

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА
УНИВЕРСИТЕТ „ПРОФ. Д-Р АСЕН ЗЛАТАРОВ“
ФАКУЛТЕТ ПО ТЕХНИЧЕСКИ НАУКИ
КАТЕДРА КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ

Живко Михайлов Томов

РАЗРАБОТВАНЕ И МОДЕЛИРАНЕ НА МЕТОДИ ЗА ПРОГНОЗИРАНЕ

А В Т О Р Е Ф Е Р А Т

на дисертационен труд за присъждане на образователна и научна степен „доктор“ по научна област 5. Технически науки, професионално направление 5.3 Комуникационна и компютърна техника, в докторска програма

„Компютърни системи и технологии“

Научни ръководители: чл. кор. проф. дмн дтн Красимир Годоров Атанасов

проф. д-р Сотир Николов Сотиров

Бургас , 2019 г.

Дисертационният труд е посветен на някои DataMining методи за предсказване и техните модели базирани на невронни и обобщени мрежи. В глава I (теоретична част) е обоснована необходимостта от изграждане на модели на обекти и процеси при научните изследвания, инженерната практика, икономическите разчети и изучаването на закономерностите в обществото. Отбелязано е, че често е невъзможно или нецелесъобразно да се провери в реални условия как реагира дадена сложна система на определени външни условия. Решаването на споменатия проблем най-често се свежда до създаване на опростено копие (модел) на изучаваната система, при което са налице съществените за целта на конкретното прогнозиране функционалност, поведение на системата и взаимодействие с околната среда. Отбелязано е, че най-често използваните видове модели са физическите и аналитичните. Приведени са плюсовете и минусите на всеки от тях. Направен е обзор на най-често използваните аналитични модели, като за илюстрация за повечето от тях са съставени несложни примери. Така в глава I акцентът е поставен не върху решаването на конкретни задачи, а върху приложимостта, особеностите, принципите и подходите при изграждането на различните подвидове аналитични модели. Обърнато е внимание на процеса на анализ на получаваните от модела резултати.

Особен акцент е поставен върху Веригите на Марков, Невронните и Обобщените мрежи, тъй като те са едни от най-удобните средства за моделиране на процеси и обекти от видове разгледани в следващите глави.

Изяснена е същността на Веригите на Марков, подчертано е удобството, което със своята простота, те предлагат на изследователя. Демонстрирани са методите за представянето им с помощта на граф (удобно при решаване на задачи от предимно теоретичен характер) и с помощта на матрица (при задачи с приложна насоченост). Когато Марковският процес е представен с помощта на граф, възлите на този граф съответстват на състоянията на системата, а дъгите – на вероятностите за преход от едно състояние във всяко възможно. Във втория случай Марковският процес се описва с матрица с n реда и n колони. Всеки ред съответства на дадено състояние, всяка позиция от този ред съдържа вероятността за преход в съответното състояние. Основополагащо положение е, че вероятностите за преход не зависят от предисторията на процеса.

Решената практическа задача илюстрира построяването на Матрицата на преходите, съдържаща вероятностите за преминаване на системата от едно състояние в друго и изчисляване на вектора на вероятностите на всяка стъпка. Показани са ред свойства на Веригите на Марков, като по-специално внимание е обърнато на сходимостта на стойностите на вектора на вероятностите за преход при нарастване на броя на стъпките.

С помощта на Веригите на Марков е удобно да се моделират процеси свързани с обработката на опашки и ред процеси протичащи паралелно в сложни системи.

Направен е обзор на развитието на Теорията на Веригите на Марков – Скритото моделиране на Марков.

Особено внимание е обърнато на случаите, в които изучаваният процес е много сложен и изследователят не разполага с достатъчно знания за него. Това не позволява да се създаде, нито модел във вид на една или няколко алгебрични функции, нито вероятностен модел. Тогава се налага да се използват подходи, позволяващи на базата на наблюдения да бъде създаден модел, който със зададена точност имитира поведението на процеса, който се анализира, ако влияещите фактори се променят в даден диапазон. В общия случай изследователят разполага със стойностите на сигналите, които постъпват в системата и тези появяващи се на нейния изход.

Съвременен инструмент за моделиране на процеси със слабо изучена природа са невронните мрежи. Те правят възможно решаването на задачи с елементи на размитост (класификация, клъстеризация), в това число и свързаните с изкуствения интелект. В глава I подробно са разгледани видовете, структурата, организацията, функционирането и приложимостта на невронните мрежи.

С оглед на особената значимост на проблема свързан с оптимизацията на Невронните мрежи са разгледани подходите за редуцирането на размерността на вектора с входните данни, включително базирания на Теорията на Интеркритериалния анализ представена от К. Атанасов. Последният се оказва особено ефективен при работа с множество критерии, когато определянето на точната им количествена оценка е проблематично.

Последната част на глава I е посветена на друг съвременен инструмент за описание на процеси протичащи паралелно – предложените от К. Атанасов Обобщени мрежи. Те са по-нататъшно развитие на Мрежите на Петри, като към последните са добавени свойства,

които позволяват да бъде разширена сферата на тяхното приложение. Подробно са разгледани видовете, структурата и функционирането на Обобщените мрежи. Показани са използваните алгоритми за движение на ядрата в обобщена мрежа. С оглед на това, че по-нататък в дисертационния труд са предложени модели базирани на Редуцирани Обобщени мрежи, в глава I са разгледани техните същност и особености.

В глава II са реализирани подробно практически задачи изискващи моделирането на сложни процеси. Построените модели са базирани на невронни мрежи. Предварителната обработка на данните, построяването на невронните мрежи, тяхното обучение и тестване е осъществено със средствата на програмната среда Matlab.

С помощта на първата мрежа е извършена клъстеризация на различни марки нефт, като на входовете на невронната мрежа са подавани предварително избрани характеристики на нефта. Обучената мрежа е способна да определи клъстера, към който се отнася дадена марка. Използвана е мрежа на Кохонен.

Разработеният MATLAB-скрипт класифицира обекти от различен характер, като така позволява на потребителя да работи с различен брой обучаващи вектори, при което размерността им също се задава по време на изпълнението. С него е извършена класификация на видовете нефт по 13 признака. След обучение на мрежата на входовете и се подава непознат вектор и тя определя на кой клъстер той принадлежи.

```
% Creating and testing a self organizing map
clear all;
Topology = [7];
Properties = input('Please enter number of components in each input vector:');
Instances = input('Please enter number of instances that will be used during creating the
network:');
[FileName Path] = uigetfile('*.txt', 'Please select file that contains input data');
PathAndFileName = strcat(Path, FileName);
% reading input data
Fid = fopen(PathAndFileName, 'r');
for i = 1:Instances
    P(i, :) = fscanf(Fid, '%g', Properties);
end;
```

```

fclose(Fid);
p = P';
clear P;
% calculating ranges in each column
for i = 1:Properties
    ranges(i, 1) = min(p(i, :));
    ranges(i, 2) = max(p(i, :));
end;
% creating and training the network
net = newsom(ranges, Topology);
net.trainParam.epochs = 2000;
net.trainParam.show = 100;
[net, tr] = train(net, p);
% user should enter the vector to determine to which cluster it pertains
for i = 1:Properties
    uv(i) = input('Please enter current element of vector to analyze: ');
end;
a = sim(net, uv);
% saving the result
[FileName Path] = uiputfile('*.TXT', 'Please select file where results will be stored');
Fid = fopen(strcat(Path, FileName), 'w');
fprintf(Fid, 'Vector to analyze: ');
fprintf(Fid, '%g ', uv);
fprintf(Fid, '\r\n');
fprintf(Fid, 'This vector relates to cluster ');
for i = 1: Topology(1)
    if a(i) == 1
        fprintf(Fid, '%d', i),
    end,
end;
fclose(Fid);

```

Приведеният скрипт създава невронна мрежа представляваща самоорганизираща се карта на Кохонен със седем неврона, разположени в едномерно пространство. Така мрежата формира седем клъстера. След обучението и, на потребителя се предлага да въведе вектор и скриптът определя към кой клъстер той принадлежи. Ако е необходимо да бъдат създадени друг брой клъстери, следва да се промени стойността на променливата Topology.

В Таблица са приведени Данните, с които е била обучена мрежата.

Следващият скрипт решава друга задача от областта на клъстеризацията – след създаване и обучение на мрежата, за всеки вектор от обучаващото множество се определя към кой клъстер той принадлежи. Мрежата отново представлява самоорганизираща се карта на Кохонен. Тя се състои от 10 неврона разположени в двумерно пространство.

```
% Creating and testing a self organizing map
clear all
Topology = [2 5];
Properties = input('Please enter number of properties:');
[FileName Path] = uigetfile('*.txt', 'Please select file that contains input data');
PathAndFileName = strcat(Path, FileName);
Instances = input('Please enter number of instances that will be used during creating the
network:');
% reading input data
Fid = fopen(PathAndFileName, 'r');
for i = 1:Instances
    c = fscanf(Fid, '%16s', 1);
    cs = size(c);
    if cs(2) < 16
        for j = cs(2) + 1 : 16
            c(j) = ' ';
        end,
    end;
    country(:,i) = c;
    P(i, :) = fscanf(Fid, '%g', Properties),
```

```

end;
fclose(Fid);
p = P';
clear P;
% calculating ranges
for i = 1:Properties
    ranges(i, 1) = min(p(i, :));
    ranges(i, 2) = max(p(i, :));
end;
% creating and training the network
net = newsom(ranges, Topology);
net.trainParam.epochs = 2000;
net.trainParam.show = 100;
[net, tr] = train(net, p);
a = sim(net, p);
% saving results
[FileName Path] = uiputfile('* .TXT', 'Please select file where results will be stored');
Fid = fopen(strcat(Path, FileName), 'w');
cs = size(Topology);
c = 1;
for i = 1 : cs(2)
    c = c * Topology(i),
end;
for i = 1: c
    fprintf(Fid, 'кльстер %d:\r\n', i);
    for j = 1 : Instances
        if a(i,j) == 1
            fprintf(Fid, '%s,\r\n', country(:,j)'),
        end,
    end;
end;
end;

```


fclose(Fid);

С помощта на мрежата са класифицирани 50 страни по демографски ориентирани данни показани в таблица. Резултатите получени без предварителната им обработка са незадоволителни. Прави впечатление, разпределянето по клъстери е излишно силно повлияно от числеността на населението на страните. Не е достъпна информация как точно извършват изчисленията използваните от скрипта функции на Matlab, но има основание да се предположи, при определяне на Евклидовото разстояние между точките съответстващи на векторите, тези функции работят със същинските стойности подавани на входа. Така влиянието на колоните (съответно характеристиките), които са изразени с големи числа, се оказва несъразмерно голямо. Затова в един и същ клъстер се оказват страни близки единствено по численост на населението и силно различаващи се по детска смъртност, брой на деца в семейството и процент на грамотните. С цел решаване на проблема е извършена предварителна нормализация на данните по колони. Така в колоните бяха получени стойности в интервала 0..1, като бяха запазени отношенията между тях.

След изпълнение на скрипта с нормализирани по колони данни в един и същ клъстер попадат страни с близки демографски характеристики, като тези измерващи се в много големи числа не доминират над останалите. Това потвърждава предположението, че функцията на Matlab `newsom`, която изгражда мрежата, не извършва предварителна нормализация на данните от обучаващото множество, въпреки че получава като параметър диапазоните, в които се намират едноименните компоненти на входните вектори. Авторът е на мнение, че това обстоятелство, въпреки че създава определено неудобство за програмирането на Matlab, носи определени предимства - позволява да бъдат задавани приоритети на различните характеристики на обектите подавани на входа на мрежата. За целта може да бъде създаден вектор с теглата на всяка характеристика, на който да се умножават входните вектори от обучаващото множество. Естествено, необходима е предварителна нормализация на входните данни по колони. Така например, ако изследователят преценява, че раждаемостта, смъртността и броят на децата в семейството имат по-голямо значение при формирането на демографската картина, той може да зададе за съответните характеристики тегла по-големи от тези за характеристики като съвкупен обществен продукт на човек, процент на градското население и процент на грамотните.

За илюстрация на казаното, на характеристиките на входните данни от споменатия пример са присвоени тегла задаващи приоритетите. След нормализацията по колони всеки компонент на векторите от обучаващото множество е умножен на съответното тегло. С така получените стойности скриптът изгради и обучи невронната мрежа, която след това определи принадлежността на всеки вектор към даден клъстер. С така обработените входни данни един и същ клъстер попадат предимно страни близки по характеристики с висок приоритет независимо с какви абсолютни стойности се измерват различните характеристики.

Изборът на метод за определяне на теглата съответстващи на приоритетите на характеристиките може да бъде предмет на бъдещи изследвания.

В глава II е реализирана оптимизация на невронна мрежа основано на редуциране на броя на входовете и, при което функционалността и не спада под необходимото ниво. Ефективен инструмент за определяне на нивото на значимост на отделните компоненти на входните вектори е Интеркритериалният анализ. Изградена е невронна мрежа за предсказване на цетановото число на суровия нефт, като първоначално е предположено, че то зависи от 8 параметъра. Използвани са 140 проби от суров петрол.

Софтуерното приложение, което изпълнява ИКА алгоритъма, връща резултатите под формата на две индексирани матрици (показани в таблици), които съдържат съответно принадлежащите и неприналежащите части на интуиционистките размити корелации открити между критериите (28 двойки). В отделни Таблици са показани връзките между двойките критерии, получени чрез прилагане на ИКА и изчисленото разстояние d_{ij} за всяка двойка критерии C_i-C_j от точка (1;0) в интуиционистки размития триъгълник.

На следващата стъпка е избрана двойката C_i-C_j с най-малкото d_{ij} , като по този начин се осигурява максимална близост на съответната точка до точка (1; 0). След това е избран третият критерий в тройката или като C_k , който е следващият най-силно корелиращ критерий с C_i , или като C_l , който е следващият най-силно корелиращ критерий с C_j , приемайки, че тройката от критерии $C_i-C_j-C_k$ или $C_i-C_j-C_l$, достига минимум на $(D_{ij} + D_{ik} + D_{jk}, D_{ij} + D_{il} + D_{jl})$. В отделна Таблица са показани двойките критерии $C_i -C_j$ в „силен положителен консонанс“, „положителен консонанс“ и „слаб положителен консонанс“.

На първата стъпка всички измервания на 140-те проби от суров петрол с 8-те критерия за всяка се анализират, за да се осигури база за сравнение на получените резултати на следващите стъпки.

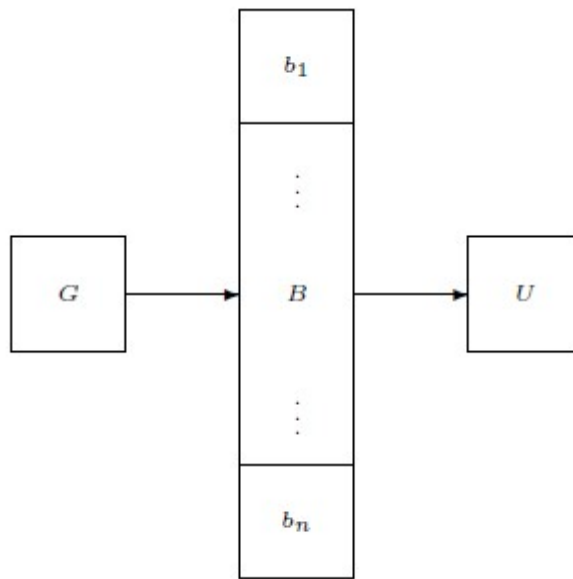
Входният вектор е разделен на три части: обучение (70%), валидиране (15%) и тестване (15%).

На първата стъпка от процеса, за да се обучи мрежата, са използвани всичките 8 критерия. След обучителния процес всички входни стойности са симулирани от невронната мрежа. Средното отклонение на всичките 140 проби е 1,8134. Корелационният коефициент r , получен с помощта на програмната среда MATLAB, е 0.97434. На всяка от следващите няколко стъпки от входните данни е премахната по една колона и са анализирани получените без нейно участие резултати. Накрая процесът води до работа с мрежа с 4 входа, като средното отклонение на всичките 140 проби е 2.19, а коефициентът на корелация - 0.95927.

Обобщавайки е отбелязано, че невронните мрежи подхождат към моделирания процес или обект като към Черна кутия. Те използват информацията от проведените наблюдения за да бъдат обучени. Поради това те са удобно средство за моделиране на процеси и обекти, за които може да бъде събрана много входно-изходна информация (например от измерване), но трудно биха могли да бъдат съставени чисто аналитични модели. Невронните мрежи са ефективни при решаване на задачи за класификация и апроксимация, когато е налично достатъчно количество информация.

В глава III са предложени Обобщени мрежи моделиращи някои процеси протичащи в системите за масово обслужване с опашка.

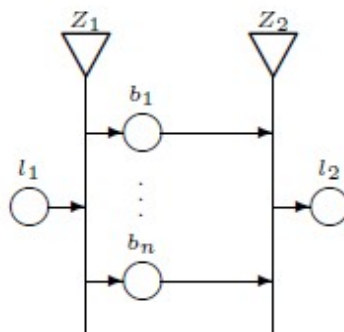
Прието е, че съществува източник на данни G (генератор на порции информация - заявки), буфер B с n клетки представящ опашка с капацитет n заявки и канал обработващ заявките U (Фиг. 3.1).



Фиг. 3.1 Модел на обслужване с опашка

При изграждането на предложените модели се използват само редуцирани обобщени мрежи без приоритети на преходите, позициите и ядрата, без времеви компоненти и капацитети на позициите кореспондиращи с буфера равни на 1. Тъй като не са зададени капацитетите на незаетите позиции, те се приемат за 1. Капацитетите на дъгите и типа на преходите не са дефинирани.

Основните дисциплини на опашките могат да бъдат представени чрез обобщена мрежа съдържаща 2 прехода, както е показано на фиг. 3.2. Отново с n е обозначен капацитетът на опашката – максималният брой на заявките в нея (представени чрез позициите на мрежата).



$$Z_1 = \langle \{l_1\}, \{b_1, \dots, b_n\}, r_1 \rangle.$$

$$r_1 = \frac{\begin{array}{c|cccc} & b_1 & \dots & b_j & \dots & b_n \\ \hline l_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \end{array}}{}$$

$$Z_2 = \langle \{b_1, \dots, b_n\}, \{l_2\}, r_2 \rangle,$$

$$r_2 = \frac{\begin{array}{c|c} & l_2 \\ \hline b_1 & s_{1,2} \\ \vdots & \vdots \\ b_j & s_{j,2} \\ \vdots & \vdots \\ b_n & s_{n,2} \end{array}}{}$$

Фиг. 3.2. Модел на обобщена мрежа

Предложените мрежи описват процеса, при който при наличие на свободен канал в СМО се прилага определена дисциплина.

Първият модел реализира дисциплина FIFO, допълнителен критерий задаващ приоритет на заявката не е предвиден. Началните характеристики на заявките се задават от генератора, който ги създава и не се променят при преминаването през мрежата.

Преходите на тази обобщена мрежа са:

$$Z_1 = \langle \{L_1\}, \{B_1, \dots, B_n\}, r_1 \rangle,$$

където:

$$r_1 = \frac{\begin{array}{c|cccc} & b_1 & \dots & b_j & \dots & b_n \\ \hline l_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \end{array}}{}$$

$r_{1,j} = b_j$ е първата празна позиция отгоре надолу сред позициите b_1, b_2, \dots, b_n ,

тук и по-долу: $1 \leq j \leq n$.

$$Z_2 = \langle \{b_1, \dots, b_n\}, \{l_2\}, r_2 \rangle,$$

където:

$$r_2 = \begin{array}{c|c} & l_2 \\ \hline b_1 & s_{1,2} \\ \vdots & \vdots \\ b_j & s_{j,2} \\ \vdots & \vdots \\ b_n & s_{n,2} \end{array} ;$$

$S_{j,2}$ = “ядрото в позиция b_j е престояло повече време в нея по сравнение с ядрата във всички други позиции.

Вторият модел използващ обобщена мрежа реализира дисциплина LIFO. Той има същото графично представяне като на Фиг. 3.2. Предикатите на този модел са:

$$r_{i,j} = b_j$$

тук b_j е първата празна позиция отгоре надолу сред

$$b_1, b_2, \dots, b_n,$$

за $j = 1, 2, \dots, n$.

$S_{j,2}$ = „Ядрото в позиция b_j се е намирало там по-кратко време по сравнение с ядрата в останалите позиции.”

Наличието на предикати на преходите в структурата на Обобщените мрежи дава възможност да бъдат описани по-сложни системи за масово обслужване, като се променя само вида на предикатите, както следва:

$r_{i,j} = „b_j$ е първата свободна отгоре надолу сред входните позиции. Удовлетворява се условието C_1 , за $j = 1, 2, \dots, n$.”

$s_{j,2} = „ядрото в позиция b_j е било в позицията по-дълго/по-кратко време по сравнение с ядрата в останалите позиции и е изпълнено условието C_2 ”.$

C_1 и C_2 са начално зададени условия.

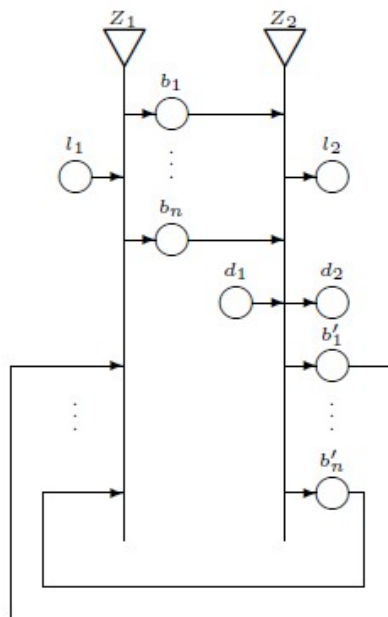
Така третият и четвъртият модел базирани на обобщена мрежа намират своето описание. Например, ако C_1 и C_2 са тавтологии, получаваме първия и втория модели.

В по-горните 4 модела не се интересувахме от характеристиките на ядрата влизаци през l и b позиции. Характеристиките се присвояваха от генератора на данни и не се променяха при преминаването през в мрежата. За да се определи времето за престой на

всяко от ядрата в позициите b_1, b_2, \dots, b_n , считаме, че ядрата получават като характеристика времето на влизане през позициите кореспондиращи с клетките на буфера.

Предложени са два модела, в които допълнителният критерий за преход на ядрата може да бъде променян по време на работа на мрежата.

Ядрата Γ_1 и Γ_2 влизат в позиции c_1 и c_2 на обобщената мрежа показана на фиг. 3.3 с начални характеристики $x_{i0}^\Gamma = \text{„критерий } C_i \text{ за избор на ядро“}$, $i = 1, 2$ съответно.



Фиг. 3.3 Пример за обобщена мрежа

Преходите имат вид:

$$Z_1 = \langle \{l_1, b'_1, \dots, b'_n\}, \{b_1, \dots, b_n\}, r_1 \rangle,$$

където:

	b_1	\dots	b_j	\dots	b_n
l_1	$r_{1,1}$	\dots	$r_{1,j}$	\dots	$r_{1,n}$
b'_1	$t_{1,1}$	\dots	$t_{1,j}$	\dots	$t_{1,n}$
\vdots	\vdots	\dots	\ddots	\dots	\vdots
b'_i	$t_{i,1}$	\dots	$t_{i,j}$	\dots	$t_{i,n}$
\vdots	\vdots	\dots	\ddots	\dots	\vdots
b'_n	$t_{n,1}$	\dots	$t_{n,j}$	\dots	$t_{n,n}$

$r_{1,j} = b_j$ е първата празна позиция отгоре надолу сред позициите b_1, b_2, \dots, b_n & x_{i0}^Γ

$$Z_2 = \langle \{b_1, \dots, b_n, d_1\}, \{l_2, d_2, b'_1, \dots, b'_n\}, r_2 \rangle,$$

където:

	l_2	d_2	b'_1	...	b'_j	...	b'_n
b_1	$s_{1,2}$	<i>false</i>	$u_{1,1}$...	<i>false</i>	...	<i>false</i>
\vdots	\vdots	\vdots	...	\ddots	...	\vdots	
$r_2 = b_j$	$s_{j,2}$	<i>false</i>	<i>false</i>	...	$u_{j,j}$...	<i>false</i> ;
\vdots	\vdots	\vdots	...	\ddots	...	\vdots	
b_n	$s_{n,2}$	<i>false</i>	<i>false</i>	...	<i>false</i>	...	$u_{n,n}$
d_1	<i>false</i>	<i>true</i>	<i>false</i>	...	<i>false</i>	...	<i>false</i>

$S_{j,2}$ = ядрото в позиция b_j е престояло в позицията повече време по сравнение с ядрата в позициите b_1, b_2, \dots, b_n & x_{20}^Γ

В случая е описана дисциплината FIFO.

Следващият модел описва друга дисциплина на опашка. Той има същото графично представяне като изображения на фиг. 3.3. Предикатите са $r_{1,j} = b_j$ е първата празна позиция отгоре надолу сред позициите

$$b_1, b_2, \dots, b_n \text{ \& } x_{10}^\Gamma$$

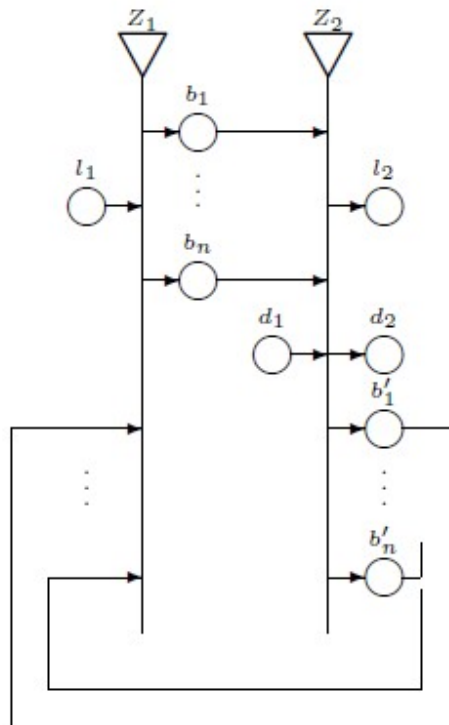
$s_{j,2}$ = „Ядрото в позиция b_j се е намирало там по-кратко време по сравнение с ядрата в мъртвите позиции.“

$$b_1, b_2, \dots, b_n \text{ \& } x_{20}^\Gamma$$

за $j = 1, 2, \dots, n$.

В общия случай дисциплината на опашката може да бъде променена във всеки момент от функционирането на обобщената мрежа.

Разгледан е модел базиран на обобщена мрежа, в която ядрата могат да променят местата си в буфера (позициите b_1, b_2, \dots, b_n), съгласно определени критерии D . Обобщената мрежа реализираща модела е показана на Фиг. 3.4.

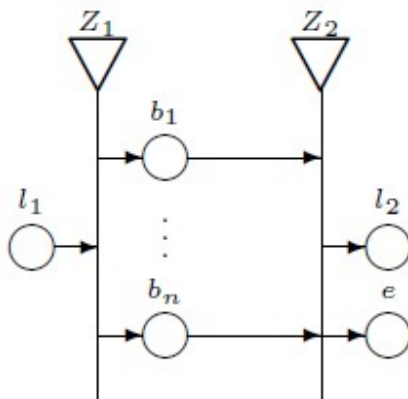


Фиг. 3.4 Пример за обобщена мрежа

В определени моменти ядрото Δ ще влезе в позиция d_1 с начални характеристики $x^{\Delta}_0 =$ „критерий D за промяна на положението на ядрото“:

Както по-горе, предложеният модел има 2 разновидности, съответстващи на базовите дисциплини на опашките (FIFO, LIFO и “С-критерия”).

Предложена е модификация на първия модел, в която някои от преминаващите през мрежата ядра могат в определен момент да я напуснат, поради това, че времето им на живот е изтекло. Такава мрежа е показана на Фиг. 3.5.



Фиг. 3.5 Пример за обобщена мрежа

Първият преход на тази мрежа ще съвпадне с прехода Z_1 на първата обобщена мрежа, а вторият преход на новата мрежа има вида:

$$Z_1 = \langle \{b_1, \dots, b_n\}, \{l_2, e\}, r_2 \rangle,$$

Където:

$$r_2 = \begin{array}{c|cc} & l_2 & e \\ \hline b_1 & s_{1,2} & v_1 \\ \vdots & \vdots & \vdots \\ b_j & s_{j,2} & v_j \\ \vdots & \vdots & \vdots \\ b_n & s_{n,2} & v_n \end{array};$$

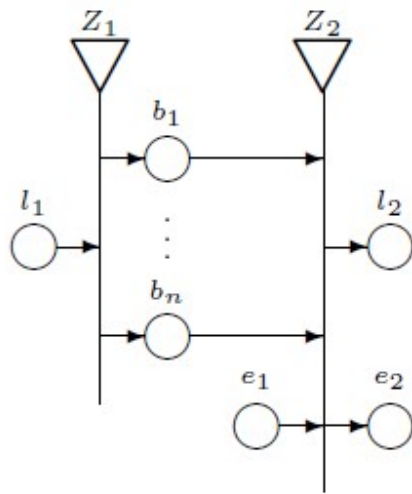
$S_{j,2}$ = “Ядрото в позиция b_j е престояло в позицията най-продължително (най-късо или в съответствие с С-критерия) време сред ядрата във всички позиции“

b_1, b_2, \dots, b_n ,”

за $j = 1, 2, \dots, n$;

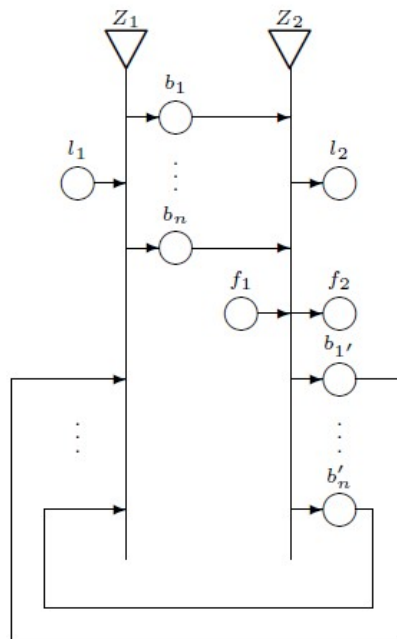
v_j = “времето на престой на текущото ядро в позиция b_j изтекло”.

В този вид обобщената мрежа е статична, тъй като времето за престой е фиксирано предварително. Ако е необходимо времето да бъде променено докато мрежата функционира, тя следва да има вида показан на Фиг. 3.6, но описанието и ще бъде подобно на по-горната. В мрежата могат да бъдат добавени позиции $c_1, c_2, c_3, c_4, d_1, d_2$ и ядрата $\Gamma_1, \Gamma_2, \Delta$, ако искаме да моделираме по-сложни дисциплини на опашката.



Фиг. 3.6 Пример за обобщена мрежа

Предложена е още една обобщена мрежа, в която ядрата могат да получават нови характеристики докато са в буфера. Един възможен вид на такава мрежа е показан на фиг. 3.7.



Фиг. 3.7 Пример за ОМ на процес с опашка

Преходите на тази мрежа имат вид:

$$Z_1 = \langle \{l_1, b'_1, \dots, b'_n\}, \{b_1, \dots, b_n\}, r_1 \rangle,$$

Където:

	b_1	\dots	b_j	\dots	b_n
l_1	$r_{1,1}$	\dots	$r_{1,j}$	\dots	$r_{1,n}$
b'_1	$true$	\dots	$false$	\dots	$false$
\vdots	\vdots	\dots	\ddots	\dots	\vdots
b'_j	$false$	\dots	$true$	\dots	$false$
\vdots	\vdots	\dots	\ddots	\dots	\vdots
b'_n	$false$	\dots	$false$	\dots	$true$

$R_{1,j} = b_j$ е първата празна отгоре надолу сред входящите позиции.

$$Z_2 = \langle \{b_1, \dots, b_n, f_1\}, \{l_2, f_2, b'_1, \dots, b'_n, r_2\} \rangle,$$

Където:

	l_2	f_2	b'_1	\dots	b'_j	\dots	b'_n
b_1	$s_{1,2}$	$false$	$true$	\dots	$false$	\dots	$false$
\vdots	\vdots	\vdots	\dots	\ddots	\dots	\vdots	
b_j	$s_{j,2}$	$false$	$false$	\dots	$true$	\dots	$false$
\vdots	\vdots	\vdots	\dots	\ddots	\dots	\vdots	
b_n	$s_{n,2}$	$false$	$false$	\dots	$false$	\dots	$true$
f_1	$false$	$true$	$false$	\dots	$false$	\dots	$false$

$S_{j,2} = b_j$ е първата от горните пълни клетки на буфера”.

В глава III е показано как веригите на Марков могат да бъдат представени с помощта на Обобщени мрежи. Най-удобно е за подобно представяне да се използват редуцирани Обобщени мрежи с преходи от вида:

$$Z = (L', L'', r),$$

Където:

$$L = \{k_1, \dots, k_m\}$$

е множеството от входни позиции на прехода, а

$$L'' = \{l_1, \dots, l_n\}$$

е множеството на изходните им позиции и r е Индексираната матрица на преходите.

Където r_{ij} е предикатът съответстващ на i -та входна и j -та изходна позиции ($1 \leq i \leq m, 1 \leq j \leq n$).

Когато стойността на този предикат е "true", ядрото от i -та входна преминава в j -та изходна позиция.

Завършената редуцирана обобщена мрежа, която ще бъде използвана, има следното формално описание:

$$(A, K, X, \Phi),$$

където:

- A е множеството на всички преходи на обобщената мрежа,
- K - множеството на ядрата, които влизат в обобщената мрежа, или се намират в определени нейни позиции с начални характеристики дадени от функцията X ,
- Φ - характеристичната функция, която присвоява нови характеристики на всяко ядро когато то преминава от входна в изходна позиция на даден преход.

Дефинирани са различни релации, оператори и операции над мрежата.

Операциите са обединение, сечение и изваждане. Релациите се използват за сравнение на структурата и резултатите от работата на 2 Обобщени мрежи.

Операторите са:

- global - променящ глобалните компоненти на мрежата,
- local – променящ компонентите на прехода,
- иерархичен – заместващ подмрежа с преход или позиция, както и обратно – заместващ преход или позиция на дадена мрежа с подмрежа, заместващ една подмрежа с друга, заместващ ядро с подмрежа,
- редуциращ - отстраняващ определени компоненти на мрежата, или определящ кой компонент е отстранен,
- разширяващ – позволяващ мрежата да бъде разширена до мрежа от даден тип,
- динамичен - променящ алгоритмите на движение на ядрата, на определяне на възможността за сливане и разцепване на ядрата, на промяна на начина за изчисление на стойността на предикатите.

Показана е възможността на обобщените мрежи за представяне на всяка верига на Марков. На всеки връх на веригата на Марков може да се съпостави преход. Когато ядрото представлящо функционирането на веригата на Марков влезе в позицията, то може да има като текуща характеристика число между 0 и 1. Нека дадено ядро е в позиция l_x . То може да има начални характеристики в зависимост от конкретния модел и споменатата по-горе

променлива r . Влизайки в други позиции, ядрото може да получи като характеристика не само нова стойност на r , но и друга информация съществена за модела.

Особено съществено е, че в обобщената мрежа могат да бъдат поставени множество ядра за определяне на начина на функциониране на веригата на Марков, при което всяко от тях може да преминава през мрежата със собствени характеристики (със собствени стойности на променливата r). Ако капацитетите на позициите са безкрайност, ядрата могат да преминават през мрежата независимо и безконфликтно, но ако капацитетите на всички или част от позициите са ограничени, то между ядрата могат да възникнат конфликти.

Съществува възможност да бъде променяна стойността на променливата определяща посоките на хода на процесите, (или посоките на движение на ядрата) чрез предикатите, които да се изчисляват (когато преходите се активират) като резултат на натрупани достатъчно ядра във входните позиции на тези преходи.

Показаната на Фиг. 3.13. обобщена мрежа моделира процеса на изчисляване на n -то просто число по няколко различни формули.

Всяко натурално число $n > 1$ има канонично представяне

$$n = \prod_{i=1}^k p_i^{\alpha_i},$$

където:

p_1, \dots, p_k са различни прости числа,

$\alpha_1, \dots, \alpha_k$ са натурални числа,

Интерес представляват предложените от К. Атанасов формули за намиране на n -то просто число.

$$p_n = \sum_{i=0}^{C(n)} sg \left(n - \sum_{j=2}^i \overline{sg}(j-1 - \varphi(j)) \right), \quad (4)$$

$$p_n = \sum_{i=0}^{C(n)} sg \left(n - \sum_{j=2}^i \overline{sg}(\psi(j) - j - 1) \right), \quad (5)$$

$$p_n = \sum_{i=0}^{C(n)} sg \left(n - \sum_{j=2}^i \overline{sg}(\sigma(j) - j - 1) \right), \quad (6)$$

$$p_n = \sum_{i=0}^{C(n)} sg \left(n - \sum_{j=2}^i \overline{sg}(j - \eta(j)) \right), \quad (7)$$

$$p_n = \sum_{i=0}^{C(n)} sg(n - \sum_{j=2}^n \overline{sg}(\tau(j) - 2)), \quad (8)$$

$$p_n = \sum_{i=0}^{C(n)} sg(n - \sum_{j=2}^n \overline{sg}(\Omega(j) - 1)), \quad (9)$$

$$\varphi(n) = \prod_{i=1}^k p_i^{\alpha_i - 1} (p_i - 1), \quad \varphi(1) = 1,$$

$$\psi(n) = \prod_{i=1}^k p_i^{\alpha_i - 1} (p_i + 1), \quad \psi(1) = 1,$$

$$\sigma(n) = \prod_{i=1}^k \frac{p_i^{\alpha_i + 1} - 1}{p_i - 1}, \quad \sigma(1) = 1,$$

$$\tau(n) = \prod_{i=1}^k (1 + \alpha_i), \quad \tau(1) = 1,$$

$$\Omega(n) = \sum_{i=1}^k \alpha_i, \quad \Omega(1) = 1$$

Предложена е обобщена мрежа моделираща паралелното изчисляване на n -то просто число по няколко формули, като заедно с това дава информация за необходимото за изчисленията компютърно време за всяка от тях. Мрежата съдържа $r + s + 4$ прехода, $3r + 3s + 10$ позиции и 4 вида ядра. В определени моменти от време ядрата a_1, a_2, \dots влизат в позиция l_1 с начална характеристика "натуралното число n ". По-нататък под "ядро" се има предвид текущото i -то ядро (с индекс i). Ядрата от вид a ще генерират ядра от видовете b и g . По време на функционирането на обобщената мрежа в позиция l_4 се намира ядро от вид d с начална и текуща характеристика "множеството обекти", като всеки обект представлява двойка " $n - p_n$ " (n – натурално число, p_n – n -то просто число).

Преходите на обобщената мрежа имат вида:

$$Z_1 = \langle \{l_1, l_4, l_9\}, \{x_1, \dots, x_{3r-2}, l_2, l_3, l_4\},$$

	x_1	\dots	x_{3r-2}	l_2	l_3	l_4
l_1	<i>false</i>	\dots	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>
l_4	$P_{4,1}$	\dots	$P_{4,3r-2}$	$P_{4,2}$	$P_{4,4}$	<i>true</i>
l_9	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>

$$\rangle,$$

където:

$P_{4,2}$ = “обектът $(n - p_n)$ на множеството d^r ”,

$P_{4,1} = \dots = P_{4,3r-2} = P_{4,4} = \neg P_{4,2}$,

Тук $\neg P$ е отрицание на P .

Ядрото a влиза в позиция l_4 и се слива с ядрото d без нова характеристика. На следващата стъпка, ако P_4 има стойност *true*, ядрото d се разцепва на 2 ядра – същото ядро d и ядрото a влязло в позиция l_2 с характеристика “простото число P_n “. Ако $P_{4,2}$ е *false* ($P_{4,1} = \dots = P_{4,3r-2} = P_{4,4} = \text{true}$), ядрото d се разцепва на $r + 2$ ядра – същото ядро d , ядрото a влязло в позиция l_3 с началната си характеристика и ядрата $b_{i,1}.. b_{i,r}$ с характеристика “натуралното число n ”.

В позиция l_7 α_{r+1} приема характеристика каноничното представяне на n и последователно се разцепва на s ядра β_1, \dots, β_s , които влизат в други изходни позиции без нови характеристики.

За всяко натурално число $i \in (1 \div r)$,

$$X_i = \langle \{x_{3i-2}, x_{3i}\}, \{x_{3i-1}, x_{3i}\},$$

	x_{3i-2}	x_{3i-1}	x_{3i}
x_{3i}	<i>false</i>	<i>true</i>	<i>true</i>
	x_{3i}	$P_{3i,3i-1}$	$P_{3i,3i}$

$$\rangle,$$

където :

$P_{3i, 3i-1}$ = „ядрото е направило n цикъла”,

$P_{3i, 3i} = \neg P_{3i, 3i-1}$, където $\neg S$ е отрицание на предиката S .

Ядрото I излиза от позиция x_{3i} и влиза в позиция x_{3i-1} с характеристика “стойността на P_n и времето за неговото изчисляване”.

Вторият преход има вида:

$$Z_2 = \langle \{l_3, l_6\}, \{l_5, l_6\},$$

	l_5	l_6	
l_3	<i>false</i>	<i>true</i>),
l_6	$P_{6,5}$	$P_{6,6}$	

където:

$P_{6,5}$ = “следващата характеристика на ядрото е 1”,

$P_{6,6} = \neg P_{6,5}$

В позиция l_6 ядрото α_{r+1} получава характеристика следващия делител на n , а в позиция l_5 - множеството на сортираните си делители.

Следващият преход има вид:

$$Z_3 = \langle \{l_5, l_7\}, \{y_1, \dots, y_{3s-2}, l_7\},$$

	y_1	...	y_{3s-2}	l_7	
l_5	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>).
l_7	<i>true</i>	...	<i>true</i>	<i>false</i>	

В позиция l_7 ядрото α_{r+1} получава характеристика каноничното представяне на n и последователно се разцепва на s ядра - b_1, \dots, b_s , всяко от последните влиза в изходните позиции без нови характеристики. Тук s е 6, тъй като моделираният процес включва 6 формули. За всяко натурално число

$$Y_4 = \langle \{y_{3i-2}, y_{3i}\}, \{y_{3i-1}, y_{3i}\},$$

	y_{3i-1}	y_{3i}	
y_{3i-2}	<i>false</i>	<i>true</i>),
y_{3i}	$P_{3i,3i-1}$	$P_{3i,3i}$	

където

$P_{3i,3i-1}$ = “ядрото е направило $C(n)$ цикъла”,

където

$$C(n) = \left[\frac{n^2 + 3n + 4}{4} \right],$$

$P_{3i,3i} = \neg P_{3i,3i-1}$

В позиция y_{3i} ядрото b_j получава нова характеристика – следващия член на сумата получава по формулата, а в позиция l_{3i-1} – стойността на P_n и времето за изчислението му.

Четвъртия преход има вид:

$$Z_4 = \langle \{x_2, \dots, x_{3r-1}, y_2, \dots, y_{3s-1}, l_9\}, \{l_8, l_9, l_{10}\},$$

	l_8	l_9	l_{10}	
x_2	<i>false</i>	<i>false</i>	<i>true</i>	
\vdots	\vdots	\vdots	\vdots	
x_{3r-1}	<i>false</i>	<i>false</i>	<i>true</i>	
y_2	<i>false</i>	<i>false</i>	<i>true</i>),
\vdots	\vdots	\vdots	\vdots	
y_{3s-1}	<i>false</i>	<i>false</i>	<i>true</i>	
l_{10}	<i>true</i>	$P_{10,9}$	<i>false</i>	

където

$P_{10,9}$ = “двойката (n, P_n) не е елемент на множеството”.

- [1]. Всички ядра от вид α и β влизат в позиция l_{10} , където се сливат в ядро γ с характеристика “множеството функции изчислили верно P_n ”. На следващата стъпка това ядро се разцепва на две γ_1 , които влизат в позиция l_8 с характеристика “минималното време за изчисляване на P_n ” и γ_2 с характеристика “двойката (n, P_n) ”, който се добавя към множеството прости числа в позиция l_4 .

ПРИНОСИ В ДИСЕРТАЦИОННИЯ ТРУД

Приносите в дисертационния труд имат научен и научно-приложен характер и могат да се формулират по следния начин:

Създадени и описани са:

- Обобщеномрежов модел на ART1 невронна мрежа
- Обобщеномрежов модел на когнитивно невронен алгоритъм за адаптивно резонансна теория 1
- Обобщеномрежов модел на алгоритъма за бавно обучение на ART2 невронната мрежа
- Обобщеномрежов модел на алгоритъма за бързо обучение на ART2 невронната мрежа
- Обобщеномрежови модели на стандартните буфери и стандартните вериги на Марков от теорията на масовото обслужване
- Обобщеномрежови модели на по-общи буфери и вериги на Марков.
- Обобщеномрежов модел за изследване на поведението на аритметични функции, чрез които може да се конструира n -тото просто число.

Реализирана е програмна симулация на обобщеномрежовия модел на ART2 невронната мрежа и да се разработи софтуер за разпознаването на лица в изображение, на рентгенови и ЯМР изображения и чрез използване на Viola-Jones алгоритъма, да се определя конкретен човек в изображение.

НАСОКИ ЗА БЪДЕЩА РАБОТА

Разработените обобщеномрежови модели дават възможност да се формулират различни идеи за техни бъдещи приложения: например, разработване на обобщеномрежов модел на процеса на криптиране текст в изображение, също както и оптимизация на ART2 алгоритъма.

В бъдеще ще се работи и по комбиниране на идеите за обобщени буфери и обобщени вериги на Марков в рамките на описаните в дисертационния труд обобщеномрежови модели.

ПУБЛИКАЦИИ СВЪРЗАНИ С ДИСЕРТАЦИОННИЯ ТРУД

- 1*. Tomov, Zh., M. Kravczak, V. Andonov, E. Dimitrov, K. Atanassov. Generalized net models of queueing disciplines in finite buffer queueing systems. Proceedings of the 16-th International Workshop on Generalized Nets, 10 Feb. 2018, Sofia, 1-9.
- 2*. Tomov, Zhivko, Vassilev, Peter, Atanassov, Krassimir, Sotirov, Sotir. Generalized Net Model of the process of n-th prime number calculation by different formulas. ANNA'18; Advances in Neural Networks and Applications 2018., VDE VERLAG GMBH · Berlin · Offenbach, 2018, ISBN:978-3-8007-4756-6, 68-72. (Scopus)
- 3*. Kacprzyk, A., V. Andonov ., Z. Tomov, K. Atanassov. On some representations and modifications of Markov chains. Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 14, 2018/2019 (in press).
- 4*. Petkov T., P. Jovcheva, Z. Tomov, S. Simeonov, S. Sotirov, A Generalized Net Model of the Neocognitron Neural Network, Flexible Query Answering Systems, Springer International Publishing, 2017, 249-258, (SJR 0.28 (Q2))
- 5*. Tomov, Z., M. Krawczak, V. Andonov, K. Atanassov, S. Simeonov. Generalized net models of queueing disciplines in finite buffer queueing systems with intuitionistic fuzzy evaluations of the tasks. Notes on Intuitionistic Fuzzy Sets, Vol. 25, 2019, No. 2, 115–122.
- 6*. Sotirov, S. V. Atanassova, E. Sotirova, L. Doukovska, V. Bureva, D. Mavrov, J. Tomov - Application of the Intuitionistic Fuzzy InterCriteria Analysis Method with Triples to a Neural Network Preprocessing Procedure - Computational Intelligence and Neuroscience, Volume 2017, (IF 2.154)