0.2, 0.1\} și $p_k = \{3, 0.3, 10, 22, 0.09, 0.1, 3, 7, 27, 30\}$, iar timpul de orientare de la un șir la șirul de așteptare k este considerat o variabilă aleatoare cu funcția de repartiție exponențială, $C_k(x) = 1 - e^{-c_k x}$, $x > 0$ cu parametrii $c_k = \{0.4, 7, 9, 17, 0.4, 3, 0.5, 25, 34, 29\}$. Valoarea stabilită $\theta_k = 0.9$, $s = 0.8$, $\varepsilon = 10^{-5}$. Rezultatele numerice sunt prezentate în Tabelul 3.8.

Tabelul 3.8. Valorile numerice ale repartiției k -perioadei de ocupare și caracteristici auxiliare

k	$h_k(s)$	$\nu_k(s)$	$\pi_k(s)$
1	0.826300	0.897435	0.699099
2	1.411732e-10	0.000000	0.046897
3	1.902816e-45	0.615705	0.001757
4	0.591946	0.011715	0.002697
5	2.105299e-16	0.002445	0.001995
6	0.037037	0.384615	1.166132e-10
7	1.943578e-18	0.425170	2.228197e-10
8	3.170704e-70	0.312500	1.264130e-10
9	1.937051e-88	0.257092	5.847553e-11
10	8.132680e-5	0.020040	6.896600e-8

3.7. Concluzii la capitolul 3

În Capitolul 3 s-a analizat conceptele de bază ale sistemelor de așteptare Polling cu priorități, care din punct de vedere matematic, sunt mai avansate decât modelele clasice și mai adecvate proceselor reale. De asemenea, s-a prezentat unele aspecte conceptuale pentru disciplina de prioritate DD, unde această disciplină este mai flexibilă decât disciplinele clasice de prioritate absolută și relativă, care sunt caracterizate printr-un nivel ridicat de conservare.

Astfel:

- s-au clasificat sistemele generalizate de așteptare cu priorități;
- s-au prezentat repartițiile perioadei de ocupare și a perioadelor auxiliare pentru fiecare strategie în stare liberă și schemele de prioritate absolută, pentru sistemul generalizat de așteptare cu priorități;
- s-au prezentat rezultate cu privire la perioada de ocupare și caracteristici auxiliare pentru prioritatea DD;
- s-au elaborat algoritmi numerici pentru determinarea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru diverse scheme și funcții de servire, pentru disciplina de prioritate DD;
- s-au prezentat exemple numerice pentru calcularea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru prioritatea DD.

CONCLUZII GENERALE ȘI RECOMANDĂRI

Concluzii generale asupra rezultatelor obținute. Problema analizată în teza de doctor *”Modele Polling cu priorități, vacanțe semi-Markoviene și servire exhaustivă”* ține de direcția de cercetare din teoria așteptării și constă în studiul modelelor matematice, modele care au un rol important în analiza, modelarea și eficientizarea diferitor procese reale, în care apar fenomene de așteptare, cum ar fi centrele de apel public, rețelele contemporane, procesele de producție, etc. Rezultatele teoretice obținute cu privire la algoritmi de modelare a lungimii virtuale a șirului de așteptare pentru sistemele Polling cu întârzieri semi-Markoviene și algoritmi de calcul a perioadei de ocupare și a caracteristicilor auxiliare pentru modelele generalizate de așteptare cu priorități, care pot fi privite ca o clasă specială a modelelor Polling, conduc la următoarele concluzii:

1. s-au studiat unele modele de așteptare clasice pentru care s-au descris domeniile de aplicare și s-au prezentat rezultate analitice [3], [8], [9].
2. s-au elaborat algoritmi și modelări numerice pentru determinarea repartiției lungimii virtuale a șirului de așteptare pentru sistemele Polling cu întârzieri semi-Markoviene pentru diverse funcții de repartiție [60], [61], [66].
3. au fost formalizate și studiate rezultatele analitice pentru perioada de ocupare și caracteristici auxiliare ale sistemelor generalizate de așteptare cu disciplina de prioritate DD (Discretionary Discipline) [6], [40], [63].
4. au fost elaborați algoritmi de calcul pentru determinarea repartiției perioadei de ocupare și a caracteristicilor auxiliare pentru sistemele generalizate de așteptare cu priorități [63].
5. au fost efectuate modelări numerice pentru diverse scheme și funcții de servire pentru determinarea caracteristicilor de performanță ale sistemelor de așteptare cu prioritatea DD și s-a stabilit viabilitatea sistemului în dependență de parametrii de intrare [63], [64].
6. s-au realizat produse program pentru algoritmi elaborați în limbajele de programare C++ și Kotlin [60], [66].

Rezultatele prezentate în teză pot servi ca suport pentru continuarea cercetărilor în această direcție și au ca scop de a prezenta algoritmi numerici pentru calculul unor caracteristici de performanță probabiliste pentru diverse legi de repartiție a timpului de servire și orientare pentru modelele studiate.

Problema științifică importantă soluționată rezidă în determinarea unor valori mai

optime ale caracteristicilor probabiliste pentru modelele Polling, rezultatele obținute atât în urma analizei modelelor de așteptare, cât și a funcțiilor de repartiție, legilor de prioritate, schemelor de servire și orientare, fapt care permite stabilirea staționarității și eficienței sistemului de așteptare.

Rezultatele obținute, din punct de vedere aplicativ, ne dau posibilitatea de a determina parametrii și indicatorii pentru estimarea calității servirii, spre exemplu, rețelele contemporane sunt deja înzestrate cu tehnologiile de rețea de tip QoS (quality of service). Aspectul aplicativ al rezultatelor prezentate este deosebit de important dat fiind că QoS și CoS (class of service) prezintă un aspect cheie în viitoarele tehnologii de rețea.

Recomandări. În calitate de sugestii ale autorului privind cercetările de perspectivă vom considera cele ce urmează:

- rezultatele obținute în teză pot fi considerate ca suport pentru continuarea cercetărilor științifice la analiza diverselor tipuri de modele de așteptare;
- aplicarea metodologiilor prezentate pentru determinarea altor caracteristici probabiliste de performanță ale sistemelor de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă, cât și pentru sistemele generalizate de așteptare cu disciplina de prioritate DD;
- depistarea unor domenii potențiale de aplicare, cum ar fi: în ingineria și tehnologia rețelelor, în medicină, în sistemele de producție, etc. care ar permite modelarea și soluționarea problemelor, în care apar fenomene de așteptare similare modelelor matematice descrise pentru sistemele de așteptare Polling cu vacanțe semi-Markoviene și servire exhaustivă, cât și pentru sistemele generalizate de așteptare cu legea de prioritatea DD.

BIBLIOGRAFIE

1. Bejan A. Modelarea timpului de orientare în sisteme de așteptare cu priorități. Teză de doctor în științe fizico-matematice. Chișinău, 2007. 117 p.
2. Bejenari D. Analiza modelelor de tip Polling cu întârzieri semi-Markoviene. Teză de doctor în științe fizico-matematice. Chișinău, 2012. 108 p.
3. Bejenari D., Mitev L. Formule aproximative pentru sisteme Polling cu timp discret. În: Revista Științifică Studii Economice, 2012, vol. VI, nr. 3-4, p. 326-331.
4. Benderschi O. Analiza sistemelor de așteptare cu priorități și trafic critic. Teză de doctor în științe fizico-matematice. Chișinău, 2009. 97 p.
5. Mișcoi Gh., Costea A. Metode bazate pe aparatul transformatelor Laplace și Laplace-Stieltjes. În: Materialele Conferinței Științifice Internaționale “Modelarea matematică, optimizare și tehnologii informaționale”. Chișinău: ATIC, 2012, p. 106-114.
6. Mișcoi Gh., Bejenari D., Usafii L. Modele semimarkoviene de servire cu priorități. În: Analele Universității Libere Internaționale din Moldova, Seria Economie. Chișinău: ULIM, 2011, vol. 11, p. 95-105.
7. Mișcoi Gh., Țicu I. R. Metoda de colorare și aplicarea ei în cercetarea modelelor fenomenelor de așteptare. În: Materialele Conferinței Științifice Internaționale “Modelarea matematică, optimizare și tehnologii informaționale”. Chișinău: ATIC, 2012, p. 99-105.
8. Mișcoi Gh., Mitev L. Metode analitice și numerice în analiza modelelor Polling. În: Materialele Conferinței Științifico-Practice Internaționale „Politici economice și financiare pentru o dezvoltare competitivă”. Chișinău: ULIM, 2013, p. 353-357.
9. Mitev L. Aplicații ale modelelor polling în sistemele informatice de comunicații. În: Revista Științifică Studii Economice, 2015, an. 9, nr. 1, p. 200-208.
10. Mitev L. Modele și metode în studierea sistemelor polling. În: Studia Universitatis, Seria științe exacte și economice, USM, 2014, nr. 2 (72), p. 34-38.
11. Abidini M.A., Boxma O.J., Resing J.A. Analysis and optimization of vacation and polling models with retrials. In: Performance Evaluation, 2016, vol. 98, p. 52-69.
12. Altman E., Kofman D. Bounds for performance measures of token rings. In: IEEE/ACM Transactions on Networking, 1996, vol. 4 (2), p. 292-299.
13. Attahiru S. Alfa. Queueing Theory for Telecommunications. London: Springer, 2010. 238p.
14. Atthahiru S. Alfa. Applied Discrete-Time Queues. New York: Springer, 2015. 383p.

15. Beekhuizen P. Performance Analysis of Networks on Chips. PhD thesis. Eindhoven University of Technology, 2010. 153 p.
16. Benderschi O. Heavy traffic analysis in queueing systems. În: Studia Universitatis, Seria Științe Exacte și Economice. Chișinău: USM, 2008, nr. 8(18), p. 38-48.
17. Benini L., De Micheli G. Networks on chips: a new SoC paradigm. In: Computer, 2002, vol. 35(1), p. 70-78.
18. Bisdikian C. A queueing model for a data station within the IEEE 802.6 MAN. In: Proceedings 17th Conference on Local Computer Networks, 1992, p. 52-61.
19. Bogunovic N. Process scheduling procedure for a class of real-time computer system. In: IEEE Trans. Ind. Electron, 1987, vol. 34, nr. 1, p. 29-34.
20. Bonald T., Proutiere A. Wireless downlink data channels: user performance and cell dimensioning. In: Proceedings Mobicom '03. USA: San Diego, 2003.
21. Boon M.A.A., Adan I.J., Boxma O.J. A two-queue polling model with two priority levels in the first queue. In: Discrete Event Dynamic Systems, 2010, vol. 20, nr. 4, p. 511-536.
22. Borst S. C. User-level performance of channel-aware scheduling algorithms in wireless data networks. In: Proceedings Infocom '03. USA: San Francisco, 2003.
23. Borst S.C. Polling systems. Amsterdam: Stichting Mathematisch Centrum, 1996. 304 p.
24. Boxma O. J., Weststrate J. A. Waiting times in polling systems with Markovian server routing. In: Messung, Modellierung und Bewertung von Rechen-systemen und Netzen. Berlin: Springer Verlag, 1989, p. 89-105.
25. Boxma O., van der Wal J., Yechiali U. Polling with batch service. In: Stochastic Models, 2008, vol. 24, nr. 4, p. 604-625.
26. Boxma O.J., Groenendijk W.P., Weststrate J.A. A Pseudoconservation Law for Service Systems with a Polling Table. In: IEEE Trans. Commun., 1990, vol. 38, nr. 10, p. 1865-1870.
27. Boxma O.J., Bruin J., Fralix B.H. Sojourn times in polling systems with various service disciplines. In: Performance Evaluation, 2009, vol. 66, nr. 11, p. 621-639.
28. Bux W. Local-area subnetworks: a performance comparison. In: IEEE Transactions on Communications, 1981, vol. 29 (10), p. 1465-1473.
29. Czerniak O., Altman E., Yechiali U. Analysis of a TCP system under polling-type reduction signal procedures. In: Proceedings of Value Tools, Pisa, Italy, 2009.
30. Czerniak O., Yechiali U. Fluid polling systems. In: Queueing Systems, 2009, vol. 63, p. 401-435.

31. Daduna H. Queueing networks with discrete time scale: Explicit expressions for the steady state behaviour of discrete time stochastic networks. In: Lecture Notes in Computer Science. Berlin: Springer, 2001, vol. 2046.
32. Dally W. J., Towles B. Route packets, not wires: on-chip interconnection networks. In: Proceedings of the Design Automation Conference, 2001, p. 684-689.
33. de Haan R. Queueing Models for Mobile Adhoc Networks. PhD thesis. University of Twente, 2009.
34. De Haan R., Boucherie R. J., van Ommeren J.-K. A polling model with an autonomous server. In: Queueing Systems, 2009, vol. 62 (3), p. 279-308.
35. Dou C., Chang J.F. Serving Two Correlated Queues with a Synchronous Server under Exhaustive Service Discipline and Nonzero Switchover Time. In: IEEE Trans. Commun., 1991, vol. 39, nr. 11, p. 1582-1589.
36. Dragalin V. P., Mishkoy Gh. K. The service with mixed priority and switchings. In: Bulletin of the URSS Academy of Sciences, 1984, nr. 3, p. 166-172 (in Russian).
37. Gaver D. P. Competitive queueing: idleness probabilities under priority disciplines. In: J. Roy. Stat. Soc. B, 1963, vol. 25(2), p. 489-499.
38. Grillo D. Polling mechanism models in communication systems - some application examples. In: Stochastic Analysis of Computer and Communication Systems, 1990, Amsterdam, North-Holland, p. 659-699.
39. Griza Iu., Korolyuk V., Mamonova A., Mishkoy Gh. Queueing Systems with semi-Markov Flow in the Series Scheme. In: Preprint. Bielefeld University, Germany, 2008, 28p.
40. Groza O., Mishkoy Gh., Mitev L., Costea A. Method of "catastrophes" and its application to analyze generalized queueing models. In: Studia Universitatis, Seria Științe Exacte și Economice. Chișinău: USM, 2012, nr. 2 (52), p. 5-12.
41. Hariharan R., Ehrlich W. K., Reeser P. K., van der Mei R. D. Performance of web servers in a distributed computing environment. In: Teletraffic Engineering in the Internet Era, 2001, p. 137-148.
42. Jaiswal N. K. Priority queues. Moscow: Mir. Russia, 1973 (in Russian).
43. Kavitha V., Altman E. Queueing in space: design of message ferry routes in static adhoc networks. In: Proceedings ITC21, 2009.
44. Kendall D.G. Some problems in the theory of queues. In: J. Roy. Statist. Soc, 1953, vol. 13(2B), p. 151-180.

45. Kleinrock L. Performance evaluation of distributed computer-communication systems. In: Queueing Theory and its Applications - Liber Amicorum for J.W. Cohen, 1988, Amsterdam, North-Holland, p. 1-57.
46. Kleinrock L., Levy H. The analysis of random polling systems. In: Journal of Operations Research, 1988, vol. 36(5), p. 716-732.
47. Klimov G. P. Time-sharing service systems. In: Theory of Probability and its Applications, 1974, vol. 19, p. 532-551.
48. Kramer G., Mukherjee B., Dixit S., Yinghua Y., Hirth R. Supporting differentiated classes of services in ethernet passive optical networks. In: Journal of Optical Networking, 2002, vol. 1(9), p. 280-290.
49. Kramer G., Mukherjee B., Pesavento G. Ethernet PON (ePON): design and analysis of an optical access network. In: Photonic Network Communications, 2001, vol. 3, p. 307-319.
50. Kramer G., Mukherjee B., Pesavento G. Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth allocation scheme in an optical access network. In: Photonic Network Communications, 2002, vol. 4, p. 89-107.
51. Levy H., Kleinrock L., Polling Systems with Zero Switch-over Periods: A General Method for Analysis the Expected Delay. In: Performance Evaluat., 1991, vol. 13, nr. 2, p. 97-107.
52. Levy H., Sidi M. Polling systems: applications, modeling, and optimization. In: IEEE Transactions on Communications, 1990, vol. 38, p. 1750-1760.
53. Levy H., Sidi M., Boxma O.J. Dominance Relations in Polling Systems. In: Queueing Systems, 1990, vol. 6, nr. 2, p. 155-171.
54. Macphée I., Menshikov M., Petritis D., Popov S. A Markov chain model of a polling system with parameter regeneration. In: Annals of Applied Probability, 2007, vol. 17, nr. 5/6, p. 1447-1473.
55. Manfield D. R. Analysis of a priority polling system for two-way traffic. In: IEEE Transactions on Communications, 1985, vol. 33, p. 1001-1006.
56. Miorandi D., Zanella A., Pierobon G. Performance evaluation of bluetooth polling schemes: An analytical approach.. In: Mobile Networks and Applications, 2004, vol. 9, p. 63-72.
57. Mishkoy Gh. K. On mixed priority service problem in systems with switching losses. In: Automatics and Remote Control, 1978, nr. 2, p. 15-21 (in Russian).

58. Mishkoy Gh., Bejenari D. Numerical solutions for Polling systems with semi-Markov switching. MDA-Workshop on Modeling and Analysis of Traffic Processing in Future Generation of Internet, Germania, Bamberg, 2009. <http://www.ktr.uni-bamberg.de>.
59. Mishkoy Gh., Bejenari D., Klimenok V. Some generalizations of Kendall and Pollaczek-Khintchin equations for Polling systems. In: Abstracts of the 19th edition of the Annual Conference on Applied and Industrial Mathematics. România, Iași, 2011, p. 72.
60. Mishkoy Gh., Bejenari D., Mitev L. Numerical algorithms regarding Polling systems with exhaustive service. In: Abstracts of the 19th edition of the Annual Conference on Applied and Industrial Mathematics. România, Iași, 2011, p. 72-73.
61. Mishkoy Gh., Bejenari D., Mitev L., Ticu I.R. Numerical solutions of Kendall and Pollaczek-Khintchin equations for exhaustive polling systems with semi-Markov delays. In: Computer Science Journal of Moldova. Chișinău: IMI, 2016, vol. 24, nr. 2(71), p. 255-272.
62. Mishkoy Gh., Krieger Udo R., Bejenari D. Matrix algorithm for Polling models with PH distribution. În: Buletinul Academiei de Științe a Republicii Moldova, Matematica. Chișinău: IMI, 2012, nr. 1, p. 70-80.
63. Mishkoy Gh., Mitev L. Performance characteristics for DD priority discipline with semi-Markov switching. In: Communications in Computer and Information Science (CCIS) Series, Springer International Publishing, 2014, p. 204-218.
64. Mishkoy Gh., Mitev L. Some questions of numerical modeling of priority discipline DD with semi-Markov switching. In: Proceedings of the 17-th International Conference on "Distributed Computer and Communication Networks (DCCN-2013): "Control, Computation, Communications. Moscow, Russia, 2013, p. 373-378.
65. Mishkoy Gh., Rykov V. V., Giordano S., Bejan A. Multidimensional Analogs of the Kendall Equation for Priority Queueing Systems: Computation Aspects. In: Automatics and Remote Control, 2008, vol. 69(6), p. 980-992.
66. Mișcoi Gh., Bejenari D., Mitev L. Numerical algorithm regarding symmetric discrete polling system. In: Proceedings of the 37th Annual Congress of the American Romanian Academy of Arts and Sciences (ARA), 2013, p. 490-492.
67. Rycov V. V., Mishkoy Gh. The new approach on study Polling systems. In: Proceedings of the 4-th International Conferences on Control Problems. Moscow, 2009, p. 1749-1758.
68. Saffer Z., Telek M. Stability of periodic polling system with BMAP arrivals. In: European Journal of Operational Research, 2009, vol. 197, nr. 1, p. 188-195.

69. Sevcik K. C., Johnson M. J. Cycle time properties of the FDDI token ring protocol. In: IEEE Transactions on Software Engineering, 1987, vol. 13, p. 376-385.
70. Shomrony M., Yechiali U. Polling systems with positive and negative customers. In: Technical Report, Department of Statistics and Operations Research. Israel, Tel-Aviv: Tel-Aviv University, 2006.
71. Shomrony M., Yechiali U. Polling systems with job failures and with station failures. In: Technical Report, Department of Statistics and Operations Research. Israel, Tel-Aviv: Tel-Aviv University, 2006.
72. Takagi H. Analysis and application of polling models. In: Performance Evaluation: Origins and Directions of Lecture Notes in Computer Science. Berlin: Springer, 2000, vol. 1769, p. 424-442.
73. Takagi H. Analysis of Polling Systems. Cambridge: MIT Press, 1986. 197p.
74. Van Arem B. Queueing Network Models for Slotted Transmission Systems. PhD thesis. University of Twente, The Netherlands, 1990.
75. Van den Berg J. L., Litjens R., Laverman J. HSDPA flow level performance: the impact of key system and traffic aspects. In: Proceedings MSWiM '04. Italy, Venice, 2004.
76. Van der Mei R. D., Hariharan R., Reeser P. K. Web server performance modeling. In: Telecommunication Systems, 2001, vol. 16, p. 361-378.
77. Van der Mei R.D. Towards a unifying theory on branching-type polling systems in heavy traffic. In: Queueing Systems, 2007, vol. 57, nr. 1, p. 29-46.
78. Van der Mei R.D., Winands E. Heavy traffic analysis of polling models by mean value analysis. In: Performance Evaluation, 2008, vol. 65, nr. 6-7, p. 400-416.
79. Van Houdt B. Numerical solution of polling systems for analyzing networks on chips. In: Proceedings of NSMC 2010. USA, Williamsburg, 2010, p. 90-93.
80. Van Vuuren M., Winands E.M.M. Iterative approximation of k -limited polling systems. In: Queueing Systems, 2007, vol. 55, nr. 3, p. 161-178.
81. Vishnevskii V. M., Dudin A. N., Klimenok V. I., Semenova O. V. Approximate Method to Study M/G/1-Type Polling System with Adaptive Polling Mechanism. In: Qual. Technol. Quant. Manag., 2012, vol. 9, nr. 2, p. 211-228.
82. Vishnevskii V. M., Semenova O. V. Mathematical Methods to Study the Polling Systems. In: Proceedings of Automation and Remote Control, 2006, vol. 67, nr. 2, p. 179-220.
83. Vishnevsky V.M., Semenova O.V. Adaptive dynamical polling in wireless networks. In: Cybernetics and Information Technologies, 2008, vol. 8, nr. 1, p. 3-11.

84. Vlasiou M., Adan I.J.B.F., Boxma O.J. A two-station queue with dependent preparation and service times. In: European Journal of Operational Research, 2009, vol. 195, nr. 1, p. 104-116.
85. Vlasiou M., Yechiali U. $M/G/\infty$ polling systems with random visit times. In: Probability in the Engineering and Informational Sciences, 2008, vol. 22, nr. 1, p. 212-245.
86. Volkovinski M. I., Kabalevsky A. N. The service with mixed priorities in the systems with switching losses. In: Automatics and Remote Control, 1975, nr. 11, p. 16-22 (in Russian).
87. Weststrate J. A. Analysis and Optimization of Polling Systems. PhD thesis. Tilburg University, 1992.
88. Wierman A., Winands E., Boxma O.J. Scheduling in polling systems. In: Performance Evaluation, 2007, vol. 64, nr. 9-12, p. 1009-1028.
89. Winands E.M.M., Adan I.J.B.F., van Houtum G.J. Meanvalue analysis for polling systems. In: QueueingSystems, 2006, vol. 54, p. 35-44.
90. Zussman G., Segall A., Yechiali U. On the analysis of the bluetooth time division duplex mechanism. In: IEEE Transactions on Wireless Communications, 2007, vol. 6(6), p. 2149-2161.
91. Вишневский В.М., Лаконцев Д.В., Семенова О.В., Шпилев С.А. Модель системы поллинга для исследования широкополосных беспроводных сетей. В: Автоматика и телемеханика, 2006, н. 12, с. 123-135.
92. Вишневский В.М., Семенова О.В. Математические методы исследования систем поллинга. В: Автоматика и телемеханика, 2006, н. 2, с. 3-56.
93. Вишневский В.М., Семенова О.В. Системы поллинга: теория и применение в широкополосных беспроводных сетях. Москва: Техносфера, 2007. 312 р.
94. Вишневский В.М., Семенова О.В., Шпилев С.А. Дуплексная система циклического обслуживания смешанных очередей. В: Автоматика и телемеханика, 2009, н. 12, с. 121-133.
95. Волковинский М., Кабалеvский А. Анализ приоритетных очередей с учетом времени переключения. Москва: Энергоиздат, 1981. 184с.
96. Гнеденко Б.В. и др. Приоритетные системы обслуживания. Москва: Изд-во Московского Университета, 1973. 447с.
97. Гнеденко Б.В., Коваленко И.В. Введение в теорию массового обслуживания. Москва: Наука, 2005. 400с.

98. Климов Г. П. Стохастические системы обслуживания. Второе издание, Москва, 2011. 242с.
99. Климов Г.П., Мишкой Г.К. Приоритетные системы обслуживания с ориентацией. Москва: Изд-во Московского Университета, 1979. 128с.
100. Мишкой Г.К. Обобщенные приоритетные системы. Кишинев: Из-во. А. Н. Молдовы, 2009. 200с.
101. Мишкой Г.К., Рыков В.В., Джиордано С., Бежан А.Ю. Многомерные аналоги уравнения Кендалла для приоритетных систем: Вычислительные аспекты. В: Автоматика и телемеханика, Москва, 2008, н. 6, с. 82-95.
102. Рыков В.В. К анализу поллинг-систем. В: Автоматика и телемеханика, 2008, н. 6. с. 90-114.
103. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. Москва: Физматлит, 2005. 320с.

ANEXE

ANEXA 1. Listingul programelor pentru determinarea repartiției lungimii șirului de așteptare pentru modelul Polling cu servire exhaustivă

Programul A1.1

```
#include<iomanip.h>
#include<conio.h>
#include<math.h>
#define eps 0.00001

double beta(int k, double s);
double ce(int k, double s);
double pi(int k, double s);
double piDelta(int k, double s);
double *b, *c, *lambda;
int n;

double beta(int k, double s)
{
    return b[k]/(s+b[k]);
}

double ce(int k, double s)
{
    return c[k]/(s+c[k]);
}

double pi(int k, double s)
{
    double result=0, resultprec;
    do
    {
        resultprec=result;
        result=beta(k,s+lambda[k]-lambda[k]*resultprec);
    }
    while(fabs(result-resultprec)>=eps);
    return result;
}

double piDelta(int k, double s)
{
    return ce(k,s+lambda[k]-lambda[k]*pi(k,s))*pi(k,s);
}

double lks(int k, double s)
{
    return (lambda[k]/(s+lambda[k]-lambda[k]*piDelta(k,s)))*(((1-
```

```

c[k])*lambda[k]*s+lambda[k]*(1-ce(k,s)))/s*s)+((1-beta(k,s)*(s-
lambda[k])+lambda[k]*s*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*s*(1-beta(k,s)))-
((1+lambda[k]*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*(1-beta(k,s))*(1-beta(k,s))));
}

```

```

void citire(double *x, int n)
{
    for(int i=0;i<n;i++) cin>>x[i];
}

```

```

void afisare(double *x, int n)
{
    cout<<"{"<<endl;
    for(int i=0;i<n;i++) cout<<setprecision(2)<<x[i]<<" ";
    cout<<"\b\b}"<<endl;
}

```

```

void algoritm()
{
    double *Pi;
    double *Pidoub;
    double *Pilks;
    double s,r;
    cout<<"Dati n=";
    cin>>n;
    cout<<"Dati coeficientii lambda:"<<endl;
    lambda=new double[n];
    citire(lambda,n);

```

```

    cout<<"Dati coeficientii b:"<<endl;
    b=new double[n];
    citire(b,n);

```

```

    cout<<"Dati coeficientii c:"<<endl;
    c=new double[n];
    citire(c,n);

```

```

    cout<<"Dati valoarea s="<<endl;
    cin>>s;
    clrscr();

```

```

    cout<<"coeficientii lambda:"<<endl;
    afisare(lambda,n);
    cout<<"coeficientii b:"<<endl;
    afisare(b,n);
    cout<<"coeficientii c:"<<endl;
    afisare(c,n);

```

```

    cout<<"rezultatele:"<<endl;
    Pi=new double[n];

```

```

for(int k=0;k<n;k++) Pi[k]=pi(k,s);
Pidoub=new double[n];
for(k=0;k<n;k++) Pidoub[k]=piDelta(k,s);
Pilks=new double[n];
for(k=0;k<n;k++) Pilks[k]=lks(k,s);
cout<<" k   pi_k(s)  pi^delta(s)  l_k(s)"
      " k   pi_k(s)  pi^delta(s)  l_k(s)"<<endl;
for(k=0;k<n/2;k++)
{
    r=k;
    cout<<setw(3)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
        setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r];
    r=k+n/2;
    cout<<setw(5)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
        setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r]<<endl;
    if(wherex()>=78)
    {
        cout<<"ENTER pentru continuare!";
        getch();
        clrscr();
    }
}
if(n%2!=0)
    cout<<setw(10)<<n<<setw(14)<<setprecision(6)<<Pi[n-1]<<
        setw(14)<<setprecision(6)<<Pidoub[n-1]<<setw(14)<<setprecision(6)<<Pilks[n-
1]<<endl;
delete [] b;
delete [] c;
delete [] lambda;
delete [] Pi;
delete [] Pidoub;
delete [] Pilks;
}

```

Programul A1.2

```

#include<iomanip.h>
#include<conio.h>
#include<math.h>
#define eps 0.00001

double beta(int k, double s);
double ce(int k, double s);
double pi(int k, double s);
double piDelta(int k, double s);
double *b, *c, *lambda, *p, *pn;
int n;

double beta(int k, double s)
{
    return b[k]/(s+b[k]);
}

```

```

}

double ce(int k, double s)
{
    return pow(c[k]/(s+c[k]), p[k]);
}

double pi(int k, double s)
{
    double result=0, resultprec;
    do
    {
        resultprec=result;
        result=beta(k,s+lambda[k]-lambda[k]*resultprec);
    }
    while(fabs(result-resultprec)>=eps);
    return result;
}

double piDelta(int k, double s)
{
    return ce(k,s+lambda[k]-lambda[k]*pi(k,s))*pi(k,s);
}

double lks(int k, double s)
{
    return (lambda[k]/(s+lambda[k]-lambda[k]*piDelta(k,s)))*(((1-
c[k])*lambda[k]*s+lambda[k]*(1-ce(k,s)))/s*s)+((1-beta(k,s)*(s-
lambda[k])+lambda[k]*s*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*s*(1-beta(k,s)))-
((1+lambda[k]*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*(1-beta(k,s))*(1-beta(k,s))));
}

void citire(double *x, int n)
{
    for(int i=0;i<n;i++) cin>>x[i];
}

void afisare(double *x, int n)
{
    cout<<"{"<<" ";
    for(int i=0;i<n;i++) cout<<setprecision(2)<<x[i]<<" ";
    cout<<"\b\b}"<<endl;
}

void algoritm()
{
    double *Pi;
    double *Pidoub;
    double *Pilks;
    double s,r;

```



```

cout<<"Dati n=";
cin>>n;
cout<<"Dati coeficientii lambda:"<<endl;
lambda=new double[n];
citire(lambda,n);

cout<<"Dati coeficientii b:"<<endl;
b=new double[n];
citire(b,n);

cout<<"Dati coeficientii p (intregi nenegative):"<<endl;
p=new double[n];
citire(p,n);
cout<<"Dati coeficientii pn:"<<endl;
pn=new double[n];
citire(pn,n);

cout<<"Dati coeficientii c:"<<endl;
c=new double[n];
citire(c,n);

cout<<"Dati valoarea s="<<endl;
cin>>s;
clrscr();

cout<<"coeficientii lambda:"<<endl;
afisare(lambda,n);
cout<<"coeficientii b:"<<endl;
afisare(b,n);
cout<<"coeficientii c:"<<endl;
afisare(c,n);
cout<<"coeficientii p:"<<endl;
afisare(p,n);
cout<<"coeficientii pn:"<<endl;
afisare(pn,n);

cout<<"rezultatele:"<<endl;
Pi=new double[n];
for(int k=0;k<n;k++) Pi[k]=pi(k,s);
Pidoub=new double[n];
for(k=0;k<n;k++) Pidoub[k]=piDelta(k,s);
Pilks=new double[n];
for(k=0;k<n;k++) Pilks[k]=lks(k,s);
cout<<" k pi_k(s) pi^delta(s) l_k(s)"
" k pi_k(s) pi^delta(s) l_k(s)"<<endl;
for(k=0;k<n/2;k++)
{
r=k;
cout<<setw(3)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r];

```

```

    r=k+n/2;
    cout<<setw(5)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
        setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r]<<endl;
    if(wheray()>=78)
    {
        cout<<"ENTER pentru continuare!";
        getch();
        clrscr();
    }
}
if(n%2!=0)
    cout<<setw(10)<<n<<setw(14)<<setprecision(6)<<Pi[n-1]<<
        setw(14)<<setprecision(6)<<Pidoub[n-1]<<setw(14)<<setprecision(6)<<Pilks[n-
1]<<endl;
    delete [] b;
    delete [] c;
    delete [] lambda;
    delete [] Pi;
    delete [] Pidoub;
    delete [] Pilks;
}

void main()
{
    clrscr();
    algoritm();
    getch();
}

```

Programul A1.3

```

#include<iomanip.h>
#include<conio.h>
#include<math.h>
#define eps 0.00001

double beta(int k, double s);
double ce(int k, double s);
double pi(int k, double s);
double piDelta(int k, double s);
double *b, *c, *lambda, *p, *pn;
int n;

double beta(int k, double s)
{
    return pow(c[k]/(s+c[k]), p[k]);
}

double ce(int k, double s)
{

```

```

    return exp(pow(pn[k],2)*s/2-s*c[k]);
}

double pi(int k, double s)
{
    double result=0, resultprec;
    do
    {
        resultprec=result;
        result=beta(k,s+lambda[k]-lambda[k]*resultprec);
    }
    while(fabs(result-resultprec)>=eps);
    return result;
}

double piDelta(int k, double s)
{
    return ce(k,s+lambda[k]-lambda[k]*pi(k,s))*pi(k,s);
}

double lks(int k, double s)
{
    return (lambda[k]/(s+lambda[k]-lambda[k]*piDelta(k,s)))*(((1-
c[k])*lambda[k]*s+lambda[k]*(1-ce(k,s)))/s*s)+((1-beta(k,s)*(s-
lambda[k])+lambda[k]*s*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*s*(1-beta(k,s))-
((1+lambda[k]*(1/b[k]))*(ce(k,s)-piDelta(k,s)))/(s*(1-beta(k,s))*(1-beta(k,s)))));
}

void citire(double *x, int n)
{
    for(int i=0;i<n;i++) cin>>x[i];
}

void afisare(double *x, int n)
{
    cout<<"{";
    for(int i=0;i<n;i++) cout<<setprecision(2)<<x[i]<<" ";
    cout<<"\b\b}"<<endl;
}

void algoritm()
{
    double *Pi;
    double *Pidoub;
    double *Pilks;
    double s,r;
    cout<<"Dati n=";
    cin>>n;
    cout<<"Dati coeficientii lambda:"<<endl;
    lambda=new double[n];

```

```

citire(lambda,n);

cout<<"Dati coeficientii p:"<<endl;
p=new double[n];
citire(p,n);
cout<<"Dati coeficientii pn:"<<endl;
pn=new double[n];
citire(pn,n);

cout<<"Dati coeficientii b:"<<endl;
b=new double[n];
citire(b,n);

cout<<"Dati coeficientii c:"<<endl;
c=new double[n];
citire(c,n);

cout<<"Dati valoarea s="<<endl;
cin>>s;
clrscr();

cout<<"coeficientii lambda:"<<endl;
afisare(lambda,n);
cout<<"coeficientii b:"<<endl;
afisare(b,n);
cout<<"coeficientii c:"<<endl;
afisare(c,n);
cout<<"coeficientii p:"<<endl;
afisare(p,n);
cout<<"coeficientii pn:"<<endl;
afisare(pn,n);

cout<<"rezultatele:"<<endl;
Pi=new double[n];
for(int k=0;k<n;k++) Pi[k]=pi(k,s);
Pidoub=new double[n];
for(k=0;k<n;k++) Pidoub[k]=piDelta(k,s);
Pilks=new double[n];
for(k=0;k<n;k++) Pilks[k]=lks(k,s);
cout<<" k   pi_k(s)   pi^delta(s)   l_k(s)"
    " k   pi_k(s)   pi^delta(s)   l_k(s)"<<endl;
for(k=0;k<n/2;k++)
{
    r=k;
    cout<<setw(3)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
        setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r];
    r=k+n/2;
    cout<<setw(5)<<(r+1)<<setw(12)<<setprecision(6)<<Pi[r]<<
        setw(12)<<setprecision(6)<<Pidoub[r]<<setw(12)<<setprecision(6)<<Pilks[r]<<endl;
    if(wherex()>=78)

```

```

    {
        cout<<"ENTER pentru continuare!";
        getch();
        clrscr();
    }
}
if(n%2!=0)
    cout<<setw(10)<<n<<setw(14)<<setprecision(6)<<Pi[n-1]<<
        setw(14)<<setprecision(6)<<Pidoub[n-1]<<setw(14)<<setprecision(6)<<Pilks[n-
1]<<endl;
    delete [] b;
    delete [] c;
    delete [] lambda;
    delete [] Pi;
    delete [] Pidoub;
    delete [] Pilks;
}

void main()
{
    clrscr();
    algoritm();
    getch();
}

```

ANEXA 2. Listingul programelor pentru determinarea perioadei de ocupare și a caracteristicilor auxiliare pentru sistemul generalizat cu prioritatea DD

Programul A2.1

```
package algorithm
import java.util.*

class Var1 {
    internal var nIterations: Int = 0
    internal var rRange: Int = 0
    internal var s: Double = 0.toDouble()
    internal var teta_k: Double = 0.toDouble()
    internal var eps = 0.00001

    internal var a: DoubleArray = DoubleArray(0)
    internal var b: DoubleArray = DoubleArray(0)
    internal var c: DoubleArray = DoubleArray(0)
    internal var pk: DoubleArray = DoubleArray(0)

    internal var sigma: DoubleArray = DoubleArray(0)
    internal var pi_: DoubleArray = DoubleArray(0)

    internal var reader = Scanner(System.`in`)
    //Locale.setDefault(Locale.US)
    var algType: Int = -1

    internal fun PI_K_1(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * PI_K_1(k - 1, s + a[k]) + sigmaDivide(k) *
            (PI_K_1(k - 1, s + a[k] * (1 - Pi_KK_1(k, s))) - PI_K_1(k - 1, s + a[k])) *
            V_K_1(k, s + a[k] * (1 - Pi_KK_1(k, s))) +
            a[k] / sigma[k] * Pi_KK_1(k, s)
        return rez
    }

    internal fun Pi_K_2(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * Pi_K_2(k - 1, s + a[k] * (1 - Pi_KK_2(k, s))) + (a[k] / sigma[k])
        * Pi_KK_2(k, s)
        return rez
    }

    private fun sigmaDivide(k: Int) = sigma[k - 1] / sigma[k]

    internal fun Pi_KK_1(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = V_K_1(k, s + a[k] * (1 - Pi_KK_1(k, s))) * Pi_KK_1(k, s)
        return rez
    }
}
```

```

    internal fun Pi_KK_1_(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_1(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
    }

    internal fun Pi_KK_2(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_2(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
    }

    internal fun H_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = B_pass_K(k, s + sigma[k - 1] * (1 - PI_K_1(k - 1, s) * V_K_1(k, s)))
    return rez
    }

    internal fun H_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = B_pass_K(k, s + sigma[k - 1] * Math.pow(1 - (sigma[k - 1] / (s + sigma[k - 1]))) *
    (1 - B_pass_K(k, s + sigma[k - 1])) * Pi_K_2(k - 1, s), (-1).toDouble()))
    return rez
    }

    internal fun SetSigma() {
    sigma = DoubleArray(rRange + 1)
    sigma[0] = 0.0
    for (k in 1..nIterations - 1) {
        sigma[k] = sigma[k - 1] + a[k]
    }
    }

    internal fun B_pass_K(k: Int, s: Double): Double {
    var rez = b[k] / (s + b[k])
    return rez
    }

    internal fun C_pas_K_(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 1.0

```

```

    val rez = c[k] / (s + c[k])
    return rez
}

    private fun returnZeroIfLessThan(k: Int): Boolean {
    if (k <= 0) {
        return true
    }
    return false
}

internal fun V_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = C_pas_K_(k, s + sigma[k - 1] * (1 - PI_K_1(k - 1, s)))
    return rez
}

internal fun citire(x: DoubleArray, n: Int) {
    for (i in 0..n - 1)
        x[i] = reader.nextDouble().toDouble()
}

internal fun InputInfo() {
    print("Introduceti nIterations valoare : ")
    nIterations = reader.nextDouble().toInt()
    rRange = nIterations
    print("Introduceti s valoare : ")

    try {
        s = reader.nextDouble().toDouble()
    } catch (e: InputMismatchException) {
        println(e.message.toString())
    }

    print("Introduceti teta_k valoare : ")
    teta_k = reader.nextDouble().toDouble()
    print("Introduceti coeficientii a : ")
    a = DoubleArray(nIterations)
    citire(a, nIterations)
    print("Introduceti coeficientii b : ")
    b = DoubleArray(nIterations)
    citire(b, nIterations)
    print("Introduceti coeficientii c : ")
    c = DoubleArray(nIterations)
    citire(c, nIterations)
}

internal fun Initialize() {
    InputInfo()
}

```



```

for (i in 0..nIterations ) {
    SetSigma()
    if (b[i] < teta_k) {
        println("H de  $\{i + 1\}$  :  $\{H\_K\_1(i + 1, s).toString()\}$ ")
        println("Niu de  $\{i + 1\}$  :  $\{V\_K\_1(i + 1, s).toString()\}$ ")
        println(" pi  $\{i + 1\}$  :  $\{PI\_K\_1(i+1,s).toString()\}$ ")
    } else {
        println("H de  $\{i + 1\}$  :  $\{H\_K\_2(i + 1, s).toString()\}$ ")
        println(" pi  $\{i + 1\}$  :  $\{Pi\_K\_2(i+1,s).toString()\}$ ")
    }
}
}

companion object {
    @Throws(InputMismatchException::class, IndexOutOfBoundsException::class)
    @JvmStatic
    fun main(vararg args: String) {
        val alg = Var1()
        alg.Initialize()
    }
}

```

Programul A2.2

```

package algorithm
import java.util.*

```

```

class Var2 {

    internal var nIterations: Int = 0
    internal var rRange: Int = 0
    internal var s: Double = 0.toDouble()
    internal var teta_k: Double = 0.toDouble()
    internal var eps = 0.00001

    internal var a: DoubleArray = DoubleArray(0)
    internal var b: DoubleArray = DoubleArray(0)
    internal var c: DoubleArray = DoubleArray(0)
    internal var pk: DoubleArray = DoubleArray(0)

    internal var sigma: DoubleArray = DoubleArray(0)
    internal var pi_: DoubleArray = DoubleArray(0)

    internal var reader = Scanner(System.`in`)
    //Locale.setDefault(Locale.US)
    var algType: Int = -1

    internal fun PI_K_1(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * PI_K_1(k - 1, s + a[k]) + sigmaDivide(k) *
            (PI_K_1(k - 1, s + a[k] * (1 - Pi_KK_1(k, s))) - PI_K_1(k - 1, s + a[k])) *

```

```

        V_K_1(k, s + a[k] * (1 - Pi_KK_1_(k, s))) +
        (a[k] / sigma[k]) * Pi_KK_1(k, s)
    return rez
}

internal fun Pi_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = sigmaDivide(k) * Pi_K_2(k - 1, s + a[k] * (1 - Pi_KK_2(k, s))) + (a[k] / sigma[k])
* Pi_KK_2(k, s)
    return rez
}

private fun sigmaDivide(k: Int) = sigma[k - 1] / sigma[k]

internal fun Pi_KK_1(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = V_K_1(k, s + a[k] * (1 - Pi_KK_1_(k, s))) * Pi_KK_1_(k, s)
    return rez
}

internal fun Pi_KK_1_(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_1(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
}

internal fun Pi_KK_2(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_2(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
//    printVals(k, rez, "Pi_KK_2")
    return rez
}

internal fun H_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = B_pass_K(k, s + sigma[k - 1] * (1 - PI_K_1(k - 1, s) * V_K_1(k, s)))
    return rez
}

internal fun H_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0

```

```

        val rez = B_pass_K(k, s + sigma[k - 1] * Math.pow(1 - (sigma[k - 1] / (s + sigma[k - 1])) *
(1 - B_pass_K(k, s + sigma[k - 1])) * Pi_K_2(k - 1, s), (-1).toDouble()))
        return rez
    }

    internal fun SetSigma() {
        sigma = DoubleArray(rRange + 1)
        sigma[0] = 0.0
        for (k in 1..nIterations - 1) {
            sigma[k] = sigma[k - 1] + a[k]
        }
    }

    internal fun B_pass_K(k: Int, s: Double): Double {
        var rez = b[k] / (s + b[k])
        return rez
    }

    internal fun C_pas_K_(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 1.0
        val rez = c[k] / (s + c[k])
        return rez
    }

    private fun returnZeroIfLesThan(k: Int): Boolean {
        if (k <= 0) {
            return true
        }
        return false
    }

    internal fun V_K_1(k: Int, s: Double): Double {
        var rez = 0.0
        rez = C_pas_K_(k, s + sigma[k - 1]) * Math.pow((1 - (sigma[k - 1] / (s + sigma[k - 1])) * (1
- C_pas_K_(k, s + sigma[k - 1])) * PI_K_1(k - 1, s)), -1.0)
        return rez
    }

    fun setPI(version: VERSION) {
        pi_ = DoubleArray(rRange)
        pi_[0] = 1.0
        for (k in 1..rRange - 1) {
            when (version) {
                Var2.VERSION.VERSION_1 -> pi_[k] = Pi_KK_1(k, s)
                Var2.VERSION.VERSION_3 -> pi_[k] = Pi_KK_2(k, s)
            }
        }
    }

    internal fun citire(x: DoubleArray, n: Int) {

```

```

    for (i in 0..n - 1)
        x[i] = reader.nextDouble().toDouble()
    }

    internal fun V_K_1_VARIANTA_2(k: Int, s: Double): Double {
        val rez = C_pas_K_(k, s + sigma[k - 1]) * Math.pow((1 - (sigma[k - 1] / (s + sigma[k - 1])))
* (1 - C_pas_K_(k, s + sigma[k - 1])) * PI_K_1(k - 1, s)), -1.0)
        return rez
    }

    internal fun InputInfo() {
        print("Introduceti nIterations valoare : ")
        nIterations = reader.nextDouble().toInt()
        rRange = nIterations
        print("Introduceti s valoare : ")

        try {
            s = reader.nextDouble().toDouble()
        } catch (e: InputMismatchException) {
            println(e.message.toString())
        }

        print("Introduceti teta_k valoare : ")
        teta_k = reader.nextDouble().toDouble()
        print("Introduceti coeficientii a : ")
        a = DoubleArray(nIterations)
        citire(a, nIterations)
        print("Introduceti coeficientii b : ")
        b = DoubleArray(nIterations)
        citire(b, nIterations)
        print("Introduceti coeficientii c : ")
        c = DoubleArray(nIterations)
        citire(c, nIterations)
    }

    fun selectAlgType() {
        println("input algorithm tipe 1,2,3,4 : ")
        algType = reader.nextDouble().toInt()
    }

    fun printVals(pas: Int, valoare: Double, name: String) {
        println("k = $pas : valoarea $name = $valoare")
    }

    internal fun Initialize() {
        InputInfo()
        for (i in 0..nIterations - 1) {
            SetSigma()
            if (b[i] < teta_k) {
                println("H de ${i+1} : ${H_K_1(i+1,s).toString()}")
            }
        }
    }

```

```

        println("Niu de ${i+1} : ${V_K_1(i+1,s).toString()}")
        println(" pi ${i + 1} : ${Pi_KK_1(i+1,s).toString()}")
    } else {
        println("H de ${i+1} : ${H_K_2(i+1,s).toString()}")
        println(" pi ${i + 1} : ${Pi_KK_2(i+1,s).toString()}")
    }
}
}
}

companion object {
    @Throws(InputMismatchException::class)
    @JvmStatic
    fun main(vararg args: String) {
        val alg = Var2()
        alg.Initialize()
    }
}
}

```

Programul A2.3

```

package algorithm
import java.util.*

```

```

class Var3 {
    internal var nIterations: Int = 0
    internal var rRange: Int = 0
    internal var s: Double = 0.toDouble()
    internal var teta_k: Double = 0.toDouble()
    internal var eps = 0.00001

    internal var a: DoubleArray = DoubleArray(0)
    internal var b: DoubleArray = DoubleArray(0)
    internal var c: DoubleArray = DoubleArray(0)
    internal var pk: DoubleArray = DoubleArray(0)

    internal var sigma: DoubleArray = DoubleArray(0)
    internal var pi_: DoubleArray = DoubleArray(0)

    internal var reader = Scanner(System.`in`)
    //Locale.setDefault(Locale.US)
    var algType: Int = -1

    internal fun PI_K_1(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * PI_K_1(k - 1, s + a[k]) + sigmaDivide(k) *
            (PI_K_1(k - 1, s + a[k] * (1 - Pi_KK_1(k, s))) - PI_K_1(k - 1, s + a[k])) *
            V_K_1(k, s + a[k] * (1 - Pi_KK_1(k, s))) +
            (a[k] / sigma[k]) * Pi_KK_1(k, s)
        return rez
    }
}

```

```

internal fun Pi_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = sigmaDivide(k) * Pi_K_2(k - 1, s + a[k] * (1 - Pi_KK_2(k, s))) + (a[k] / sigma[k])
* Pi_KK_2(k, s)
    return rez
}

private fun sigmaDivide(k: Int) = sigma[k - 1] / sigma[k]

internal fun Pi_KK_1(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = V_K_1(k, s + a[k] * (1 - Pi_KK_1_(k, s))) * Pi_KK_1_(k, s)
    return rez
}

internal fun Pi_KK_1_(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_1(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
}

internal fun Pi_KK_2(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_2(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
}

internal fun H_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = B_pass_K(k, s + sigma[k - 1]) +
        ( sigma[k - 1]/(s+sigma[k-1]) ) *
        ( 1 - B_pass_K(k,s+sigma[k-1]) ) *
        PI_K_1(k-1,s) * V_K_1(k,s)
    return rez
}

internal fun H_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = B_pass_K(k, s + sigma[k - 1] * Math.pow(1 - (sigma[k - 1] / (s + sigma[k - 1])) *

```

```

(1 - B_pass_K(k, s + sigma[k - 1])) * Pi_K_2(k - 1, s), (-1).toDouble()))
    return rez
}

internal fun SetSigma() {
    sigma = DoubleArray(rRange + 1)
    sigma[0] = 0.0
    for (k in 1..nIterations - 1) {
        sigma[k] = sigma[k - 1] + a[k]
    }
}

internal fun B_pass_K(k: Int, s: Double): Double {
    var rez = b[k] / (s + b[k])
    return rez
}

internal fun C_pas_K_(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 1.0
    val rez = c[k] / (s + c[k])
    return rez
}

private fun returnZeroIfLesThan(k: Int): Boolean {
    if (k <= 0) {
        return true
    }
    return false
}

internal fun V_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = C_pas_K_(k, s + sigma[k - 1] * (1 - PI_K_1(k - 1, s)))
    return rez
}

fun setPI(version: VERSION) {
    pi_ = DoubleArray(rRange)
    pi_[0] = 1.0
    for (k in 1..rRange - 1) {
        when (version) {
            Var3.VERSION.VERSION_1 -> pi_[k] = Pi_KK_1(k, s)
            Var3.VERSION.VERSION_3 -> pi_[k] = Pi_KK_2(k, s)
        }
    }
}

internal fun citire(x: DoubleArray, n: Int) {
    for (i in 0..n - 1)
        x[i] = reader.nextDouble().toDouble()
}

```

```

}

internal fun V_K_1_VARIANTA_2(k: Int, s: Double): Double {
    val rez = C_pas_K_(k, s + sigma[k - 1]) * Math.pow((1 - (sigma[k - 1] / (s + sigma[k - 1])))
* (1 - C_pas_K_(k, s + sigma[k - 1])) * PI_K_1(k - 1, s)), -1.0)
    return rez
}

internal fun InputInfo() {
    print("Introduceti nIterations valoare : ")
    nIterations = reader.nextDouble().toInt()
    rRange = nIterations
    print("Introduceti s valoare : ")
    try {
        s = reader.nextDouble().toDouble()
    } catch (e: InputMismatchException) {
        println(e.message.toString())
    }

    print("Introduceti teta_k valoare : ")
    teta_k = reader.nextDouble().toDouble()
    print("Introduceti coeficientii a : ")
    a = DoubleArray(nIterations)
    citire(a, nIterations)
    print("Introduceti coeficientii b : ")
    b = DoubleArray(nIterations)
    citire(b, nIterations)
    print("Introduceti coeficientii c : ")
    c = DoubleArray(nIterations)
    citire(c, nIterations)
}

fun selectAlgType() {
    println("input algorithm tipe 1,2,3,4 : ")
    algType = reader.nextDouble().toInt()
}

fun printVals(pas: Int, valoare: Double, name: String) {
    println("k = $pas : valoarea $name = $valoare")
}

internal fun Initialize() {
    InputInfo()
    for (i in 0..nIterations - 1) {
        SetSigma()
        if (b[i] < teta_k) {
            println("H de ${i+1} : ${H_K_1(i+1,s).toString()}")
            println("Niu de ${i+1} : ${V_K_1(i+1,s).toString()}")
            println(" pi ${i + 1} : ${PI_K_1(i+1,s).toString()}")
        } else {
            println("H de ${i+1} : ${H_K_1(i+1,s).toString()}")

```



```

        println(" pi ${i + 1} : ${Pi_K_2(i+1,s).toString()}")
    }
}
}

companion object {
    @Throws(InputMismatchException::class)
    @JvmStatic
    fun main(vararg args: String) {
        val alg = Var3()
        alg.Initialize()
    }
}
}

```

Programul A2.4

package algorithm

import java.util.*

```

class Var4 {
    internal var nIterations: Int = 0
    internal var rRange: Int = 0
    internal var s: Double = 0.toDouble()
    internal var teta_k: Double = 0.toDouble()
    internal var eps = 0.00001

    internal var a: DoubleArray = DoubleArray(0)
    internal var b: DoubleArray = DoubleArray(0)
    internal var c: DoubleArray = DoubleArray(0)
    internal var pk: DoubleArray = DoubleArray(0)

    internal var sigma: DoubleArray = DoubleArray(0)
    internal var pi_: DoubleArray = DoubleArray(0)

    internal var reader = Scanner(System.`in`)
    //Locale.setDefault(Locale.US)
    var algType: Int = -1
    //correct
    internal fun PI_K_1(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * PI_K_1(k - 1, s + a[k]) + sigmaDivide(k) *
            (PI_K_1(k - 1, s + a[k] * (1 - Pi_KK_1(k, s))) - PI_K_1(k - 1, s + a[k])) *
            V_K_1(k, s + a[k] * (1 - Pi_KK_1(k, s))) +
            (a[k] / sigma[k]) * Pi_KK_1(k, s)
        return rez
    }

    internal fun Pi_K_2(k: Int, s: Double): Double {
        if (returnZeroIfLesThan(k)) return 0.0
        val rez = sigmaDivide(k) * Pi_K_2(k - 1, s + a[k] * (1 - Pi_KK_2(k, s))) + (a[k] / sigma[k])
    }
}

```

```

* Pi_KK_2(k, s)
//   printVals(k, rez, "Pi_K_2")
   return rez
}

private fun sigmaDivide(k: Int) = sigma[k - 1] / sigma[k]

internal fun Pi_KK_1(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = V_K_1(k, s + a[k] * (1 - Pi_KK_1_(k, s))) * Pi_KK_1_(k, s)
//   printVals(k, rez, "Pi_KK_1")
    return rez
}

internal fun Pi_KK_1_(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_1(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
}

internal fun Pi_KK_2(k: Int, s: Double): Double {
    var pi = 0.0
    var piPrec = 0.0
    do {
        piPrec = pi
        pi = H_K_2(k, s + a[k] - a[k] * pi)
    } while (Math.abs(pi - piPrec) >= eps)
    val rez = (pi + piPrec) / 2
    return rez
}

internal fun H_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = B_pass_K(k, s + sigma[k - 1] * (1 - PI_K_1(k - 1, s) * V_K_1(k, s)))
    return rez
}

internal fun H_K_2(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 0.0
    val rez = B_pass_K(k, s + sigma[k - 1] * Math.pow(1 - (sigma[k - 1] / (s + sigma[k - 1])) *
(1 - B_pass_K(k, s + sigma[k - 1])) * Pi_K_2(k - 1, s), (-1).toDouble()))
    return rez
}

internal fun SetSigma() {
    sigma = DoubleArray(rRange + 1)
}

```

```

    sigma[0] = 0.0
    for (k in 1..nIterations - 1) {
        sigma[k] = sigma[k - 1] + a[k]
    }
}

internal fun B_pass_K(k: Int, s: Double): Double {
    var rez = b[k] / (s + b[k])
    var sol = Math.pow(rez, pk[k])
    return sol
}

internal fun C_pas_K_(k: Int, s: Double): Double {
    if (returnZeroIfLesThan(k)) return 1.0
    val rez = c[k] / (s + c[k])
    return rez
}

private fun returnZeroIfLesThan(k: Int): Boolean {
    if (k <= 0) {
        return true
    }
    return false
}

internal fun V_K_1(k: Int, s: Double): Double {
    var rez = 0.0
    rez = C_pas_K_(k, s + sigma[k - 1]) * Math.pow((1 - (sigma[k - 1] / (s + sigma[k - 1]))) * (1
- C_pas_K_(k, s + sigma[k - 1])) * PI_K_1(k - 1, s)), -1.0)
    return rez
}

internal fun citire(x: DoubleArray, n: Int) {
    for (i in 0..n - 1)
        x[i] = reader.nextDouble().toDouble()
}

internal fun InputInfo() {
    print("Introduceti nIterations valoare : ")
    nIterations = reader.nextDouble().toInt()
    rRange = nIterations
    print("Introduceti s valoare : ")
    try {
        s = reader.nextDouble().toDouble()
    } catch (e: InputMismatchException) {
        println(e.message.toString())
    }

    print("Introduceti teta_k valoare : ")
    teta_k = reader.nextDouble().toDouble()
}

```

```

    print("Introduceti coeficientii a : ")
    a = DoubleArray(nIterations)
    citire(a, nIterations)
    print("Introduceti coeficientii b : ")
    b = DoubleArray(nIterations)
    citire(b, nIterations)
    print("Introduceti coeficientii c : ")
    c = DoubleArray(nIterations)
    citire(c, nIterations)
}

fun selectAlgType() {
    println("input algorithm tipe 1,2,3,4 : ")
    algType = reader.nextDouble().toInt()
}

fun printVals(pas: Int, valoare: Double, name: String) {
    println("k = $pas : valoarea $name = $valoare")
}

internal fun Initialize() {
    InputInfo()

    println("introduceti vectorul Pk")
    pk = DoubleArray(nIterations)
    citire(pk, nIterations)
    for (i in 0..nIterations - 1) {
        SetSigma()
        if (b[i] < teta_k) {
            println("H de ${i + 1} : ${H_K_1(i + 1, s).toString()}")
            println("Niu de ${i + 1} : ${V_K_1(i + 1, s).toString()}")
            println("pi ${i + 1} : ${PI_K_1(i+1,s).toString()}")
        } else {

            println("H de ${i + 1} : ${H_K_2(i + 1, s).toString()}")
            println("pi ${i + 1} : ${Pi_K_2(i+1,s).toString()}")
        }
    }
}

companion object {
    @Throws(InputMismatchException::class, IndexOutOfBoundsException::class)
    @JvmStatic
    fun main(vararg args: String) {
        val alg = Var4()
        alg.Initialize()
    }
}

```

DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII

Subsemnata, declar pe răspundere personală că materialele prezentate în teza de doctorat sunt rezultatul propriilor cercetări și realizări științifice. Conștientizez că, în caz contrar, urmează să suport consecințele în conformitate cu legislația în vigoare.

Mitev Lilia

Semnătura

Data:

CURRICULUM VITAE



Numele de familie și prenumele:	Mitev (Usatîi) Lilia
Data nașterii:	27.11.1986
Cetățenia:	Republica Moldova
Studii:	<i>Licență:</i> Universitatea Pedagogică de Stat "Ion Creangă" din Chișinău, 2004-2009, specialitatea Informatică și limba engleză, calificarea Licențiat în informatică; <i>Masterat:</i> Universitatea Pedagogică de Stat "Ion Creangă" din Chișinău, 2009-2011, specialitatea Tehnologii informaționale și de comunicație în instruire, calificarea master în Științe ale educației; <i>Doctorat:</i> Universitatea Academiei de Științe a Moldovei, Institutul de Matematică și Informatică al AȘM, 2012-2015, specialitatea 112.03-Cibernetică matematică și cercetări operaționale.
Domeniile de interes științific:	Teoria așteptării, teoria probabilităților și statistică matematică.
Participări în proiecte științifice naționale și internaționale:	- Proiectul A.Ș.M. (Academy of Sciences of Moldova) – STCU (The Science & Technology Center in Ukraine), cifrul 13.820.08.06 STCU.F/5854: <i>Sisteme cu priorități cu schimb semi-Markov și control în rețele complexe</i>, 2013-2014; - Proiectul pentru Tineri Cercetatori, cifrul 13.819.18.05A: <i>Modele de așteptare semi-Markov</i>,

2013-2014;

3. Proiectul A.Ș.M - Bielorusia, cifrul 10.820.06BF: *Cercetarea sistemelor de așteptare cu priorități în sisteme contemporane de diversificare a resurselor informaționale*, 2010-2011;

4. Proiectul A.Ș.M - Germania, cifrul 09.820.08.0IGF: *Analiza și modelarea procesării traficului informațional în viitoarele generații de rețele*, 2009.

Participări la foruri științifice:

1. Conferința Științifică Internațională a doctoranzilor „Tendințe contemporane ale dezvoltării științei: viziuni ale tinerilor cercetători”, Chișinău, 10 martie 2015;

2. The International Conference "The Third Conference of Mathematical Society of the Republic of Moldova dedicated to the 50th anniversary of the foundation of Institute of Mathematics and Computer Science "IMCS-50", Chișinău, August 19-23, 2014;

3. The International Conference "Mathematics Days in Sofia" – MDS 2014, Sofia, Bulgaria, July 07-10, 2014;

4. Conferința Științifică Internațională a doctoranzilor „Tendințe contemporane ale dezvoltării științei: viziuni ale tinerilor cercetători”, Chișinău, 10 martie 2014;

5. Conferință științifică "UNIVERSITAS EUROPAEA XXI: Știința Universitară în contextul Integrării Europene", Chișinău, 18 octombrie 2013;

6. The 17-th International Conference on "Distributed Computer and Communication Networks (DCCN-2013): "Control, Computation, Communications", Moscow, Russia, October 07-10, 2013;

7. The International Conference "The 21st Conference on Applied and Industrial Mathematics" - CAIM 2013, București, Romania, September 19-22, 2013;

8. The International Conference "Mathematics & Information Technologies: Research and Education

- (MITRE-2013)”, Chişinău, August 18-22, 2013;
9. The International Conference ”The 37th Annual Congress of the American Romanian Academy of Arts and Sciences (ARA)”, Chişinău, June 04-09, 2013;
10. Conferinţa Ştiinţifico-Practică Internaţională „Politici economice şi financiare pentru o dezvoltare competitivă”, Chişinău, 12 aprilie 2013;
11. Conferinţa Ştiinţifică Internaţională „Strategii de dezvoltare socio-economică a societăţii în condiţiile globalizării”, Chişinău, 15-17 octombrie 2012;
12. The International Conference ”The 20th Conference on Applied and Industrial Mathematics” - CAIM 2012, Chişinău, August 22-25, 2012;
13. Conferinţa Ştiinţifică Internaţională “Modelarea matematică, optimizare şi tehnologii informaţionale”, Chişinău, 19-23 martie 2012;
14. The International Conference ”The 19th Conference on Applied and Industrial Mathematics” - CAIM 2011, Iaşi, România, September 22-25, 2011;
15. The International Conference “Mathematics & Information Technologies: Research and Education” - MITRE-2011, Chişinău, August 22-25, 2011;
16. The International Conference ”The 7th Congress of the Romanian Mathematicians”, Braşov, Romania, June 29-July 25, 2011;
17. Conferinţa Ştiinţifică Internaţională „Probleme şi perspective de dezvoltare a potenţialului economic şi managerial al Republicii Moldova în condiţiile de criză”, Chişinău, 21 aprilie 2011;
18. The International Conference ”The 18th Conference on Applied and Industrial Mathematics” - CAIM 2010, Iaşi, România, October 14-17, 2010;
19. Conferinţa Ştiinţifică Internaţională “Modelare

	Matematică, Optimizare și Tehnologii Informaționale”, Chișinău, 24-26 martie, 2010;
	20. The International Conference on "Information Technologies, Systems and Networks", Chișinău, February 25-26, 2010.
Lucrări științifice publicate:	Articole în reviste științifice – 4, Articole în culegeri științifice – 13, Teze ale comunicărilor științifice – 12, Ghiduri didactice – 1.
Premii, mențiuni, distincții:	1. Premiul anual "Vladimir Andrunachievici" pentru tineri al Institutului de Matematică și Informatică al Academiei de Științe a Moldovei, 2015; 2. Bursa oferită de Federația Mondială a Oamenilor de Știință, Elveția, iulie 2014 – iunie 2015; 3. Bursa de Excelență a Guvernului pentru doctoranzi, 2014; 4. Premiului pentru Tineri Cercetători pentru cea mai bună prezentare în cadrul "the Third Conference of Mathematical Society of the Republic of Moldova" - "IMCS-50", 2014; 5. Bursa de merit de gradul II pentru masteranzi, 2010; 6. Diploma for excellent presentation of the report at "The International Conference "Informațional Technologies, Systems & Networks" ITNS, 2010.
Cunoașterea limbilor:	<i>Limba română</i> - maternă, <i>Limba engleză</i> - bine, <i>Limba rusă</i> - bine.
Date de contact:	Adresa: Chișinău, str. Academiei 5, IMI al AȘM Telefon: 079213150 Email: liliausate@yahoo.com