

Изследване на алгоритми
за конструиране
на обобщеномрежови модели

Вася Красиминова Атанасова

АВТОРЕФЕРАТ

на дисертационен труд, представен за присъждане
на образователна и научна степен „доктор“ по научна специалност
01.01.12 „Информатика“, професионално направление
4.6 “Информатика и компютърни науки”

Научен ръководител: акад. Иван П. Попчев

Дисертационният труд е обсъден и допуснат до защита на разширено заседание на секция „Интелигентни системи” на ИИКТ-БАН, състояло се на 20 март 2013 г.

Дисертационният труд съдържа 127 страници, в които 30 фигури, 15 таблици и библиография, включваща 153 заглавия.

Защитата на дисертационния труд ще се състои на от часа в зала на блок на ИИКТ-БАН на открито заседание на научното жури в състав:

1.
2.
3.
4.
5.

Материалите по защитата са на разположение на интересуващите се при секретаря на НС на ИИКТ-БАН, ул. „Акад. Георги Бончев”, бл. 2.

Автор:

Вася Красиминова Атанасова

Заглавие:

Изследване на алгоритми за конструиране на обобщеномрежови модели

Характеристика на дисертационния труд

Понятието „обобщена мрежа“ се появява преди 30 години. През това време то е изследвано и като математически обект и като средство за моделиране на реални процеси. Както се вижда от двете библиографии, публикувани през 2002 и 2007 г., през годините интересът към понятието нараства и се проявява в повече от 700 публикации с автори от над 20 страни. Въпреки това, теоретичните проблеми, които стоят пред изследователите на обобщените мрежи, все още са твърде много, а и непрекъснато се появяват нови идеи и проблеми.

Настоящият дисертационен труд е посветен на съществуващи и новопоявили се теоретични проблеми, на някои възможни приложения на обобщените мрежи, както и на средствата за разпространяване на информацията за обобщени мрежи. Направен е преглед на развитието на алгоритмите за движение на ядра в преход и на функционирането на цяла мрежа, като са обосновавани пунктовете, в които могат тези два алгоритъма да се оптимизират по отношение на броя стъпки за изпълнение. За някои видове интуиционистки размити обобщени мрежи за първи път са описани как трябва да се модифицират стандартните алгоритми в съответствие със спецификите на тези разширения.

Предложени са две нови разширения на понятието „обобщена мрежа“, за които също са представени съответно модифицираните спрямо тях алгоритми за функциониране. Доказани са теореми за консервативност на тези разширения.

Използваната в труда терминология от областта на обобщени мрежи е заимствана от [2, 41, 48], а терминологията от областта на интуиционистки размитите множества следва възприетата в [50]. За край на доказателство на теорема ще бъде използван знакът \square .

Цели и задачи

Основните цели на дисертационния труд могат накратко да се обобщят по следния начин:

1. Да се дефинират разширения на понятието „обобщена мрежа“ и да се изследват относно консервативността им спрямо стандартната дефиниция. Да се построят алгоритмите за движение на ядрата в тях на ниво отделен преход и цяла мрежа.
2. Да се разработят обобщеномрежови модели на метаевристични алгоритми и да се изследва приложимостта им за оптимизиране на отделни аспекти на работата на самите обобщени мрежи. Да се разработят модели на алгоритмите за разрешаване на конфликти в системи с контрол на версиите.

За постигане на тези цели са формулирани следните задачи:

1. Да се проучат съществуващите модификации на обобщените мрежи, в това число и алгоритмите за движение на ядрата в тях. Да се построят алгоритми за движение на ядрата в съществуващите досега четири вида интуиционистки размити обобщени мрежи.
2. Да се дефинира и изследва разширението интуиционистки размити обобщени мрежи от пети тип, да се докаже неговата консервативност и да се построят алгоритмите за движение на ядрата на ниво отделен преход и цяла мрежа.

3. Да се дефинира и изследва разширението „обобщени мрежи с обеми на ядрата“, да се докаже неговата консервативност и да се построят алгоритмите за движение на ядрата на ниво отделен преход и цяла мрежа.
4. Да се предложи решение на проблем за представимостта на преходите, в обобщените мрежи, формулиран през 1991 г.
5. Да се опише обобщеномрежовия модел, отразяващ паралелната работа на алгоритъма на мравките и на генетичен алгоритъм, оптимизиращи работата на реален процес.

Апробация на резултатите

Апробацията на резултатите е осъществена в рамките на представяния на доклади на няколко международни конференции, в статии в научни списания и тематични сборници. Резултатите се явяват постижения и по няколко научни проекта, в чиито колективи авторът се явява член или бенефициент:

- Проект с Фонд „Научни изследвания“ № ДИД-02-29 / 2009 „Моделиране на процеси с фиксирани правила на развитие“;
- Проект с Фонд „Научни изследвания“ № ДТК-02-44 / 2009 „Ефективни Монте Карло методи за големи научно-изследователски задачи“;
- Проект с Фонд „Научни изследвания“ № БИН-02-09 / 2009 „Проектиране и разработка на интуиционистки размити логически средства в информационните технологии“;
- Проект с ЕСФ и MOMH № BG051PO001-3.3.04/40 „Изграждане на висококвалифицирани млади изследователи по съвременни информационни технологии за оптимизация, разпознаване на образи и подпомагане вземането на решения“.
- Проект по Royal Society, International Joint Project № JP100372 „Обобщеномрежови модели и интуиционистки размити множества в интелигентните системи“.

Списък на публикациите по дисертационния труд

1. Atanassova, V., Georgiev, P., Generalized net models of conflict resolution approaches in version control systems. Part 2: Copy-Modify-Merge, *Proc. of 10th Int. Workshop on Generalized Nets*, 5 Dec. 2009, 14–21, ISSN: 1313-6860.
2. Atanassova, V., Generalized net models of conflict resolution approaches in version control systems. Part 1: Lock-Modify-Unlock, *Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Vol. II: Applications*, Warsaw, 2010, 13–24, ISBN: 978-8389475305.
3. Atanassova, V., Design of training tests on generalized nets, *Proc. of 5th IEEE Conf. on Intelligent Systems*, 7–9 July 2010, London, UK, 327–330, ISBN: 978-1-4244-5163-0.
4. Atanassova, V. Utilizing the model graphic structures when teaching generalized nets. *Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Vol. II: Applications*, 2010, 7–17, ISBN: 978-83-89475367.
5. Атанасова, В. Уики технологии в помощ на науката. Примерът с Ifigenia.org, *Сборник от 8^{ма} Национална младежка научно-практическа сесия на ФНТС*, 10–11 май 2010, 83–88, ISSN: 1314-0698.
6. Atanassov, K., Dimitrov, D., Atanassova, V., Algorithms for Tokens Transfer in Different Types of Intuitionistic Fuzzy Generalized Nets, *Journal of Cybernetics and Information Technologies*, Vol. 10, 2010, No.4, 22–35, ISSN: 1311-9702 (print), 1314-4081 (online).

7. Atanassov, K., V. Atanassova, P. Chountas, A. Shannon. Generalized nets with places having intuitionistic fuzzy capacities. *Notes on Intuitionistic Fuzzy Sets*, Vol. 17, 2011, No. 4, 21–28, ISSN: 1310-4926.
8. Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov. A generalized net with an ACO-algorithm optimization component, *Lecture Notes in Computer Science*, Vol. 7116, 2012, 190–197, ISSN: 0302-9743.
9. Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5. Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: Monte Carlo Methods and Applications, *Proc. of the 8th IMACS Seminar on Monte Carlo Methods*, August 29 – September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, 39–46, eBook ISBN: 9783110293586.
10. Atanassova, V., On Intuitionistic Fuzzy Approach to Generalized Net Prognostics. In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications*, SRI-PAS, Warsaw, 2012, 1–12, ISBN: 978-8389475411.
11. Atanassova, V. The minimal solution of a problem in generalized nets, *Proc. of the 6th IEEE Int. Conference on Intelligent Systems*, 6–8 Sept. 2012, Sofia, Bulgaria, Vol. II, 159–163, ISBN: 978-1-4673-2277-5.
12. Atanassova, V. Generalized Nets with Volumetric Tokens, *Comptes rendus de l'Academie Bulgare des Sciences*, Vol. 65, 2012, No. 11, 1489–1498, ISSN: 1310-1331.

Списък на забелязани цитирания на публикациите по дисертационния труд

Забелязани са общо 7 цитирания на 4 публикации.

Atanassov, K., D. Dimitrov, V. Atanassova, Algorithms for Tokens Transfer in the Different Types of Intuitionistic Fuzzy Generalized Nets. Cybernetics and Information Technologies, Vol. 10, 2010, No. 4, 22–35, ISSN 1311-9702 (print), 1314-4081 (online).

Цитирана в:

1. Todorova, M., K. Kanev, Educational Framework for Verification of Object-oriented Programs, The Joint International Conference on Human-Centered Computer Environments HCCE'2012, 8-13 March, Hamamatsu, Japan, 2012, 23-27, ISBN 978-1-4503-1191-5.
2. Тодорова, М. Построяване на коректни обектно-ориентирани програми чрез изграждане на техни обобщеномрежови модели, Годишник на секция “Информатика” към Съюза на учените в България, том 4, 2011, 1-28, ISSN 1313-6852.
3. Todorova, M. Model checker of object-oriented programs based on generalized nets, In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications*. (K. Atanassov, M. Baczynski, J. Drewniak, J. Kacprzyk, M. Krawczak, E. Szmidt, M. Wygalak, S. Zadrozny, Eds.), SRI-PAS, Warsaw, 2012, 309-320, ISBN: 9788389475411.
4. Todorova, M. Verification of Procedural Programs via Building there Generalized Nets Models. Proceedings of the 41st Spring Conference of the Union of Bulgarian Mathematicians, Mathematics And Education In Mathematics, 2012, 259-265, http://www.math.bas.bg/smb/2012_PK/tom_2012/pdf/259-265.pdf

Atanassova, V. Design of Training Tests on Generalized Nets. –In: Proc. of the 5th IEEE Int. Conf. on Intelligent Systems, 7–9 July 2010, London, UK, 327–330, ISBN 978-1-4244-5163-0.

Цитирана в:

5. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov A generalized net with an ACO-algorithm optimization component, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 7116, 2012, 190–197.

Цитирана в:

6. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5: Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: Monte Carlo Methods and Applications, Proceedings of the 8th IMACS Seminar on Monte Carlo Methods, August 29 – September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, 39–46, eBook ISBN: 9783110293586.

Цитирана в:

7. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Съдържание на дисертационния труд

Дисертационният труд е в обем от 122 страници и се състои от увод, четири глави, заключение, декларация за оригиналност на резултатите, библиография, списък на публикациите по дисертационния труд, списък на забелязаните цитирания на публикации по дисертационния труд и списък на проектите с участие на докторанта. Дисертационният труд включва 30 фигури и 15 таблици, а библиографията към него – 153 заглавия.

Глава 1. Накратко за обобщените мрежи

В тази глава са дадени кратки исторически бележки за появата и развитието на ОМ, основните дефиниции, свързани с тях, и алгоритмите за движение на ядрата в тях.

1.1. Обобщени мрежи – поява и развитие

Понятието ОМ е дефинирано от К. Атанасов през 1982 г. В обзора са дадени данни за основните дисертации, свързани с ОМ, както и за публикациите, в т.ч. монографиите в тази област.

Изследванията в теоретичен план, са свързани с изучаване на свойствата на ОМ като математически обекти. Накратко са описани идеите от алгебричния, топологическия, логическия, операторния, методологическия и програмния аспекти на теорията на ОМ.

Важна част от изследванията по ОМ са свързани с изучаване на *редуцираните ОМ*, т.е. ОМ, от които са отстранени част от компонентите, и на *разширенията на ОМ*, т.е. ОМ, към които са добавени нови компоненти.

През 1983 г. К. Атанасов дефинира понятието интуиционистки размито множество (ИРМ), започва да изучава свойствата му и да търси приложенията му. Така възниква идеята за съчетаване на ИРМ и ОМ – през 1985 г. се появява първото разширение на ОМ – *интуиционистки размити ОМ*. През годините са дефинирани други над 20 различни разширения на ОМ: цветни ОМ, ОМ с глобална памет, ОМ с интервално време за активирание, ОМ със сложна структура, ОМ с оптимизационен компонент, двупосочни ОМ, ОМ с ядра, имащи време на живот и други.

През последните 20 години основното внимание на специалистите, интересувани се от ОМ, е насочено към възможностите за тяхното приложение. Например, описани са над 500 ОМ-модела на процеси на поставяне на медицински диагнози в областта на нефрологията, неврологията, общата и вътрешна медицина. Създадени са и ОМ модели на функциониране на отделните системи в човешкото тяло.

Едни от най-интересните приложения на ОМ са свързани с областта на изкуствения интелект. През 1991 г. пред конференция на Института по микропроцесорна техника К. Атанасов излиза с доклад, в който предлага езикът на ОМ да се превърне в език за универсално описание на отделните области на изкуствения интелект. През следващите 20 г. голяма част от резултатите на специалистите, работещи в областта на ОМ, са свързани с реализиране на тази идея.

По идея на А. Shannon в началото на века започва интензивна работа по описание с ОМ на основните процеси, протичащи в учебно заведение, като акцент пада на процесите, свързани с електронно обучение. Резултати се съдържат публикации с участието на специалисти от Австралия, Белгия, Великобритания, Индия, Италия, Мароко, Полша, Португалия, САЩ, Словакия, Южна Корея и други. Описани са и ОМ-модели на процеси на разпознаване и синтез на образи и говор и на вземане на решения. Досега за голяма част от основните видове невронни мрежи е показано, че тяхното функциониране и резултатите от работата им може да се опише с ОМ – по една универсална ОМ за всеки от отделните видове невронни мрежи.

1.2. Формални дефиниции

Дадени са формалните дефиниции на понятията *преход* и *ОМ*, съгласно основните публикации в тази област. Тези дефиниции са изложени в параграф 2.2.

1.3. Алгоритми за функциониране на переход и обобщена мрежа

Алгоритмите за движение на ядрата в ОМ, също както и дефиницията, са се променяли във времето неколкостранно. В дисертационния труд са изложени най-новите версии на алгоритмите за движение на ядрата в отделен переход и в цяла ОМ.

Ще отбележим, че тук, в стъпка (А04) на първия от двата алгоритъма, е предложена за първи път модификация, която цели допълнително увеличаване на бързодействието на алгоритъма: тя се състои в проверка дали ядрата могат или не могат да се разцепват и в случай, че не могат, процесът на проверка на предикатите се преустановява при намиране на първия предикат с вярностна стойност „истина“.

От прегледа на литературата става ясно, че досега не са дискутирани от методологична гледна точка отделните аспекти на дефиницията и алгоритмите за движение на ядра в переход и мрежа. Затова авторът реши да проведе проучване сред студенти по дисциплината „Обобщени мрежи“ в Техническия университет – София, Университета „Проф. д-р Асен Златаров“ – Бургас, и Бургаския свободен университет доколко се усвояват тези знания. В Глава 3 са дискутирани резултатите от проучванията, а в Глава 2 се съдържат някои коментари, изясняващи дефинициите и отделните стъпки на двата алгоритъма, които в явен вид никога не са правени в досегашните публикации, а биха били полезни за по-доброто разбиране на този апарат за математическо моделиране.

Глава 2. Развитие на теорията на обобщените мрежи

Глава 2 съдържа пет резултата на автора, свързани с теорията на ОМ.

В точка 2.1. е решен проблем, формулиран през 1991 г., отнасящ се до представимостта на переходите на ОМ. Този резултат е докладван пред международната конференция на IEEE “Intelligent Systems” 2012 г., [74].

В точка 2.2. е предложено разширение на понятието ОМ, наречено „ОМ с обеми на ядрата“. Даден е алгоритъм за движение на ядрата в новия вид ОМ и е формулирана и доказана теорема, че това разширение на ОМ е консервативно. Резултатите са публикувани в [71].

Текстът на точка 2.3. следва и подобрява публикация [62] с участието на автора, където за първи път е описан алгоритъм за движение на ядрата в интуиционистки размити ОМ от I, II, III и IV вид на ниво переход и цяла мрежа.

Нов, пети, вид интуиционистки размити ОМ е описан в точка 2.4. на базата на [58]. Даден е алгоритъм за движение на ядрата в новия вид ОМ и е формулирана и доказана теорема, че това разширение на ОМ е консервативно.

В точка 2.5 е описана публикуваната в [73] процедура за събиране на статистически данни, извлечени от симулация на ОМ модел. Детайлно е описан методът за използване на тази информация за прогностични цели. Използвана е интуиционистка

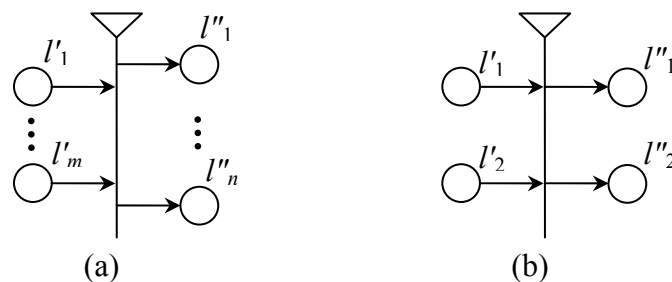
размитост, за да се съставят оценки на преходите на ядрата по време на симулациите, които след това се интерпретират като вероятности за преход на ядрата по време на активиранията на мрежата за прогностична цел.

2.1. Представимост на преходите

Тук се дискутира проблем за представимостта на преходите, предложен от К. Атанасов през 1991 г. Идеята зад този открит проблем произлиза от статия на С. Böhm и G. Jakorini. През 2005 г. е предложено едно решение на проблема от В. Тасева, но тук предлагаме нов подход, който води до ново – доказано минимално – решение.

В. Тасева е въвела означението $H_{m,n}$, за да означи ОМ преход с m входни и n изходни позиции. Специален случай на $H_{m,n}$ е преходът $H_{2,2}$ (Фиг. 2.1).

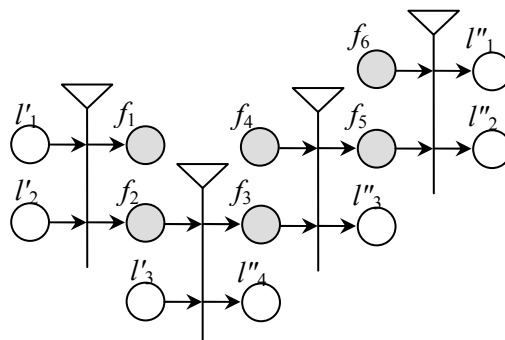
Нерешеният проблем е: да се конструират алгоритми за представяне на произволен преход от вида $H_{m,n}$ посредством преходи от вида $H_{2,2}$.



Фигура 2.1. Преход от вида $H_{m,n}$ (a) и специалният му случай, преход от вида $H_{2,2}$ (b)

К. Атанасов и Е. Димитров показват, че всяка Е-мрежа може да бъде представена като композиция от преходи, принадлежащи само към типа $H_{2,2}$. Впоследствие, К. Атанасов показва, че този резултат е в сила и в случая на ОМ. Той разглежда три конфигурации от композициите на преходи от типа $H_{2,2}$. В. Тасева разглежда тези и конструира нови подобни конфигурации и им дава имената, които се използват и в настоящия дисертационен труд, а именно: **X**, **V**, **A**, **I** и **\A**-конфигурации.

Да започнем с един пример. Да разгледаме преход от типа $H_{3,4}$, т.е. такъв като показания на Фиг. 2.1. (a) при $m = 3$ и $n = 4$. Представянето му посредством композиция само от $H_{2,2}$ преходи е показано на Фиг. 2.3., на която при представянето с $H_{2,2}$ преходи на прехода $H_{3,4}$ се появяват три нови прехода и шест нови позиции, отбелязани в сиво, които до този момент не са съществували. Позициите са наречени „фиктивни“ и отбелязани с f_1, \dots, f_6 . Тези „фиктивни“ позиции са въведени само и единствено с цел да се запазят изискванията на нотацията на $H_{2,2}$ преходите. Можем да отбележим още, че тази конфигурация е от типа **V**.



Фигура 2.3. Представяне на един $H_{3,4}$ преход с четири $H_{2,2}$ прехода

В. Тасева въвежда следните две функции: $T(\mathbf{S}, m, n)$ и $P(\mathbf{S}, m, n)$, които изчисляват броя на преходите и „фиктивните“ позиции на дадена конфигурация, представена с $H_{2,2}$ преходи. \mathbf{S} тук отговаря на произволен символ от множеството $\{\mathbf{X}, \mathbf{V}, \mathbf{\Lambda}, l, \backslash\}$. Тасева доказва, че за $m, n > 2$:

$$\begin{aligned} T(\mathbf{X}, m, n) &= T(\mathbf{V}, m, n) = T(\mathbf{\Lambda}, m, n) \\ &= T(l, m, n) = T(\backslash, m, n) = m + n - 3 \end{aligned} \quad (1)$$

и

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) \\ &= P(l, m, n) = P(\backslash, m, n) = 2m + 2n - 8. \end{aligned}$$

Тя показва още, че броят на „фиктивните“ позиции може да бъде редуциран, ако се приложи обединение на позиции. Това обединение може да се приложи само към двойки, съставени от една „фиктивна“ изходна позиция и една „фиктивна“ входна такава. В примера по-горе, такива обединения са възможни само за позиции f_1 и f_4 или между f_1 и f_6 , съответно. Ако се приложи такава оптимизация, броят на преходите остава същият, докато броят на „фиктивните“ позиции се свежда до:

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) \\ &= P(l, m, n) = P(\backslash, m, n) = 2m + n - 6, \end{aligned} \quad (2)$$

където $m \geq n$. Следователно, за всяко $n > 2$ новото представяне съдържа по-малко „фиктивни“ позиции.

В дисертационния труд е показано, че m и n участват във функциите T и P симетрично, с което формулата на Тасева в (2) се подобрява до:

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) = P(l, m, n) \\ &= P(\backslash, m, n) = 2 \cdot \max(m, n) + \min(m, n) - 6. \end{aligned} \quad (3)$$

В дисертационния труд е предложена нова конфигурация за представянето с $H_{2,2}$ преходи. Новостта на представения подход е в наличието на дъги за обратна връзка, които допринасят и за по-стегнатия вид на модела. Ще наричаме това ново представяне **F**-представяне (F от “feedback”, „обратна връзка“).

Да започнем със случая когато $m = n \geq 3$. Следователно, един $H_{m,n}$ преход като показания на Фиг. 2.1. (а) може да бъде **F**-представен като композиция от $H_{2,2}$ преходи, както е показано на Фиг. 2.4.

Веднага се вижда, че за това **F**-представяне броят на преходите и броят на „фиктивните“ позиции са равни съответно на:

$$T(\mathbf{F}, m, m) = m, \quad (4)$$

$$P(\mathbf{F}, m, m) = m. \quad (5)$$

Формулирана и доказана е следната теорема.

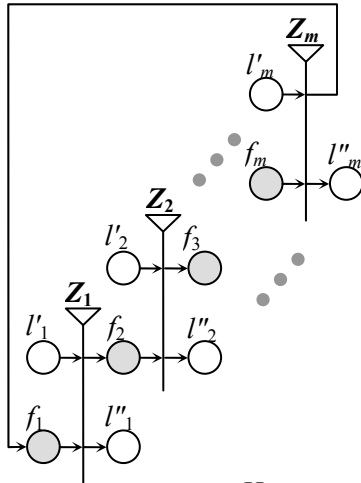
Теорема: За всеки две естествени числа m, n е в сила, че:

$$T(\mathbf{F}, m, n) = \max(m, n), \quad (6)$$

$$P(\mathbf{F}, m, n) = 2 \cdot \max(m, n) - \min(m, n), \quad (7)$$

и това са минималните брой преходи и брой „фиктивни“ позиции, необходими за **F**-представянето.

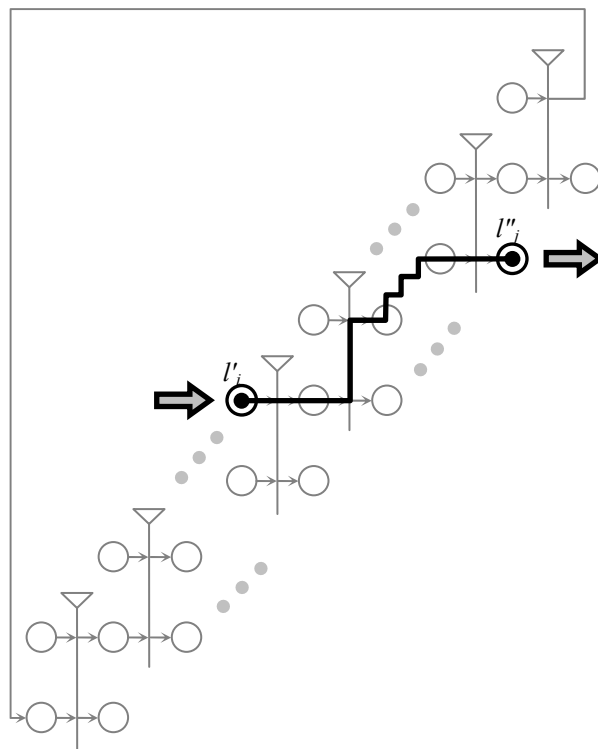
Нека допуснем, че $m \geq n$ и нека $m = n + k$. Случаят $k = 0$ е илюстриран на Фиг. 2.4. Веднага се вижда, че в този случай равенства (4) и (5) съвпадат с равенства (6) и (7), съответно.



Фигура 2.4. **F**-представяне на един $H_{m,m}$ преход с $H_{2,2}$ преходи

Теоремата се доказва по индукция.

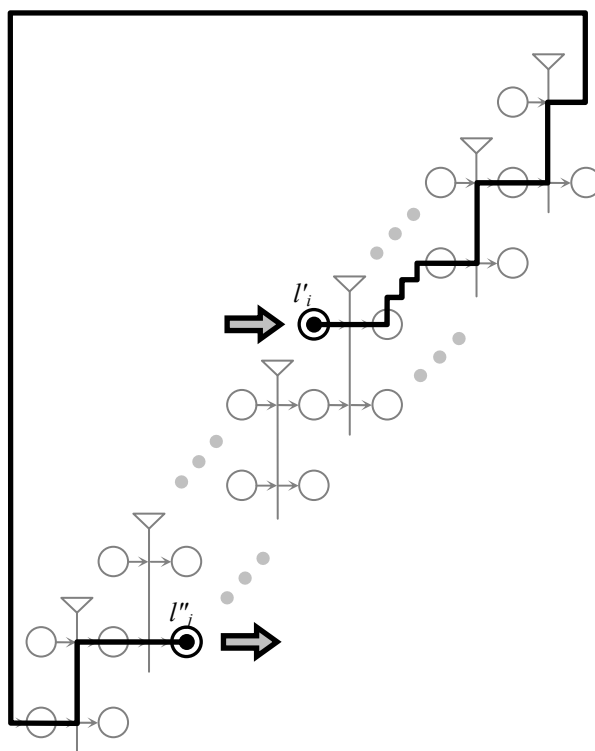
Ще отбележим само, че преходите в **F**-представянето осигуряват, че движението на ядрата при него напълно съответства на движението на ядрата в оригиналния $H_{m,n}$ преход. Този факт добре се илюстрира от Фиг. 2.8. и 2.9. Двете демонстрират маршрутите на ядро, което влиза през i -тата входна позиция на $H_{m,n}$, т.е., l'_i , и напуска през неговата j -та изходна позиция l''_j .



Фигура 2.8. Движение на ядро от l'_i към l''_j когато $i \leq j$

Фиг. 2.8. разглежда случая, когато $i \leq j$, а случаят, при който $i > j$, е илюстриран на Фиг. 2.9. При него наличието на дъги за обратна връзка ефективно се използва, за да се насочи ядро от i -тата входна позиция към крайната изходна позиция на m -тия преход, оттам обратно към първия преход на композицията, и най-сетне към j -тия изход. Това е и

принципната разлика между нашия подход и досега съществуващите подходи. Благодарение на нея, настоящата конструкция съдържа по-малък брой преходи и по-малък брой „фиктивни“ позиции. В допълнение, по индукция се показва, че този брой е минималният възможен.



Фигура 2.9. Движение на ядро от l'_i към l''_j когато $i > j$

В [74] е публикувана основната част от този текст, разширена тук с граничните случаи, когато $m \in \{1, 2\}$ или $n \in \{1, 2\}$.

2.2. Обобщени мрежи с ядра с обеми

Съществуват реални ситуации, при които процеси, моделирани с апарата на ОМ, изискват ОМ-ядрата, които са динамичните носители на информация за модела, да проявяват някои допълнителни специфични характеристики като „обем“, „размер“ или „тегло“. Когато се конструира един ОМ-модел, ядрата получават разнообразни характеристики в числово-буквен вид, отразяващи специфични аспекти на моделирания процес, и те често представляват записи в база данни. Конкретните стойности на тези характеристики на ядрата обикновено нито повлияват, нито са повлияни от статичните компоненти на мрежата (позиции, преходи или дъги) или съветните им свойства като приоритет или капацитет.

Разгледана е примерна ситуация, при която ядрата на ОМ проявяват свойства, които имат отношение към свойства на ОМ-инфраструктурата, което води до необходимостта от дефиниране на ново разширение на понятието ОМ.

Тъй като новопредложеното разширение на ОМ, наречено **обобщена мрежа с ядра с обеми** (ОМЯО, *generalized net with volumetric tokens, GNVT*) изисква модификация на дефиницията на ОМ преход, ще дадем общата дефиниция, както е дадена от К. Атанасов, съпоставяйки я със статията на автора, [71], и поставяйки специален акцент върху разликите (отбелязано в сиво).

Формално, всеки преход, както на ОМ, така и на ОМЯО, се задава чрез наредена седморка от вида:

$$Z = \langle L', L'', t_1, t_2, R, S, \square \rangle,$$

където:

- (a) $L' = \{l'_1, \dots, l'_i, \dots, l'_m\}$ е крайно непразно множество от входните позиции на прехода;
- (b) $L'' = \{l''_1, \dots, l''_j, \dots, l''_n\}$ е крайно непразно множество от изходните позиции на прехода;
- (c) t_1 е текущ момент на активиране на прехода;
- (d) t_2 е текуща стойност на продължителността на активното състояние на прехода;
- (e) R е условие на прехода, задаващо се чрез индексирана матрица от вида:

$$R = \begin{array}{c|cccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & r_{i,1} & \dots & r_{i,j} & \dots & r_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & r_{m,1} & \dots & r_{m,j} & \dots & r_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$, където r_{ij} е предикат, съответстващ на i -та входна позиция на прехода и j -та изходна позиция на прехода.

- (f) S е индексирана матрица на капацитети на дъгите във вида

$$S = \begin{array}{c|cccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & s_{1,1} & \dots & s_{1,j} & \dots & s_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & s_{i,1} & \dots & s_{i,j} & \dots & s_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & s_{m,1} & \dots & s_{m,j} & \dots & s_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$, където $s_{i,j}$ ($1 \leq i \leq m$, $1 \leq j \leq n$) е естествено число, по-голямо или равно на нула или знака за безкрайност, ∞ , и означава броя ядра, които могат да преминат през дъгата между i -тата входна и j -тата изходна позиция в рамките на едно активиране на прехода. Този компонент от дефиницията съдържа разликите между стандартната ОМ и новата ОМЯО. В стандартните ОМ, винаги когато едно ядро премине през дадена дъга, чийто капацитет $s_{i,j} \neq \{0; \infty\}$, този капацитет намалява с 1. В случая с ОМЯО капацитетът на дъгата намалява с числовата стойност, изразяваща характеристиката „обем“ на ядрото. Можем, следователно, да смятаме, че стандартните ОМ ядра в ОМЯО са ядра с обем, равен на 1.

- (g) \square е булев израз, задаващ типа на прехода.

Продължаваме с формалната дефиниция на ОМЯО. Тя, макар и с малко, се различава от стандартната дефиниция на ОМ. Наредената четворка

$$E_{VT} = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, K_{VT}, \pi_K, \theta_K, V_{VT} \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle,$$

се нарича ОМЯО, ако:

- (a) A е множество от всички преходи в мрежата.
- (b) π_A е функция, задаваща приоритетите на преходите.
- (c) π_L е функция, задаваща приоритетите на позициите.
- (d) c е функция, задаваща капацитетите на позициите.
- (e) f е функция, която изчислява вярностната стойност на предикатите в условието на прехода.
- (f) θ_1 е функция, задаваща следващия момент от време, когато даден преход Z може да бъде активиран.
- (g) θ_2 е функция, която задава продължителността на активното състояние на даден преход Z .
- (h) K е множеството от стандартните ядра в ОМ.
- (i) K_{VT} в новопредложената модификация е множеството на ядрата на ОМЯО, които проявяват характеристиката “обем”.
- (j) π_K е функция, която задава приоритетите на ядрата.
- (k) θ_K е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа.
- (l) V_{VT} в новопредложената модификация е функцията, която присвоява на ядрата от множеството K_{VT} обеми, които са естествени числа, и стойност 1 на ядрата от множеството K (като това важи и за E , и за E_{VT}).
- (m) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала.
- (n) t° е елементарната времева стъпка на фиксираната времева скала.
- (o) t^* е продължителност на функционирането на обобщената мрежа.
- (p) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата.
- (q) Φ е характеристична функция, която задава нова характеристика на всяко ядро при придвижването му от входна към изходна позиция на даден преход.
- (r) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа.

Новопредложената модификация на понятието ОМ като ОМЯО изисква от нас да представим и надлежно модифицирания алгоритъм за функциониране на отделен ОМ преход в даден момент от време.

Отново, по-удобно е да представим и сравним двата алгоритъма за случая на стандартна ОМ и модифицирания случай, предложен от автора в [71] (даден в сиво). Като цяло, разликата се състои във факта, че в случая на стандартна ОМ преминаването на ядрото през прехода води до намаляване на капацитетите на изминатата дъга и на изходната позиция с 1, докато в случая с ОМЯО тези капацитети намаляват с числото, представляващо обема на придвиженото ядро.

За тази цел ще използваме дефиницията на функцията $\bar{c}(l, time)$, която дава броя на текущо наличните ядра в позиция l в момент от време $time$.

Алгоритъм А за ОМЯО

- (A01) Входните и изходните позиции на прехода се подреждат по приоритет.
- (A02) Ядрата във входните позиции се подреждат в списък по приоритет. Във всяка позиция има по още един аналогичен списък, който на този етап е все още празен.
- (A03) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . На елементите на R^* се присвоява стойност „0“, ако те съответстват на:
- празна входна позиция, т.е. не съдържаща ядра, които потенциално могат да преминат до изходните позиции на прехода, следователно всички клетки от R^* по съответния ред имат стойност „0“;
 - пълна изходна позиция, т.е. нямаща капацитет да поеме ядра, които потенциално могат да пристигнат от входните позиции на прехода, следователно всички клетки от R^* по съответния стълб имат стойност „0“;
 - входна и изходна позиции с текущ капацитет на дъгата между тях равен на нула, т.е. клетката от R^* , намираща се на съответния ред и стълб има стойност „0“, т.е. от $s_{i,j} = 0$ следва $R^*_{i,j} = 0$.
- (A04) Започваме с входната позиция с най-висок приоритет, съдържаща най-малко едно ядро, което не е било придвижено към изходна позиция на текущата времева стъпка. Взимаме от нея ядрото с най-висок приоритет, да го означим с α . Последователно проверяваме за всяка от изходните позиции $l'' \in L''$ дали е в сила $V_{VT}(\alpha) > c(l'') - \bar{c}(l'', time)$, т.е. дали обемът на ядрото α е строго по-голям от текущо наличния капацитет на изходната позиция, изразено като разликата между максималния и текущо заетия капацитет на позицията. **Ако да**, тогава изходната позиция не може да приеме толкова обемно ядро (не и по време на текущо активното състояние на прехода), следователно, се присвоява стойност „0“ на съответната клетка от матрицата R^* . **В противен случай**, проверяваме дали текущият капацитет на дъгата между съответната входна и съответната изходна позиция е по-малък от $V_{VT}(\alpha)$. **Ако да**, присвояваме стойност „0“ на съответната клетка от матрицата R^* . **В противен случай**, ако ядрото може да се разцепва, проверяваме всички предикати от реда на матрица R , които съответстват на разглежданата позиция. Последователно в реда записваме 1 на местата, които съответстват на предикат, имащ вярностна стойност „истина“ и 0 на тези, чийто съответстващ им предикат има вярностна стойност „лъжа“. Ядрото от входната позиция се разцепва на толкова ядра, колкото са единиците с реда и новите ядра се преместват в съответните изходни позиции на прехода, като веднага получават своите нови характеристики. Ако ядрото във входната позиция не може да се разцепва, проверяваме вярностната стойност на предикатите в реда от ИМ R , съответстващ на входната позиция докато намерим първия предикат, имащ вярностна стойност „истина“. Ядрото отива в съответната му изходна позиция, където то получава нова характеристика. Ще отбележим, че изходната позиция, в която влиза ядрото е с най-високия приоритет сред позициите, в които то би могло да влезе.
- (A05) Ако за дадено ядро всички предикати имат вярностна стойност „лъжа“, това ядро остава във входната си позиция, но се премества във втория ѝ списък и то повече не участва в процесите по време на текущото активиране на прехода.

- (A06) Капацитетът на всяка изходна позиция, където са влезли ядра по силата на стъпка (A04), намалява със стойностите на обемите на ядрата. Когато максималният брой ядра, които могат да постъпят в разглежданата изходна позиция, бъде достигнат, тогава на всички клетки в съответния стълб на матрицата R^* се присвоява стойност „0“.
- (A07) Капацитетът на всяка дъга, през която преминава ядро, намалява със стойността на неговия обем. Когато капацитетът на дъгата стане 0, на клетката в матрицата R^* , която съответства на тази дъга, се присвоява стойност „0“.
- (A08) Ако има входна позиция с приоритет, по-нисък от този на горната, от която все още не са придвижени ядра, преминаваме към стъпка (A04). В противен случай отиваме към стъпка (A09).
- (A09) Текущата стойност t' на моделното време се увеличава с t° .
- (A10) Проверява се достигнат ли е момент $t_1 + t_2$.
- (A11) Ако отговорът на стъпка (A10) е „не“, се отива към стъпка (A04), ако е „да“ – край на функционирането на прехода.

Алгоритъмът за функциониране на цяла ОМЯО (наричан *Алгоритъм В*) е идентичен с алгоритъма за функциониране на стандартна ОМ, даден от К. Атанасов.

ОМ G се нарича *консервативно разширение на ОМ H* , ако по някои от своите компоненти (вид на ядрата, сложност на характеристиките, и т.н.) G представлява разширение на H , но резултатът от работата на G е идентичен с резултата от работата на H .

Ползата от въвеждането на консервативни разширения на ОМ се изразява във възможността с тях да се работи по-лесно и удобно в сравнение с ОМ, дефинирани по стандартната дефиниция. Доказана е:

Теорема: ОМЯО е консервативно разширение на ОМ, т.е. за всяка ОМЯО съществува една ОМ, която представя функционирането и резултатите на ОМЯО.

Новото разширение на понятието ОМ, по аналогия с някои други разширения като интуиционистки размитите ОМ (ИРОМ), предлага широки възможности за по-нататъшни интересни изследвания и приложения. Процесите от горния пример могат да бъдат по-добре моделирани с ОМЯО, отколкото със стандартна ОМ. Новото понятие може да се използва и за проектиране на нови варианти на други добре познати алгоритми, като метаевристичния оптимизационен алгоритъм на мравките. Връзките между ОМЯО и други ОМ разширения също ще са обект на бъдещи изследвания и дискусии.

2.3. Алгоритми за движение на ядра в интуиционистки размити обобщени мрежи

Интуиционистки размитите множества (ИРМ, *intuitionistic fuzzy sets, IFSs*), дефинирани през 1983 г., са разширения на размитите множества. Те имат две степени: на принадлежност (μ) и на непринадлежност (ν), такива, че сумата им може да е по-малка от 1, т.е. може да бъде дефинирана и трета степен на неопределеност $\pi = 1 - \mu - \nu$. Над ИРМ са дефинирани разнообразие от операции, релации и оператори от модален, топологичен и др. тип.

Първите две разширения на ОМ, предложени през 1985 г., са наречени *интуиционистки размити обобщени мрежи* (ИРОМ, *intuitionistic fuzzy generalized*

nets, IFGNs) от първи и втори тип. През 2001 г. са описани и още две разширения. Тук се дават подробно алгоритмите за движение на ядра в преход и в цяла мрежа за всеки от четирите типа мрежи.

На всяко съждение (в класическия смисъл) можем да присвоим вярностна стойност: „истина“ – означавана с 1, или „лъжа“ – с 0. В случая с размитата логика вярностната стойност може да е реално число от интервала $[0, 1]$ и може да се нарече „степен на вярност“ на дадено съждение. В случая на ИРЛ може да бъде добавена още една стойност, „степен на невярност“, която също е реално число в интервала $[0, 1]$. Така две реални числа, $\mu(p)$ и $\nu(p)$, се присвояват на съждението p , като е в сила и следното условие:

$$\mu(p) + \nu(p) \leq 1.$$

Степента на неопределеност се дефинира като

$$\pi(p) = 1 - \mu(p) - \nu(p).$$

Нека над множество от съждения S е дефинирана оценъчна функция f (по аналогия с т. (е) от дефиницията на ОМ), така че:

$$f(p) = \langle \mu(p), \nu(p) \rangle.$$

В дисертационния труд са дадени дефинициите за преход на ИРОМ-1 и на цяла ИРОМ-1. Следвайки статия [62], ще въведем алгоритмите за движението на ядрата в ИРОМ-1 на ниво преход. В сиво са маркирани новите стъпки в алгоритмите.

Алгоритъм А за ИРОМ-1

- (A01) Входните и изходните позиции на прехода се подреждат по приоритет.
- (A02) Ядрата във входните позиции се подреждат в списък по приоритет. Във всяка позиция има по още един аналогичен списък, който на този етап е все още празен.
- (A03) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . Присвоява се стойност „ $\langle 0, 1 \rangle$ “ (отговаряща на стойността „лъжа“), на всички елементи от индексираната матрица, които се намират:
 - (a) в ред, отговарящ на празна входна позиция; или
 - (b) в стълб, отговарящ на запълнена изходна позиция; или
 - (c) в клетка (i, j) , за която $s_{i,j} = 0$, т.е. текущият капацитет на дъгата между i -тата входна и j -тата изходна позиция е нула.
- (A04) Сортираните позиции се обхождат последователно, по реда на приоритета им, започвайки с позицията с най-висок приоритет, която има поне едно ядро и през която не е осъществен преход на текущата времева стъпка. За ядрото с най-висок приоритет (от първия списък) определяме дали то може да се разцепва или не. Този факт се определя от динамичния оператор, дефиниран над конкретната ОМ, и в частност над дадената ИРОМ-1. Ако не е дефиниран такъв оператор, ще приемем, че ядрото може да се разцепва толкова пъти, колкото е необходимо. След това се проверяват предикатите, отговарящи на съответния ред от индексираната матрица R . Ако ядрото не може да се разцепва, проверката приключва с откриването на първия предикат с вярностна стойност, различна от „ $\langle 0, 1 \rangle$ “. В противен случай, трябва да изчислим вярностните стойности на всички предикати в реда, за които елементите на R^* са различни от нула.

(A05) В зависимост от изпълнението на оператора за позволяване или забраняване на разцепването на ядрата в мрежата, ядрото от стъпка **(A04)** ще премине или до всички изходни позиции, където му е позволено да премине, или до точно онази от тях, която има най-висок приоритет, следвайки едно от седемте условия по-долу, където за предиката $r_{i,j}$ е в сила, че $f(r_{i,j}) = \langle \mu_{i,j}, \nu_{i,j} \rangle$.

C1 $\mu_{i,j} = 1, \nu_{i,j} = 0$ (случаят на стандартната ОМ);

C2 $\mu_{i,j} > \frac{1}{2} (> \nu_{i,j})$;

C3 $\mu_{i,j} \geq \frac{1}{2} (\geq \nu_{i,j})$;

C4 $\mu_{i,j} > \nu_{i,j}$;

C5 $\mu_{i,j} \geq \nu_{i,j}$;

C6 $\mu_{i,j} > 0$;

C7 $\nu_{i,j} < 1$, т.е. поне $\pi_{i,j} > 0$.

Кое условие ще се използва, зависи от желанието на потребителя. Условието се задава за всеки преход преди активирането на ИРОМ-1. Ако едно ядро не може да премине през даден преход на тази времева стъпка, ядрото се премества във втората група ядра на съответната входна позиция. Ядрата, които се влезли в позицията след активирането на прехода, също се преместват във втората група.

(A06) Стойностите на характеристичната функция на изходните позиции (една или повече), в които са влезли ядра (съгласно стъпка **(A05)**), се изчисляват и следният запис се присвоява като следваща характеристика на ядрото:

„стойност на функцията Φ за съответното ядро, $\mu_{i,j}, \nu_{i,j}$ “

(A07) Присвояваме стойностите „ $\langle 0, 1 \rangle$ “:

- (a) на всички редове на R^* , за които съответната изходна позиция (откъдето ядрото излиза на стъпка **(A05)**) е вече празна; или
- (b) на всички стълбове в R^* , които са запълнени в резултат на придвижването на ядрата на стъпка **(A05)**; или
- (c) в онези клетки от индексирания матрица, които отговарят на дъги между дискутираните входна и изходни позиции, за които капацитетите на дъгите са станали нула в резултат от придвижването на ядрата.

(A08) Текущият брой на ядрата във всички входни позиции за текущия преход намалява с 1 за всяко ядро, което ги е напуснало на тази времева стъпка. Ако текущият брой на ядрата за дадената входна позиция е нула, клетките от съответният ред от индексирания матрица R^* получават стойност „ $\langle 0, 1 \rangle$ “.

(A09) Капацитетите на всички изходни позиции, в които е постъпило ядро по силата на стъпка **(A04)**, намаляват с 1. Ако е достигнат максималният брой ядра за дадена изходна позиция, клетките от съответния стълб на индексирания матрица R^* стават „ $\langle 0, 1 \rangle$ “.

(A10) Капацитетите на всички дъги, през които е преминало ядро, намаляват с 1. Ако капацитетът на дъга е достигнал нула, на съответната клетка от индексирания матрица R^* , която отговаря на тази дъга, се присвоява стойност „ $\langle 0, 1 \rangle$ “.

(A11) Ако продължава да има ядра във входните позиции, които подлежат на трансферирание, и има свободни изходни позиции и има дъги с ненулеви капацитети,

то алгоритъмът продължава със стъпка (A12); в противен случай продължава със стъпка (A14).

- (A12) Текущото моделно време t се увеличава с t° .
- (A13) Текущият момент от време равен или по-голям ли е от $t_1 + t_2$? Ако отговорът е „не“, връщане към стъпка (A04), в противен случай алгоритъмът продължава към стъпка (A14).
- (A14) Край на функционирането на прехода.

В *интуиционистки размитите обобщени мрежи от втори тип* (ИРОМ-2, по дефиниция ядрата са заменени от някакви количества вещество, които „протичат“ в мрежата. Това обстоятелство налага някои разлики между определенията на ИРОМ-1 и ИРОМ-2. В дисертационния труд са показани тези разлики.

По-долу ще приведем само точките от дефиницията, които се различават от тези на стандартната ОМ:

- (d) c е функция, даваща капацитетите на позициите, т.е. $c : L \rightarrow \mathcal{R}$, където \mathcal{R} е множеството на реалните (а не естествените) числа.
- (h) K е множеството на ядрата в мрежата, които тук представляват количества вещество.
- (o) Φ е характеристичната функция, която присвоява нови характеристики на всяка позиция, когато ядро извърши преход от изходна позиция към нея, и както е в случая с ИРОМ-1, добавя същата характеристика към степените на вярност и невярност на предиката, който разрешава или не преминаването на ядрото през прехода.
- (p) b е функция, даваща максималния брой характеристики, които дадена позиция (а не ядро) може да получи.

Всички операции и релации, които се дефинират над ОМ, могат да се дефинират и над ИРОМ-2.

В дисертационния труд за първи път се дава формалния алгоритъм за ИРОМ-2. Той, разбира се, е базиран на алгоритмите за трансфер на ядра в стандартни ОМ и ИРОМ-1, но сега има някои съществени разлики и първата от тях е, че преминаването на ядрата се извършва само на една стъпка, но с продължителност, определена от втория темпорален компонент на прехода, ако такъв съществува.

Общият алгоритъм (който ще означаваме с **Алгоритъм А**) за преминаване на ядра през преход в даден момент от време $t_1 = \text{TIME}$ (текущия момент от време на ИРОМ-2) е, както следва.

Алгоритъм А за ИРОМ-2

- (A01) Непразните входни позиции и незапълнените изходни позиции се сортират по приоритет в намаляващ ред. За разлика от **Алгоритъм А за ИРОМ-1**, всяка позиция може да съдържа само определено количество вещество, което е аналог на единично ядро.
- (A02) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . Присвоява се стойност „0,1“ (отговаряща на стойността „лъжа“), на всички елементи от индексираната матрица, които се намират:
 - (a) в ред, отговарящ на празна входна позиция; или

- (b) в стълб, отговарящ на запълнена изходна позиция; или
- (c) в клетка (i, j) , за която $s_{i,j} = 0$, т.е. текущият капацитет на дъгата между $i^{-\text{тата}}$ входна и $j^{-\text{тата}}$ изходна позиция е нула.

(A03) Сортираните позиции се обхождат последователно по приоритета им, започвайки от позицията с най-висок приоритет, която съдържа някакво количество вещество. Във всяка от обходените позиции, се проверяват предикатите, отговарящи на съответния ред от индексиранията матрица R^* . Ако количеството вещество не може да се раздели, проверката приключва с намирането на първия предикат, чиято вярностна стойност е различна от „ $\langle 0, 1 \rangle$ “; в противен случай трябва да изчислим вярностните стойности на всички предикати на реда, за които елементите на R^* са различни от различни от „ $\langle 0, 1 \rangle$ “.

(A04) В зависимост от изпълнението на оператора за позволяване или забраняване на разцепването на ядрата в мрежата, ядрото от стъпка **(A03)** ще премине или до всички позволени за него изходни позиции, или до точно онази от тях, която има най-висок приоритет, следвайки едно от следните седем условия, където за предиката $r_{i,j}$ е в сила, че $f(r_{i,j}) = \langle \mu_{i,j}, \nu_{i,j} \rangle$.

C1 $\mu_{i,j} = 1, \nu_{i,j} = 0$ (случаят на стандартната ОМ);

C2 $\mu_{i,j} > \frac{1}{2} (> \nu_{i,j})$;

C3 $\mu_{i,j} \geq \frac{1}{2} (\geq \nu_{i,j})$;

C4 $\mu_{i,j} > \nu_{i,j}$;

C5 $\mu_{i,j} \geq \nu_{i,j}$;

C6 $\mu_{i,j} > 0$;

C7 $\nu_{i,j} < 1$, т.е. поне $\pi_{i,j} > 0$.

Условието, което ще бъде използвано, се определя за всеки преход преди началото на активирането на ИРОМ-2.

За разлика от предния алгоритъм, тук е в сила следното специфично изискване:

$$\sum_j \mu_{i,j} \leq 1.$$

Причината да се формулира това условие е, че количеството вещество в $i^{-\text{тата}}$ входна позиция ще бъде доставено до $j^{-\text{тата}}$ изходна позиция в зависимост от вярностната стойност $\mu_{i,j}$. В $i^{-\text{тата}}$ входна позиция ще остане толкова количество вещество, колкото отговаря на степента на невярност $\nu_{i,j}$. По дъгата, свързваща $i^{-\text{тата}}$ входна с $j^{-\text{тата}}$ изходна позиция, ще се задържи останалото количество вещество, отговарящо на допълнението

$$\pi_{i,j} = 1 - \mu_{i,j} - \nu_{i,j}.$$

(A05) Стойностите на характеристичната функция на изходните позиции (една или повече), в които са влезли ядра (съгласно стъпка **(A04)**), се изчисляват и следният запис се присвоява като следваща характеристика на ядрото:

„стойност на функцията Φ за съответното ядро, $\mu_{i,j}, \nu_{i,j}$ “

(A06) Присвояваме стойностите „ $\langle 0, 1 \rangle$ “:

- (a) на всички редове на R^* , за които съответната изходна позиция (откъдето ядрото излиза на стъпка **(A05)**) е вече празна; или
- (b) на всички стълбове в R^* , които са запълнени в резултат на придвижването на ядрата на стъпка **(A05)**; или

- (с) в онези клетки от индексиранията матрица, които отговарят на дъги между дискутираните входна и изходни позиции, за които капацитетите на дъгите са станали нула в резултат от придвижването на ядрата.

- (A07)** Текущото количество вещество във всяка входна позиция за текущия преход намалява с количествата вещество, отговарящи на сумата $\sum_j \mu_{i,j}$. Ако капацитетът на позицията стане нула, то в клетките на съответния ред от индексиранията матрица R^* се записва стойност „ $\langle 0, 1 \rangle$ “.
- (A08)** Текущото количество вещество във всяка изходна позиция за текущия преход се увеличава с количествата вещество, отговарящи на сумата $\sum_i \mu_{i,j}$. Ако максималният капацитет на позицията бъде достигнат, то в клетките на съответния стълб от индексиранията матрица R^* се записва стойност „ $\langle 0, 1 \rangle$ “.
- (A09)** Капацитетите на всички дъги, свързващи i -тата входна с j -тата изходна позиция, през които е преминало вещество, намаляват пропорционално на сумата $\mu_{i,j} + \pi_{i,j}$. Ако максималният капацитет на дъгата бъде достигнат, в клетката от индексиранията матрица R^* , която отговаря на тази дъга, се записва стойността „ $\langle 0, 1 \rangle$ “.
- (A10)** Ако продължава да има ядра във входните позиции, които подлежат на трансфериране, и има свободни изходни позиции и има дъги с ненулеви капацитети, то алгоритъмът продължава със стъпка **(A11)**, в противен случай продължава със стъпка **(A13)**.
- (A11)** Текущото моделно време t се увеличава с t° .
- (A12)** Текущият момент от време равен или по-голям ли е от $t_1 + t_2$? Ако отговорът е „не“, връщане към стъпка **(A03)**, в противен случай алгоритъмът продължава към стъпка **(A13)**.
- (A13)** Край на функционирането на прехода.

Общият алгоритъм за функциониране на ИРОМ-2 съвпада с алгоритъма за функциониране на ИРОМ-1.

Представени са ИРОМ от трети (ИРОМ-3) и от четвърти вид (ИРОМ-4). Те са разширения съответно на ИРОМ-1 и ИРОМ-2. Тук, характеристиките на ядрата или, съответно, позициите ще бъдат оценявани в смисъла на интуиционистки размитите множества, т.е. те ще получават за стойности ИР двойки, които представляват степените на вярност и невярност (валидност и невалидност) на характеристиките, т.е., тези характеристики тук имат вида

$$x_{ci}^\alpha = \left\langle \bar{x}_{ci}^\alpha, \mu(r_{i,j}), \nu(r_{i,j}), \mu(x_{ci}^\alpha), \nu(x_{ci}^\alpha) \right\rangle,$$

където \bar{x}_{ci}^α е стандартната характеристика на ядрото.

В ИРОМ-4 ядрата представляват някакви количества вещество, което се движи из мрежата.

Стойностите на предикатите в условието на прехода могат да бъдат ИР двойки, т.е. тези предикати могат да се характеризират със степени на вярност и невярност, както в случаите на ИРОМ-1 и ИРОМ-3.

Функциите в първия компонент на ИРОМ-4 – $\pi_A, \pi_L, c, f, \theta_1, \theta_2$ – са сходни с тези в ИРОМ-2 и удовлетворяват същите условия.

Съществената разлика между ИРОМ-2 и ИРОМ-4 от една страна и останалите видове ОМ, от друга, е множеството K на ОМ-ядра и функциите, свързани с него. Тук

елементите на K са някакви “количества”, които имат само начални характеристики и не получават текущи по време на функционирането на мрежата. Функцията θ_K определя моментът от време, когато дадено ядро ще постъпи в мрежата, както при стандартните ОМ. Темпоралните компоненти също са както при другите ОМ.

За двата нови вида ОМ, ИРОМ-2 и ИРОМ-4, функцията Φ има нов смисъл, подобен на този в ИРОМ-3: тя присвоява на всяка позиция следната характеристика: количеството ядра в съответната позиция в различните моменти от време и степените на валидност и невалидност на тази характеристика. Както при ОМ, така и тук моделът може да бъде разширен: може да даде и други данни за моделирания процес (например моментите от време на постъпване на „количествата вещество“ в позициите).

Подобно на ИРОМ-2 и ИРОМ-3, функцията Φ в ИРОМ-4 също се оценява. При все това, тук за всяка позиция p , тази функция се оценява с две реални числа $\mu(x_{cu}^p)$ и $\nu(x_{cu}^p)$, такива че $\mu(x_{cu}^p), \nu(x_{cu}^p) \in [0, 1]$ и $\mu(x_{cu}^p) + \nu(x_{cu}^p) \leq 1$. Така горните условия имат вида:

- C1 $\mu(x_{cu}^p) = 1, \nu(x_{cu}^p) = 0$ (случаят на стандартната ОМ);
- C2 $\mu(x_{cu}^p) > \frac{1}{2} (> \nu(x_{cu}^p))$;
- C3 $\mu(x_{cu}^p) \geq \frac{1}{2} (\geq \nu(x_{cu}^p))$;
- C4 $\mu(x_{cu}^p) > \nu(x_{cu}^p)$;
- C5 $\mu(x_{cu}^p) \geq \nu(x_{cu}^p)$;
- C6 $\mu(x_{cu}^p) > 0$;
- C7 $\nu(x_{cu}^p) < 1$, т.е. поне $\pi(x_{cu}^p) > 0$.

и позициите ще получат характеристики, които имат вида

$$x_{cu}^p = \langle \bar{x}_{cu}^p, \mu(r_{i,j}), \nu(r_{i,j}), \mu(x_{cu}^p), \nu(x_{cu}^p) \rangle,$$

където \bar{x}_{cu}^p е характеристиката на позицията от ИРОМ-2.

2.4. Интуиционистки размити обобщени мрежи с интуиционистки размити капацитети на позициите

Както се посочва и в т. 2.3. досега са конструирани четири типа ИРОМ. Тук, следвайки [58], ще бъде предложен нов, пети, тип ИРОМ. За краткост ще го означим с ИРОМ-5. Тези нови ИРОМ ще съдържат позиции, имащи ИР капацитети.

Като основа на настоящата дефиниция отново ще използваме определението на стандартната ОМ. На съответното място ще отворим (със сив фон) разликите между стандартната и новата дефиниция. Промяната в стандартната дефиниция е малка, но тя повлиява вида на някои други ОМ параметри, а също и на Алгоритъма за движение на ядра в даден преход, и ще ги обсъдим по-долу.

Наредената четворка

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^o, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

се нарича ИРОМ-5, ако:

- (a) A е множество от всички преходи в мрежата;
- (b) π_A е функция, задаваща приоритетите на преходите, т.е., $\pi_A: A \rightarrow \mathcal{N}$, където $\mathcal{N} = \{0, 1, 2, \dots\} \cup \{\infty\}$;

(с) π_L е функция, задаваща приоритетите на позициите, като L е множеството от всички позиции на обобщената мрежа; $\pi_L: L \rightarrow \mathcal{N}$, където $L = \text{pr}_1A \cup \text{pr}_2A$ и с pr_iX е означена $i^{\text{-тара}}$ проекция на n -мерното множество X , $n \in \mathcal{N}$, $n \geq 1$, $1 \leq k \leq n$;

(d) В дефиницията на стандартна ОМ, c е функция, задаваща капацитетите на позициите, т.е. $c: L \rightarrow \mathcal{N}$. Тук, за всяка позиция $p \in L$ се определя интуиционистки размитата тройка $\langle c(p), \mu_c(p), \nu_c(p) \rangle$. Смисълът ѝ е следният. Нека $\bar{c}(p)$ е текущият брой ядра, които стоят в позиция p , т.е. $c(p) \geq \bar{c}(p)$ и нека $[x]$ е цялата част от реалното положително число x . Сигурно е, че $[\mu_c(p) \cdot \bar{c}(p)]$ на брой ядра с най-високи приоритети ще имат възможността да напуснат позиция p , $[\nu_c(p) \cdot \bar{c}(p)]$ е броят ядра, които няма да имат възможност да напуснат позиция p при текущото активиране на прехода, а за останалите

$$\bar{c}(p) - [\mu_c(p) \cdot \bar{c}(p)] - [\nu_c(p) \cdot \bar{c}(p)] \geq 0$$

на брой ядра не е ясно дали ще напуснат или не позиция p . Тази промяна в дефиницията на стандартна ОМ ще се отрази върху дефиницията на θ_2 от т. (g) по-долу, и следователно върху стойностите на t_2 в т. (с) от дефиницията на стандартен ОМ-преход в Глава 1.

(e) f е функция, която изчислява вярностната стойност на предикатите в условието на прехода (за описаната тук ОМ нека функцията f има стойност „лъжа“ или „истина“, т.е. стойност от множеството $\{0; 1\}$;

(f) θ_1 е функция, задаваща следващия момент от време, когато даден преход Z може да бъде активиран, т.е. $\theta_1(Z, t) = t'$, където $\text{pr}_3Z = t$, $t' \in [T, T + t^*]$ и $t \leq t'$. Стойността на тази функция се изчислява в момента, когато преходът спре да функционира.

(g) В дефиницията на стандартна ОМ, θ_2 е функция, задаваща продължителността на активното състояние на даден преход Z , т.е. $\theta_2(Z, t) = t'$, където $\text{pr}_4Z = t \in [T, T + t^*]$ и $t' \geq 0$. Стойността на тази функция се изчислява, когато преходът започне да функционира. Тук за прехода Z , съдържащ позиция $p \in L$ от точка (d), $\theta_2(Z)$ ще бъде числото, за което трябва да важат следните неравенства:

$$\theta_2(Z, t) \leq \max_{p \in \text{pr}_1Z} [\bar{c}(p) \cdot \mu_c(p)], \quad (8)$$

$$\theta_2(Z, t) \leq \max_{p \in \text{pr}_1Z} (\bar{c}(p) - [\bar{c}(p) \cdot \nu_c(p)]). \quad (9)$$

Комбинацията на двете неравенства е коректна, защото за всяка позиция p

$$\bar{c}(p) - [\bar{c}(p) \cdot \mu_c(p)] - [\bar{c}(p) \cdot \nu_c(p)] \geq \bar{c}(p) - \bar{c}(p) \cdot \mu_c(p) - \bar{c}(p) \cdot \nu_c(p) \geq 0.$$

(h) K е множеството от ядрата на ОМ;

(i) π_K е функция, която задава приоритетите на ядрата т.е. $\pi_K: K \rightarrow \mathcal{N}$;

(j) θ_K е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа, т.е. $\theta_K(\alpha) = t$, където $\alpha \in K$, $t \in [T, T + t^*]$;

(k) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала;

(l) t° е елементарната времева стъпка на фиксираната времева скала;

(m) t^* е продължителност на функционирането на обобщената мрежа;

- (п) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата;
- (о) Φ е характеристична функция, която задава нова характеристика на всяко ядро при придвижването му от входна към изходна позиция на даден преход;
- (р) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа, $b : K \rightarrow \mathcal{N}$.

Когато ИРОМ-5 съдържа само част от гореописаните компоненти, тя се нарича редуцирана ИРОМ-5. Всички операции, релации и оператори над стандартни ОМ, могат да бъдат трансформирани за случая на ИРОМ-5.

Конструиранят алгоритъм за преминаване на ядра в рамките на един преход след моментът от време $t_1 = TIME$ (текущият момент от време в ИРОМ-5) е същият като за стандартна ОМ.

Както се вижда, ИРОМ-5 е разширение на стандартната ОМ, защото ако положим $\mu_c(p) = 1$ за всяка позиция p (и следователно, $\nu_c(p) = \pi_c(p) = 0$) и ако пропуснем ограничения (8) и (9), то ще получим стандартната ОМ.

Формулирана и конструктивно доказана е следната теорема.

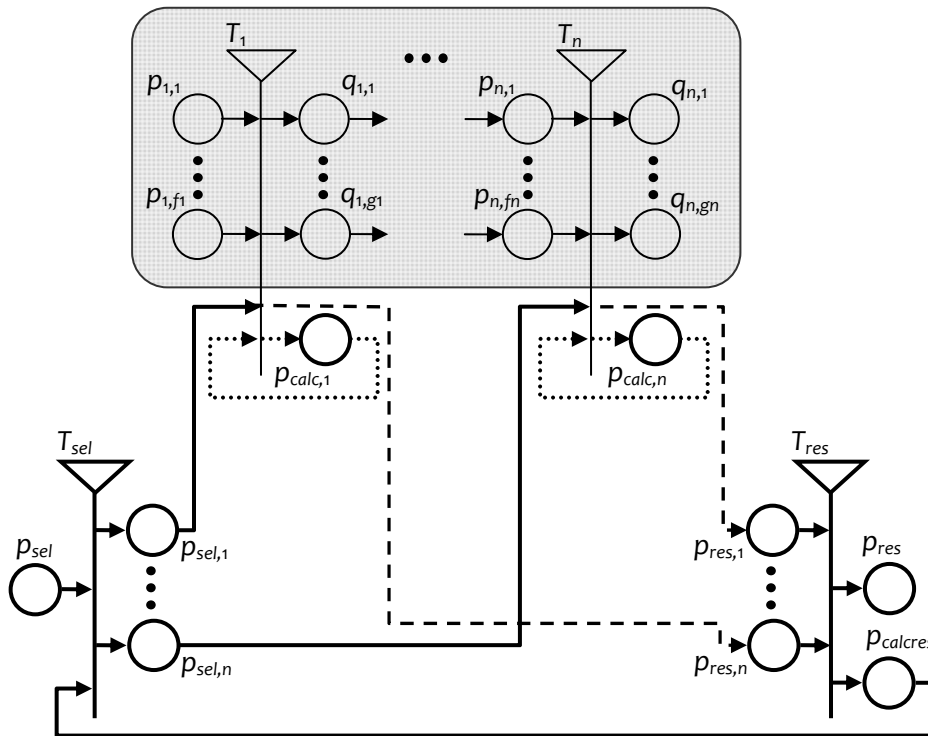
Теорема: За всяка ИРОМ-5 съществува стандартна ОМ, която описва функционирането и резултатите от нейната работа. И така двата типа ОМ са еквивалентни, т.е. ИРОМ-5 са консервативни разширения на ОМ. Накрая ще споменем, че всички съществуващи до момента ОМ-разширения са консервативни.

2.5. Интуиционистки размит подход към прогнозирането чрез обобщени мрежи

Нека имаме абстрактен модел на реален процес, конструиран с апарата на ОМ. Моделът се състои от n на брой преходи, T_1, \dots, T_n и множество от позиции, входни и изходни, съвкупно означавани с $p_{i,j}$ и $q_{i,k}$ където $i \in \{1, \dots, n\}$, $j \in \{1, \dots, f_i\}$, $k \in \{1, \dots, g_i\}$, където f_i и g_i съответно представляват броя на входните и на изходните позиции на i -тия преход. На Фиг. 2.15. абстрактният ОМ модел е схематично илюстриран в сиво и разширен с допълнителни инфраструктурни елементи, които отразяват новото в текущо представения модел.

Тези допълнителни инфраструктурни елементи са следните:

- два нови прехода T_{sel} и T_{res} ,
- $3n + 3$ нови позиции:
 - p_{sel} (входна позиция за прехода T_{sel} и за цялата ОМ),
 - $p_{sel,1}, \dots, p_{sel,n}$,
 - $p_{calc,1}, \dots, p_{calc,n}$ които са n позиции за междинни изчисления, прикачени към всеки от n -те прехода на оригиналната ОМ на моделирания процес,
 - $p_{res,1}, \dots, p_{res,n}$,
 - $p_{calcres}$,
 - p_{res} (изходна позиция за прехода T_{res} и за цялата ОМ).



Фигура 2.15. Предложеното разширение на абстрактния OM модел

Ще коментираме един важен аспект на нейните преходи и позиции: техните приоритети. Първото съображение, което трябва да се направи, е, че двата нови прехода T_{sel} и T_{res} имат, съответно, най-високия и най-ниския приоритети измежду всички преходи. Второто съображение е, че в цялата мрежа имаме общо четири нива на приоритет на позициите:

- На първото ниво позициите с най-висок приоритет са p_{sel} , $p_{sel,1}$, ..., $p_{sel,n}$ и $p_{calcres}$, т.е. входните и изходните позиции към прехода с най-висок приоритет T_{sel} .
- Второто ниво съдържа съвкупността от позициите на стандартната OM, която описва реалния процес, а именно позиции $p_{i,j}$ и $q_{i,k}$, а $i \in \{1, \dots, n\}$ е броят на преходите, и $j \in \{1, \dots, f_i\}$, $k \in \{1, \dots, g_i\}$. На Фиг. 2.15, това са позициите в оцветената в сиво област.
- Третото ниво съдържа позициите $p_{calc,i}$, $i \in \{1, \dots, n\}$, които представляват входове или изходи на прехода T_i .
- Четвъртото ниво съдържа позициите с най-нисък приоритет в новата мрежа, $p_{res,1}$, ..., $p_{res,n}$ и p_{res} , т.е. входните и изходните на прехода с най-нисък приоритет T_{res} .

Така описаните допълнителни елементи от статичната природа на OM формират нов контур за динамичните елементи (ядрата), които пренасят метаинформация относно оригинално моделирания процес. Да означим тези ядра с α , β и $\gamma_1, \dots, \gamma_n$.

Едно α -ядро влиза в глобалната входна позиция за новокопструирувания OM модел p_{sel} с характеристика текущия избор (*selection*) на потребителя за намерението му при пускането на модела: дали целта на пускането е извършване на симулация, или е с цел прогнозиране. Следователно, ядрото α влиза в позицията с точно една от следните две характеристики: „симулация“ или „прогнозиране“.

Този първи избор е от ключово значение за изпълнението на оригиналния модел, понеже го определя:

- дали информацията за преминаването на ядрата само ще бъде извличана, статистически обработвана и съхранявана в допълнителните позиции $p_{calc,1}$, ..., $p_{calc,n}$, $p_{calcres}$ ако потребителят е избрал да извърши симулация,

- или събраната информация от предишни симулации ще се използва (без промени) за пускане на модела с прогностична цел.

Така различното поведение на новата ОМ, предопределено от избора, направен в първата входна позиция, диктува, че тези два сценария трябва да бъдат разгледани поотделно. За начало са направени наблюдения относно времевата скала и елементарната времева стъпка на новия модел и са дадени някои нужни работни дефиниции.

Ще казваме, че ядро, намиращо се във входна позиция l_i , *подлежи* на трансфер към изходна позиция l_j винаги когато в (i, j) -тата клетка на индексирания матрица на предикатите на прехода стойността е или “истина” (безусловен преход), или някакъв предикат $W_{i,j}$, чиято вярностна стойност текущо се оценява на „истина“ (условен преход). Винаги когато дадено ядро подлежи на трансфер, то ще премине през прехода стига да разполага с нужното време на текущата времева стъпка. Тази допълнителна зависимост от времето означава, че някои ядра, които иначе подлежат на трансфериране през прехода, може да не успеят да завършат тази процедура в текущия момент от време, т.е. те ще останат във входната позиция до момента на следващата активация на прехода.

Както може веднага да се види, и множеството от „успешните“ ядра (да означим кардиналното му число с μ) и множеството на „неуспешните“ ядра (съответно с кардинално число ν) са подмножества на множеството на всички ядра във входните позиции на дадения преход (да означим кардиналното му число с τ). Онова, което не се вижда непосредствено, е че сумата на кардиналните числа на двете множества може и да не се допълва до τ . Възможно е някои ядра във входни позиции да подлежат на преминаване към изходни позиции (поради верността на изчислените предикати в индексирания матрица на прехода), но да не са могли да преминат поради ограниченото времетраене на активното състояние на прехода. Следователно,

$$0 \leq \frac{\tau - \mu - \nu}{\tau} \leq 1.$$

Тази ситуация прилича на случая на интуиционистки размитите степени на принадлежност (μ/τ), непринадлежност (ν/τ) и неопределеност (допълнението на първите две до 1).

Моделът е описан първо в сценария, при който той се пуска с цел *симулация*, а после – в сценария, при който се пуска с цел *прогнозиране*. Като общо правило са нужни голям брой симулации преди да се проведе проиграване на модела с цел прогноза, което да претендира за правдоподобност.

Сценарий 1: Симулация

Когато потребителят избере да стартира модела със симулационна цел, от него се извлича информация за поведението на оригинално моделирания процес. За целта разглеждаме няколко променливи, свързани с ядрата, циркулиращи в оригиналния ОМ модел. Първата променлива съответства на общия брой на ядрата (означили сме ги с τ), които стоят във входните позиции на всеки преход. Друга променлива съответства на броя на онези ядра, които успешно са извършили преминаването към изходни позиции по време на текущото активно състояние на съответния преход. Тези ядра, да ги наречем μ -ядра, не само са подлежали на трансфер, но и също са имали достатъчно време, за да го осъществят. Поддържаме също така и променлива за броя на ядрата, които не са подлежали на трансфер, по причина че изчислението на предикатите в съответните клетки от индексирания матрица на предикатите на прехода са дали в резултат стойност „лъжа“. Избираме да ги означаваме с ν . Тези τ -, μ - и ν -ядра:

- се преброяват постъпково в позиции $p_{calc,i}$ за всеки преход T_i ($i \in \{1, \dots, n\}$) на оригинално моделирания процес, за всяко активизиране на преходите в течение на дадената симулация, и

- се събират в позиция $p_{calcres}$ в края на всяка симулация. Дотогава докато моделът се пуска за симулационни цели, информацията от оригиналната ОМ ще се събира в позиция $p_{calcres}$.

По-формално, видът на първия преход T_{sel} е:

$$T_{sel} = \langle \{p_{sel}, p_{res}\}, \{p_{sel_1}, \dots, p_{sel_n}\}, IM_{sel} \rangle,$$

$$IM_{sel} = \begin{array}{c|ccc} & p_{sel,1} & \dots & p_{sel,n} \\ \hline p_{sel} & true & \dots & true \\ p_{calcres} & true & \dots & true \end{array}.$$

Това означава, че в началото на симулацията α -ядрото в позиция p_{sel} се разцепва на n идентични копия, които безусловно преминават към всички позиции $p_{sel,1}, \dots, p_{sel,n}$ на преход T_{sel} , които представляват n допълнителни входни позиции на n -те прехода на моделирания процес.

Едновременно с това в позиция $p_{calcres}$, която е втората входна позиция на T_{sel} , стои едно подлежащо на разцепване β -ядро със следната характеристика:

$$x^\beta = "R, VAL_1, VAL_1^*, VAL_2, VAL_2^*, \dots, VAL_n, VAL_n^* ",$$

където:

- R е поредният номер на текущото пускане на модела.
- VAL_1, \dots, VAL_n са n -те масива със съхранени стойности за текущото пускане на модела, които имат вида:

$$VAL_i = " \left\langle R, A_{R,i}, \sum_{s=1}^{A_{R,i}} \tau_{R,i,s}, \sum_{s=1}^{A_{R,i}} \mu_{R,i,s}, \sum_{s=1}^{A_{R,i}} \nu_{R,i,s} \right\rangle ",$$

където $i \in \{1, \dots, n\}$ е поредният номер на прехода, а $A_{R,i}$ е общият брой на активиранията на прехода T_i по време на $R^{\text{ТОТО}}$ пускане на ОМ модела. Третият параметър, а именно

$$\sum_{s=1}^{A_{R,i}} \tau_{R,i,s},$$

отговаря на общия брой τ на ядрата във всички входни позиции на T_i , натрупани по време на $A_{R,i}$ брой активирания на прехода по време на $R^{\text{ТОТО}}$ пускане на модела. Подобно е значението и на четвъртия и петия параметри, където μ отговаря на онези от ядрата, които подлежат на трансфер и успешно преминават от входна към изходна позиция по време на всичките $A_{R,i}$ активирания на прехода по време на $R^{\text{ТОТО}}$ пускане на модела. Накрая, ν отговаря на броя на ядрата във входната позиция, които не подлежат на преминаване към изходна позиция.

- VAL_1^*, \dots, VAL_n^* са n масива, които кумулативно натрупват сборовете на съответните стойности на τ -, μ - и ν -ядрата по време на всички предишни пускания на модела, във вида:

$$VAL_i^* = " R, \sum_{s=1}^R A_{i,s}, \sum_{s=1}^R \tau_{i,s}, \sum_{s=1}^R \mu_{i,s}, \sum_{s=1}^R \nu_{i,s} ",$$

където R представляват общия брой на пусканията (симулациите) направени дотук, а останалите стойности са сборовете на съответните стойности за едно пускане. В началото, когато $R = 0$, $VAL_i = " \langle 0, 0, 0, 0, 0 \rangle "$ и $VAL_i^* = " \langle 0, 0, 0, 0, 0 \rangle "$. След това, когато R е естествено число по-голямо от 0, $VAL_i = " \langle R, 0, 0, 0, 0 \rangle "$ и характеристиката на β преди поредната симулация е: $x^\beta = " \langle R, \langle R, 0, 0, 0, 0 \rangle, VAL_1^*, \langle R, 0, 0, 0, 0 \rangle, VAL_2^*, \dots, \langle R, 0, 0, 0, 0 \rangle, VAL_n^* \rangle "$.

Както вече споменахме, двете ядра α и β се разцепват на по n ядра ($\alpha_1, \dots, \alpha_n$ и β_1, \dots, β_n) по време на преминаването им от съответните входни позиции на прехода T_{sel} до изходните позиции $p_{sel,1}, \dots, p_{sel,n}$. Ядрата α_i имат идентични характеристики като ядрото α , а именно „избор“. Ядрата β_i имат характеристики „ VAL_i, VAL_i^* “, които са съответните части от характеристиката на ядрото β . В изходните позиции всяко ядро α_i се слива със съответното β_i -ядро в n нови ядра: γ_i . Началната характеристика на γ_i е „ x^α, x^{β_i} “, където:

- x^α е характеристиката на ядрото α („симулация“), и
- x^{β_i} е характеристиката на ядрото β_i , а именно „ VAL_i, VAL_i^* “ (в началния момент тези масиви се състоят само от нули).

И така n -те γ -ядра безусловно преминават към n -те входни позиции $p_{sel,1}, p_{sel,2}, \dots, p_{sel,n}$ на n -те преходи, които съставляват оригинално моделирания процес. Да опишем формално само i -тия от тези n прехода след модификациите, въведени от новопредложения подход. За по-лесно разбиране сегментът от индексирания матрица на предикатите на прехода, който отговаря на оригиналния модел, също е оцветен в сиво, както на Фиг. 2.15.

$$T_i = \langle \{p_{i,1}, \dots, p_{i,fi}, p_{sel,i}, p_{calc,i}\}, \{q_{i,1}, \dots, q_{i,gi}, p_{calc,i}, p_{res,i}\}, IM_i \rangle,$$

$IM_i =$	$q_{i,1}$	\dots	$q_{i,gi}$	$p_{calc,i}$	$p_{res,i}$
$p_{i,1}$		\dots		<i>false</i>	<i>false</i>
\vdots	\vdots	$W_{i,j,k}$	\vdots	\vdots	\vdots
$p_{i,fi}$		\dots		<i>false</i>	<i>false</i>
$p_{sel,i}$	<i>false</i>	\dots	<i>false</i>	<i>true</i>	<i>false</i>
$p_{calc,i}$	<i>false</i>	\dots	<i>false</i>	$W_{calc,i}$	$\neg W_{calc,i}$

където:

- $W_{i,j,k}$ са съответните предикати на оригинално моделирания процес (от чийто конкретен вид се абстрахираме).
- $W_{calc,i}$ = „времетраенето на функционирането на оригиналния ОМ модел още не е приключило“.
- $\neg W_{calc,i}$ е отрицанието на предиката $W_{calc,i}$.

Ядрото γ_i , което стои в позиция $p_{sel,i}$, безусловно постъпва в позиция $p_{calc,i}$ и не подлежи на директен преход към позиция $p_{res,i}$. То може да достигне позиция $p_{res,i}$ единствено ако е направило поне един цикъл през позиция $p_{calc,i}$ (позицията за междинни изчисления на стойностите на масивите „ VAL_i, VAL_i^* “, които отразяват движението на ядрата по време на симулацията). Следователно, както предвижда предикатът $W_{calc,i}$, ядрото ще цикли в позиция $p_{calc,i}$ дотогава, докато функционира оригиналният ОМ модел, като така запазва цялата информация за преминаването на ядрата по време на всички възможни активирания на преход T_i . В момента, в който оригиналният модел спре да функционира, ядрото γ_i , което е циклило в позиция $p_{calc,i}$, записвайки данните за „успешните“ и „неуспешните“ преминавания на ядра през прехода T_i , се придвижва към позиция $p_{res,i}$ получавайки характеристиката „ VAL_i, VAL_i^* “, където съответните стойности на масивите представляват натрупването на тази информация по време на всички активирания на прехода в рамките на цялата симулация на оригиналната ОМ. Това е възможно, тъй като приоритетът на позициите $p_{calc,i}$ е по-нисък от приоритетите на позициите $p_{i,1}, \dots, p_{i,fi}, q_{i,1}, \dots, q_{i,gi}$.

Дискутираме как ще продължат да се придвижват ядрата γ_i във втория новодобавен преход T_{res} . Отбелязваме, че поради най-ниския приоритет на този преход, той ще се

активира само когато цялостното функциониране на оригиналната ОМ е напълно завършило, т.е. след края на симулацията. Формалната дефиниция на прехода е следната:

$$T_{res} = \langle \{p_{res,1}, \dots, p_{res,n}\}, \{p_{res}, p_{calcres}\}, IM_{res} \rangle,$$

$$IM_{res} = \begin{array}{c|cc} & p_{res} & p_{calcres} \\ \hline p_{res,1} & W_{sel} & true \\ \dots & \dots & \dots \\ p_{res,n} & W_{sel} & true \end{array}.$$

и предикатът $W_{sel} = „x^\alpha$ е прогнозиране“. Следователно, при първия сценарий този предикат има вярностна стойност „лъжа“. Ядрата γ_i от входните позиции $p_{res,i}$ безусловно преминават към изходната позиция $p_{calcres}$, където се сливат в едно ядро, β , с характеристика: $x^\beta = " \bar{R}, \overline{VAL_1}, \overline{VAL_1^*}, \overline{VAL_2}, \overline{VAL_2^*}, \dots, \overline{VAL_n}, \overline{VAL_n^*} "$, където $\bar{R} = R + 1$, $\overline{VAL_i} = \langle \bar{R}, 0, 0, 0, 0 \rangle$, $\overline{VAL_i^*} = \overline{VAL_i} + VAL_i$.

Така окончателните данни за трансферите на ядрата по време на последната симулация на модела не се пазят поотделно, а се обединяват с данните, получени от всички предишни проигравания на модела. При първия сценарий когато целта на проиграването е „симулация“, както е зададено в характеристиката на началото ядро α , към позиция p_{res} на прехода T_{res} не преминават никакви γ_i ядра. Достигането на тази изходна позиция е възможно само когато целта, с която е пуснат моделът, е „прогнозиране“, т.е. вторият сценарий.

Сценарий 2: Прогнозиране

По време на втория сценарий, функционирането на първия преход T_{sel} е същото както по време на първия сценарий. Основната разлика между двата сценария е в преходите T_i ($i \in \{1, \dots, n\}$). Преди да ги опишем формално, нека да означим с $r_{i,j,k}$ една случайна променлива от интервала $[0; 1]$, която е свързана с прехода T_i . Тази променлива, отговаряща на j -тата входна позиция и k -тата изходна позиция на i -тия преход, ще се използва за определяне на типа на новите предикати $V_{i,j,k}$, които ще заменят оригиналните предикати, разгледани в първия сценарий.

В зависимост от целта на прогностичното проиграване на модела, има два различни варианта за развитието на процесите в оригиналния ОМ модел. Тези варианти могат условно да се нарекат „размит“ случай и „интуиционистки размит“ случай. И в двата случая формалното представяне на прехода е както следва:

$$T_i = \langle \{p_{i,1}, \dots, p_{i,fi}, p_{sel,i}, p_{calc,i}\}, \{q_{i,1}, \dots, q_{i,gi}, p_{calc,i}, p_{res,i}\}, IM_i \rangle,$$

$$IM_i = \begin{array}{c|ccc|cc} & q_{i,1} & \dots & q_{i,gi} & p_{calc,i} & p_{res,i} \\ \hline p_{i,1} & & \dots & & false & false \\ \vdots & \vdots & V_{i,j,k} & \vdots & \vdots & \vdots \\ p_{i,fi} & & \dots & & false & false \\ \hline p_{sel,i} & false & \dots & false & true & false \\ p_{calc,i} & false & \dots & false & W_{calc,i} & \neg W_{calc,i} \end{array},$$

където:

- $W_{calc,i} = „времето на функциониране на оригиналната ОМ не е приключило“.$
- $V_{i,j,k}$ са новите предикати, заменящи тези на оригинално моделирания процес.

- **Размит случай:**

$$V_{i,j,k} = "r_{i,j,k} \in \left[0; \frac{\sum_{s=1}^R \mu_{i,s}}{\sum_{s=1}^R \tau_{i,s}} \right] "$$

(използваните означения са като гореописаните). В този вариант, продължителността на функциониране на прехода T_i тук съвпада с продължението на функциониране на T_i в оригиналната мрежа. Това означава, че по време на работата на модела с цел прогнозиране могат да преминат същият брой ядра както в оригиналната мрежа.

- **Интуиционистки размит случай:** В този случай, преходът T_i ще бъде активен толкова на брой времеви стъпки, колкото е общият брой на ядрата в неговите входни позиции. Предикатът $V_{i,j,k}$ ще получи вида си по следното правило:

$$\left\{ \begin{array}{l} \text{ако } r_{i,j,k} \in \left[0; \frac{\sum_{s=1}^R \mu_{i,s}}{\sum_{s=1}^R \tau_{i,s}} \right], \quad \text{то } V_{i,j,k} = \text{„true“ и ядрото преминава} \\ \text{ако } r_{i,j,k} \in \left(\frac{\sum_{s=1}^R \mu_{i,s}}{\sum_{s=1}^R \tau_{i,s}}; \frac{\sum_{s=1}^R \mu_{i,s} + \sum_{s=1}^R \nu_{i,s}}{\sum_{s=1}^R \tau_{i,s}} \right), \quad \text{то } V_{i,j,k} = \text{„false“ и ядрото не} \\ \text{ако } r_{i,j,k} \in \left(\frac{\sum_{s=1}^R \mu_{i,s} + \sum_{s=1}^R \nu_{i,s}}{\sum_{s=1}^R \tau_{i,s}}; 1 \right), \quad \text{то } V_{i,j,k} = \text{„false“ и ядрото не} \end{array} \right. \begin{array}{l} \text{към изходна позиция (предикатът} \\ \text{е „истина“)} \\ \text{преминава (предикатът е „лъжа“)} \\ \text{преминава (поради недостатъчно} \\ \text{време)} \end{array}$$

Посредством това правило можем да имитираме реално функциониране на Алгоритъма за движение на ядра в преход (описан в Глава 1).

Накрая третият преход T_{res} запазва вида си, описан в първия сценарий, но сега предикатът $W_{sel} = \text{„}x^\alpha \text{ е прогнозиране“}$ се оценява като „истина“. Това променя поведението и характеристиката на ядрата γ_i . Всяко от тях се разцепва на две идентични ядра γ'_i и γ''_i . Ядрата γ'_i с характеристики „ VAL_i, VAL_i^* “ преминават към позиция p_{res} и се сливат там с ядрото δ , получаващо характеристиката: „ $VAL_1^*, VAL_2^*, \dots, VAL_n^*$ “, която позволява на потребителя да проследява назад и да разбира условията, предопределили разволя на прогностичното проиграване на модела.

Ядрата γ''_i , също с характеристики „ VAL_i, VAL_i^* “, преминават към позиция p_{calres} и се сливат там в ядро β без да получават нова характеристика. Така при бъдещи пускания на модела, без значение от целта на пускането, това ядро β може отново да бъде използвано като вход в прехода с най-висок приоритет T_{sel} .

Така конструираният модел описва процеса на събиране и натрупване на статистически данни за преминаването на ядра по време на симулации на произволни ОМ

модели. Използвайки тези данни, наред с интуиционистки размити оценки, можем да използваме тази конструкция за съставяне на прогнози относно поведението на модела.

Представените тук идеи подсказват конструирането на още по-детайлен подход към събирането на статистически данни, когато информацията се събира не на нивото на преходи, както е тук, а на нивото на входните и изходните позиции на прехода. Подобно детайлизиране ще бъде обект на бъдещи изследвания.

Глава 3. Приложения на обобщените мрежи

В тази глава ще бъдат описани някои нови приложения на ОМ.

В точка 3.1., следваща доклад [77], са описани две обобщени мрежи. Първата от тях представлява универсалната обобщена мрежа G_{ASOA} за класа алгоритми за оптимизация по метода на мравките, описваща начина на функциониране и резултатите от тяхната работа. Дискутираме как да се построи ОМ, която включва G_{ASOA} като оптимизационен компонент, т.е. като нейна подмрежа, чиито резултати от работата определят оптималното движение на ядрата в новата мрежа.

В точка 3.2., имаща за основа доклад [78], с ОМ е описан модел на хибриден метаевристичен подход, комбиниращ метода на мравките с генетични алгоритми. Целта е решения, получени с метода на мравките, да се подадат като начална популация за генетичния алгоритъм, с което отчасти да се контролира иначе случайния ѝ характер и да се постигнат по-добри и по-бързи резултати.

В точка 3.3. са представени моделите на двата най-общи и разпространени подхода за разрешаване на редакционни конфликти в системи с контрол на версиите. Двата модела са публикувани съответно в [72, 79].

3.1. Обобщеномрежов модел с оптимизационен компонент универсалната обобщена мрежа за класа алгоритми по метода на мравките

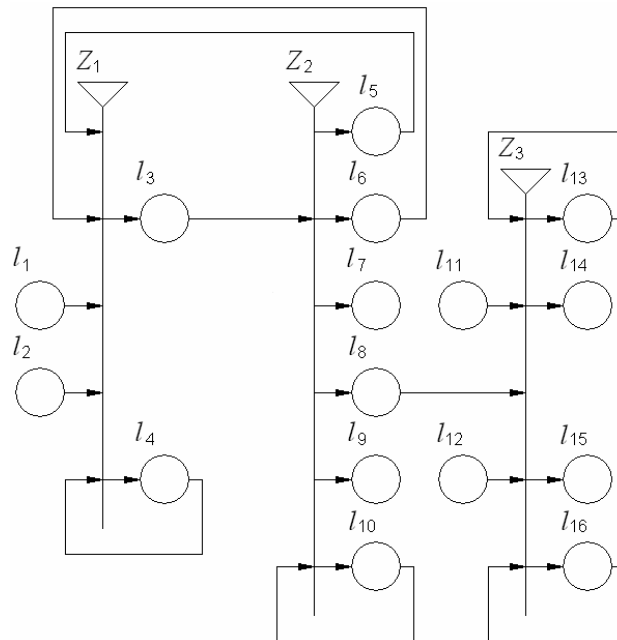
В доклад [77] с участието на автора за първи път е предложена оптимизационна задача, моделираща посредством ОМ алгоритъма за оптимизация по метода на мравките. Този компонент осигурява, че ОМ функционира оптимално по отношение на резултатите от изпълнението на алгоритъма на мравките.

Оптимизацията по метода на мравките (*Ant Colony Optimization, ACO*) е метаевристичен метод, предложен от М. Дориго през 1992 г. АСО алгоритмите се прилагат към задачи от реалния живот и промишлеността, при които за кратък срок се изискват задоволително добри решения. АСО открива добри решения на оптимизационни задачи с ограничителни условия, каквато задача е например Задачата за раницата. За разлика от точните методи, метаевристичите позволяват да се решават мащабни задачи чрез предоставянето на удовлетворителни решения за разумно кратко време.

Конструирана е ОМ, която е универсална за всички стандартни АСО-алгоритми. Тази ОМ описва как множество от n изкуствени мравки се придвижват по върховете на граф на АСО-алгоритъм, който се състои от m на брой върха и k на брой ребра. Да допуснем, че предварително е известно дали и къде в графа е останало някакво количество феромон от предишна симулация на процеса.

Нека означим така описаната ОМ с $G_{АСОА}$. Тя има три прехода, 16 позиции (вж Фиг. 3.1) и четири типа ядра (α , β , γ и ε). В ядрото α е заложено да се съдържа информация относно разположението на изкуствените мравки в графа. Ядрото β съдържа информация за броя на върховете (m) и броя на ребрата (k) на графа. Ядрото γ съдържа структурата на графа, т.е. матрицата му на инцидентност, докато ядрото ε съдържа информация на кои върхове/дъги при предишна симулация на процеса е бил положен феромон от мравките, и в какви количества. Всички тези ядра постъпват във входните позиции на ОМ с определени начални характеристики, които са описани по-долу:

- ядрото α постъпва в позиция l_1 с начална характеристика: „ n -мерен вектор, чиито елементи са разположенията на мравките в графа“;
- ядрото β влиза в позиция l_2 с начална характеристика: „ $\langle t$ -мерен вектор с елементи върховете на графа или l -мерен вектор с елементи ребрата на графа; целева функция“;
- ядрото γ влиза в позиция l_{11} с начална характеристика: „структура на графа с t върха и l ребра“;
- ядрото ε влиза в позиция l_{12} с начална характеристика: „начални данни за местоположението и количеството положен феромон“.



Фигура 3.1. ОМ модел на АСО

Ще опишем трите прехода последователно. Първият преход Z_1 има вида:

$$Z_1 = \langle \{l_1, l_2, l_4, l_5, l_6\}, \{l_3, l_4\}, r_1 \rangle,$$

	l_3	l_4
l_1	true	false
l_2	false	true
l_4	false	true
l_5	true	false
l_6	true	false

Ядрото α , намиращо се или в позиция l_1 , или в позиция l_5 , или в позиция l_6 , постъпва в позиция l_3 с характеристика: „вектор на текущите стойности на функцията на прехода $\langle \varphi_{1,cu}, \varphi_{2,cu}, \dots, \varphi_{n,cu} \rangle$ “, докато ядрото ε остава да цикли в позиция l_4 ,

получавайки характеристиката: „нов t -мерен вектор с елементи върховете на графа, или нов l -мерен вектор с елементи ребрата на графа“.

Вторият преход Z_2 има вида:

$$Z_2 = \langle \{l_3, l_{10}\}, \{l_5, l_6, l_7, l_8, l_9, l_{10}\}, r_2 \rangle,$$

$$r_2 = \begin{array}{c|ccccc} & l_5 & l_6 & l_7 & l_8 & l_9 & l_{10} \\ \hline l_3 & W_{3,5} & W_{3,6} & W_{3,7} & true & W_{3,9} & W_{3,10} \\ l_{10} & false & false & true & true & W_{10,9} & W_{10,10} \end{array},$$

и предикатите в r_2 имат следния смисъл:

- $W_{3,5}$ = „Текущата итерация не е свършила“,
- $W_{3,6} = W_{3,10} = \neg W_{3,5} \vee \neg W_{10,9}$,
- $W_{10,7}$ = „Текущото най-добро решение е по-лошо от глобалното най-добро решение“,
- $W_{10,9}$ = „Вярностната стойност на $C_1 \vee C_2 \vee C_3$ е истина“,
- $W_{10,10} = \neg W_{10,9}$,

а използваните в предиката $W_{10,9}$ изрази C_1 , C_2 и C_3 са следните условия за край:

- C_1 = „Постигнат е максимален брой итерации (изчислително време)“,
- C_2 = „Постигнат е определен брой итерации без да е подобрен резултатът“,
- C_3 = „Ако е известна долна/горна граница, то текущите резултати са близо до тази граница (напр. по-малко от 5%)“.

Ядрото α от позиция l_3 постъпва в позиция l_5 с характеристика: „ $\langle S_{1,ci}, S_{2,ci}, \dots, S_{n,ci} \rangle$ “, където $S_{i,ci}$ е текущото частично решение за текущата итерация, извършена от i -тата мравка ($1 \leq i \leq n$).

Ако предикатът $W_{3,6}$ е истина, то ядрото α се разцепва на три ядра: същото α и ядрата α' и α'' :

- Ядрото α постъпва в позиция l_6 с текуща характеристика: „нов n -мерен вектор с елементи новите местоположения на мравките“;
- Ядрото α' постъпва в позиция l_8 с последната характеристика на ядрото α (преди разцепването);
- Ядрото α'' постъпва в позиция l_{10} с текуща характеристика наредената двойка: „ \langle най-доброто решение за текущата итерация; пореден номер на итерацията \rangle “.

Когато предикатът $W_{10,9}$ е истина, ядрото α'' може да постъпи в позиция l_9 , където получава характеристиката „най-добро получено решение“.

В позиция l_7 постъпва онова от двете ядра от позиция l_{10} , което има по-лоши стойности като текуща характеристика, докато в позиция l_{10} остава да цикли ядрото, имащо за текуща характеристика най-добрите стойности.

Третият преход Z_3 има вида:

$$Z_3 = \langle \{l_8, l_{11}, l_{12}, l_{13}, l_{16}\}, \{l_{13}, l_{14}, l_{15}, l_{16}\}, r_3 \rangle,$$

$$r_3 = \begin{array}{c|cccc} & l_{13} & l_{14} & l_{15} & l_{16} \\ \hline l_8 & false & false & false & true \\ l_{11} & true & false & false & false \\ l_{12} & false & false & false & true \\ l_{13} & W_{13,13} & W_{13,14} & false & false \\ l_{16} & false & false & W_{16,15} & W_{16,16} \end{array},$$

и предикатите в r_3 имат следния смисъл:

- $W_{13,14} = W_{16,15} =$ „Вярностната стойност на израза $C_1 \vee C_2 \vee C_3$ е истина“,
- $W_{13,13} = W_{16,16} = \neg W_{13,14}$.

Ядрата γ от позиция l_{11} и β от позиция l_{12} с гореописаните характеристики постъпват, съответно, в позиции l_{13} и l_{16} без нова характеристика. Ядрото α от позиция l_8 постъпва в позиция l_{16} и се обединява там с ядрото β (като новото ядро отново е под името β), което получава характеристика: „стойност на функцията, актуализираща феромона съобразно стойностите на целевата функция“. Ядрата β и γ постъпват, съответно, в позиции l_{14} и l_{15} без да получават нови характеристики.

На базата на дефиницията на ОМ, строим нов тип ОМ – ОМ с оптимизационни компоненти; да го означим с *GNOС*. Ядрата в новия тип ОМ се придвижват съобразно класическите алгоритми, базирани на проверката на вярностните стойности на предикатите, определящи условията на преходите. Нека E е произволна ОМ и нека искаме да управляваме придвижването на ядрата в нея по оптимален начин, на база решенията, получени от АСО алгоритъма. Можем да обединим (с операцията обединение между ОМ) мрежите E и $G_{АСОА}$ и в новата ОМ можем да организираме оптималния начин за придвижване на ядрата.

Според алгоритъма за придвижване на ядрата, придвижването на ядрата се осъществява на всяка времева стъпка от функционирането на дадената ОМ в рамките на активното състояние на прехода, т.нар. „абстрактен преход“, който е обединение от всички активни ОМ-преходи в текущия момент. По тази причина можем да опишем функционирането само на един ОМ-преход и в частност, на абстрактния преход.

Нека е даден преходът Z от мрежата E . Всяка от неговите входни позиции (в термините на АСО алгоритъма) ще съответства на един начален връх на графа. Нека определим всички ОМ-ядра във входовете на Z . Броят им нека е началната характеристика на ядрото α от $G_{АСОА}$ (във вида на началните върхове на дадения граф). Информацията за ядрата, които съдържат тези ядра ще бъде зададена като начална характеристика на ядрото β на същата ОМ. Информацията за графа може да бъде въведена в ОМ $G_{АСОА}$ преди началото на процеса и ще бъде съхранявана по време на цялата симулация.

Като се имат предвид характеристичните функции на изходните позиции на прехода Z , можем да определим онези от върховете на графа, които имат достатъчно високи тегла.

Всяко ядро от входните позиции на Z ще съответства на „мравка“, която трябва да търси път до върха от графа, който има максимално тегло.

Компонентите t_1 , t_2 и M от ОМ прехода не са необходими (например, можем да приемем, че стойностите на t_1 са последователни естествени числа, $t_2 = 1$, а елементите от M са равни на 1. Типът на прехода има вида на дизюнкция.

На базата на определените по-горе характеристики на ядрата от $G_{АСОА}$, тези ядра преминават през мрежата и в резултат определят маршрутите на отделните мравки (а следователно и на ядрата от Z) от началните върхове на графа до върховете му с максимални тегла (които отговарят съответно на входните и изходните позиции на прехода Z). Така, използвайки решенията от мрежата $G_{АСОА}$, можем да определим начинът, по който да преминават ядрата от входните към изходните позиции на прехода Z .

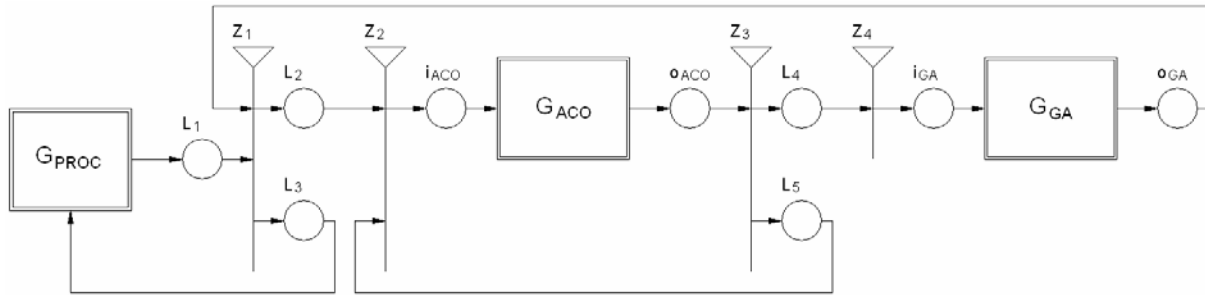
Тази процедура може да се използва за управление на всеки един от преходите на ОМ E . В този случай в $G_{АСОА}$ ще поставим съответния брой α - и β -ядра, така че техният брой да отговаря на броя преходи в ОМ E . Описаната процедура е универсална по своята природа. Приложима е към всяка ОМ.

Предложената тук процедура може да намери разнообразни приложения, например в спешната медицина: възможно е да се конструират ОМ модели на организацията и управлението на звено за спешна медицинска помощ, а използвайки оптимизационен компонент по метода на мравките може да се търси оптималност при вземането на решения в звеното.

3.2. Обобщена мрежа, синхронизираща хибриден метаевристичен алгоритъм

Описан е ОМ модел, базиращ се на два от най-добре представящите се метаевристични подхода: алгоритмите за оптимизация по метода на мравките и генетичните алгоритми. Накратко са представени генетичните и хибридните метаевристични алгоритми.

Следвайки публикацията ни [78], конструираме ОМ мрежа, описваща хибридният алгоритъм между АСО и ГА. Процесът в конкретно решаваната оптимизационна задача се описва в рамките на мрежата G_{PROC} , вж. Фиг. 3.2.



Фигура 3.2. ОМ с подмрежи G_{PROC} , G_{ACO} и G_{GA}

Всяка от мрежите G_{ACO} и G_{GA} има по една входна и една изходна позиции. Нека ги наричаме, съответно, i_{ACO} ; o_{ACO} ; i_{GA} ; o_{GA} . Нека ядрото α от мрежата G_{PROC} постъпва в позиция l_1 с начална характеристика: „текущо описание на задачата (граф на задачата, ограничителни условия, и т.н.)“.

Първият преход има вида:

$$Z_1 = \langle \{l_1, o_{GA}\}, \{l_2, l_3\}, r_1 \rangle,$$

$$r_1 = \begin{array}{c|cc} & l_2 & l_3 \\ \hline l_1 & true & false \\ o_{GA} & W_{GA,2} & W_{GA,3} \end{array},$$

и предикатите в индексиранията матрица имат вида:

- $W_{GA,2}$ = „Необходима е нова итерация“,
- $W_{GA,3}$ = $\neg W_{GA,2}$.

Ядрото α от позиция l_1 постъпва в позиция l_2 с новата характеристика: „инициализация на графа на задачата (начално разположение и количество феромон)“.

Когато вярностната стойност на предиката $W_{GA,2}$ е истина, α -ядрото от позиция o_{GA} постъпва в позиция l_2 с характеристика: „обновяване на феромона, съобразно текущите решения“, докато в позиция l_3 постъпва без да получи нова характеристика.

Вторият преход има вида:

$$Z_2 = \langle \{l_2, l_5\}, \{i_{ACO}\}, r_2 \rangle,$$

$$r_2 = \begin{array}{c|c} & i_{ACO} \\ \hline l_2 & true \\ l_5 & true \end{array}.$$

Всяко едно от α -ядрата от позиции l_2 или l_5 постъпва в позиция i_{ACO} без да получи нова характеристика. То преминава през мрежата G_{ACO} и, напусайки я през позиция o_{ACO} , получава характеристика: „текущи решения на АСО алгоритъма (генерирани популации)“:

Третият преход има вида:

$$Z_3 = \langle \{o_{ACO}\}, \{l_4, l_5\}, r_3 \rangle,$$

$$r_3 = \frac{\quad}{o_{ACO} \mid \begin{array}{cc} l_4 & l_5 \\ W_{ACO,4} & W_{ACO,5} \end{array}},$$

където предикатите в индексиранията матрица имат вида:

- $W_{ACO,4} = \text{„Условието за край е изпълнено“}$,
- $W_{ACO,5} = \neg W_{ACO,4}$.

Когато вярностната стойност на предиката $W_{ACO,4}$ е истина, ядрото α постъпва в позиция l_4 с характеристика: „Представяне на текущите решения (популации) във вид, подходящ за ГА“, докато в позиция l_5 постъпва без да получи нова характеристика.

Четвъртият преход има вида:

$$Z_4 = \langle \{l_4\}, \{i_{GA}\}, r_4 \rangle,$$

$$r_4 = \frac{\quad}{l_4 \mid \begin{array}{c} i_{GA} \\ true \end{array}}.$$

Ядрото α от позиция l_4 постъпва в позиция i_{GA} с характеристика „Текуща популация (решения) на ГА“.

Предложеният нов хибриден метаевристичен подход комбинира две метаевристички, *Ant Colony Optimization* и генетични алгоритми, комбинирайки предимствата на двата съставни подхода. Подходът е приложен към решаване на проблем за параметрична идентификация на процес на култивиране на бактерията *E. coli*, и резултатът, приет за участие в предстоящата 9th International Conference on Large Scale Scientific Computation '2013, показва, че хибридният алгоритъм се нуждае от почти два пъти по-малко изчислително време и приблизително пет пъти по-малки популации от агенти, в сравнение със стандартните алгоритъм на мравките и генетичен алгоритъм.

3.3. Обобщеномрежови модели на механизми за разрешаване на редакционни конфликти

Контролът на версиите е управлението на промените в документи, програми и друга информация, съхранена под формата на компютърни файлове. Обичайно контрол е необходим и използван в многопотребителски среди като платформи за разработка на софтуер, уикита и системи за управление на съдържанието, където екип от хора могат да си сътрудничат в работата върху споделено множество от файлове. Всяка версия се асоциира с времева марка (*timestamp*), потребител (потребителско име или IP-адрес) извършил редакцията, както и други незадължителни данни. Версиите могат да бъдат съхранявани, възвръщани и (при някои файлови типове) сливани.

При функционирането на системите с контрол на версиите важен проблем за решаване е конкурентният достъп до общото съдържание от страна на множество потребители, поради възможностите за конфликти на редакции и инцидентна загуба на информация. Редакционните конфликти възникват, когато един потребител започне да редактира документ, но при опита да съхрани версията си се оказва, че друг потребител междуременно вече е променил документа.

Проблемът с конкурентния достъп за редактиране (*write access*) е фундаментален за всички системи с контрол на версиите и споделени файлови хранилища. С известни вариации, съществуват два генерални подхода за разрешаването на този проблем:

- „Заклучи–Промени–Отключи“ (*Lock–Modify–Unlock, LMU*), и
- „Копирай–Промени–Слей“ (*Copy–Modify–Merge, CMM*).

Механизмът LMU е линейно ориентиран и смятан за песимистичен. Състои се в заключване на файла, така че в един момент от време само един потребител да има достъп за редактиране до него. Заедно с ползата от лесната реализация на този подход, той има и недостатъци. Той може да се окаже рестриктивен и да доведе до загуба на време и производителност.

Подходът СММ е граф-ориентиран и смятан за оптимистичен. При него всеки потребител има равнопоставен достъп и при отваряне за редактиране на даден файл системата създава за този потребител локално работно копие. Така потребителите имат възможност да работят едновременно и независимо един от друг. Накрая системата се опитва автоматично да слее в една окончателна версия отделните редакции, и в случай на неуспех оставя на потребителите ръчно да се справят с това.

Всеки от двата подхода има своите предимства и недостатъци и уместността на избора между тях зависи от няколко фактора като типа файлове в хранилището, броя файлове в хранилището, размера на допринасящата потребителска общност, средното време за извършване на редакция, средният обем извършени промени в байтове, и регистрираната честота на възникващи редакционни конфликти при тези условия.

Двата ОМ модела съдържа статични преходи, означени с Т и индекс L или С (съответно за моделите LMU или СММ) и номерирани с поредния номер на прехода отляво надясно. Всички позиции, с изключение на DB и ID са маркирани със съответните действия, които се случват в тях, и отново индексирани с L или С. Двата модела неминуемо съдържат общи позиции (Open, Edit, Cancel, Wait, Save, Close), но вторият съдържа и няколко допълнителни позиции (Done, Confl, Approve, Merge). Позициите DB (за базата данни на хранилището) и ID (за генератора на идентификатори на поредни версии) нямат индекс, тъй като в двата модела между тях няма семантични разлики.

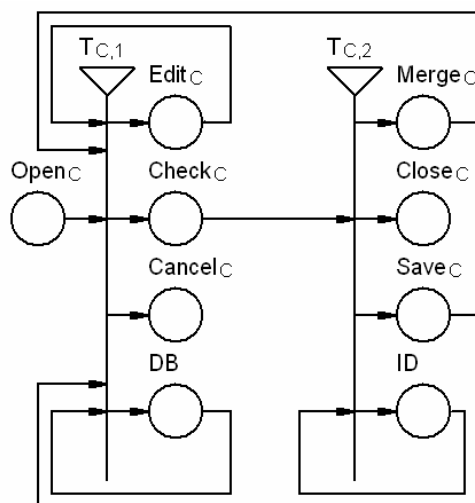
И двата ОМ модела съдържат три типа ядра, които подлежат на разцепване и сливане, и тези ядра са носители на динамичната природа на моделите:

- ϵ -ядро, което отговаря на редакторите (потребителите), които осъществяват достъп до даден файл. Това ядро отначало влиза в мрежата през позицията с етикет Open.
- δ -ядро, което стои само в позиция DB и представлява базата данни с файлове в хранилището. Отделните инстанции на файлове са ядра, означени с φ , които веднъж щом се отделят от разделящото се ядро δ , могат да се придвижват в останалата част от мрежата.
- i -ядро, стоящо само в позиция ID и представляващо генератора на идентификатори на версии. Отделните идентификатори на файлови версии са естествени числа, които в ОМ модела са представени чрез ядра, означени с v . Веднъж щом се отделят от разделящото се ядро i , те могат да се придвижват из останалата част от мрежата.

ОМ моделът на подхода LMU (Фиг. 3.6) е по-простият от двата. Той съдържа два прехода, девет позиции и гореописаните три вида ядра.

Всяко ϵ -ядро влиза в мрежата през позиция $Open_L$ с начална характеристика: „Потребителско име (IP-адрес) на потребителя. Поискан файл. Момент от време на достъп до файла T_{Open} “.

Едно ядро δ постоянно цикли в позиция DB с характеристика: „Списък от файловете в базата данни, асоциирани със състоянието им Заключен/ Отключен и идентификатора на текущата версия“. Първоначално всички файлове в базата данни са в състояние „отключен“, но по време на функционирането на модела всички файлове, които бъдат отворени за редактиране променят състоянието си на „заклучен“. Докато ядрото δ представлява цялата база от данни, отделните екземпляри на файлове ще бъдат представяни с φ -ядра.



Фигура 3.6. OM модел на „Заклучи–Промени–Отключи“

Ядро i перманентно цикли в позиция ID с характеристика: „Брой на редакциите на файла, внесени в системата, считано от даден момент“.

Отделните идентификатори на версии ще бъдат представени с v -ядра.

Първият преход $T_{L,1}$ има четири входни и четири изходни позиции

$$T_{L,1} = \langle \{Wait_L, Open_L, Save_L, DB\}, \{Wait_L, Cancel_L, Edit_L, DB\}, M_{L,1} \rangle$$

$M_{L,1} =$	$Wait_L$	$Cancel_L$	$Edit_L$	DB
$Wait_L$	$P_{L,1,3}$	$P_{L,1,4}$	$P_{L,1,1}$	false
$Open_L$	$P_{L,1,2}$	false	$P_{L,1,1}$	false
$Save_L$	false	false	false	true
DB	$P_{L,1,2}$	false	$P_{L,1,1}$	true

където предикатите $P_{L,1,1}, \dots, P_{L,1,4}$ имат следните интерпретации.

- $P_{L,1,1} =$ „Състоянието на поискания файл от текущия потребител е „Отключен“.
- $P_{L,1,2} = \neg P_{L,1,1} =$ „Състоянието на поискания файл от текущия потребител е „Заклучен“.
- $P_{L,1,3} =$ „Потребителят чака файла да бъде отключен.“
- $P_{L,1,4} =$ „Потребителят престава да чака файла да бъде отключен.“

Когато вярностната стойност на предиката $P_{L,1,1}$ е „истина“, ядро ε напуска позиция Open или Wait и влиза в позиция Edit. Едновременно с това ядро δ в позиция DB се разцепва на едноименното ядро δ и едно ново ядро φ , което представлява файл от хранилището, поискан за редактиране от текущия потребител. Ядрото φ влиза в позиция Edit с текуща характеристика: „Име и съдържание на поискания файл. Други характеристики (незадължителни)“.

Ядрото δ прави един цикъл и се връща в позиция DB и актуализира характеристиката си, като променя в характеристичната таблица състоянието на файла, съответстващ на φ , от „отключен“ на „заклучен“. За отбелязване е, че състоянието на файла е важна характеристика, която формира разликата между предикатите $P_{L,1,1}$ и $P_{L,1,2}$.

Когато ядрото ε от позиция $Open_L$ и ядрото φ от позиция DB влязат в позиция $Edit_L$, те се сливат в едно ядро под името ε , което получава характеристиката: „Потребителско име. Име на файла. Момент T_{Lock} на заключване на файла. Състояние: Заклучен“.

Предикатът $P_{L,1,2}$ става „истина“, когато едно ε ядро влезе в мрежата през позиция $Open_L$, изисквайки файл от хранилището, но съответният запис от характеристичната таблица на δ -ядрото в позиция DB гласи, че поисканият файл е заключен за

редактиране. Следователно, δ -ядрото остава в позиция DB без да се разцепва на нови ядра, докато ядрото ε преминава в позиция $Wait_L$ с характеристика: „Момент T_{Wait} , когато потребителят е започнал да чака файла да бъде отключен.“

Предикатът $P_{L,1,3}$ е „истина“ дотогава, докогато редакторът чака файла да бъде отключен, т.е. ядрото ε цикли в позиция $Wait_L$ без да получава нови характеристики.

Предикатът $P_{L,1,4}$ става „истина“ когато редакторът реши да се откаже от понататъшно чакане файлът да бъде отключен, като така ядрото ε от позиция $Wait_L$ преминава в позиция $Cancel_L$ с новата характеристика: „Текущ момент от време $T_{Cancel} \cdot Time_{Wait}$ “, където $Time_{Wait}$ е времето, прекарано напразно в чакане файлът да бъде отключен, и $Time_{Wait} = T_{Cancel} - T_{Wait}$.

Веднъж щом ядрото φ влезе в позиция $Save_L$, на следващия момент от функционирането на ОМ то ще премине безусловно в позиция DB (откъдето и съответната клетка от индексиранията матрица $M_{L,1}$ е „истина“, вместо някакъв предикат) с характеристика, която ще бъде коментирана по-късно.

Вторият преход $T_{L,2}$ има две входни и три изходни позиции

$$T_{L,2} = \langle \{ID, Edit_L\}, \{ID, Close_L, Save_L\}, M_{L,2} \rangle,$$

	ID	Close _L	Save _L
ID	$P_{L,2,2}$	false	$P_{L,2,2}$
Edit _L	false	$P_{L,2,1}$	$P_{L,2,2}$

където предикатите $P_{L,2,1}$ и $P_{L,2,2}$ имат следните интерпретации:

- $P_{L,2,1} =$ „Потребителят иска да приключи работата си по файла без да запази редакцията си“.
- $P_{L,2,2} =$ „Потребителят иска да приключи работата си по файла и да запази редакцията си“.

Когато едно ядро ε (слято, както беше описано по-горе) влезе в позиция $Edit_L$, то ще остане в тази позиция дотогава, докогато редакторът работи по файла. Когато потребителят приключи работата си по файла, ядрото ще напусне позиция $Edit_L$ защото или предикатът $P_{L,2,1}$ или предикатът $P_{L,2,2}$ ще са станали „истина“, в зависимост от конкретния сценарий: потребителят може или да реши да затвори файла без да запазва редакцията си, или може да избере да съхрани промените (което и обичайно се случва).

Когато предикат $P_{L,2,1}$ е „истина“, т.е. редакторът избере да напусне без да запазва промените по файла, ядрото ε ще се разцепи на две ядра: същото ядро ε и ново ядро φ . Новополученото ядро ε ще премине към позиция $Close_L$ с характеристиката: „Момент T_{Close} на затваряне на файла. $Time_{Close}$ “, където $Time_{Close} = T_{Close} - T_{Lock}$ е продължителността от време, в което файлът е бил заключен, въпреки че не са настъпвали в него редакционни промени.

Новополученото ядро φ ще влезе в позиция $Save_L$ с характеристика: „Съдържание на файла: Непроменено. Идентификатор на версията: Непроменен. Състояние на файла: Отключен“.

Когато предикат $P_{L,2,2}$ е „истина“, т.е. потребителят избере да съхрани нанесените промени, се случват три действия. Първо, генераторът на идентификатори на поредни версии ще подаде съответния идентификатор (естествено число), който да бъде присвоен на новосъхранената файлова версия, и после ще увеличи това число с 1 (така че следващата версия на файл от хранилището да получи различен идентификатор). За тази цел ядрото i (с текуща характеристика естественото число $ID_{Current}$), стоящо в позиция ID, ще се разцепи на две ядра: новото ядро v и същото ядро i . Ядрото v ще премине в позиция $Save_L$ с характеристика: „Текуща стойност на ядрото i : $ID_{Current}$ “.

Ядрото i , получено след разцепването, ще постъпи в позиция ID с характеристика: „ $ID_{Current} + 1$ “.

От друга страна, ε -ядрото от позиция $Edit_C$ отново ще се разцепи на две ядра: същото ядро ε и ново ядро φ (което отговаря на означеното преди ядро φ дошло от позиция DB). Новополученото ядро ε ще премине в позиция $Close_C$ с характеристиката: „Момент T_{Save} на съхраняване на файла. $Time_{Save}$ “, където $Time_{Save} = T_{Save} - T_{Lock}$ е продължителността на времето за редактиране на файла, през което той е бил в състояние „заклучен“ за други потребители. (Разликата между $Time_{Close}$ и $Time_{Save}$ е само в тяхната семантика.)

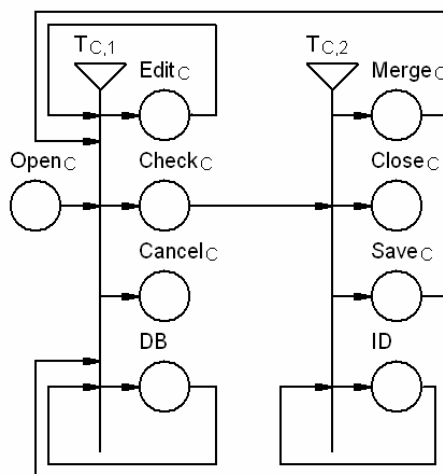
Новополученото ядро φ влиза в позиция $Save_C$, където се слива с ядро ν отново под името φ и получава характеристика: „Съдържание на файла: Променено. Идентификатор на версията: $ID_{Current}$. Състояние на файла: Отключен“.

Най-сетне ще се върнем на характеристиката на ядрото φ , получено при безусловния си преход от позиция $Save_C$ обратно в позиция DB . На практика:

- то отразява факта, че файлът бива отключен, без значение бил ли е редактиран или не; и
- в случай на съхранени промени, то добавя нова версия към историята на файла и я прави текущо видима по подразбиране за всички потребителите на системата.

Така ядрото φ от позиция $Save_C$ постъпва в позиция DB , където се слива с наличното там ядро δ с характеристики, зависещи от конкретните стойности на характеристиките, носени от φ ядрото на предходната стъпка от функционирането на модела.

Подобно на ОМ модела на LMU, и моделът тук също оперира с описаните три типа ядра, които могат да се разцепват и сливат: ε , δ и ν -ядра.



Фигура 3.7. ОМ модел на подхода „Копирай–Промени–Слей“

Всяко ε -ядро влиза в мрежата през позиция $Open_C$ с начална характеристика: „Потребителско име (IP-адрес) на потребителя. Поискан файл. Времева марка (timestamp) на момента от време на достъп до файла“.

Едно ядро δ постоянно стои в позиция DB с характеристика: „Списък от файловете в базата данни, съдържащ: (1) Идентификатор на файла. (2) Идентификатор и времева марка на текущата версия на файла. (3) Списък на копията на файла, асоциирани с идентификатор на копието, идентификатор на потребителя и идентификатор и timestamp на версията на файла, която е била текуща към момента на генериране на копието“.

Отново ядрото δ представлява цялата база данни, отделните инстанции на файлове от базата данни ще бъдат представени като φ -ядра. Ядрото ν и тук цикли постоянно в позиция ID с характеристика: „Брой редакции на файла, внесени в системата, считано от даден момент“. Отделните идентификатори на версии ще бъдат и тук представени с ν -ядра.

Първият преход $T_{C,1}$ има следния вид:

$$T_{C,1} = \langle \{Edit_C, Merge_C, Open_C, Save_C, DB\}, \{Edit_C, Check_C, Cancel_C, DB\}, M_{C,1} \rangle$$

	$Edit_C$	$Check_C$	$Cancel_C$	DB
$Edit_C$	$P_{C,1,2}$	$P_{C,1,4}$	$P_{C,1,3}$	$P_{C,1,3}$
$Merge_C$	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
$Open_C$	$P_{C,1,1}$	<i>false</i>	<i>false</i>	<i>false</i>
$Save_C$	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>
DB	$P_{C,1,1}$	$P_{C,1,5}$	<i>false</i>	<i>true</i>

и предикатите в индексиранията матрица имат вида:

- $P_{C,1,1}$ = „Потребителят получава копие на файла от хранилището, отворен в режим на редактиране.“
- $P_{C,1,2}$ = „Потребителят продължава да редактира файла.“
- $P_{C,1,3}$ = „Потребителят спира да редактира и не иска да запази редакцията си.“
- $P_{C,1,4}$ = „Потребителят спира да редактира и иска да запази редакцията си.“
- $P_{C,1,5}$ = „Засечен е конфликт на редакции.“

Когато предикатът $P_{C,1,1}$ е истина, ядрото ε напуска позицията $Open_C$ и влиза в позиция $Edit_C$. Същевременно, ядро δ в позиция DB се разцепва на същото ядро δ и едно ново ядро φ , което представлява копие на файла от хранилището, поискан от потребителя за редактиране. Ядрото φ влиза в позиция $Edit_C$ с текуща характеристика: „{1} *Файл*. {2} *Идентификатор, времева марка и съдържание на текущата версия на файла*“.

Ядро δ прави един цикъл и се връща в позиция DB , където обновява характеристичната таблица с добавянето на нов запис в третата колона за новосъздаденото копие на поискания файл φ , който запис се означава с ε и се характеризира с идентификатора на текущата версия на файла, моментът от време (времева марка) на генериране на копието и потребител, поискал копието.

Така когато ε -ядрото от позиция $Open_C$ и φ -ядрото от позиция DB влязат в позиция $Edit_C$, те се сливат като едно ε -ядро, което получава характеристика: „{1} *Потребител*. *Идентификатор на файл*. {2} *Идентификатор на версията, времева марка и съдържание на най-скорошната версия на файла*. {3} *Времева марка и съдържание на копието на файла*.“

Когато предикатът $P_{C,1,2}$ получи вярностна стойност „истина“, то това означава, че потребителят продължава да редактира по извлеченото при него персонално копие на файла, т.е. ε -ядрото продължава да цикли в позиция $Edit_C$. На всеки цикъл в позиция $Edit_C$, ядрото ε обновява своята характеристика, особено в последната си част „Съдържание на копието на файла“.

Когато предикатът $P_{C,1,2}$ получи вярностна стойност „лъжа“, то това означава, че може да стане истина или предикатът $P_{C,1,3}$ или $P_{C,1,4}$: потребителят решава да прекрати редактирането на файла. Кой от двата предиката ще стане истина, зависи от това дали потребителят иска да съхрани редакцията си или не. Ако потребителят иска да откаже редактирането без да запази промените си, става истина предикатът $P_{C,1,3}$, а ядрото ε от позиция $Edit_C$ се разцепва на две: същото ядро ε и началното ядро φ . Ядрото ε влиза в позиция $Cancel_C$ без никаква характеристика, докато ядрото φ влиза в позиция DB и се слива там с ядрото δ , обновявайки характеристичната таблица с премахването на съответния запис за копието от съответния файл, поискан от съответния потребител.

Най-сетне потребителят може да реши да съхрани редакцията си, което означава предикат $P_{C,1,4}$ да стане „истина“, което става причина ε -ядрото от позиция $Edit_C$ да се придвижи до позиция $Conflict_C$, запазвайки последната си характеристика. Там е от значение каква ще бъде стойността на предиката $P_{C,1,5}$. Предикатът $P_{C,1,5}$ съответства

на проверка на системата дали е настъпил конфликт на редакции за дискутираното копие на файла, или не. Тази проверка лесно се прави в позиция **DB**, предвид факта, че характеристичната таблица на ядрото δ поддържа информация не само за идентификаторите на файловете версии, но и за времевите марки на текущата файлове версия и на всяко от произведените копия на файла, когато той е бил отворян за редактиране. Когато нова версия на файла успешно се съхрани в базата данни, времевата марка на текущата (обновена) версия на файла също се обновява, и ако вече са били генерирани копия на файла, техните времеви марки са по-малки като стойност от стойността на времевата марка на обновената версия на файла, следователно предикатът $P_{C,1,5}$ става истина. Това води до разцепването на δ -ядрото в позиция **DB** на две ядра: оригиналното δ -ядро, носител на пълната информация за системната база данни, и едно ново ядро φ_{cur} , което преминава в позиция **Conflict_C** с характеристиката: „Идентификатор на текущата версия, времева марка и съдържание на файла“. Там ядрото φ_{cur} се слива с ядрото ε , което е пристигнало по силата на стойността „истина“ на предиката $P_{C,1,4}$. Така, позиция **Conflict_C** съответства на момента, в който системата засича конфликт на редакции и изисква от потребителя ръчно да го разреши.

Така в позиция **Conflict_C** новополученото след сливането ядро ε получава комбинираната характеристика: „{1} Потребител. Идентификатор на файл. {2} Идентификатор на версията, времева марка и съдържание на файловата версия, която е била текуща към момента на генерирането на копие. {3} Времева марка и съдържание на файловото копие. {4} Идентификатор на версията, времева марка и съдържание на текущата файлове версия.“ На практика, това съответства на момента от време, когато потребителят е получил известие за конфликта на редакции.

Позиция **Conflict_C** е единствената позиция в мрежата, която свързва двата прехода в модела, т.е. се явява изходна позиция за преход $T_{C,1}$ и входна за втория преход $T_{C,2}$, който се представя по следния начин:

$$T_{C,2} = \langle \{Conflict_C, ID\}, \{Merge_C, Close_C, Save_C, ID\}, M_{C,2} \rangle$$

	Merge _C	Close _C	Save _C	ID
Conflict _C	$P_{C,2,1}$	$P_{C,2,2}$	$P_{C,2,2}$	false
ID	false	false	$P_{C,2,2}$	true

и предикатите в индексиранията матрица имат вида:

- $P_{C,2,1} =$ „Потребителят продължава да редактира.“
- $P_{C,2,2} = \neg P_{C,2,1}$.

Ако предикатът $P_{C,2,1}$ е „истина“, това отговаря на сценария, при който потребителят е избрал да продължи да редактира. Резултатът е, че ε -ядрото от позиция **Conflict_C** преминава в позиция **Merge_C**, където, както знаем от първата матрица на предикатите $M_{C,1}$, то безусловно преминава в позиция **Edit_C**.

В обратния случай, ако предикат $P_{C,2,1}$ е лъжа, т.е. е истина предикатът $P_{C,2,2}$, това отговаря на ситуацията, когато системата позволява на потребителя да съхрани редакцията си, тъй като не е установен конфликт на редакции. Как точно това ще се случи в термините на ОМ, зависи от стойността на предиката $P_{C,1,5}$ от по-горе:

- Ако предикатът $P_{C,1,5}$ е „лъжа“, т.е. не е засечен конфликт на редакции, то пред потребителя няма технически пречки да съхрани своите промени по файла. Следователно ε -ядрото от позиция **Conflict_C** се разделя на две ядра: ядро ε , което представлява редактора, и ядро φ_{new} , което представлява новогенерираната версия на файла. По същото време i -ядрото в позиция **ID** също се разцепва на две: същото ядро i и ново ядро v , което представлява поредния идентификатор на файлова версия. Ядрото i продължава да цикли в позиция **ID**, като текущата му характеристика се увеличава с 1. Ядрото v преминава от позиция **ID** в позиция **Save_C**, където се слива с ядрото φ_{new} , което получава характеристиката:

„Потребител. Идентификатор на файл. Идентификатор на файлова версия. Времева марка. Редактирано съдържание на файла“, където идентификаторът на версията е получен от характеристиката на ν -ядрото, а времевата марка отговаря на текущия момент от време, когато новата версия на файла е съхранена. Нещо повече, както се вижда от индексиранията матрица $M_{C,1}$ на първия преход, от позиция $Save_C$ винаги се преминава към позиция DB (стойност „истина“), с което тези данни се добавят като нов запис в историята на файла в базата данни. Самото ядро ε преминава от позиция $Conflict_C$ към позиция $Close_C$ с малко по-кратка характеристика: *„Потребител. Идентификатор на файл. Идентификатор на версията. Времева марка“*, която на практика съответства на затварянето на сесията за редактиране и добавяне на нов запис към списъка на извършените от потребителя действия в системата (в случай, че системата поддържа такъв дневник).

- Ако предикатът $P_{C,1,5}$ е „истина“, т.е. е бил засечен редакционен конфликт в системата, но потребителят въпреки това не иска да продължи да редактира, което означава, че ε -ядрото от позиция $Conflict_C$ се разцепва на две ядра: ядро ε , което преминава в позиция $Close_C$ без никаква характеристика и ядро φ_{new} , което преминава в позиция $Save_C$ с характеристиката: *„Премахване от базата данни на съответния запис за текущото копие на файла, генерирано от този потребител“*. Веднъж щом се премахне от характеристикната таблица, това файлово копие повече няма да влияе на останалите генерирани от други потребители локални копия на същия файл и няма да причинява при никого от тях конфликти на редакции. Както се вижда в матрицата на преходите на първия преход, преминаването от позиция $Save_C$ към позиция DB е безусловно „истина“.

В заключение, може да се отбележи, че при симулация на двата модела анализът на резултатите може да помогне на разработчиците на системи с контрол на версиите да направят верния избор, когато решават кой от двата подхода да внедрят в продуктите си.

Глава 4. Уики уебсайт с приложение за обобщени мрежи

Дискутирана е реализацията на уебсайт, изграден от автора с мрежовата технология уики, посветен на областите ИРМ и ОМ. Сайтът Ifigenia.org, е първият такъв тематично специализиран уебсайт и още от самото си създаване започва да предоставя следните възможности:

- изграждане на учебен корпус от контекстуално свързан понятиен апарат (по примера на онлайн енциклопедията Уикипедия),
- база данни от публикации,
- платформа за търсене на партньорства и провеждане на научни дискусии,
- среда за съвместна разработка на научни текстове, проектни предложения, софтуерна документация и други.

През април 2010 г., след направено разширение на ядрото на софтуера, сайтът започва да предлага и опцията за провеждане на онлайн тестове по ОМ, и то в непосредствена близост до учебното съдържание в рамките на уикито.

4.1. Уики като технология и философия

Уики е технология за създаване на колекции от взаимосвързани Web 2.0 страници, проектирана така, че всеки да може да ги редактира, като използва само платформено независим браузър, като всички версии на страниците се съхраняват еднозначно в базата данни. Сайтовете, изградени на уики технологията, често се използват от различни потребителски общности за улесняване на онлайн базираната комуникация и сътрудничество. Дискутирани са основните характеристики на технологията и философията на уики.

4.2. Ifigenia.org – среда за електронно обучение по обобщени мрежи

Първоначалната идея за създаването на Ifigenia.org е продиктувана от нуждата да се изгради интегрален онлайн-базиран информационен ресурс за ОМ и ИРМ. Основната инсталация е направена с версия 1.13.2 на MediaWiki, версия 5.0.51a-24+lenny2 на MySQL и версия 5.2.6-1+lenny3 на PHP. По време на периода на тестване и конфигуриране на инсталацията са изпробвани разнообразни разширения на ядрото. В резултат, ядрото на MediaWiki в Ifigenia.org е надградено със разширения, достъпни от специалната страница “Version”. Бяха направени необходимите изпитания, които да осигурят, че всички скриптове и шаблони в уикито работят коректно под различни браузъри и техни версии. Сайтът съдържа два типа страници – с публичен и с ограничен достъп. Освен страници с дефиниции на понятията от тематиката на сайта и електронни копия на публикации, в Ifigenia.org се поддържа и документация на разработваните програмни продукти за ОМ.

Ще отбележим, че ОМ са прилагани за моделирането на разнообразни приложни аспекти на образованието, например изпитване и оценяване на студенти, оценяване на преподаватели, използване на платформи за електронно обучение, управление на информационните и материалните потоци в дигитален университет, и др. Въпреки това доскоро не бяха правени опити за обратното: да се приложат добре утвърдени практики на преподаване и тестване към обучението по теорията на ОМ. Този факт отваря възможности за нови, плодотворни и интересни направления на изследване в областта.

В началото на 2010 г. в [69] авторът предложи новия, *дидактически*, аспект на теорията на ОМ. От една страна той има за цел да адресира и обоснове необходимостта от системен подход към обучението по ОМ като критична предпоставка за формирането на добри специалисти по този апарат за математическо моделиране, а от друга – да очертае посоките за развитие и на методологическия аспект на теорията на ОМ. Като първа стъпка в тази посока бяха предложени стандартизирани обучителни тестове, които взимат предвид особеностите на теорията на ОМ и нивото на настоящата студентска аудитория.

Тестовите бяха изпробвани в реални условия със студенти от Технически университет – София, Университет „Проф. д-р Асен Златаров“ – Бургас и Бургаски свободен университет, където от години се четат курсове по ОМ. За отделните типове задачи беше направен анализ на обучителните цели, следващ таксономията на В. Bloom, както и анализ на най-често допусканите грешки, съдържащ както полезна обратна връзка за провежданото обучение, така и идеи, свързани с методологическия аспект на теорията на обобщените мрежи. Резултатите от това проучване са публикувани в [75] и ще бъдат накратко представени по-долу.

През април 2010 г. към ядрото на Ifigenia.org беше инсталирано разширението *Quiz*, което позволява всички тези типове тестови задачи да бъдат интегрирани на практика в уикито и да се използват за провеждане на електронни тестове.

Когато се изготвят оценителни тестове за обучителните курсове по ОМ, следва да се спазват основните принципни препоръки за тестовите формати. Без претенции за изчерпателност, някои от най-популярните типове тестове са въпросите с множество отговори, въпросите от типа „истина или лъжа“, тест за попълване на пропуснати думи, тест за намиране на съответствия, въпрос със свободен отговор и други. Авторът смята тези тестови типове за подходящи средства за оценяване на обучаемите по ОМ.

Представен е списък на примерни тестови задачи от различните типове, описани в [69]. Всеки тип е представен с по три въпроса, като са дадени и верните отговори.

На базата на изследването на литературата по ОМ, може да се твърди, че графичните структури на ОМ моделите не играят централната роля в процеса на моделиране чрез ОМ. Те имат само спомагателна роля за илюстриране на модела, тъй като не съдържат и не представят цялата необходима информация за функционирането на модела, която информация се съдържа в съпътстващите индексирани матрици на предикатите и капацитетите на преходите, описанията на началните характеристики на ядрата и характеристичните функции, които определят следващите им стойности и т.н.

Графичната структура на обобщеномрежовия модел обаче е неговото най-непосредствено и потребителски привлекливо представяне, тъй като тя предлага с един поглед разбиране на последователността на модела и относителната му сложност, изразени с броя на преходите, позициите, дъгите (прави и обратни), както и възможните повтарящи се елементи, в които се съдържат потенциални възможности за опростяване и оптимизиране на модела.

Този визуален подход с възможността за непосредствено представяне и разбиране на теорията на ОМ е особено полезен при начинаещи обучаеми или такива, които нямат предварителни познания по изкуствен интелект, математическо моделиране или програмиране, но се нуждаят от този апарат за решаването на практически задачи от тяхната област на компетенции.

Дотук авторът е формулирал два вида задачи, целящи да се изпитат нивата на знаене и разбиране на обучаемите при изграждане на графичните структури на ОМ (Таблица 4.3., задача 3), и при прилагане на граф-операторите, дефинирани над ОМ.

Тези тестови задачи са били апробирани в реални изпити и резултатите са публикувани в [75]. За отбелязване е, че трите групи обучаеми са подредени според времето, когато са обучавани и изпитвани, и че обратната връзка от всяка група обучаеми е вземана предвид при обучението на всяка следваща група.

Резултатите от теста са оформени в таблици и е направен коментар относно най-типичните допускани от обучаемите грешки по двата вида задачи.

Особено предизвикателство би било сравняването на тестови резултати за една конкретна група обучаеми с представянето им при изграждане на реални ОМ модели, тъй като двете дейности оценяват различни когнитивни равнища, според йерархията на В. Bloom на образователните цели: знаене → разбиране → приложение → анализ → синтез → оценка. Използвайки тестове, могат да се адресират и оценят само трите най-ниски от общо шестте когнитивни равнища. За сравнение, процесът на конструиране на ОМ модел изисква в добавка и умения от трите най-високи когнитивни равнища.

С помощта на разширението *Quiz* на ядрото на софтуера на Ifigenia.org, *MediaWiki*, става възможна реализацията на всички описани типове тестови задачи в електронен тест, подходящ за самооценяване и оценяване. Следвайки документацията, накратко е представен синтаксисът на разширението *Quiz* за различните типове въпроси.

Разширението дава допълнителна възможност да се задават и въпроси, чиито отговори са точни числа или числа в определени интервали, или с +/- определено отклонение, зададено в проценти. Възможно е и фино регулиране на точките, които се дават или отнемат при верни/грешни отговори, като и настройка на тегловите коефициенти на отделните отговори в рамките на въпрос и на отделните въпроси в рамките на теста.

Понастоящем базата от данни на Ifigenia.org предлага свободен достъп до над 50 теми от понятийния апарат на ИРМ и ОМ и до повече от 300 публикации от тези области в PDF формат (част от които специално цифровизирани за качване в сайта, тъй като до този момент са били трудно достъпни, макар и основополагащи за изследванията в тези две научни области). В рамките на различни проекти с участието на автора тази база понастоящем е, а и в бъдеще ще бъде допълвана.

Синтезираният опит по изграждането на Ifigenia.org може да се използва при създаване и на други подобни уебсайтове, целящи да отговорят на потребността от информационни ресурси, разнообразие от услуги и насърчаване на общостен дух, [7].

Заклучение – основни резултати

Дисертационният труд включва дефиниции на две нови разширения на понятието „обобщена мрежа“ и доказателствата, че те са консервативни. Дадени са алгоритми за движение на ядрата в петте дефинирани до сега интуиционистки размити обобщени мрежи, както и решение на един открит проблем, съществуващ в теорията на обобщените мрежи повече от 20 години. Представени са възможностите, оптимизационните процедури по алгоритъма на мравките да се използват за оптимизиране на движението на ядрата в обобщени мрежи, както и обратно – чрез обобщени мрежи да се оптимизира работата на алгоритъма на мравките и генетичните алгоритми при решаване на конкретни проблеми.

При изготвянето на дисертационния труд се появиха редица идеи за бъдещи изследвания. Например, проблемът за представимостта на произволни обобщеномрежови преходи чрез композиция на $H_{2,2}$ преходи, от една страна, и проблемът за консервативността на разширенията на обобщените мрежи, от друга, навеждат на въпроса за възможностите да се конструират идентични („синонимни“) по обхват и действие обобщеномрежови модели, които обаче се различават по някои от своите елементи: инфраструктурни елементи (брой преходи и позиции, функции на приоритета и капацитета им), логически елементи (индексирани матрици с предикати за условията на преходите, тип на преходите), динамични елементи (ядра, начални стойности на характеристиките на ядрата и характеристични функции), темпорални елементи (времева скала, елементарна времева стъпка, продължителност на изчисление) и т.н. На практика въпросът се свежда до това, с кои от (незадължителните) компоненти в дефиницията на ОМ може да се направи компромис, в какви ситуации и на каква цена (от гледна точка на изчислително време и памет). Този проблем не е бил обект на разглеждане до момента.

Освен от чисто методологически интерес така поставеният въпрос е важен и от гледна точка на новопредложения дидактически аспект на теорията на обобщените мрежи. Изясняването му би дало възможност както за формиране на задълбочено разбиране у обучаеми и потенциални бъдещи специалисти по моделиране с обобщени мрежи за „относителната тежест“ на всеки от компонентите в дефиницията, така и за повече адекватност, гъвкавост и стандартизираност при изграждането на обобщеномрежови модели на реални процеси.

Приносителите в настоящия дисертационен труд са от научен, научно-приложен и приложен характер.

Научните приноси като цяло са ориентирани към дефиниране на нови разширения на понятието „обобщена мрежа“, доказване на тяхната консервативност и построяване на алгоритмите за движение на ядрата в тях на ниво отделен преход и цяла мрежа.

1. За съществуващите интуиционистки размити обобщени мрежи от всеки един дефинираните досега четири типа са построени липсващите до момента алгоритми за движение на ядрата в техните преходи и в целите мрежи.
2. Дефиниран е нов, пети, вид на интуиционистки размити обобщени мрежи. Доказана е теорема, че този вид обобщена мрежа е консервативно разширение на стандартното понятие за обобщена мрежа, т.е. функционирането и резултатите от работата ѝ може да се опише в термините на стандартна обобщена мрежа. Построени са алгоритмите за движение на ядрата в интуиционистки размитите обобщени мрежи от пети вид на ниво отделен преход и цяла мрежа.
3. Дефинирана е обобщена мрежа с обеми на ядрата. Доказана е теорема, че този вид обобщена мрежа е консервативно разширение на стандартното понятие за обобщена мрежа. Описани са съответните за това разширение два алгоритъма за движение на ядрата – на ниво отделен преход и на ниво цяла мрежа.
4. Предложено е цялостно решение на проблема за представимост на преходите в обобщените мрежи, формулиран през 1991 г., който изисква да се намери процедура, еднозначно представяща произволно сложен преход като композиция от преходи с два входа и два изхода. Доказано е, че предложеното решение е минималното възможно решение.
5. Формулирано е предложение как чрез присвояването на интуиционистки размити оценки на проведени обобщеномрежови симулации да се съставят прогнози за бъдещото изпълнение на обобщеномрежов модел.

Научно-приложните приноси са следните:

1. Предложена е универсалната обобщена мрежа за класа алгоритми за оптимизация по метода на мравките, описваща начина на функциониране и резултатите от тяхната работа. Построена е обобщена мрежа, която включва предложената мрежа като оптимизационен компонент, т.е. подмрежа, чиито резултати от работата определят оптималното движение на ядрата в новата мрежа.
2. Създаден е обобщеномрежов модел, отразяващ паралелната работа на реален процес, на алгоритъм на мравките и на генетичен алгоритъм. Съчетани в един хибриден метаевристичен алгоритъм, тези двата алгоритъма оптимизират работата на реалния процес в рамките на обобщеномрежовия модел.
3. Създадени са и два обобщеномрежови модела на основните алгоритми за предотвратяване на редакционни конфликти в системи с контрол на версиите, а именно „Заклучи–Промени–Отключи“ и „Копирай–Промени–Слей“.

Приложният принос се състои в предлагането на цялостно изложение за нов дидактичен аспект на теорията на обобщените мрежи и в показаните началните стъпки по реализацията му в уеб среда, изградена с технологията уики. За усилията си в тази посока докторантът получава II награда на VIII национална научно-практическа сесия на Федерацията на научно-техническите съюзи (2010 г.) и Наградата на БАН „Иван Евстратиев Гешов“ за най-млади учени до 30 години в областта на техническите науки (2011 г.).

Библиография

Номерацията в библиографията към автореферата следва номерацията в библиографията към дисертационния труд.

- [7] Атанасова, В. Уики технологии в помощ на науката. Примерът с Ifigenia.org, Сборник от 8^{ма} Национална младежка научно-практическа сесия на ФНТС, 10-11 май 2010, 83–88.
- [58] Atanassov, K., V. Atanassova, P. Chountas, A. Shannon. Generalized nets with places, having intuitionistic fuzzy capacities. *Notes on Intuitionistic Fuzzy Sets*, Vol. 17, 2011, No. 4, 21–28.
- [62] Atanassov, K., D. Dimitrov, V. Atanassova, Algorithms for tokens transfer in different types of intuitionistic fuzzy generalized nets. *Cybernetics and Information Technologies*, Volume 10, No 4, 2010, 22–35.
- [69] Atanassova, V., Design of Training Tests on Generalized Nets, Proc. of 5th International IEEE Symposium ‘Intelligent Systems’, 7–9 July 2010, London, UK, 327–330, 2010.
- [71] Atanassova, V. Generalized Nets with Volumetric Tokens, *Comptes rendus de l'Academie Bulgare des Sciences*, Vol. 65, 2012, No. 11, 1489–1498.
- [72] Atanassova, V., Generalized net models of conflict resolution approaches in version control systems. Part 1: Lock-Modify-Unlock, *Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*. Vol. II: Applications, 2010, 13–24.
- [73] Atanassova, V., On Intuitionistic Fuzzy Approach to Generalized Net Prognostics. In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*. Volume II: Applications. (K. Atanassov, M. Baczynski, J. Drewniak, J. Kacprzyk, M. Krawczak, E. Szmidt, M. Wygralak, S. Zadrozny, Eds.), SRI-PAS, Warsaw, 2012, 1–12, ISBN: 9788389475411
- [74] Atanassova, V. The Minimal Solution of a Problem in Generalized Nets, Proc. of the 6th IEEE Int. Conference on Intelligent Systems ‘2012, 6–8 Sept. 2012, Sofia, Bulgaria, Vol. II, 159–163.
- [75] Atanassova, V. Utilizing the Model Graphic Structures when Teaching Generalized Nets, In: *Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*, Vol. 2, 2010, 7–17.
- [77] Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov. A generalized net with an ACO-algorithm optimization component, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 7116, 2012, 190–197.
- [78] Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5. Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: *Monte Carlo Methods and Applications*, Proc. of the 8th IMACS Seminar on Monte Carlo Methods, August 29 – September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, 39–46.
- [79] Atanassova, V., Georgiev, P., Generalized net models of conflict resolution approaches in version control systems. Part 2: Copy–Modify–Merge, Proc. of 10th Int. Workshop on Generalized Nets, 5 Dec. 2009, 14–21.