

әл-Фараби атындағы Қазақ ұлттық университеті

ӘОЖ 004.4'423

Қолжазба құқығында

МУРЗАХМЕТОВ АСЛАНБЕК НУРБЕКОВИЧ

**Екі деңгейлі сөздік және геометриялық интерпретация негізінде
инновациялар жүйесін қалыптастыру және оңтайландыру**

6D070300 – Ақпараттық жүйелер

Философия докторы (PhD) дәрежесін
алу үшін дайындалған диссертация

Отандық ғылыми кеңесші:
физика-математика ғылымдарының докторы,
профессор м.а., Дюсембаев А.Е.
әл-Фараби атындағы Қазақ ұлттық
университеті, Алматы, Қазақстан

Шетелдік ғылыми кеңесші:
техникалық ғылымдарының докторы,
профессор Барахнин В.Б.
Ақпараттық және есептеу
технологиялары федералды зерттеу
орталығы, Новосібір, РФ

Қазақстан Республикасы
Алматы, 2022

МАЗМҰНЫ

НОРМАТИВТІК СІЛТЕМЕЛЕР	3
БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР	4
КІРІСПЕ	5
1 ИННОВАЦИЯНЫҢ ТАРАЛУ ЕРЕКШЕЛІКТЕРІ ЖӘНЕ НЕГІЗГІ ТАПСЫРМАЛАРЫ	11
1.1 Инновация диффузиясы теориясының жалпы түсініктері.....	11
1.1.1 Эверест Роджерстың диффузиялық моделі.....	13
1.1.2 Френк Басстың диффузиялық моделі.....	14
1.1.3 Naming Game моделі.....	15
1.2 Инновациялар жүйесін оңтайландырудағы қайта құрылымдау түсінігі...	17
1.2.1 Қайта құрылымдау әдістері.....	19
1.3 Бетті сақтау және оңтайландыру стратегиялары.....	22
1.3.1 Бетті алмастыру алгоритмдері.....	24
1.3.2 Инновациялар сөздігін оңтайландыру мәселелері.....	28
2 WORKING SET СТРАТЕГИЯСЫ НЕГІЗІНДЕ ИННОВАЦИЯЛАР СӨЗДІГІН ОҢТАЙЛАНДЫРУ	31
2.1 Working set стратегиясының сипаттамасы.....	31
2.2 Инновациялар сөздігін оңтайландыру алгоритмі.....	32
2.2.1 Инновациялар сөздігін оңтайландырудағы негізгі шектеулер.....	34
2.2.2 Беттерге және ұғымдарға сілтемелер жолы.....	37
2.2.3 Траекторияның бақылау күйі.....	40
2.3 Есептеу процесінің геометриялық интерпретациясы.....	41
2.3.1 Инновациялар сөздігін оңтайландыру тапсырмасының функционалын тұрғызу.....	47
2.3.2 Модельді оңтайландыру.....	51
2.3.3 Функционалды бағалау.....	52
3 ЭПИДЕМИЯ ҚАУШІН ТӨМЕНДЕТУ МАҚСАТЫНДА МУЛЬТИАГЕНТТІ ЖҮЙЕ ТОПТАРЫН ҚАЙТА ҚҰРЫЛЫМДАУ МОДЕЛІ	54
3.1 Агенттердің байланыс кезінде эпидемияны жұқтыруы мүмкіндігі бар мультиагентті жүйенің топтарын қайта құру моделін тұрғызу.....	54
3.2 Әлеуметтік қоғамда инновацияның таралу ақпараттық жүйесін сипаттау	59
3.2.1 Әлеуметтік қоғамда инновацияның таралуын имитациялық модельдеу	61
3.2.2 Жасушалы автомат кеңістігінде инновацияның таралуын талдау.....	64
ҚОРЫТЫНДЫ	69
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	71
ҚОСЫМША А - Авторлық құқық куәліктері	81
ҚОСЫМША Ә - Енгізу актілері	86
ҚОСЫМША Б - Бағдарлама листингі	88

НОРМАТИВТІК СІЛТЕМЕЛЕР

Диссертацияда келесі стандарттарға сәйкес сілтемелер қолданылды:
Қазақстан Республикасы МЖМБС 5.04.034-2011. «Қазақстан Республикасы Мемлекеттік жалпыға білім беру стандарты. Жоғары оқу орнынан кейінгі білім. Докторантура.» Негізгі ережелер ҚР БҒМ бекітілген. 17.06.2011 ж. №261. Астана, 2011 ж.

«Диссертацияны безендіру нұсқаулығы», Қазақстан Республикасы БҒМ ЖАК 28 қыркүйек 2004 жыл. №377-3 ж.

ГОСТ 7.32-2001. Ғылыми зерттеу жұмысының есебі. Безендіру ережелері мен құрылымы.

ГОСТ 7.1-2003. Библиографиялық таспа. Библиографиялық сипаттау.

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

WS	– Working set (жұмысшы жиын)
CPU	– Central process unit
FIFO	– First in last out
LIFO	– Last in first out
LRU	– Least recently used
PFF	– Page fault frequence
NFU	– Not frequently used
NRU	– Not recently used
БАҚ	– Бұқаралық ақпарат құралдары
NG	– Naming Game
SVM	– Support vector machine
ОЖ	– Операциялық жүйе
I/O	– Input /Output
РҒА СБ	– Ресей ғылым академиясының Сібір бөлімі
DICR	– Distributed information-computer resources (Үрестірілген ақпараттық-есептеуіш ресурстар)

КІРІСПЕ

Зерттеу тақырыбының өзектілігі. Қазіргі таңда білім мен ғылым саласындағы серпілістер және өзгерістер оларды жаңа көз қараспен қарауға және дамудың жаңа деңгейіне өтуге себепші болып отыр. Мұндай серпіліс пен өзгерістердің бірден бір себебі түрлі салаларда инновациялық технологияларды кеңінен пайдалану болып табылады. Инновацияның қалыптасу және таралу механизмі бұрыннан белгілі болғанына қарамастан, бұл саладағы зерттеулер өзекті болып табылады және кейбір мәселелер жеткілікті түрде шешілмегендіктен оларды терең зерттеу қажет. Сонымен қатар, инновациялар жүйесіндегі жаңа тұжырымдар мен құралдардың пайда болуына байланысты, олар үнемі одан әрі дамытуды талап етеді. Инновация ұғымын американдық ғалым Й. Шумпетер өзінің «Экономикалық даму теориясы» еңбегінде алғаш рет «жаңашыл» және «жаңалық» түсінігі ретінде енгізді. Содан кейін түрлі ғалымдар біршама зерттеулер жүргізіп, инновацияға өздігінше ұғымдар бере бастады. Атап айтсақ, Ф. Валента және Л. Волдачек «инновация бұл – өзгеріс» десе, Э.Уткин инновацияны нақты «объект» деп санайды. Ал, Ф. Никсон – «іс-шаралардың жиыны», С. Валдайцев «жаңа өнімді игеру» деген ұғым берген [1]. Жалпы айтқанда, инновация – бұл әлеуметтік құбылыс, яғни адамдардың қандай да бір қоғамда қалыптасқан дәстүрлерді өзгертуге, жаңалықтарға ұмтылу, бір кезеңнен екіншісіне өту құлқын анықтайды. Қазақ тілінде осы түпнұсқа қолданылып келеді [2].

Инновацияның қалыптасуын, таралуын және қоғамға әсер етуі ықпалын зерттеу бойынша қазіргі таңда біршама тәжірибе жинақталған. Шетелдік ғалымдар Э. Роджерс [3] және Ф. Басс [4] ХХ ғасырдың 60-шы жылдары әлеуметтік жүйелерге инновациялардың ену үрдістерін және қоғамдағы өзгерістерді түсіндіретін математикалық модельдер және тұжырымдарды ұсынды. П. Дойчманн, У. Даниельсон [5], А. Барончелли [6], А.М. Федотов [7], Ф.А. Мурзин [8], В.Б. Барахнин [9] сынды ғалымдар инновациялардың таралу үрдістерін, оның ықпалын және сөздік жұмысын зерттеді. Сонымен қатар, отандық ғалымдар А.Е. Дюсембаев [10], Ж.А. Тусупов [11] зерттеулерінде сөздік жұмысы қарастырылған. Алайда бұл зерттеулерде инновациялар сөздігінің жұмысын оңтайландыру тапсырмасы туралы айтылмаған. Мұндай тапсырма үлкен қызығушылық тудырады, әсіресе сөздік көлемі үлкен болған жағдайда. Себебі, сөздік бірнеше ондаған микросекунд ішінде жүздеген немесе одан да көп машиналық нұсқауларды орындайды. Бетті жүктеу уақыты бірнеше ондаған миллисекундқа жуық. Мұндай жағдайлар сөздік беттерінде өзара байланысқан сілтемелердің қателігіне, яғни бет қателігіне әкелуі мүмкін. Бет қателігінің болуы жады жүйесінің өнімділігін 98% дейін төмендетуі мүмкін. Сондықтан беттік жадыны басқару жүйесінің, оның ішінде инновациялар сөздігінің басты міндеттерінің бірі – бет қателігінің пайда болу жиілігін азайту. Оның шешімі әдетте бетті алмастыру алгоритмін дұрыс таңдаумен байланысты. Осындай шешімдердің бірі – сегментациялау тапсырмасы үшін бағдарламалық кодты қайта құру арқылы WS (Working set, қазақша «жұмысшы жиын») стратегиясын

қолдану (А.Е. Дюсембаев). Оңтайландыру процесін басқару теориясында жұмысшы жиын ұғымы ерекше рөл атқаратындығына күмән жоқ, сондықтан WS стратегиясы қайта құрылымдау мәселесін шешуде ерекше қызығушылық тудырады. Осы себепті, WS стратегиясы ерекше назарға ие. Қайта құрылымдау тапсырмасы әдетте бір немесе бірнеше мақсатқа сәйкес процестерді бөліктерге (блоктар, бөлімдер, беттер және т.б.) бөлу немесе ұйымдастыру болып табылады. Таңдалған тақырыптың өзектілігі келесі мәселелерге негізделеді:

- инновациялар сөздігі үшін оңтайландыру тәсілдері мен модельдердің жоқтығы;
- түпкілікті шешімдердің болмауы (refactoring, restructuring);
- инновациялар сөздігінің үлкен өлшемдігі (Big Data мәселесі);
- сөздіктегі ұғымдардың нашар орналасуы инновациялар жүйесінің өнімділігін төмендеуіне әкелуі мүмкін;
- жүйе өнімділігінің төмендеуі бет қателігінің көптігінен болуы мүмкін;
- агенттер арасында эпидемия қауіпін азайту мақсатында мультиагентті жүйені оңтайландыру моделін құру мүмкіндігі;
- алғаш рет инновациялар сөздігінің жұмысын оңтайландыру тапсырманы қойылды және алғаш рет есептеу процесіннің геометриялық интерпретациясы негізінде модельді оңтайландыру мүмкіндігі көрсетілді.

Осыған байланысты диссертацияда жүргізілген зерттеулер өзекті болып табылады.

Зерттеу жұмысының мақсаты: инновациялар жүйесін қалыптастыру және оңтайландыру, оңтайландырылған жүйені мультиагентті жүйенің топтарын қайта құрылымдау тапсырмасына қолдану.

Зерттеудің тапсырмалары: Қойылған мақсатқа жету үшін диссертациялық жұмыста келесі тапсырмаларды шешу қарастырылады:

- инновациялар жүйесін сипаттау бойынша әдебиеттер мен технологияларды талдау және инновацияның қалыптасу механизмін анықтау;
- инновациялар жүйесін қалыптастыру;
- есептеу процестері мен геометриялық интерпретацияға негізделген инновациялар сөздігін оңтайландыру моделін құру;
- құрылған модельді мультиагентті жүйенің топтарын қайта құрылымдау тапсырмасына қолдану;

Зерттеу нысаны: инновациялар жүйесі, инновациялар жүйесін оңтайландыру, мультиагентті жүйе.

Зерттеу пәні: инновациялар жүйесін қалыптастыру, инновациялар сөздігін оңтайландыру моделі және оңтайландыру моделін қолдану.

Зерттеу әдісі: зерттеуде қойылған тапсырмаларды шешу үшін диссертацияда келесі әдістер қолданылады: дискретті оңтайландыру әдістері, стахостикалық бағдарламалау әдістері, операцияларды зерттеу әдістері, геометриялық интерпретация принциптері.

Ғылыми жаңалығы:

1. Инновациялар жүйесін қалыптастыру;

2. есептеу процестері мен геометриялық интерпретация негізінде инновациялар сөздігін оңтайландыру моделі;
3. құрылған модельді мультиагентті жүйенің топтарын қайта құрылымдау үшін қолдану.

Жұмыстың теориялық және практикалық маңыздылығы:

Диссертацияда қойылған және шешілген тапсырмалар ақпараттық жүйелердің іргелі бағытына жатады. Жұмыстың практикалық маңыздылығы инновациялар сөздігінің жұмысын оңтайландыру кезінде құрылған модельдерді іс жүзінде қолдануға болатындығымен анықталады. Сонымен қатар, инновациялар сөздігін қалыптастыру және оңтайландыру мәселесі үшін де маңызды. Себебі мұндай сөздік үлкен өлшемді болу мүмкін, бұл өз кезегінде үлкен деректер мәселесіне алып келеді. Яғни, сөздік жұмысын оңтайландыру тапсырмасы заңды болып табылады. Біздің жағдайда инновациялар сөздігі екі деңгейлі: сөздіктің бірінші деңгейі – бұл статикалық жады, мұнда сөздіктің өзі орналасқан және сөздік инновациялар немесе ұғымдар орналасқан беттерге бөлінген. Сөздіктің екінші деңгейі – динамикалық жады, оның беттерінде статикалық жады беттерінің көшірмелері орналасқан. Инновацияларды қолданылтын сала аясында анықтау, оны кластерлеу және деңгейі бойынша сипаттама беру жоғарыда айтылған мәселелерді шешу жолын біршама жеңілдетпек. Диссертациялық жұмысты орындау барысында инновациялар сөздігін оңтайландырудың сызықтық емес моделі, соның ішінде функционал және шектеулер жүйесі анықталды. Зерттеу барысында алынған нәтижелер құрылған модельдің ерекшеліктерін ескеретін практикалық оңтайландыру алгоритмін құруға негіз береді және туыстас есептерді шешу үшін қолданыла алады.

Жұмыстардың едәуір бөлігі инновацияның диффузия құбылысын зерттеуге және қолданбалы аспектілерге арналған. Яғни, алынған нәтижелер инновацияның қалыптасу механизмін зерттеуге, қоғамдағы инновацияның қалыптасуын және таралуын, саяси пікірлердің, идеялардың туындауын және таралуын және олардың қоғамға әсерін сараптау барысында қолдануға мүмкіндік береді.

Зерттеушінің жеке үлесі: Зерттеуде қойылған мақсат пен тапсырмаларды шешу көп еңбектенуді қажет етті. Зерттеу жұмыстары әл-Фараби Қазақ ұлттық университетінің базасында, Ақпараттық және есептеу технологиялары федералды зерттеу орталығының (Новосібір, Ресей) базасында жүргізілді. Диссертациялық жұмыста инновациялар сөздігінің жұмысын оңтайландыру зерттелді және математикалық моделі құрылды, зерттеулер аясында инновация түсінігі енгізілді және инновацияның қалыптасу механизмі сипатталынды, бірегей комбинаторлық кеңістігін пайдалана отырып есептеу процесінің геометриялық интерпретациясы табылды және инновациялардың таралу эксперименті сипатталынды.

Қорғауға шығарылған негізгі тұжырымдар:

– алғаш рет жұмысшы жиын стратегиясы негізінде инновациялар сөздігінің жұмысын оңтайландыру тапсырмасы қойылды;

– біздің зерттеулер аясында инновация түсінігі енгізілді және инновацияның қалыптасу механизмі сипатталынды;

– инновациялар сөздігін оңтайландырудың математикалық моделі құрылды:

а) инновациялар сөздігін оңтайландыру тапсырмасының функционалы (соның ішінде $h \geq 1$ эксперименті бойынша эмпирикалық функционал) және шектеулер жүйесі табылды;

б) бірегей комбинаторлық кеңістік (Булеан) шыңдары бойынша кездейсоқ кезу ретінде есептеу процесінің геометриялық интерпретациясы табылды;

с) есептеу процесінің геометриялық интерпретациясы негізінде инновациялар сөздігін оңтайландыру тапсырмасының өлшемін азайту мүмкіндігі көрсетілді;

– оңтайландыру тапсырмасының бағалау функционалы құрылды;

– эпидемия қауіпін төмендету мақсатындағы мультиагентті жүйе топтарын қайта құрылымдау тапсырмасы үшін қолданылуы көрсетілді.

Ғылыми жұмыс нәтижелерін апробациялау: Зерттеу тақырыбы бойынша ғылыми нәтижелері келесідей халықаралық ғылыми конференцияларда жарияланды: ISIS 2017 «The 18th International Symposium on Advanced Intelligent Systems», Тэгу, Оңтүстік Корея, 2017; «Integration of the scientific community to the global challenges of our time», Осака, Жапония, 2017; «Әуезов оқулар», Шымкент, 2017; «Интеллектуалдық ақпараттық және коммуникациялық технологиялар - «Қазақстан – 2050» стратегиясы аясында үшінші индустриалды революцияны жүзеге асырудың құралы», Астана, 2016; «Ғылым және Білім», Астана, 2015. Сонымен қатар, диссертациялық жұмыс нәтижелері әл-Фараби атындағы Қазақ ұлттық университетінің «Ақпараттық технологиялар» факультетінде, «Ақпараттық жүйелер» кафедрасында және Ақпараттық және есептеуіш технологиялар институтының ғылыми семинарларында талқыланды.

Диссертациялық жұмысты орындау барысында 20 мақала жарияланды, соның ішінде Scopus және Web of Science базаларына енгізілген журналдарда 4 мақала, Қазақстан Республикасы Білім және Ғылым министрлігінің білім және ғылым саласындағы бақылау комитеті ұсынған журналдарында 5 мақала, халықаралық конференциялар жинақтарында 5 мақала, халықаралық журналда 1 мақала жарияланды және зияткерлік меншік нысанын енгізу бойынша 5 куәлік алынды.

Жарияланымдар:

1. Fedotov A.M., Barakhnin V.B., Murzakhmetov A.N., Milyuk I.R. Modelling of process information dissemination and its impact dynamics to mass consciousness. Journal of Theoretical and Applied Information Technology. 15th December 2020. Vol.98. No 23. pp. 3691-3702
2. Fedotov A.M., Murzakhmetov A.N., Dyusembaev A.E. Expansion of ideas and processes in social and biological communities. Eurasian Journal of Mathematical and Computer Applications. 2018, Vol. 6, iss. 4, pp. 17-28.

3. Murzakhmetov A.N., Dyusembayev A.E., Umbetov U.U., Abdimomynova M.M., Shekeyeva K. Study of the innovations diffusion on the base of Naming Game mathematical model. *Compusoft an International Journal of Advanced Computer Technology*. 2020, Vol. 9(1), pp. 3547-3551.
4. Grishko M.V., Murzakhmetov A.N. Making investment decisions based on econometric analysis in the conditions of Kazakhstan stock market // *Bulletin of the National academy of sciences of the Republic of Kazakhstan*, Iss. 2, Mar 2015, pp. 250-256.
5. Fedotov A.M., Murzakhmetov A.N., Dyusembayev A.E., Grishko M.V. Model of innovation diffusion in the social systems under the influence of the media and interpersonal communication. *ISIS 2017-The 18th International Symposium on Advanced Intelligent Systems*, October 11-14, 2017, EXCO in Daegu, South Korea. pp. 916-922.
6. Fedotov A.M., Murzakhmetov A.N., Dyusembayev A.E., Grishko M.V. To Analysis of the Model of Innovation Diffusion in the Social Systems under the Influence of the Media and Interpersonal Communication. *INFORMATION*, March 2018. Vol. 21, iss. 3, pp. 1187-1196.
7. Murzakhmetov A.N. Research Imitation Multiagent Models of the Securities Market. II International scientific-practical conference "INTEGRATION OF THE SCIENTIFIC COMMUNITY TO THE GLOBAL CHALLENGES OF OUR TIME" Vol.2, March 7-9, 2017, Osaka, Japan. pp. 46-52.
8. Umbetov U.U., Murzakhmetov A.N. Modeling the influence of internet news on the decision-making process of investors. *Proceedings of international scientific-practical conference "Auezov Readings, 3-14 April Shymkent, 2017. Vol 5*, pp. 318-321.
9. Мурзахметов А.Н., Федотов А.М., Гришко М.В., Дюсембаев А.Е. Әлеуметтік-экономикалық қоғамдарда инновацияның таралуын модельдеу. *ҚР ҰҒА Хабарлары, физика-математика сериясы, №6 (316)*, 2017, 39-44 б.
10. Мурзахметов А.Н. Құнды қағаздар қоржынның тиімді аумағын есептеу жүйесі. *ҚазККА Хабаршысы №2 (101)*, 2017, 241-248 б.
11. Мурзахметов А.Н. Интернет жаңалықтарының мәтіндерін автоматты талдау моделі. *Д. Серікбаев атындағы ШҚМТУ хабаршысы, математика, физика, химия, информатика сериясы, №2*, 2017, 112-116 б.
12. Мурзахметов А.Н., Бапанов А.А. Прогнозирования на основе эконометрических модели. «Интеллектуалдық ақпараттық және коммуникациялық технологиялар - «Қазақстан – 2050» стратегиясы аясында үшінші индустриалды революцияны жүзеге асырудың құралы» III Халықар.ғыл. – практ. конфер. Баяндамалар жинағы. 2-3 маусым 2016 жыл. Астана, 2016.
13. Бапанов А.А., Мурзахметов А.Н. Современные классификаций онтологий. Сборник материалов X Международной научной конференции студентов и молодых ученых "Наука и образование - 2015", Астана, 2015. стр. 425-427.

14. Мурзахметов А.Н., Бейсов Н.К. Жасушалы автомат кеңістігінде ақпараттың таралуын талдау. ҚазҰТЗУ хабаршысы, техникалық ғылымдар сериясы, №6 (136), 2019, 462-465 б.
15. Murzakhmetov A.N., Infante Moro A., Dyusembayev A.E. Social media and networks as a tool to analysis of tourists preferences and requirements. Vestnik KazNRTU, №2 (138), 2020, pp. 227-231.
16. Гришко М.В., Мурзахметов А.Н., Дюсембаев А.Е. Модель принятия решений для Казахстанского фондового рынка на основе эконометрического анализа. Свидетельство депонирования объектов интеллектуальной собственности, авторское произведение. Регистрационный №002334, 29 апреля 2015 г.
17. Мурзахметов А.Н., Дюсембаев А.Е. Построение модели оптимизации работы словаря инновации на основе стратегий WS. Авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізімге мәліметтерді енгізу туралы куәлік, 2020 жылға 30 желтоқсан, №14230.
18. Дюсембаев А.Е., Мурзахметов А.Н. Особенности модели оптимизации работы словаря инноваций при стратегии рабочего множества. Авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізімге мәліметтерді енгізу туралы куәлік, 2021 жылғы 29 сәуір, №17094.
19. Дюсембаев А.Е., Мурзахметов А.Н. Построение модели реорганизации групп мультиагентной системы с возможным инфицированием агентов при контактах. Авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізімге мәліметтерді енгізу туралы куәлік, 2022 жылғы 21 ақпан, №23780.
20. Дюсембаев А.Е., Мурзахметов А.Н. Информационная система формирования и оптимизации словаря инноваций. Авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізімге мәліметтерді енгізу туралы куәлік, 2022 жылғы 28 наурыз, №24565.

Диссертацияның құрылымы және көлемі: Диссертация кіріспеден, үш бөлімнен, қорытынды, әдебиеттер тізімі және қосымшалардан тұрады, жалпы саны 103 бет машиналық мәтіні, оның ішіне 22 сурет, 159 пайдаланылған әдебиеттер және 15 бет қосымша кіреді.

1 ИННОВАЦИЯНЫҢ ТАРАЛУ ЕРЕКШЕЛІКТЕРІ ЖӘНЕ НЕГІЗГІ ТАПСЫРМАЛАРЫ

Бұл бөлімде инновацияның таралу ерекшеліктері, жалпы тұжырымдары және инновацияның таралу модельдері, сонымен қатар, алмастырудың кейбір әдістері қарастырылған.

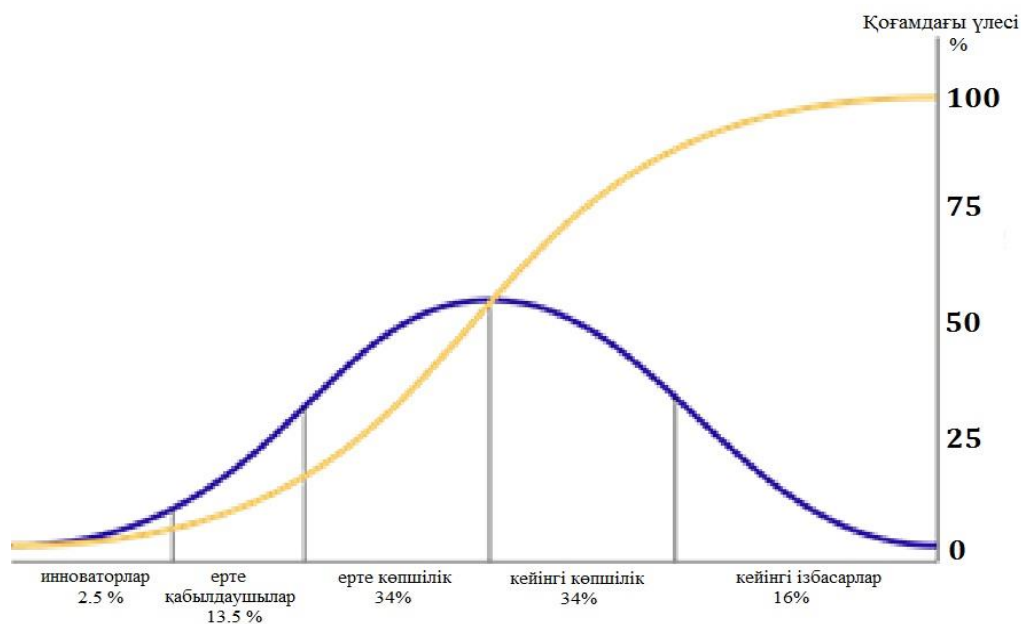
1.1 Инновация диффузиясы теориясының жалпы түсініктері

Әлеуметтік қауіпсіздікті анықтайтын процестердің динамикасын сипаттау үшін әлеуметтік жүйедегі тиісті процестердің таралу жылдамдығын ескеру қажет. Әлеуметтік жүйеде, соның ішінде әлеуметтік қауіпсіздікті анықтайтын процестердің кейбірін ретінде сипаттауға болады. Жалпы, инновация дегеніміз – жаңашылдықты, идеяны, тәжірибе немесе объекті сезінетін жеке тұлға немесе топ» [12]. Жаңашылдық дегеніміз - белгілі бір әлеуметтік пән үшін жаңа идеялар, өнімдер, шешімдер, технологиялар және т.б. Диффузия - бұл уақыт өте келе әлеуметтік жүйенің мүшелері арасында белгілі бір арналар арқылы инновацияның таралу процесі. Сонымен қатар, диссертациялық жұмыста ақпарат ретінде, инновация, идея және пікір сияқты түніктерді білдіреді.

Инновация диффузиясының теориясы әлеуметтік жүйеде инновациялардың таралуын қарастырады. Осы саланың зерттеушілері әлеуметтік жүйенің мүшелері жаңашылдықты қандай ықтималдылықпен қабылдайтындығын және оның таралу жылдамдығын түсіндіруге тырысуда. Инновацияны әлеуметтік жүйеге инноваторлар енгізеді, содан кейін біртіндеп инновация туралы ақпаратты бір-біріне беретін көптеген агенттер қабылдайды, осыған байланысты әлеуметтік жүйенің табиғаты маңызды рөл атқарады. Инновацияның таралуына көптеген факторлар әсер етеді: агент сипаттамалары, инновациялық сипаттамалар және әлеуметтік жүйенің сипаты. Диффузияны қарапайым түрде зерттеу - бұл әлеуметтік жүйе агенттерінің инновацияны қабылдауға әсер ететін осы және басқа факторлардың өзара әрекеттесуін зерттеу.

Инновациялардың диффузиясы туралы тұжырымдама 19 ғасырда өмір сүрген француз ғалымы және әлеуметтанушысы Габриэль Тардтың теориялық жұмыстарына негізделген. Г. Тард өзінің «Имитация заңдары» кітабында [13] инновацияларды қабылдау заңдылықтарын сипаттайтын *S* тәрізді қисық тұжырымдамасын алға тартты және қоғамдық пікірдің көшбасшылық тұжырымдамасын ұсынды.

Әлеуметтік өзгерістердің таралу процесінің динамикасы дәстүрлі түрде *S*-тәрізді қисықпен модельделеді және келесідей кезеңдерден тұрады: инноваторлар (инновацияны бірінші қабылдап және қолдана бастаған инноваторлар), ерте қабылдаушылар (инновацияны көп ұзамай қабылдап, қолдана бастаған адамдар), ерте көпшілік (инноваторлар мен алғашқы ізбасарлардан кейінгі инновацияны қабылдаушылар), кейінгі көпшілік (инновацияның кеңінен таралғаннан кейін барып қабылдағандар) және кейінгі ізбасарлар (соңғы болып қабылданған). Аталған топтар 1.1-суретте көрсетілген:



Сурет 1.1 – Инновацияның таралу динамикасы

Инновацияның негізгі сипаттамаларын және оны таралуымен байланысты контекстті қарастырайық. Әлеуметтік жүйенің әрбір мүшесінде (агент) инновацияны қабылдау және қолдану туралы шешім қабылдау қажеттілігі туындайды. Бұл жағдайда инновацияны қабылдау процесі келесі кезеңдерден өтеді:

- 1) білім - агент инновациямен таныс, бірақ ол туралы толық ақпараты жоқ;
- 2) сенімділік - агент қызығушылық танытады, инновацияға жағымды немесе қолайсыз қатынасты қалыптастырады, қосымша ақпаратты іздейді;
- 3) шешім - агент инновацияны қабылдаудың артықшылықтары мен кемшіліктерін өлшейді, оны пайдалану, қабылдау немесе қабылдамау туралы шешім қабылдайды;
- 4) тестілеу/енгізу - агент инновацияны қолданады;
- 5) растау - агент нәтижелерді бағалайды және одан әрі пайдалану туралы шешім қабылдайды.

Инновацияның таралуын модельдеу барысында көптеген зерттеушілер эпидемиялардың таралу динамикасы теориясынан алынған әдістерді қолданады. Көптеген танымал модельдер келесі болжамдарға негізделген:

- 1) әлеуетті тұтынушылардың саны (яғни нарық өлшемі) белгіленген, бірақ әр сәтте барлық тұтынушыларды екі топқа бөлуге болады - инновацияны қолданып жүргендер және оны әлі қабылдамағандар;
- 2) жаңа өнім туралы ақпарат тұтынушылар арасындағы жеке байланыстар арқылы ғана таратылады;
- 3) уақыт өте келе инновация өзінің әлеуетті нарығын толығымен толтырады;
- 4) диффузия жылдамдығына қазіргі таралу деңгейі әсер етеді, яғни. инновацияны қабылдаған адамдардың саны және инновацияны қолданудың қазіргі және шекті деңгейлері арасындағы алшақтық.

1.1.1 Эверест Роджерстың диффузиялық моделі

Роджерс өзінің "Диффузия инноваций" атты жұмысында түрлі инновациялардың қоғамда қабылдану деңгейлерін зерттеді. Тардтың S-тәрізді қисық сызық тұжырымдамасын Э. Роджерс былай сипаттайды: «Біріншіден, жаңа идеяны санаулы адамдар ғана қабылдайды, содан кейін инновацияны көптеген адамдар қабылдайды, соңына келгенде, қабылдау қарқыны бәсеңдейді». Классикалық сипатқа ие болған бұл монографияда Роджерс белгілі уақыт аралығында инновациялардың таралуына байланысты мыңдаған ғылыми жұмыстарды талдап, олардың барлығында келесідей заңдылықтар болған:

- 1) кез-келген жаңашылдық;
- 2) ақпараттың адамнан адамға берілуі;
- 3) қоғамдық немесе әлеуметтік орта;
- 4) уақыт факторы.

Оның зерттеу нәтижелері бойынша қоғам мүшелерінің шешім қабылдау графигі 5 бөлікке бөлінген қоңырау тәрізді қисыққа (қалыпты үлестірім) ұқсас. Э.Роджерс стандартты ауытқуларды ескере отырып әрбір сегментке атау және шамалас бағасын берді: инноваторлар (тәуекелге бейім, барлық жаңашылдыққа ашық жеке тұлғалар) - 2,5%; ерте ерушілер (инновация жайлы ақпарат көзі, олардың пікірімен санасады) - 13,5%; ерте көпшілік (жаңа идеяларды және технологияларды бірінші болып қабылдағысы келмейтін адамдар санаты) - 34%; кеш көпшілік (инновацияның артықшылығына сенбей, шүбә көзбен қараушылар) - 34%; артта қалушылар (өткенге, тарихқа байланған және қандай да бір жаңашылдықты баяу қабылдайтын қоғам мүшелері) - 16% [14].

Э. Роджерс мәлімдеуінше инновациялардың диффузия теориясы жалпы психологиямен, атап айтқанда әлеуметтік оқыту теориясымен тығыз байланысты. Э. Роджерс сонымен қатар 500-ден астам диффузиялық процестерді талдай отырып инновацияларды қабылдау процесінің келесі бес кезеңін анықтады [15]:

- 1) хабардар болу - жеке тұлға жаңа идея туралы біледі, бірақ ақпарат жеткіліксіз;
- 2) қызығушылық - жеке тұлға идеяға қызығушылық танытады және қосымша ақпарат іздейді;
- 3) бағалау - жеке тұлға инновацияның қазіргі және болашақтағы пайдасын бағалай отырып шешім қабылдайды;
- 4) тестілеу;
- 5) ассимиляция.

Диффузия жылдамдығы, Э. Роджерс бойынша, бес негізгі фактормен анықталады:

- 1) жаңа заттардың салыстырмалы артықшылықтары;
- 2) қоршаған ортамен, қолданыстағы құндылықтармен және өткен тәжірибемен үйлесімділік;
- 3) дамудың күрделілігі;
- 4) түпкілікті шешім қабылдағанға дейін тестілеу мүмкіндігі;

5) коммуникациялық көріну - инновацияның нәтижелерін басқа адамдар көре алатын және бағалай алатын дәреже.

Инновациялардың диффузия теориясына сәйкес кез-келген инновация (мысалы, жаңа идея, техника, технология) таралады, яғни ол белгілі бір болжамды модельге сәйкес қоғамда таралады. Графикалық түрде бұл модель S-тәрізді қисық түрінде ұсынылған.

А. Бандураның оқыту теориясына сәйкес [16], инновациялардың таралуы символдық модельдеу, сендіру, әлеуметтік ояту және мотивация сияқты ұғымдардың призмасы арқылы қарастырылады. Инновацияның таралуы және қабылдау процестеріне келетін болсақ, зерттеулер нәтижесі бойынша, инновациялар туралы ақпарат қоғамдық байланыс арналары арқылы таратылатындығын көрсетті. БАҚ тұлғааралық қарым-қатынасқа қарағанда әлдеқайда жылдам. Дегенімен, тұлғааралық қарым-қатынас көбінесе бұқаралық ақпарат құралдары ұсынатын жаңа идеяны немесе басқа инновацияны қабылдауға әсер ететін негізгі факторлардың бірі болып табылады. БАҚ-ты зерттеумен айналысатын зерттеушілер инновациялар диффузиясының бүкіл процесін іс жүзінде аяқтайтын төрт негізгі кезеңді анықтайды. Олар: білім, сендіру, шешім және оның дұрыстығын растау. Жеке авторлар жаңалықтардың немесе ақпараттың таралуын зерттеді және дәл осы зерттеулер медиа әсерін түсіну үшін ерекше қызығушылық тудырады. П. Дойчман және В. Дэниэлсон [5, р. 348-355] жаңалықтардың диффузиясын зерттей келе, жаңалық басқа жаңалықтарға қарағанда тезірек тарала бастайды.

Бірінші және негізгі жіктеу параметрі – бәсекелес инновациялардың саны (модель өлшемі). Бір инновацияның таралуын немесе екі немесе одан да көп инновациялардың бәсекелестік сипаттамасын қарастыруға болады. Егер бір тәуелсіз инновация зерттелетін болса, онда бұл диффузиялық модельдер, егер n ($n = 2, 3, \dots$) - алмастыру моделі.

Келесі параметр - инновация идеяларын қолдану, еліктеу және олардың инновациялардың таралуын зерттеуде үйлесуі. Жаңашылдық пен еліктеу арасындағы айырмашылықты алдымен Шумпендер жасады, кейіннен оның идеяларын Э. Мэнсфилд ресімдеді [17, 18].

«Инновация» фирманың (немесе әлеуетті тұтынушының) басқа фирмалардың (тұтынушылардың) іс-әрекетін ескерместен инновацияны қабылдау қабілеттілігін, ал «еліктеу» - инновацияның таралуына қарай қарыз алу жылдамдығын көрсетеді.

1.1.2 Френк Басстың диффузиялық моделі

Басс моделі бойынша инновациялық өнімді қабылдаушылар санының артуы екі әсермен түсіндіріледі:

- жарнама әсері;
- тұлғааралық коммуникация әсері.

Роджерске қарағанда Басс адамдар санатын 5 емес тек 2 ғана бөлді: 1) инновацияны бірінші кезекте өздері татып көретін және ол жайлы түрлі хабарламалардан білетін адамдар; 2) инновацияны бірінші санаттағы адамдардан

біліп, олардың пікіріне құлақ асатын адамдар. Мұнда жарнаманың әсері едәуір төмендейді. Көп жағдайда бірінші санаттығы адамдардың пікірі қызық, себебі новатор үшін бүгінгі қалыпты жағдай ертен көпшілік үшін қалыпты жағдайға айналады. Төменде Басс моделі көрсетілген:

$$\frac{dF}{dt} = \left(p + q \frac{F(t)}{N} \right) (N - F(t))$$

мұнда $F(t)$ – t уақыт аралығына дейін инновацияны қабылдаған агенттер саны;

N – максималды мүмкін болатын агенттер саны;

p – сыртқы әсер ету коэффициенті (жарнама, БАҚ);

q – ішкі әсер ету коэффициенті (тұлғааралық коммуникация) [19].

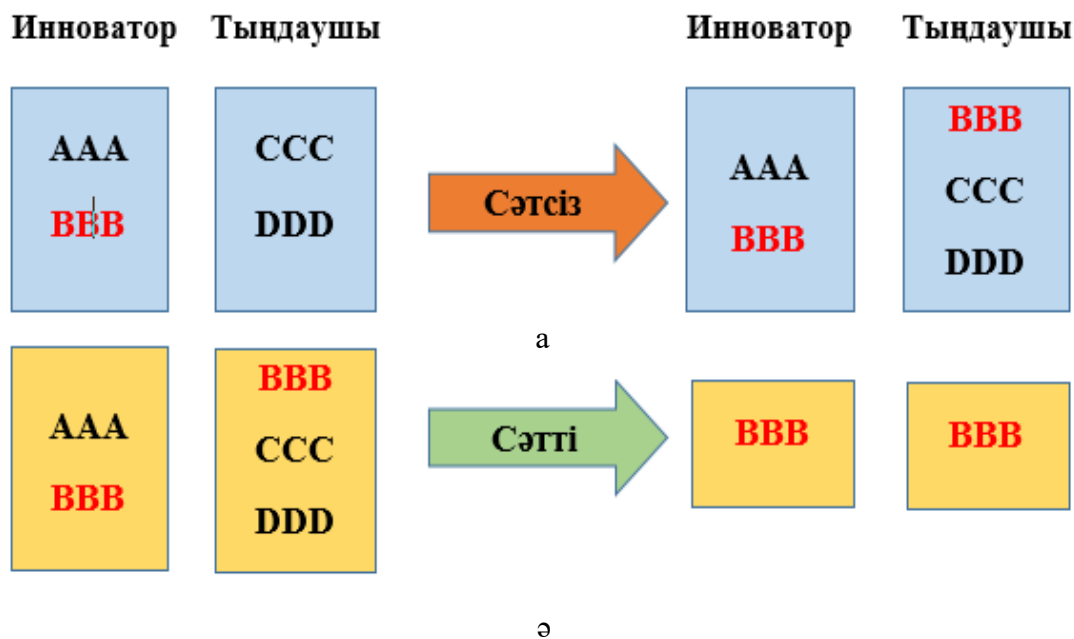
Басс моделі біршама кеңейтулерге ие, дегенімен өзінің бастапқы түрінде ол агенттердің кері өсімін және агенттер аудиториясына бәсекелестікті ескеру мүмкіндігіне қабілетсіз.

1.1.3 Naming Game моделі

Интернеттің қарқынды дамуымен біз түрлі құрылғылар арқылы ғаламдық желіге қосылып, адамдар бір біріне жақындай түсті. Түрлі қосымшалар арқылы адамдар жылдам ақпарат алмасып, түрлі мәселелерді талқылай алады, олармен бөлісе, қабылдай алады. Кейбіреулер, мәселені талқыға салып, өз ойын ортаға салады, біреулер олармен келісіп, енді біреулер қарсытлық білдіріп жатады. Мұндай сан-алуан ақпараттармен бірге инновациялар да таралады. Мәселен, Bitcoin, Selfie, МООС сияқты жаңа сөздер мен қысқартулар пайда болды. Бұлар өз кезегінде әлеуметтік қоғамдағы инновациялардың таралу үрдістерін зерттеуге әртүрлі ғылыми қауымдастықтардың қызығушылықтарын арттырды. Бұл үрдістер биологияда [20], әлеуметтік ғылымда [21], пікірлер немесе келіссөздердің таралуы [22-26], жасанды интеллектте [27] әртүрлі салаларда [28, 29] қоғамды қалыптастыру сияқты жұмыстарда мұқият зерттелген және пайдаланылған. Кешенді желілер мен әлеуметтік динамиканы біріктіре отырып, Naming Game (NG) математикалық модельдеу және бәтуға келу процестерін модельдеудің тиімді тәсілін ұсынады. Атап айтқанда, NG – бұл кейбір атаулар туралы келісімге келуге тырысатын агенттердің өзара әрекеттесуін зерттейтін модель. Соңғы уақытта NG кездейсоқ графикалық желілер [30-32], кішігірім желілер [33-36] және әртүрлі күрделі желілік модельдерде [37-39] зерттелді. Naming Game моделі мүмкіндігінше аз есептеу қуатын пайдаланады және осылайша танымдық күрделілік пен өнімділік үшін төменгі сызықты белгілейді. Тілдік өзін-өзі ұйымдастырудың басқа модельдерінен айырмашылығы, агенттер жеке сөздердің сәттілік дәрежесі туралы ақпаратты сақтамайды және ең жақсы сөз немесе крест-ситуациялық оқуды таңдау сияқты зияткерлік эвристиканы пайдаланбайды.

Naming Game моделінің ережесі бойынша N агенттері қатысады және олар өз сөздігіне M сөздерді жүктеуге тырысады, осылайша бір агент басқа біреудің назарын аудара алады. Сөздер – бұл адамдар, физикалық нысандар, қатынастар,

веб-сайттар, суреттер, музыка файлдары сияқты кез-келген нысандар бола алады. Барлық агенттердің $t=0$ уақытында бос сөздігі бар. Әр кадамда ($t=1,2,\dots$) кездейсоқ екі ойыншы таңдалады, олардың біреуі баяндаушы, ал екіншісі тыңдаушы ретінде қарастырылады. Алғашында, егер сөздік бос болса, ол сөздіктен жаңа сөз құрастырады, содан кейін оны тыңдаушыға белгісіз заттың атауы ретінде береді. Бірақ егер баяндаушының жадында бірнеше сөздер болса, ол кездейсоқ тыңдаушыға айту үшін жадынан бір сөз таңдап алады. Егер берілетін сөз тыңдаушының жадында болса, онда екі агент те келісімге келеді де екі агент те осы сөзді сақтайды.



а – келісім сәтсіз; ә – келесім сәтті

Сурет 1.2 – Минималды NG өзара әрекеттесуі

Егер тыңдаушының жадында баяндаушының айтылған сөзі болмаса, бұл әңгіме сәтсіз болатындығын ескере отырып, тыңдаушы бұл сөзді үйреніп, оны жадына қосады. Бұл процесс бір сөз арқылы түпкілікті келісімге келгенге дейін жалғасады немесе байланысқа қажетті уақыт өткеннен кейін нәтиже бермейді. NG моделінің өзара әрекеттесуі келесі ережелерге бағынады (сурет 1.2):

– баяндаушы/тыңдаушы өз сөздігінен кездейсоқ бір сөз таңдайды (егер сөздік бос болса жаңа сөз құрады) және оны тыңдаушыға береді;

– егер тыңдаушының сөздік тізімінде баяндаушы жіберген сөз болса және бұл сөз баяндаушы таңдаған объектімен байланысты болса, онда өзара әрекеттесу сәтті болады, екі ойыншы да өз сөздік қорында тек жеңіске жететін сөзді ғана сақтайды, қалған сөздердің бәрін жояды (сәттілік);

– егер тыңдаушының сөздік тізімінде спикер жіберген сөз болмаса, өзара әрекеттесу сәтсіз болады, тыңдаушы жаңа сөз өз сөздігіне қосып, тізімді жаңартады (сәтсіздік).

Бұл модель бірқатар болжамдар жасауға қабілетті. Негізінде, әр ойыншы барлық басқа ойыншылармен ойнай алады, яғни өзара әрекеттесу желісінің құрылымында нақты негізгі топология жоқ. Келесі 1.2-суретте көрсетілгендей, біз қарапайымдылық үшін «AAA, BBB» т.с.с әріптерді пікір білдіру үлгісіне көрсету үшін қолданамыз.

Федотов А.М., [40-45] Мурзин Ф.А., [46-50] Барончелли А. [51-54] жұмыстарында инновация дуффузиясы, Naming Game сөздігі қарастырылған.

1.2 Инновациялар жүйесін оңтайландырудағы қайта құрылымдау түсінігі

Компьютердің өнімділігін бағалау және есептеу процесін басқаруға байланысты мәселелер ауқымы өте кең және жеке ғылыми сала ретінде қалыптасуда, сондықтан бағдарламалар көбінесе зерттеудің негізгі объектісі ретінде қарастырылады.

Есептеу процесерін басқару теориясындағы қызықты мәселенің бірі – бағдарламаны қайта құрылымдау болып табылады. Бұл бағдарламалаудың классикалық мәселесіне, негізінен А.А. Ляпунов [55], Ю.И. Янов [56], С.С.Лавров [57] және т.б. ғалымдардың іргелі жұмыстарының нәтижесінде қалыптасқан зерттеулер саласына жатқызуға болады. Сонымен қатар, бағдарламаны оңтайлы қайта құрылымдау туралы алғашқы жарияланған жұмыстардың бірі 1964-1966 жылдардағы F.H. Dearnly, G.V. Hewell [58] J. Dennis [59], A. Schurman [60, 61], C.V. Ramamoorthy [62] зерттеулері. Бұл саладағы зерттеулер келесі жұмыстарда [63-65] қарастырылған. Бағдарламаны қайта құрылымдау тапсырмасы J. Kral жұмысында [66] жады ұяшықтарында жазылған, ұзындықтары бірдей тізбектелген адрестер бойынша бөлу ретінде қарастырылады және әрбір сегмент тек көршілес командалардан тұрады деп түсіндіріледі. Сонымен бірге, кез-келген уақытта жедел жадыда бағдарламаның кез-келген орындалуында тек бір ғана сегмент болады деп болжанады. Осы болжамдар бойынша сегменттердің қайта жазылуын азайту алгоритмі беріледі. Сонымен бірге, бұл жұмыстарда қайта құрылымдау мәселесін жалпы түрде құру әрекеті көрсетілген, яғни графикалық есеп түрінде ұсынылған. J. Dennis жұмысында [59, p. 753-755] қайта құрылымдаудың мазмұнды сипаттамасы беріледі және оның жүзеге асырылуы егжей-тегжейлі талқыланады.

Жоғарыда айтылғандай, қайта құрылымдау тапсырмасы әдетте бір немесе бірнеше мақсатқа сәйкес тізбектелген бағдарламаны бөліктерге (блоктар, бөлімдер, беттер және т.б.) бөлу болып табылады. Жоғарыда айтылғандай, статикалық жадысы бар жүйелер үшін мұндай мақсат болуы мүмкін: бет қателігінің орташа санын азайту, бағдарламаға бөлінген негізгі жақтың орташа мөлшері, бағдарламаның орташа орындалу уақыты, сонымен қатар, кеңістік және уақыт функционалдығын азайту және т.б. Параллельді жүйелер үшін қайта құрылымдаудың негізгі мақсаттарының бірі - тізбектелген бағдарламаны бөліктерге бөлу, яғни бағдарламаны параллельді тармақтарға бөлу. Жадысы бар жүйелер үшін «қайта құрылымдау» терминімен бірге «қайта құру», «рефакторинг» [67, 68], «рестракиринг» [69, 70] терминдері қолданылады.

«Есептеу жүйелерін оңтайландыру» жеке ғылыми сала ретінде қалыптастыру және дамыту барысында қайта құрылымдау мәселесін шартты түрде үш тәсілге бөлуге болады: дискретті аналитикалық модельдердің синтезіне негізделген тәсіл; диаграмма, сонымен қатар кластерлік тәсілдер, олар негізінен белгісіз қателіктерімен тек жуық шешімдерді алуға мүмкіндік береді. Нақты шешімдерді табу тек ерекше жағдайларда ғана кездеседі.

Бастапқы кезеңде (60-шы жылдар), белгілі себептермен, сол кезде белгілі дискретті оптимизацияның кейбір әдістері, кластерлік талдау және графикалық теория қайта құрылымдау мәселесін шешу үшін тікелей қолданылды. Сонымен қатар, көптеген жағдайларда басты мақсат мәселенің «инженерлік бөлігін» шешу, яғни есептеу жүйелерінің аппараттық құрамының тиімділігін арттыру болатын. Кейінірек қайта құрылымдау алгоритмдерін синтездеу кезінде олар инженерлік басымдықтарға емес, математикалық объектілер сияқты бағдарламалардың қасиеттері мен ерекшеліктеріне назар аудара бастады және дәл сол қасиеттерді, мысалы, аппараттық-логикалық, комбинаторлық сипаттамаларды зерттей бастады. Комбинаторлық мәселе ретінде жалпы жағдайда қайта құрылымдау мәселесі *NP*-толық есептер класына жатады және есептеу тәжірибелерінде олардың қасиеттерін көрсетеді. Қайта құрылымдау кезінде бағдарламалардың орналасуы, сондай-ақ алмастыру стратегиясы сияқты бағдарламалардың маңызды қасиетін ескеруге бағытталған бірқатар жұмыстар жүргізілді.

WS стратегиясы бар статикалық жады жүйелерге арналған сөздіктерді қайта құрылымдау модельдері (жұмысшы жиын) Журавлевтің операторлық теориясының бағдарламаны сегменттеу мәселесінде қолдануға негізделген модельдер болып табылады. Мұндай модельдердің болуы қайта құрылымдау мәселесінің нақты ε - дәл шешімдердің құрылуына кепілдік беретін жағдайларды анықтауға мүмкіндік береді. Бұл жерде *WS* стратегиясы әр түрлі белсенді алмасу алгоритмдері тобының негізгі өкілі. Жинақталған тәжірибелік және теориялық материал негізінде теориялық және жүйелік бағдарламалау саласындағы маман Д.Ферраридің пікірі бойынша: «Жұмысшы жиын - стратегиясы есептеу жүйелерінде бұрын-соңды қолданылатын ең сәтті өзін-өзі баптау құралдарының бірі болып табылады» [71]. Есептеу процесін басқару теориясында жұмысшы жиын ұғымы ерекше рөл атқаратындығына күмән жоқ, сондықтан *WS* стратегиясы қайта құрылымдау мәселесін шешу де ерекше қызығушылық тудырады. Осы себепті, *WS* стратегиясы ерекше назарға ие. Ол үшін жалпы шектеулер жағдайында, *WS* стратегиясының ерекшеліктерін пайдалана отырып, қайта құрылымдау мәселесінің функционалды нысанын және рұқсат етілген қайта құрылымдау жиынын белгілейтін шектеулер жүйесін жазуға болады.

Жалпы, жұмыста Журавлевтің операторлық теориясын негігі тапсырмаға қолдану негізінде сөздікті қайта құрылымдау мәселесін шешудің жаңа тәсілі сипатталған. Диссертациялық жұмыста осы бағытта келтірілген нәтижелерді операторлық теорияны қолданбалы бағытта дамыту ретінде қарастыруға болады. Математикалық модельдер, әдістер, мәселелердің мазмұны, сондай-ақ осы модельдер мен әдістерді қолдану нәтижелерінің есептеу жүйелерінің жұмысына

нақты әсер етуі, қайта құрылымдау мәселесі дискреттік оңтайландыру тұрғысынан зерттеу объектісі ретінде қызығушылық тудырады.

Дж. Фон Нейманның 50 жыл бұрын өзі жасаған компьютер жұмысын басқарудағы жадты басқарудың шешуші рөлі туралы былай деген: «Уақыт өте келе есептеу құрылғыларының қарапайым базасы жақсарған сайын, компьютер сәулетшісінің тезистері өзектілігін жоғалтпады, керісінше ол жаңа мазмұнмен толықтырылды».

1.2.1 Қайта құрылымдау әдістері

Бұл бөлімде сөздік жұмысын оңтайлы қайта құрылымдау әдістері туралы жалпы ақпарат берілген. Осы тақырыпта қол жетімді қысқаша шолу [72], сондай-ақ кейбір бөліктері [73] 20 жылдан астам уақыт бұрын жазылып, 1965 жылдан бастап 70 жылдар басына дейінгі жұмыстар қамтылғанын атап өткен жөн. Бүгінгі күні қайта құрылымдау тапсырмасын оңтайлы есебін шешудің үш тәсілі бар:

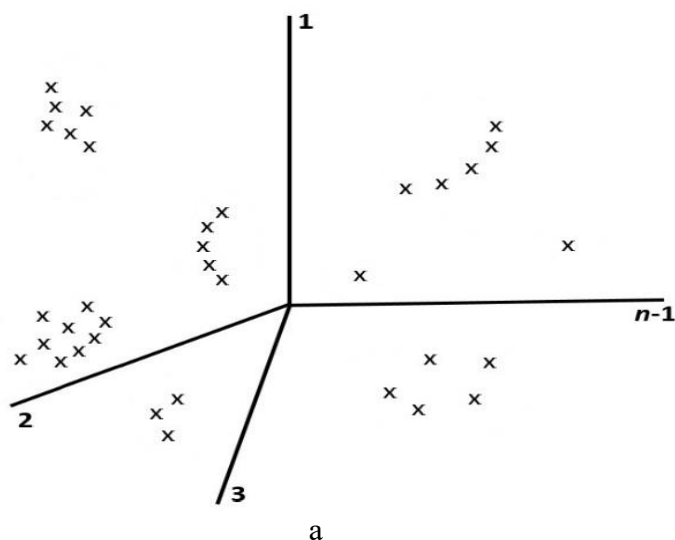
1. *Комбинаторлық-аналитикалық тәсіл* бағдарламаны қайта құрылымдау мәселесіне арналған дискретті аналитикалық модельдерді синтездеуге (яғни функционалдық және шектеулер синтезінде), содан кейін дискретті бағдарламалау әдістерін қолдануға негізделген. Бұл бағытқа [74-78] жұмыстары кіреді. Бұл тәсілдің басты ерекшелігі, әдетте, қайта құрылымдау мәселесінің функционалы және рұқсат етілген қайта құрылымдау жиынын анықтайтын шектеулерді дәл жазу мүмкін емес және жуық шешімнің қателігін теориялық тұрғыдан бағалау қиын (әсіресе функционалдық құрамға кіретін коэффициенттер болғандықтан).

2. *Графикалық тәсіл* А.А. Ляпунов, Ю.И. Янов, Л.А. Калужин, Р. Карп [79-82] жұмыстарына негізделген. Мұнда әдеттегі тәсіл, блоктар шындарға сәйкес келетін, шеттердің арасындағы бақылауды беруді басқару; шыңның салмағы - блок өлшемі, қабырғаның салмағы - көршілес шындар арасындағы бақылауды өткізудің жалпы саны. Графиктің шындарын кез-келген ішкі жиынға түсетін шындардың жалпы салмағы бет өлшемінен аспайтындай етіп кесу керек, ал шеттерінен тыс жиектердің жалпы салмағы минималды болады. Графикалық теорияда бұл мәселе белгілі (ол *NP*-толық), және оны шешудің әртүрлі әдістері ұсынылды. Айта кету керек, мұндай тұжырым әр түрлі ішкі жиындарға түсетін блоктар негізгі жадта бір уақытта бірнеше рет болуы мүмкін екендігін ескермейді. Басқаша айтқанда, бұл тұжырым сөздіктің орындалуының кез-келген сәтінде негізгі жадта сөздіктің тек бір беті болған кезде дәл осындай жағдайды көрсетеді.

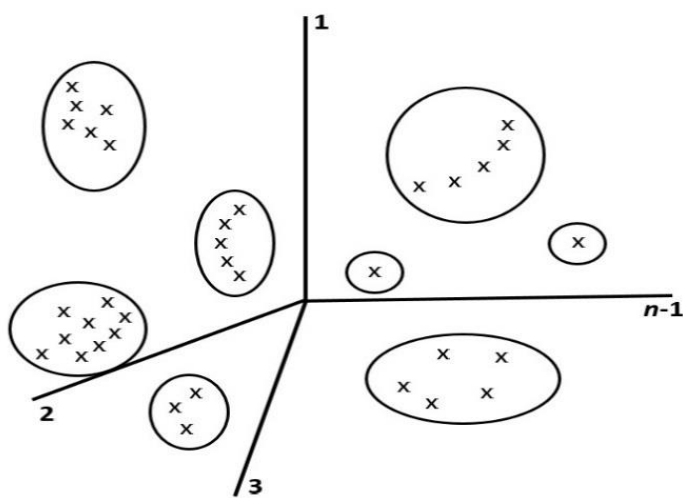
Осылайша, қайта құрылымдау мәселесінің графикалық нұсқауы беттерді ауыстыру стратегиясын әлсіз ескереді және, былайша айтқанда, айтарлықтай нақтылауды қажет етеді. Соған қарамастан, қайта құрылымдау мәселесінің бастапқы сызбасын қолдана отырып, зерттеулердің үлкен циклі жүргізілді. Осылайша алынған шешімдер сөздіктің тұрақты резидентінің белгіленген мөлшерін немесе сілтеме деректерінде (сілтеме жолдары) беттердің орташа санын азайтуға мүмкіндік берді. Беттерде орналасқан ұғымдардың күйін жақсарту үшін әртүрлі эвристикалық әдістер қолданылады, әсіресе граф

шындарының салмағын есептеу кезінде (жақындық матрицасының элементтері) [83-85].

3. *Класстерлік тәсіл* кластерлік талдау мәселесі ретінде қайта құрылымдау мәселесін интерпретациялауға негізделген [86-91]. Мұнда бағдарламаның әр блогы жақындық матрицасының қатарына сәйкес келеді, яғни сөздіктің әр ұғымы ($n-1$) - өлшемді кеңістіктің нүктесі болып табылады (сурет 1.3).



а



ә

а – класстерлік тәсіл; ә – класстерлеу барысы

Сурет 1.3 - ($n-1$) - өлшемді кеңістіктің нүктесі

Үлкен ішкі қосылыстары бар ұғымдар бір кластерде орналасуы керек [92, 93]. Өз кезегінде, кластер, әдетте, бетке сәйкес келеді, бұл көп жағдайда негізгі жадта сөздік беттерінің «бекітілген» ішкі жиындары болуы мүмкін. Осы ішкі жиындардағы беттер арасындағы өзара сілтемелер қайта құрылымдау модельдерінде беттің бұзылуына әкелетін сілтемелер ретінде қарастырылмауы керек. Осы әдісті қолдану арқылы алынған қайта құрылымдау мәселесінің

шешімдері, теориялық қатесі белгісіз болатын, тек жуықталған шешімдер болып табылады.

Айта кетейік, тәсілдерді бірнеше бөлікке бөлу өте ерікті және көбінесе оңтайлы қайта құрылымдау есептерін шешуде негізгі тапсырманы график түрінде белгілей отырып, оны шешу үшін аралас әдіс қолданылады. Мұнда жиі кездесетін нәрсе, уақыт өте келе қайта құрылымдау туралы мәселені шешкенде, оптималды шешім беретін жылдам алгоритмдерге қарағанда артықшылық бере бастады. Нәтижесінде болжалды шешім қатесін теориялық бағалауда үлкен қиындықтар туындайды. Алайда, ұсынылған алгоритмдердің авторлары жүргізген эксперименттік тексеру, ең болмағанда, анықтамалық мәліметтер бойынша олардың практикалық тиімділігін растайды. Бір қызығы, 1967 жылы L.W. Coe [94] жұмысындағы тәжірибиелік нәтижелер бетті сәтті қайта құру жүйесінің жұмысына едәуір әсер ететінін көрсететін (үлкен FORTRAN бағдарламасы үшін бет қателіктерінің санын 50% азайтқан). Бұл жұмыс зерттеушілердің назарын аударды және әртүрлі саладағы мамандардың қайта құрылымдау мәселесіне үлкен қызығушылығын тудырды.

Қосалқы жад мен негізгі арасындағы алмасу бүкіл беттерде жүзеге асырылады, алмасу кезінде орталық процессор (CPU) басқа бағдарламаның орындаушы командаларына ауысады. Егер бағдарламаны орындау барысында жедел жадыға кірмейтін бет сілтеме пайда болса, онда бет қателігі деп аталатын беттің үзілуі орын алады. Бағдарламаның орындалуы, кем дегенде, қажетті бетті жүктеуге қажет уақытқа үзіледі [95]. Жедел жадыға қажетті бетті жүктемес бұрын, жедел жадыдан жойылатын сөздік беттерінің бір бөлігін (ауыстыру стратегиясына байланысты) анықтау керек, содан кейін осы беттерде жады бос болып саналады. Егер жойылатын беттің мазмұны жедел жадыда болу кезінде бұрмаланса, жүйе өзгертілген беттің осы беттің түпнұсқасын сақтау кезінде қосалқы жадқа қайта жазылуын қамтамасыз етеді. Әйтпесе, бетті көмекші жадқа қайта жазудың қажеті жоқ. Жедел жадыдан шығарылатын бетті немесе беттер тобын таңдауды алгоритм немесе стратегия ауыстыру деп аталатын бір алгоритмге сәйкес орындалады. Әрине, әрдайым бұл бетті жедел жадыдан алып тастаған жөн, бұл сөздік жақын арада қажет болады, ал ауыстыру алгоритмінің өзі өте қарапайым болуы керек. Көптеген операциялық жүйелер (ОЖ) бетті сегменттеуде статикалық жадыны пайдаланады. Қажетті өнімділікті қамтамасыз ету үшін ОЖ жадысының менеджері жедел жадтағы жаңа ақпаратты сақтайа отырып, жақын арада қандай логикалық адрестерге қол жеткізілетінін білуге тырысады. Бұл жерде бастысы бетті ығыстыру алгоритмінде жүзеге асырылған стратегияны сәтті таңдау болып табылады.

Статикалық мекенжайды салыстыру бет кестесінің көмегімен жасалады. Әрбір статикалық бет үшін кестедегі жазба негізгі жадтағы сәйкес бет жақтауының нөмірін, сондай-ақ жадқа кіруді басқаруға арналған бет атрибуттарын қамтиды. Қалаған бет жадта болмаса немесе жадқа кіруге рұқсат берілмегенде не болады? Әрине, операциялық жүйеге қандай да бір жолмен болған жағдай туралы хабарлау керек. Бұл әдетте ерекше жағдайлар механизмін қолдану арқылы жасалады. Статикалық бетке осындай қоңырау шалуға

тырысқанда, бет қателігінің (*page fault*) белгілі бір түрін өңдеуге арналған бұйрықтардың арнайы реттілігіне әкелетін бетте қателіктер пайда болады. Бет қателігі әр түрлі жағдайларда орын алуы мүмкін: жедел жадта ешқандай бет болмаған кезде, “тек оқуға” арналған атрибуты бар бетке жазуға тырысқанда немесе “тек орындау” атрибуты бар бетті оқуға немесе жазуға тырысқанда. Осы жағдайлардың кез-келгенінде операциялық жүйенің бөлігі болып табылатын бетті бұзу өңдегіші шақырылады.

Бізді қызықтыратыны бет қателігінің нақты нұсқасы - жоқ бетке сілтеме жасау, өйткені оны өңдеу көбінесе бет жүйесінің өнімділігін анықтайды. Бағдарлама негізгі жадта жоқ статикалық бетке кіргенде, операциялық жүйе негізгі жадтан бет таңдап, оған сыртқы жадтан статикалық беттің көшірмесін жылжытуы керек және бет кестесінің сәйкес элементін өзгертуі керек. Есептеуіш жүйенің өнімділігін жақсартуға бет қателігі жиілігін азайту, сонымен қатар оларды өңдеу жылдамдығын арттыру арқылы қол жеткізуге болады. Жедел жадыда жоқ бетке тиімді қолжеткізу уақыты келесілер арқылы қалыптасады:

- ерекше жағдайларды өңдеу (бет қателігі);
- екінші жадтан бетті оқу (жүктеу), кейде негізгі жадта орын жеткіліксіз болған кезде беттің біреуін негізгі жадтан екінші жадқа көшіру керек, яғни бетті ауыстыру қажет);
- осы бет қателігіне себеп болған процесті жалғастыру.

1.3 Бетті сақтау және оңтайландыру стратегиялары

Жадыны басқарудың ішкі жүйесі келесі стратегияларды жүзеге асырумен байланысты:

Таңдау стратегиясы (fetch policy) – бетті екінші жадтан негізгіге қайта жазу. Таңдаудың екі негізгі нұсқасы бар – сұраныс бойынша және белсенді. Сұрай бойынша таңдау алгоритмінің процесі дискіде жоқ бетке сілтеме жасалған кезде орындалады. Оның орындалуы бетті дискіден физикалық бетке жүктеу және бетке тиісті кесте жазбасын түзету. Таңдау алгоритмі алдын-ала оқуды орындайды, яғни ерекше жағдайды тудырған бетке қосымша, оның айналасындағы бірнеше беттер де жадқа жүктеледі (әдетте іргелес беттер кезекпен сыртқы жадта орналасады және оларды бір дискіден оқуға болады). Мұндай алгоритм деректер мен кодтардың едәуір көлемімен жұмыс істеу кезінде пайда болатын ерекше жағдайлардың көптігімен байланысты шығындарды азайтуға арналған; Сонымен қатар, дискінің жұмысы оңтайланады.

Орналастыру стратегиясы (placement policy) – кіріс бетті негізгі жадтың қай бөліміне орналастыруын анықтайды. Егер жүйе сегментті түрде ұйымдасқан болса, онда динамикалық бөлу стратегиясына ұқсас стратегия қажет.

Алмастыру стратегиясы (replacement policy) – жедел жадтан орын босату үшін сыртқы жадқа қандай бетті ығыстыру керектігін қарастырады. Тиісті бетті алмастыру алгоритмінде жүзеге асырылған орынды алмастыру стратегиясы қажетті ақпаратты жадта сақтауға және сол арқылы бет қателігі жиілігін азайтуға мүмкіндік береді. Алмастыру әр процеске бөлінген ұғымдардың санын ескеруі керек. Сонымен қатар, алмастырылатын бет алмастыруды түрткі болған

процестің құрамына кіреме немесе оны негізгі жадтағы барлық ұғымдардың ішінен таңдалуы керек пе екенін шешу қажет.

Алмастыру кезінде бетті негізгі және қосымша жадтың арасына екі рет алмастыру керек. Орын алмастыру процесін модификациялау битін қолдана отырып оңтайландыруға болады. Егер бетке кем дегенде бір байт жазылған болса, компьютер модификациялау битін орнатады. Алмастыруға үміткерді таңдаған кезде модификацияланған бит тексеріледі. Егер бит орнатылмаса, бұл бетті дискіге көшірудің қажеті жоқ, себебі оның көшірмесі дискіде бар. Мұндай әдіс «тек оқуға» арналған беттерге де қолданылады, олар ешқашан өзгертілмейді. Бұл схема бет қателігін өңдеу уақытын қысқартады.

Бетті алмастыру алгоритмдерінің алуан түрлері бар. Олардың барлығы жергілікті және ғаламдық болып бөлінеді. Жергілікті алгоритмдер, ғаламдықтан айырмашылығы, әр процес үшін белгіленген немесе динамикалық реттелген беттер санын таратады. Процесс өзіне тиесілі барлық беттерді біткенде, жүйе басқа процесстердің бетінен емес, өзіне тиесілі беттің бірін физикалық жадтан жояды. Ерекше жағдай болған жағдайда, алмастырудың ғаламдық алгоритмі ешқандай процедураға қарамастан, кез-келген физикалық бетті шығарумен қанағаттанады.

Ғаламдық алгоритмдердің бірнеше кемшіліктері бар. Біріншіден, олар кейбір процестерді басқа процестердің әрекетіне сезімтал етеді. Мысалы, егер жүйеде бір процесс бір уақытта көптеген жады беттерін қолданатын болса, онда барлық қосымшалар олардың жұмысына жадының болмауына байланысты қатты баяулайды. Екіншіден, дұрыс жұмыс істемейтін сөздіктің бір бөлігі бүкіл жүйенің жұмысына кедергі келтіруі мүмкін және көп жадыны алуға тырысады. Сондықтан көп тапсырмалы жүйеде кейде күрделі алгоритмдерді қолдану қажет болады. Жергілікті алгоритмдерді пайдалану операциялық жүйеде әр процеске бөлінген физикалық ұғымдардың тізімін сақтауды қажет етеді. Бұл беттер тізімін кейде процесс резиденттерінің жиыны деп атайды.

Алгоритмнің тиімділігі, әдетте, жадыға сілтемелердің нақты бір тізбегі бойынша бағаланады, олар үшін бет қателігінің саны есептеледі. Бұл бірізділік сілтеме жолы деп аталады. Кездейсоқ сандар сенсорының көмегімен немесе белгілі бір жүйені бақылау арқылы сілтемелер жолын жасанды түрде жасай аламыз. Соңғы әдіс тым көп сілтемелер береді және олардың санын азайту үшін 2 әрекет жасауға болады:

– белгілі бір бет өлшемі үшін сілтемелер мекен-жайларды емес, олардың нөмірлерін ғана есте сақтауға болады;

– бір бетте бірдей орындалатын бірнеше сілтемелерді бір рет тіркеуге болады.

Жоғарыда айтылғандай, процессорлардың көпшілігінде жадқа кірудің кейбір статистикасын жинауға мүмкіндік беретін қарапайым құралдары бар. Бұл құралдар әдетте бет кестесінің әр элементі үшін екі арнайы жалаушадан тұрады. Сілтеме жалаушасы (сілтеме биті) осы бетке кіру болған кезде автоматты түрде орнатылады және жоғарыда талқыланған өзгерту жалаушасы (битті өзгерту) осы бетке жазған кезде орнатылады. Амалдық жүйе белсенді түрде пайдаланылатын

беттерді бөлектеу үшін осындай жалаулардың орнатылуын қайта-қайта тексеріп отырады, содан кейін осы жалаулардың мәндері қалпына келтіріледі.

1.3.1 Бетті алмастыру алгоритмдері

Бетті алмастырудың танымал алгоритмдері: RANDOM, FIFO (алдымен бірінші болып шығады), LIFO (ең алғаш шыққан), LRU (ең аз қолданылған), WS (жұмысшы жиын).

FIFO алгоритмі жедел жадыға алғашқы жүктелген бетті анықтайды және бұл бетке қол жеткізу жиілігін есепке алмайды. LIFO алгоритмі, керісінше, жедел жадыға соңғы түскен бетті анықтайды, сонымен қатар жиілікті ескермейді. Екі жағдайда да, жойылған бет негізгі жадтан шығарылғаннан кейін бірден қажет болуы әбден мүмкін.

Бетке кіру жиілігін ескеретін алгоритмдерге LRU алмастыру алгоритмі жатады. Осыған сәйкес, ең аз қолданылған бет жойылады. LRU алгоритмі практикалық тиімділікке ие. Алайда, LRU алгоритмі кей жағдайларда төмен көрсеткіштерге ие.

FIFO алгоритмі, First In – First Out, яғни бірінші кіру-бірінші шығу деген мағынаны білдіреді. Әр бетке уақыт белгісі қойылады. Бұған беттің кезегін құру арқылы қол жеткізіледі. Ауыстыру үшін ең ескі бет таңдалады. Өкінішке орай, бұл стратегия белсенді қолданылған беттерді, мысалы, файлды өңдеу кезінде мәтіндік процессор сөздік беттерін алмастыруға әкелуі мүмкін. Белсенді беттерді ауыстыру кезінде бәрі дұрыс жұмыс істегенімен, бет қателігі бірден пайда болуы мүмкін.

Билэди аномалиясы – бір қарағанда, жадының беттер саны қаншалықты көп болса, бет қателігі де соғұрлым аз болатыны анық. Бірақ бұл әрдайым бола бермейді. Билэди мен оның әріптестері анықтағандай, процеске бөлінген ұғымдар санын көбейту, іс жүзінде беттерге жасалған сілтемелер қателігінің ұлғайуына әкеледі. Бұл құбылыс «Билэди аномалиялары» немесе «FIFO аномалиялары» деп аталады. Билэди аномалиясының нәтижелерінің бірі басқа алгоритмдердің арасында бет қателігінің минималды жиілігіне ие болатын оңтайлы алгоритмді іздеу болды. Мұндай алгоритм табылды. Бұл қарапайым: ұзақ уақыт пайдаланылмайтын бетті алмастыру болатын. Осы бетке бірінші сілтеме жасалмастан бұрын орындалатын нұсқаулықтардың саны әр бетке белгіленуі керек. Нұсқаулықтар саны ең жоғары бетке ығыстырылады. Бұл алгоритмді сипаттау оңай, бірақ оны орындау мүмкін емес. ОЖ келесі сілтеменің қай бетке болатынын білмейді. (Бұрын мұндай проблемалар процестерді жоспарлау кезінде туындаған – SJF алгоритмі).

LRU алгоритмі, яғни ең ұзақ пайдаланылмаған бетті ығыстыру. Бұл алгоритмі оңтайлы алгоритміне жақындау және эвристикалық ережеге негізделген алгоритм.

FIFO мен оңтайлы алгоритмнің басты айырмашылығы – біреуі артқа, екіншісі алға қарайды. Егер болашақты болжау үшін өткен шақты қолдану қажет болса, онда ұзақ уақыт пайдаланылмаған бетті алмастырудың мағынасы бар. Бұл

тәсіл ең аз қолданылған алгоритм (LRU) деп аталады. Алгоритмнің жұмысы 1.4-суретте көрсетілген.

	0	1	2	3	0	1	4	0	1	2	3	4	
				3	3	3	3	3	3	2	2	2	
			2	2	2	2	4	4	4	4	3	3	
		1	1	1	1	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	0	0	0	4	
	р	р	р	р			р	0		р	р	р	Page faults = 8

Сурет 1.4 – LRU алгоритм жұмысының мысалы

LRU жақсы, бірақ орындау тарапынан қиын алгоритм. Жадта барлық беттердің байланыстырылған тізімі болуы керек, оның басында жақында пайдаланылған беттер сақталады. Сонымен қатар, бұл тізім жадқа сілтеме орындалған сайын жаңартылуы керек. Мұндай тізімдегі беттерді іздеуге көп уақыт қажет. Д. Таненбаум LRU алгоритмінің арнайы 64-биттік сілтегішпен орындалу нұсқасын қарастырды, ол әр нұсқаулық орындалғаннан кейін автоматты түрде жоғарылайды және беттің кестесінде беттің әр сілтемесі үшін көрсеткіш мәні енгізілетін сәйкес өріс бар. Бет қателігі пайда болған кезде, осы өрістің ең кіші мәні бар бет жүктеледі. Оңтайлы алгоритм де, LRU алгоритмі де Билэди аномалиясынан зардап шекпейді [96].

NFU (Not Frequently Used) алгоритмі, яғни сирек қолданылатын бетті ығыстыру. Көптеген заманауи процессорлар LRU алгоритмін жүзеге асыру үшін тиісті жабдықты қамтамасыз етпейтіндіктен, LRU алгоритміне жақын, бірақ арнайы қолдауды қажет етпейтін алгоритмді пайдаланғысы келеді. LRU алгоритміне жақын және алгоритмнің бағдарламалық қамтамасыздандыруы бар алгоритм ол – NFU алгоритмі. Ол бастапқыда нөлге тең, әр бір бетке бағдарламалық есептегіштерді қажет етеді. Әрбір үзіліс кезінде (әр нұсқаудан кейін емес) операциялық жүйе жадтағы барлық беттерді қарап шығады және сілтеме жалаушасы орнатылған әр беттің есептегіш мәнін арттырады және сілтеме жалаушасын қалпына келтіреді. Осылайша, есептегіш бойынша ең төменгі мәні бар бет, мысалы, сирек кіретін бет, ығыстыруға үміткер болып табылады. NFU алгоритмінің басты кемшілігі – бұл ештеңені ұмытпайды. Мысалы, біраз уақыт өте жиі сілтеме жасалған, содан кейін сілтемелер тоқтатқан бет әлі де жадтан жойылмайды, өйткені оның есептеуіш көрсеткіші үлкен мөлшерде болуы мүмкін. Мысалы, бірнеше өту компиляторларында бірінші өту кезінде белсенді қолданылған беттер ұзақ уақытқа үлкен есептегіш мәндерін сақтап, болашақта пайдалы беттерді жүктеуге кедергі келтіреді. Дегенімен, қазіргі таңда алгоритмді аздап өзгертуге және есептегіш мәнін «ұмытып кетуге»

мүмкіндік беретін технологиялар бар. Әрбір үзіліс кезінде санауыштың мазмұны 1 битке оңға жылжытылса жеткілікті, содан кейін ол кіру жалаушасы бар беттер үшін көбейтіледі. Алгоритмнің тағы бір тұрақты кемшілігі – бұл беттерді сканерлеу процесінің ұзақтығы.

Ары қарай бірнеше алгоритмдерді атап өтуге болады. Мысалы, Second-Chance алгоритмі – бұл ескі беттің сілтеме жалаушасын (сілтеме биті) талдай отырып, жиі пайдаланылатын беттердің жоғалуын болдырмайтын FIFO алгоритмінің модификациясы. Егер жалауша қойылса, онда бет ығыстырылмайды, бірақ оның жалаушасы қалпына келтіріліп, бет кезектің соңына ауыстырылады. FIFO алгоритмінен ерекшелігі осында. Егер бастапқыда барлық беттерге жалаушалар қойылса, яғни барлық беттерге сілтеме жасалса, Second-Chance алгоритмі FIFO алгоритміне айналады. Бұл алгоритм Multics және BSD Unix-те қолданылды.

NRU (Not Recently-Used) алгоритмі Macintosh компьютерінде қолданылады, мұнда модификацияланған биттер мен сілтемені талдау негізінде зардап шеккен беттер таңдалады. Беттерді буферлеуге негізделген қызықты стратегиялар VAX / VMS және Mach жүзеге асырылған.

Сондай-ақ, көптеген басқа алмастыру алгоритмдері бар. Алмастырудың әртүрлі алгоритмдерінің егжей-тегжейлі сипаттамасын [97] және басқалардан көруге болады.

Трешинг (Thrashing). Процесстегі блоктар санын минимумға дейін азайту теориялық тұрғыдан мүмкін болса да, белсенді пайдаланылатын беттердің белгілі бір саны бар, оларсыз процесс көбінесе бет қателігін тудырады. Бет қателігінің жоғары жиілігі трешинг деп аталады. Егер командаларды орындаудан гөрі беттерді жүктеуге көбірек уақыт кетсе, процесс трешинг күйінде болады. Мұндай критикалық жағдай алмастырудың нақты алгоритмдеріне қарамастан туындайды.

Көбінесе трешинг нәтижесі есептеу жүйесінің өнімділігін төмендеуі болып табылады. Жағымсыз сценарийлердің бірі келесідей болуы мүмкін. Алмастырудың жаһандық алгоритмінде блоктар жетіспейтін процесс басқа процестерден блоктарды ала бастайды, сәйкесінше олар да солай жұмыс істей бастайды. Нәтижесінде барлық процестер екінші жад құрылғысына сұраныс кезегіне түседі (күту режимінде) және дайын күйдегі процесс кезегі бос болады. CPU жүктемесі азаяды. Бұған операциялық жүйе мультипрограммалау дәрежесін жоғарылату арқылы жауап береді, бұл одан әрі трешинг және процессор жүктемесінің одан әрі төмендеуіне әкеледі. Осылайша, жүйенің өткізу қабілеті трешинг себебінен азаяды. Ғаламдық алгоритмдерді қолдану кезінде пайда болатын трешинг әдісін жергілікті алмастыру алгоритмдерін қолдану арқылы шектеуге болады. Жергілікті алмастыру алгоритмдерінде, егер процестердің бірі трешингке ұшыраса да, бұл басқа процестерге әсер етпейді. Алайда, ол жүктеу қырылғысына кезекте тұрып көп уақыт жұмсайды, бұл басқа процестердің беттерін алмастыруды қиындатады.

Трешинг сияқты қиын жағдай алмастырудың нақты алгоритмдеріне қарамастан туындайды. Теориялық тұрғыдан трешингтің жоқтығына кепілдік

беретін жалғыз алгоритм жоғарыда қарастырылған оңтайлы алгоритм болып табылады, бірақ ол іс жүзінде қолданылмайды.

Сонымен, трешинг - бұл бет қателігінің жоғары жиілігі. Оны бақылап отыру керек. Ол жоғары болған кезде процесс блоктарға мұқтаж болады. Бет қателігінің қалаған жиілігін орната отырып, одан блоктарды қосу немесе азайту арқылы процестің көлемін реттеуге болады. Бүкіл процесті жүктеу орынды болуы мүмкін. Босатылған блоктар бет қателіктері басқа процестерге бөлінеді. Трешингтің алдын алу үшін процесске қажетінше көп блоктарға көп бөлу керек. Бірақ оның қаншалықты блок қажет екенін қайдан білуге болады? Процесс қанша блок қолданатындығын анықтауға тырысу керек. Бұл мәселені шешу үшін П. Деннинг жергілікті принципті қолдануға негізделген жұмысшы жиын моделін қолданды. WS алгоритмі жоғарыда қарастырылған алгоритмдерден біршама ерекшеленеді: осы алгоритмге сәйкес жұмысшы жиын деп аталатын беттер сөздік жұмысын орындау кезінде негізгі жадта орналасады, оның ерекшеліктері көптеген жағдайларда сөздіктің статикалық жадының ағымдағы орналасуын, яғни сөздіктің жергілікті бөлігін жақындастыруға мүмкіндік береді.

Әрине, жүйенің тиімділігі тұрғысынан, алмастыру алгоритмдері мүмкіндігінше бетте кем қамтамасыз етуі керек. Бірақ бұл мәнді ауыстыру стратегиясына ғана емес, сонымен қатар кездейсоқ мәліметтерге де байланысты екенін ескеру керек. Беттің көптеген қателіктерін тудыратын сөздік жұмысының тиімділігін күрт төмендеуі мүмкін. Ауыстыру алгоритмдерін модельдеу кезінде, көптеген жағдайларда олар оғаш бағдарламаның есептегіштеріне негізделеді, олардың көшірмелері жедел жадыда орналасқан әр статикалық бетте өзінің санауышы болады. Бастапқы сәтте барлық есептегіштер нөлге орнатылады. Бет әр бүр жүктелген сайын оның есептегіші нөлге келтіріледі, ал жедел жадыда қалған беттердің есептегіштері 1-ге көбейеді. Минималды есептегішпен бетті жою LIFO алгоритмін қолдану арқылы бет таңдауына сәйкес келетіндігін, максималды санауышы бар бетті жою беттің таңдауына сәйкес келеді. LRU алгоритмінің жұмысын модельдеу кезінде жедел жады орналасқан беттерге кіру жиілігін де ескеру қажет. Статикалық жадыны мұндай ұйымдастыру (бағдарламалаушыларға белгілі бір ыңғайлылық беретініне қарамастан, мысалы, стек құрылымдарымен, динамикалық кестелермен, деректер массивтерімен жұмыс жасау) физикалық жадыны ұтымды пайдаланумен байланысты бірқатар мәселелерді көтереді, олардың ішіндегі ең өткірі - жадыны бөлу мәселесі. Бұл мәселе, сонымен қатар, сөздік беттері ішіндегі мекен-жай кеңістігінің толық толтырылмағандығына байланысты бетті ұйымдастыру жағдайында пайда болады. Қосалқы және негізгі жад арасындағы сөздік беттерінің динамикалық алмасуы кез-келген уақытта негізгі жад фрагменттелген жинақ болып табылады, мұнда сөздік алып жатқан аудандар бос аудандармен ауысады. Бөліну мәселесін шешудің қызықты теориялық тәсілі сандар теориясының негізінде ұсынылды, дәлірек айтқанда, бөлу сандарының ендірілу теориясының нәтижелерін қолдану [98, 99] жұмыстарында көрсетілген. Мұндағы тағы бір маңызды мәселе – динамикалық алмасуы жұмыстарын нақты уақыт режимінде қолдану және тиімді алгоритм қолдану.

1.3.2 Инновациялар сөздігін оңтайландыру мәселелері

Заманауи «рестрактчеринг» [100-104], «рефакторинг» [105, 106] және сол сияқты кез-келген бағдарламаны, ал біздің жағдайда сөздікті оңтайландыру, сөздік жұмысына жағымды әсер етеді. Бұл әсіресе нашар құрылымдалған және жиі іске қосылатын сөздіктерде байқалады. Зерттеушілер сөздікті түрлендірген кезде мәселенің практикалық тұрғыдан оңтайлы шешімін табуға тырысуда. Қалай болғанда да, оңтайландыру сөздіктің дұрыстығын, орналасуын және т.с.с. қасиеттерін бұзбайды.

Бүгінгі күнге дейін жоғарыда аталған мәселенің түпкілікті шешімі болмауы ғалымдардың қызығушылығын және соған байланысты зерттеулерді тудыруда. Сөздік жұмысын оңтайландыру мәселесінің көптеген ресми тұжырымдары күрделі (*NP*-күрделі) экстремалды тапсырмаларға, соның ішінде комбинаторлық мәселелерге әкеледі, олар тәжірибе үшін эвристикалық тәсілдерді қарастыруға және зерттеуге мәжбүр. Екінші жағынан, *NP* сыныбындағы бірқатар мәселелер, соның ішінде біз үшін ең қызықты мәселе соңғы 60 жыл ішіндегі зерттеулер. Бұл жерде *NP*-кластағы басқа да мәселелердің жарқын мысалы – сатушының есебі, ол қарапайымдылығына қарамастан айтарлықтай математикалық мазмұнға ие, олардың нақты жағдайлары *P* класына жатады.

Біздің тапсырмаға қайта оралсақ, қайта құру бойынша бірқатар зерттеулер, оның ішінде бізді қызықтыратын тақырып кластерлік техникаға негізделген [107, 108] және т.б. тәжірибелердің жетілгеніне қарамастан, функционалдылық тұрғыдан кез-келген жағдайда белгісіз дәлдікпен жуықтас шешімдерді береді, яғни кластерлік тәсіл бағаланбайды, шешім нақты (оңтайлы) немесе бастапқы есептің ең жақсы шешімдері белгісіз дәл (оңтайлы) мәні немесе оған жақын шешімі алынады. Журавлевтің классификацияға негізделген тәсіліне негізделген кластерлік тұжырымдаманы қолданатын тағы бір тәсіл [109-111] ұсынылды.

Зерттеулерде негізгі тапсырманың шешімі бастапқы жағдайға байланысты әртүрлі болуы мүмкін. Бұл бөлімде сөздікті қайта құру мәселесіне назар аударамыз, ол үшін кластер әдісі сөздіктің іс-әрекетін жақсартуға мүмкіндік береді, бірақ нақты (оңтайлы) шешімге белгісіз жақындау арқылы. Тәжірибе жүзіндегі сәттілікке қарамастан, жоғарыда аталған мәселе әлі де ашық күйде және инженерлік қызығушылықтан басқа, сонымен қатар математикалық мазмұнға да ие. Соңғысы шамамен 70-ші жылдардан 90-жылдарға дейін қарқынды зерттеулермен ұзақ тарихқа ие. Кейінірек, мультикомпьютерлік және мультипроцессорлы жүйелерге бағытталған және алдыңғы тәжірибелерге негізделген қайта құрылымдау жұмыстары жақсы нәтижелерге қол жеткізгеннен кейін, жүйенің жұмысын жақсартуға мүмкіндік беретін, сонымен бірге бастапқы мәселенің белгісіз оңтайлы шешімін зерттеу жұмыстары біршама бәсеңдеді. Осылайша, мәселенің нақты (оңтайлы) шешімін немесе нақты шешім табу мүмкіндігін зерттеуге бет бұрғанан кейін, біздің мақсатымыз негізгі тапсырманың аналитикалық моделін құру болып табылады. Басқаша айтқанда, бастапқы тапсырманы тиісті концептуалды ұсынумен бірге, біздің мақсатымыз

жұмысшы жиын тұжырымдамасын қолдана отырып, тұжырымға сәйкес келетін функционалды және шектеулерді табу болып табылады, бұл сәйкес экстремалды есеп үшін алгоритм құруға мүмкіндік береді. Сонымен қатар, ұсынылған тәсіл мен нәтижелерді түсінуге оңай болу үшін есептеу процесінің геометриялық түсініктемесін де беру пайдалы. 50 жылдық зерттеулер жүргізілсе де, біз үшін нақты (оңтайлы) шешімді қалай құру туралы мәселесі әлі күнге дейін ашық болып қала береді. Біздің ойымызша, нәтижелер мультикомпьютерлік және мультипроцессорлы жүйелер үшін ұқсас мәселерді зерттеу үшін де пайдалы болмақ, мысалы, SVM тәрізді көпкомпьютерлік жүйелер. Сонымен қатар, алынған нәтижелерден WS тәрізді жылдам және арзан алмасу алгоритмін құру мүмкіндігі пайда болады.

Статикалық жадтың ішкі жүйесі бос беттер блоктарының резерві болған кезде тиімді жұмыс істейді. Трешинг болмаған кезде жүйені қолдауды қамтамасыз ететін алгоритмдер жиі оятып, жадының күйін тексеретін фондық процестердің бөлігі ретінде жүзеге асырылады (олар көбінесе қызметтер деп аталады). Егер бос блоктар аз болса, олар алмастыру стратегиясын өзгерте алады. Олардың міндеті - жүйені ең жақсы күйде ұстау. Мұндай процестің мысалы ретінде фондық процесс бола алады – Unix ОЖ-нің көптеген нұсқаларында қолданылатын және жұмысшы жиынды қолдану негізінде жүктеу алгоритмінің жеңіл нұсқасын жүзеге асыратын беттер жинағы. Бұл қызметтер жұмыс процестерінің құрамына кірмейтін беттерді шығарады. Ол бос беттер тізіміндегі беттердің саны белгіленген төменгі шегіне жеткенде белсенді жұмыс істей бастайды және беттерді өз стратегиясына сәйкес итермелеуге тырысады. Windows операциялық жүйелерінде мұндай рөлді жұмыс жиыны менеджері атқарады, ол секундына бір рет шақырылады немесе бос жадтың көлемі белгілі шектеуден төмен түссе, жадыны басқарудың жалпы саясаты мен жұмысшы жиынды қолдау үшін жауап береді. Жадты қолдаумен байланысты операциялық жүйенің функцияларын жүзеге асыру – беттер кестелерін жүргізу, мекен-жайларды аудару, бет қателіктерін өңдеу, ассоциативті жадты басқару және басқалары – процестің адрестік кеңістігін ыңғайлы көрсетуді қамтамасыз ететін мәліметтер құрылымымен тығыз байланысты. Бұл құрылымдардың форматы белгілі бір ОЖ-нің жабдықтары мен ерекшеліктеріне байланысты [112]. Көбінесе ОЖ процесінің статикалық жадысы бес типті сегменттерге бөлінеді: деректер, стек, ортақ және жадқа түсірілген файлдардың сегменті.

Ұғымдар – бұл бетте орналасқан, процестің статикалы мекен-жай кеңістігіндегі үздіксіз аудандар (Linux-та оларды аудандар деп атайды). Әр аймақ бірдей қорғаныс режимі бар беттер жиынтығынан тұрады. Виртуалды кеңістіктегі аймақтардың арасында бос аймақтар болуы мүмкін. Бұл әсіресе төмен деңгейлі бағдарламалау кезінде байқалады. Атап айтқанда, жеке жад аймақтарын атауға болады және процестер арасында мәліметтер алмасу үшін қолданылады. Егер олардың атауы (пароль) белгілі болса, екі процесс ортақ жад аумағы арқылы байланыса алады. Бұл әдетте статикалық жад интерфейсін құрайтын арнайы сілтемелердің көмегімен орындалады (мысалы, тар және ұртар). Орындалатын файлды жүктеу (exec system call), әдетте, оның бөліктерін

(код, деректер) процестің адрестік кеңістігінің сәйкес сегменттеріне көрсету (mapping) арқылы жүзеге асырылады. Алғашқы нұсқауды орындауға тырысқанда, жүйе кодтың қажетті бөлігі жадта жоқ екенін анықтайды, содан кейін бет қателігі пайда болады және кодтың осы бөлігін дискіден жүктейді. Әрі қарай процедура барлық бағдарлама жедел жадыға жүктелгенге дейін қайталаанады.

Жад менеджерінің дұрыс жұмысы, абстрактілі статикалық жады моделін таңдауға және оны аппараттық қамтамасыз етуге байланысты іргелі мәселелермен қатар көптеген ұсақ бөлшектермен қамтамасыз етіледі. Мұндай компоненттің мысалы ретінде беттер жүйесін жадта локализациялауды қарастыруға болады, бұл беттік жүйені қолдау I/O буферлерін, басқа да маңызды деректер мен кодтарды жадта сақтауға мүмкіндік беретін кейбір беттерге рұқсат беру қажеттілігіне әкелетін жағдайларда қолданылады. Статикалық жад жүйесі I/O ішкі жүйесімен қайшы келуі мүмкін жағдайды қарастырайық. Мысалы, процесс буферге кіруді сұрайды және оның аяқталуын күтеді. Бұл бет қателігін тудыруы мүмкін, басқару басқа процеске ауысады және нөлдік емес ықтималдықпен бірінші процесс кіруі керек болған беттің жүктелуіг тудырады. Мұндай жағдайлар қосымша бақылауды қажет етеді, әсіресе егер енгізу/шығару тікелей жадқа қол жетімділік (DMA) механизмін қолдану арқылы жүзеге асырылса. Бұл мәселені шешудің бір әдісі – деректерді буферге енгізу, содан кейін оны пайдаланушы кеңістігіне көшіру. Екінші шешім – беттің атрибуттарының бөлігі болып табылатын арнайы локализациялау битінің көмегімен беттерді жадыға локализациялау. Локализацияланған бетті ауыстыру мүмкін емес. Локализация биті енгізу-шығару операциясынан кейін қалпына келтіріледі. Локализация битінің тағы бір қолданылуы бетті қалыпты ауыстыру кезінде орын алуы мүмкін. Ұзақ күткеннен кейін жүйе маңызды емес процессті өз қолына алып, дискіден қажетті бетті ығыстырады. Егер оны тез арада басымдыққа ие процесс алмастырса, соңғысы жаңадан жүктелген төмен басымдыққа ие бетті оңай ауыстыра алады, өйткені оған сілтемелер болмаған. Локализациялау битін пайдалану барысында оны өшіруді ұмытып кетсе қауіпті болуы мүмкін. Егер бұл жағдай орын алса, бет қолданылмайды. Сондықтан нақты уақыттағы жүйелер статикалық жадты ешқашан пайдаланбайды. Беттерді локализациялау жүйесінен басқа, жадыны басқару процесінде пайда болатын басқа да қызықты мәселелер бар. Мәселен, бет қателігіне себеб болған мәлімдемені қайта орындау қиын болуы мүмкін. Кейінге қалдырылған жадты бөлу алгоритмдері де қызығушылық тудырады, дегенімен бұл диссертациялық жұмыста қарастырылатын тапсырмалар шеңбері оларды толығырақ қарастыруға мүмкіндік бермейді.

Сөздік құрылымын жақсартудың тағы бір қызықты жолы – бағдарламаның рефакторингі [105, р. 32-42; 106, р. 14-16]. Түрлі пікірлерге қарамастан, біздің жағдайда сөздікті қайта құрылымдау және сөздікті қайта өңдеу бірдей естіледі, бірақ айырмашылықтар бар. Негізгі айырмашылық – рефакторинг үшін әзірленген әдіс объектіге бағытталған кодтарды түрлендіруде қиын бағытқа ие.

2 WORKING SET СТРАТЕГИЯСЫ НЕГІЗІНДЕ ИННОВАЦИЯЛАР СӨЗДІГІН ОҢТАЙЛАНДЫРУ

Бұл бөлімде сөздікті оңтайландыруға бағытталған зерттеулер мен олардың нәтижелері ұсынылған. Инновациялар сөздігін оңтайландыру мәселесі бойынша сәйкес модельдерді тереңрек қарастыру осы зерттеулерді жүргізуге бірден-бір себепкер болып отыр. Біздің жағдайда, бағалау функционалын азайту мақсатында сөздік беттеріне ұғымдарды қайта орналастыру NP -қиын мәселе болып табылады және олар үшін кластерлік пен эвристикалық тәсілдерге негізделген әдістерді қолдануға болады. Бөлімде есептеу процесінің геометриялық интерпретациясы табылған, ол жұмыстағы процесстерді толықтай анықтауға және қайта құрылымдаудың математикалық моделін, соның ішінде функционал мен шектеулерді құруға көмектеседі және оларға қолайлы шешімдер беріледі. Модель және оның ерекшеліктері одан әрі алгоритмді құру бойынша оңтайлы немесе ε - оңтайлы шешімдер алуға негіз береді және алгоритмнің кейбір қасиеттері қарастырылған.

2.1 Working set стратегиясының сипаттамасы

Working set, қазақ тілінде жұмысшы жиын, моделі - информатикадағы белгілі бір уақыт аралығында процеске қажетті жад көлемін анықтайтын ұғым. Бұл модельді 1968 жылы АҚШ ғалымы Питер Деннинг (Peter Denning) өзінің «The Working set model for program behavior» [113] атты жұмысында ұсынған. Диссертациялық жұмысты орындау барысында Working set моделінің ерекшеліктеріне сүйе отырып, тапсырманы шешу үшін қолданылды. Сонымен қатар, диссертацияда Working set атауының қазақша аудармасы «Жұмысшы жиын» деп беріледі.

Процестер жадта қажетті беттерсіз жұмыс жасай бастайды. Нәтижесінде, машинаның алғашқы нұсқауы орындалған кезде, сөздіктің бір бөлігін жүктеуді қажет ететін бет қателігі пайда болады. Келесі бет қателігі ғаламдық айнымалыларды локализациялау кезінде, екіншісі стек үшін жады бөлу кезінде пайда болады. Процесс өзіне қажет беттердің көпшілігін жинағаннан кейін, бет қателігі сирек кездеседі. Осылайша, белсенді түрде қолданылатын беттер жиыны бар (P_1, P_2, \dots, P_n), бұл белгілі бір t уақыт аралығында нәтижелі жұмыс істеуге мүмкіндік береді және көптеген бет қателіктерін болдырмайды. Бұл беттер жиыны процестің жұмысшы жиын деп аталады. Жұмысшы жиындағы беттердің саны T параметрімен анықталады, T функциясы төмендемейді және салыстырмалы түрде аз. Кейде T процесс бақыланатын жұмысшы жиын терезесінің өлшемі деп аталады. Көптеген мекенжайлармен жүйелі түрде жұмыс істейтін тест бағдарламасын жазу оңай, бірақ, көптеген нақты процестер осылай әрекет етпейді, дегенімен жергілікті меншікті көрсетеді. Есептеудің кез-келген кезеңінде процесс аз беттермен жұмыс істейді. Процесс жұмыс істеп тұрған кезде ол бір жұмысшы жиыннан екіншісіне ауысады. Сөздік бір-біріне сәйкес келуі мүмкін бірнеше жұмысшы жиыннан тұрады. Мысалы, процедура шақырылған кезде, оның жергілікті және ғаламдық айнымалы мәндері бар

беттерден тұратын, процедураның нұсқаулары бар жаңа жұмысшы жиынын анықтайды. Оны аяқтағаннан кейін процесс осы жұмыс жиынынан кетеді, бірақ оған жаңа процедуралық шақыру кезінде оралуы мүмкін. Осылайша, жұмысшы жиыны сөздік мәліметтерімен анықталады. Егер процесс жұмысшы жиынға қажетті мөлшерден аз блок бөлсе, онда ол трешинг күйінде болады [114].

Сілтемелердің орналасу принципі жұмыс процестерінің жиынына жиі ауысуға жол бермейді. Ресми түрде мұны былайша білдіруге болады. Егер $(t-T, t)$ уақыт аралығында сөздік $W(t, T)$ беттеріне сілтеме жасаса, онда T мәнін дұрыс таңдау арқылы $(t, t+T)$ уақыт аралығында осы сөздіктің сол беттерге қол жетімді болуы әбден мүмкін. Басқаша айтқанда, жергілікті ұстаным бойынша егер адам болашаққа тым алыс қарамаса, онда оны өткен шақ арқылы дәл болжай алуға болады. Уақыт өте келе процестің жұмысшы жиыны өзгеруі мүмкін (беттердің құрамы жағынан да, олардың саны жағынан да).

Жұмысшы жиынның маңызды қасиеті - оның мөлшері. ОЖ әр процесті жұмысшы жиынына сәйкес келетін блоктардың жеткілікті мөлшерімен қамтамасыз етуі керек. Егер блоктар әлі де сақталатын болса, онда басқа процесс басталуы мүмкін. Егер процестердің жұмысшы жиынын жадта сақтау мүмкін болмаса және трешинг басталса, онда процестердің біреуін дискіге жүктеуге болады. Процестерді жадқа орналастыру туралы шешім оның мөлшеріне негізделуі керек. Бастапқы процестер үшін бұл шешім эвристикалық түрде қабылдануы мүмкін. Процесс барысында жүйе мыналарды анықтай алуы керек: процесс жұмыс жиынын кеңейтеді немесе жаңа жұмыс жиынына ауысады. Егер бет атрибутына t_i соңғы пайдалану уақытын қоссақ (беттің i нөмірі үшін), онда t беті T уақытында параметрмен анықталған жұмысшы жиынға жатады [115-117].

Бұл тәсілді жүзеге асырудың тағы бір жолы процесс нәтижесінде туындаған бет қателігі санын бақылауға негізделуі мүмкін. Егер процесс көбінесе бет қателігін тудырса және жады толы болмаса, жүйе оған бөлінген блоктар санын көбейте алады. Егер процесс белгілі бір уақыт ішінде ерекше жағдайларды тудырмаса және генерация деңгейі белгілі бір шектен төмен болса, онда процесс блоктардың санын қысқартуға болады. Бұл әдіс тек процестің құрамына кіретін беттер жиынтығының көлемін басқарады және оны бетті алмастырудың қандай-да бір стратегиясымен толықтырылуы керек. Бір жұмысшы жиыннан екіншісіне ауысу кезінде жүйенің тоқтап қалуына қарамастан, ұсынылған шешім ешқандай қосымша жүйелік конфигурацияны қажет етпестен, әр бір процес үшін ең жақсы өнімділікті қамтамасыз ете алады.

2.2. Инновациялар сөздігін оңтайландыру алгоритмі

Сөздік жұмыс орындалған кезде олардың беттеріне сілтемелер тізбегі пайда болады. Сонымен қатар, сөздіктің орындалуының кез келген сәтінде сөздіктің барлық беттері негізгі жадта орналаспайды, бірақ олардың кейбіреулері ғана (резиденттер жиыны) орналасады. Егер қазіргі уақытта сілтеме жасалған бет жедел жадыда болмаса, онда бет қателігі орын алады және сөздік жұмысы көмекші жадтан керекті бетті жүктеуге қажет уақытқа үзіледі. Егер ағымдағы уақытта жедел жадыда орналасқан бетке сілтеме болса, бет сәтсіздігі орын

алмайды және сөздік жұмысы үзілмейді. Сөздік жұмысын орындау кезінде бет қателіктер саны жалпы жүйенің жұмысына әсер етеді және беттерді ауыстыру стратегиясына арналған кездейсоқ мәліметтер функциясы болып табылады. Беттің көптеген ақауларын тудыратын сөздік жұмысына байланысты есептеу жүйелерінің жұмысында күрт жоғалудың белгілі мысалдары бар. Статикалық жады бар есептеу жүйелерін оңтайландыру әдістерінің бірі - ең алдымен жүйелік бағдарламаларға, яғни тұрақты (жиі) жұмыс істейтін сөздіктерге бағытталған оңтайлы қайта құрылымдау әдісі (сөздікті қайта құрылымдау).

Сөздікті қайта құрылымдау әдісін қолдану сөздіктің белгілі бір санына есептелген бет қателіктерінің орташа санын азайтуы мүмкін. Сонымен, [118] жұмысында сипатталған осындай эксперименттердің саны 50, ал [119, 120] жұмыстарында эксперименттер саны 1-ге тең (аспаптық эксперимент).

Көптеген жұмыстарда қайта құрылымдау кезінде сөздік жұмысын оңтайландыру негізінен белсенді беттерде орналасқан ұғымдарды байланыстыру функцияларындағы «бейтараптандыруға» байланысты. Сонымен қатар, белсенді беттердің арасындағы қателікке әкелмейтін байланыстарды «бейтараптандырған» жөн. Бұл беттер, егер олардың саны 1-ден көп болса, атап айтқанда, алмастыру стратегиясымен анықталады.

Енгізілген мәліметтер сапасының кенеттен өзгеруі және осы деректерді қою кезіндегі қателіктер қайта құрылымдау мәселесінің оңтайлы шешімдерінің тұрақтылығын зерттеу міндетін шешуге әкеледі. WS бетті алмастыру алгоритмі, алмасу стратегиясы ретінде пайдаланылады, ол сөздік жиынтығының маңызды ішкі қасиетін, яғни жергілікті меншікті ескеруге мүмкіндік береді. Жұмысшы жиын стратегиясына сәйкес, сөздік жұмысы орындалуының кез-келген сәтінде белгіленген уақыт интервалында өңделген сөздік беттері белсенді болып саналады. Бұл стратегия уақыттың әр түрлі белсенді беттермен алмасу алгоритмдері тобының негізгі өкілі. Бұл топтың алгоритмдері, мысалы, VMIN, PFF, GWS, SWS және олардың модификациялары жұмысшы жиын тұжырымына негізделген [121].

WS стратегиясының әмбебап сипаты сонымен қатар сөздік жұмысының кез-келген уақытында жұмыс беттерінің жиыны сөздікке қажет негізгі жадының көлемін бағалауға мүмкіндік береді және осындай шараны атқаратын жұмысшы жиынның мөлшері болып табылады. Жұмысшы жиынның мөлшері, беттің бұзылу жиілігіне қарағанда, статикалық жадыны бөлу алгоритміне тәуелді емес.

Беттің бұзылу жиілігі – бұл, бір жағынан, кездейсоқ мәліметтері бар сөздікті сипаттайтын көрсеткіш, екінші жағынан - жүйелік алгоритм – беттерді ауыстырудың алгоритмі (стратегиясы).

Сонымен, алмастыру алгоритмдерінің ішінде теориялық және эксперименттік зерттеулердегі басты объект (есептеу процесін басқару теориясында ерекше рөл атқаратындығының арқасында) WS стратегиясы деп айтуға болады. Жұмыстардың үлкен сериялары жұмысшы жиынның қасиеттерін, оның сипаттамаларын, бағдарламалық орналасу феноменімен және бағдарламалардың трансформациялық өзгерісімен байланысын зерттеуге арналған. Мұнда көптеген жағдайларда қосымша талдау қажет етеді. Осы

жұмыстардың барлығын талқылау осы диссертациялық жұмыстың шеңберіне кірмейді. Бұл мәселенің нақты шешімдерін алу үшін WS алмасу стратегиясымен сөздікті қайта құрылымдау мәселесін зерттеу өте қызықты және маңызды екені сөзсіз. WS стратегиясы жағдайында қайта құрылымдау мәселесі белгілі әдістердің қолданылуын шектейтін бір ерекшелігі бар екенін атап өткен жөн, бұл жерде белсенді сөздік беттерінің саны уақыт өте келе өзгеріп отырады. Бұл жағдай қайта құрылымдау мәселесін шешуге болатын шешімдер жиынын сипаттауды қиындатады. WS стратегиясы үшін, жарияланымдардың көптігіне қарамастан, негізгі міндеттің нақты формасы да, мүмкін шешімдер жиынын анықтайтын шектеулердің формасы да ұзақ уақыт белгілі болмады. WS стратегиясымен қайта құрылымдау жағдайында, Д.Феррари бағдарламаның деректерді пайдалана отырып, CWS қайта құрылымдау алгоритмін қолдану арқылы отырып жүйелік бағдарламалардың орналасуын жақсартуды ұсынды [122, 123]. Алайда, CWS алгоритмі де, оның негізінде құрылған бағдарламалық жасақтаманы қайта құрылымдау процесі де эвристикалық процедуралар болып табылады.

2.2.1 Инновациялар сөздігін оңтайландырудағы негізгі шектеулер

Қайта құрылымдауды талап ететін сөздік ұғымдары $i_{j_1}, i_{j_2}, \dots, i_{j_n}$ тәрізді n ұғымдардан тұрсын, қысқаша i_1, i_2, \dots, i_n олар алдын ала белгіленген және статикалық жадының S_1, S_2, \dots, S_p беттері бойынша $p > 1$ болып орналасқан (сурет 2.1). Бет қателігінің көп мөлшерде пайда болуы біршама мәселе тудырады, бұл сөздік құрылымының нашарлығына байланысты болуы мүмкін және сөздікпен қоса жүйенің де өнімділігінің төмендеуіне әкеледі. Сөздіктің мұндай жағымсыз сипаттамасының себебі сөздік бөліктері әртүрлі топтағы әртүрлі бағдарламашылармен жазылған болуы мүмкін, ал кейбіреулері тіпті таныс емес болуы да ықтимал. Сонымен қатар, кейбір жағдайларда сөздіктің кейбір бөліктері уақыт өте келе жоғалады, кейде сөздіктің маңызды бөлшектерін қалпына келтіру мүмкін емес.

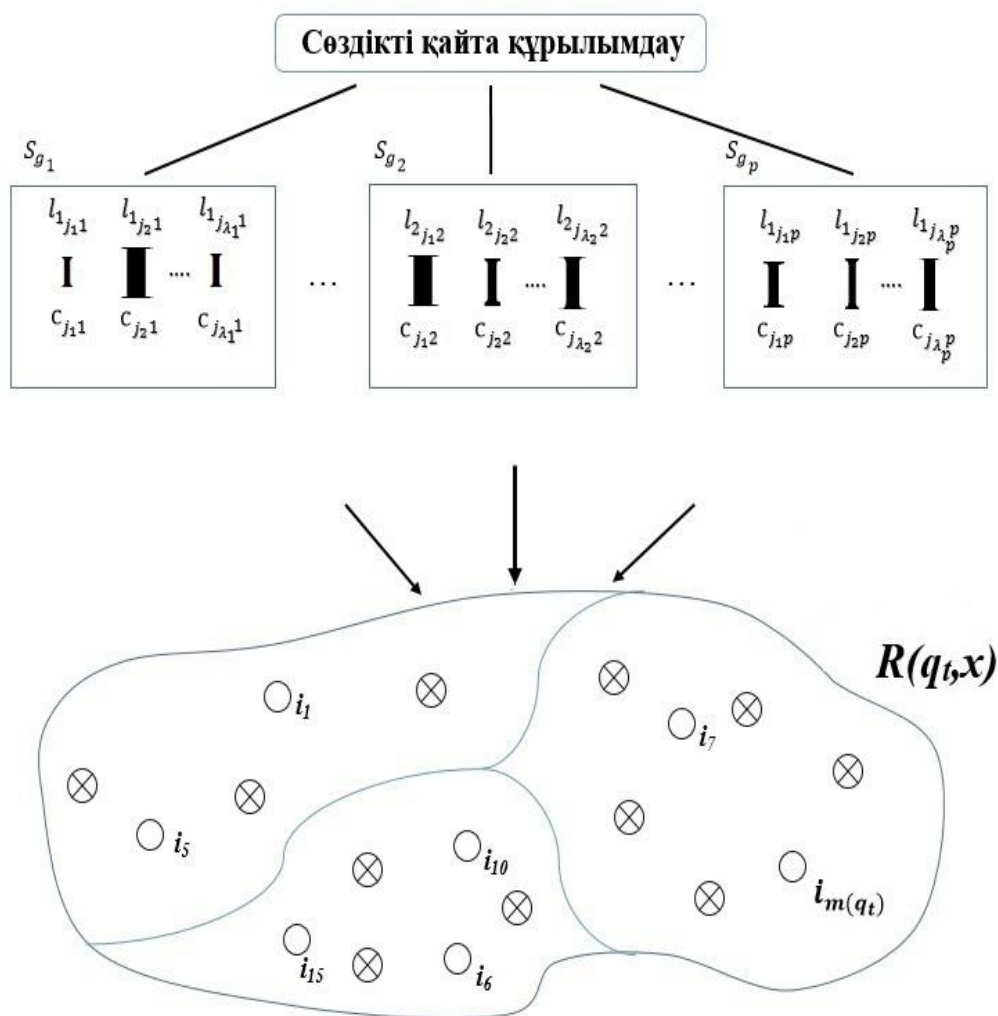
Сонымен, v_r дегеніміз r беттің ұзындығы, $r = 1, 2, \dots, p$ және әр бір ұғым $i_i, i = 1, 2, \dots, n$ ұзындығына ие және ол екінші деңгейде, яғни динамикалық жадында өңделу барысында өзгермейді. Осылайша, жүйе беттері көпөлшемді. Инновациялар (ұғымдар) бір-бірімен екінші деңгейлі жадыда, яғни сөздіктің динамикалық жадында өзара әрекеттеседі, мұнда инновациялар (ұғымдар) өңделеді және қалыптасады. Мұндағы i_1, i_2, \dots, i_n ұғымдары S_1, S_2, \dots, S_p беттеріне орналасқан, мысалы, $x = (x_{ri})_{p \times n}$ бульдік матрицасы арқылы, мұндағы $x_{ri} = 1$ элементі егер i нөмірлі ұғым r номерлі бетке тиесілі болса және кері жағдайда $x_{ri} = 0$ болады. Мұндай типті барлық матрицалар X арқылы белгіленеді.

Біздің негізгі тапсырманың тұжырымына сәйкес, мақсатты функционалды азайту мақсатында S_1, S_2, \dots, S_p статикалық жады беттері бойынша ұғымдарды жылжыту белгілі NP-қиын мәселе болып табылады. Осылайша, бұл мәселе теорияның қолданбалы мәселесі ретінде қарастырылса да, зерттеу

қызығушылығын арттыратын және қосымша немесе тіпті негізгі ынталандырушы бола алатын іргелі аспектіге ие.

Біздің тапсырмаға орала отырып, кездейсоқ деректерді $D = \{\theta\}$ деп қарастырайық және алмасу стратегиясы функционалды мәнге де әсер етеді. Диссертацияда алмасу стратегиясы ретінде жұмысшы жиын стратегиясын (WS) таңдаудың негізгі себебінің бірі, WS стратегиясы инвариант қасиетін береді, яғни ұғымдарды сөздік беттері бойынша өзара қанша алмастырғанымен инновация өзгермейді.

Біз t сәтіндегі q_t күйін еске отырып, соңғы k сілтемелер жиыны ретінде көрсетеміз. Мұнда біз мультисет ретінде $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$ белгілеуін қолданамыз. Сәйкес жиын үшін, яғни қайталанбайтын ұғымдар үшін біз $in = (i_1, i_2, \dots, i_{m(q_t)})$ белгісін қолданамыз және $i_1 < i_2 < \dots < i_{m(in)}$ реттелген жиын деп қарастырамыз. Мұнда біз жазбадан t индексін алып тастай аламыз, себебі біз үшін маңызды емес егер $q_1 = \{i_1, i_2, i_3\}$ және $q_2 = \{i_2, i_1, i_3\}$ немесе $q_3 = \{i_3, i_1, i_2\}$ және т.б. жайлы айтар болсақ. Сонымен, біз ары қарай бұл белгілеулерді орнына реттелген (i_1, i_2, i_3) , яғни $in = (i_1, i_2, i_3)$ деп белгілейміз, мұндағы $i_1 < i_2 < i_3$.



Сурет 2.1 – Сөздікті қайта құрылымдау графы

Сөздіктің бақылау күйі $in = (i_1, i_2, \dots, i_{m(in)})$ түріне ие, мұндағы i_j - ұғым нөмірі ($j = 1, 2, \dots, m(in)$), ол i және оларды \circ деп белгіленген. Сурет 2.1 көрсетілгендей жұмыс жиыны мұнда $in = (i_1, i_5, i_6, i_7, i_{10}, i_{15}, i_{m(in)})$.

Сурет 2.1 сәйкес көрсетілген тағы бір \otimes символы бар, ол жұмысшы жиынның сәйкес бетіне тиесілі басқа ұғым (немесе оның нөмірі) бар екенін білдіреді. Әрине, q_t бақылау күйінен 2.1 суретіндегі \otimes немесе \circ деп белгіленген ұғымға сілтемесі бет қателігін бермейді. 2.1 суретінде R жұмысшы жиын ретінде $R(in, x)$ жұмысшы жиын көрсетілген, ол in бақылау күйі және x матрицасынан туындайды. Әрі қарай, жоғарыдағы көріністің кеңейтімі мен нақтылауын береміз. $x \in X$ матрицасында табиғи шектеулер ((a)-(c) шектеулері) бар, ары қарай олардың тұжырымдамалық тұрғыдан сипатталуы, содан кейін ресми мәлімдеме мен нақтылаулары келтіріледі. Мұндай типтік функциялар стандартты, мысалы стохастикалық бағдарламалау және үлгіні тану өрістері үшін. Практикалық тәсілдер бойынша [124, 125], дәлірек айтқанда, екінші функционалды сөздікті қайта құрылымдауға арналған шығындар ретінде қабылдаймыз.

Функционал: негізгі тапсырманың функционалы ретінде біз сөздіктегі бір экспериментте болған бет қателігі санының математикалық күтілімін аламыз. Қосымша тапсырманың функционалы ретінде біз сөздіктің $h \geq 1$ бетіндегі қателіктердің орташа мәнін аламыз.

Шектеу (a): кез-келген бетке тиесілі ұғымдардың жалпы ұзындығы осы беттің ұзындығынан аспайды.

Шектеу (b): сөздіктің кез-келген ұғымы тек бір ғана бетке тиесілі.

Шектеу (c): сөздікті орындау барысында пайда болған кез-келген жұмысшы жиынның жалпы ұзындығы алдын-ала белгілі жүйенің тұрақты мәнінен аспайды.

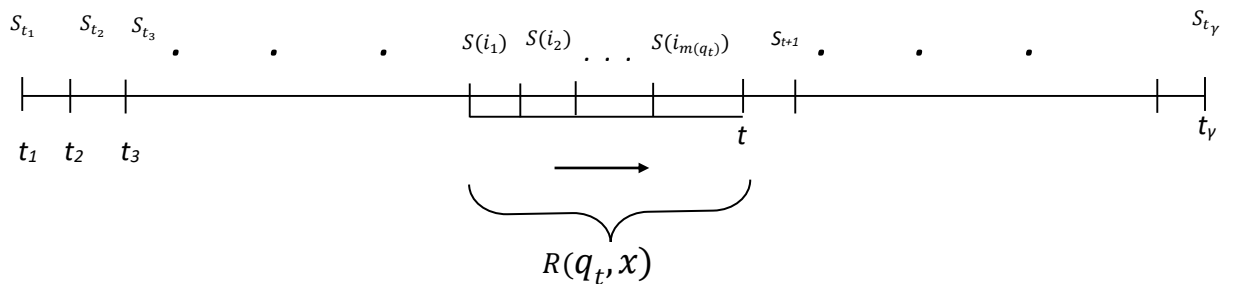
Ескерту. (a)-(c) шектеулері матрицамен белгіленуі керек, бұл i_1, i_2, \dots, i_n ұғымның S_1, S_2, \dots, S_p беттері бойынша бөлінуін анықтайтын $x = (x_{ri})_{p \times n}$.

Сөздікті қайта құрылымдауға арналған жұмыстардың көптігіне карамастан, жоғарыда аталған мәселенің нақты (оңтайлы) шешімін алу тапсырмасы келесі жұмыстарда қарастырылған [126, 127] (Қосымша А.)

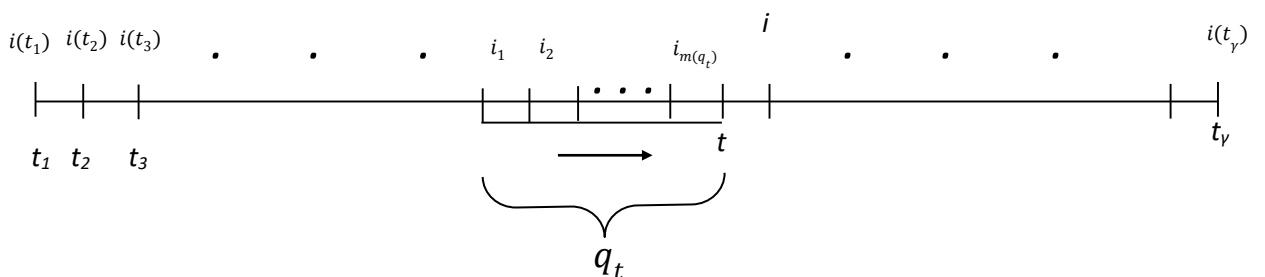
Іс жүзінде, оңтайландыру мәселесін шешкенде, әдетте қайта құрылымдалған сөздікпен жұмыс істейтін деректер жиыны нақты немесе жиі өзгермейді деп санаймыз. Оның орындалмауы сөздікті қайта-қайта құрылымдау қажеттілігіне әкелуі мүмкін. Сөздікке енгізілген мәліметтердің кенеттен өзгеруі сияқты жағдайдың зардаптарын өтеу үшін әзірленген тиімді тәсілдердің бірі - оның параметрлері бұзылған жағдайда ғана қайта құрылымдау мәселесін қарастыру. Егер сөздіктің кіріс деректер жиынының өзгеруі қолайсыздық тудырмаса, онда қайта құрылымдау мәселесі қойылмайды. Алдыңғы бөлімде инновациялар сөздігін оңтайландыру тапсырмасының функционалы және шектеулер жүйесі қарастырылады; Сонымен қатар есептеу процесінің геометриялық интерпретациясы негізінде инновациялар сөздігін оңтайландыру тапсырмасының өлшемін азайту мүмкіндігі көрсетіледі.

2.2.2 Беттерге және ұғымдарға сілтемелер жолы

Сөздік беттеріне және сол беттерде орналасқан ұғымдарға сілтемелер жолы t ось бойымен біртіндеп солдан оңға қарай жылжитын k ұяшықтарға бөлінген. Сурет 2.2 және 2.3 сәйкес көрсетілгендей $t_1, t_2, t_3, \dots, t_\gamma$ сәттері t_0 бастап t_γ сәтімен аяқталатын іріктеу уақыттары болып табылады, бұл $\theta \in D$ кездейсоқ мәліметтермен эксперименттің соңғы сәті. Сурет 2.2 көрсетілгендей t осінің жоғарғы жағындағы $S_t, t=1,2,3,\dots,\gamma$ дегеніміз бұл беттер (немесе олардың нөмірлері) болып табылады, олар үшін $k=4$ берілген және $x \in X$ бекітілген. Сурет 2.3 үшін $i(t_j), j = 1, 2, \dots, t_\gamma$ белгісі - бұл сөздікті орындау кезінде сілтеме жасалған $k=4$ және $x \in X$ бірдей болатын 2.2 суреттегі беттер нөмірлеріне сәйкес келетін ұғымдар нөмірлері. Сурет 2.2 сәйкес сүзгінің мазмұны, t сәтінде q_t бақылау күйімен және $x = (x_{ri})_{p \times n}$ матрицасынан туындаған $R(q_t, x)$ жұмысшы жиынын құрайтын беттердің нөмірлері болып табылады. $S(i_j)$ беті бұл $q_t, j = 1, 2, \dots, m(q_t)$ бақылау күйінің i_j ұғымын қамтитын бет. Біздің жағдайда жұмысшы жиынның басқа түрі жоқ. Осылайша, $R(q_t, x)$ мультижиыны ретінде $\{S(i_1), S(i_2), \dots, S(i_{m(q_t)})\}$ болады, бірақ іс жүзінде $R(q_t, x)$ сәйкес келетін жиын қайталанбауы керек. Айталық, $R(in, x)$ дегеніміз $R(q_t, x)$ мәніне сәйкес келетін қайталанбайтын беттердің белгіленуі болсын. Мұнда, $R(q_t, x)$ мәнінен $R(in, x)$ мәніне өту q_t параметріндегі қайталануларды өшіру арқылы оңай орындалады.



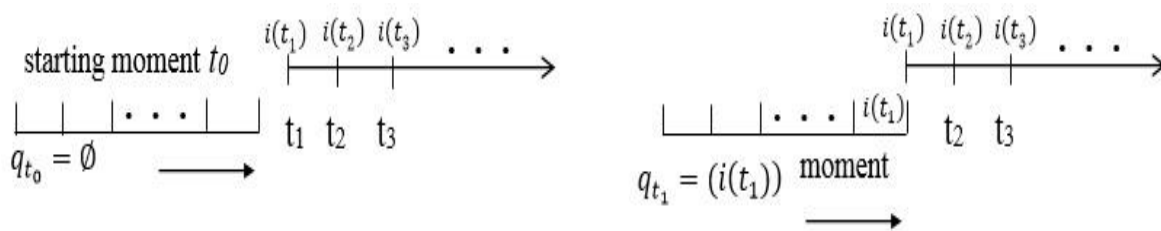
Сурет 2.2 – $R(q_t, x)$ жұмысшы жиыны бар сөздік беттеріне сілтемелер жолы



Сурет 2.3 – Сөздік ұғымдарына сілтемелер жолы

Сурет 2.3 сүйене отырып, q_{t+1} келесі бақылау күйі $q_t \cup \{i\}$ ретінде құрылғанын байқауға болады, яғни $q_{t+1} = \{i_2, i_3, \dots, i_{m(q_t)}, i\}$. Мұндай оқиға, яғни, i мәнін бұғаттау үшін q_t бақылау күйін жіберуді, $q_t \rightarrow i$ деп белгілей

аламыз. t_0 уақытының алғашқы сәтінде бізде беттерге сілтемелер жолы, ұғымдар және бос сүзгі бар. Бұл келесі 2.4 суретіне сәйкес көрсетілген.



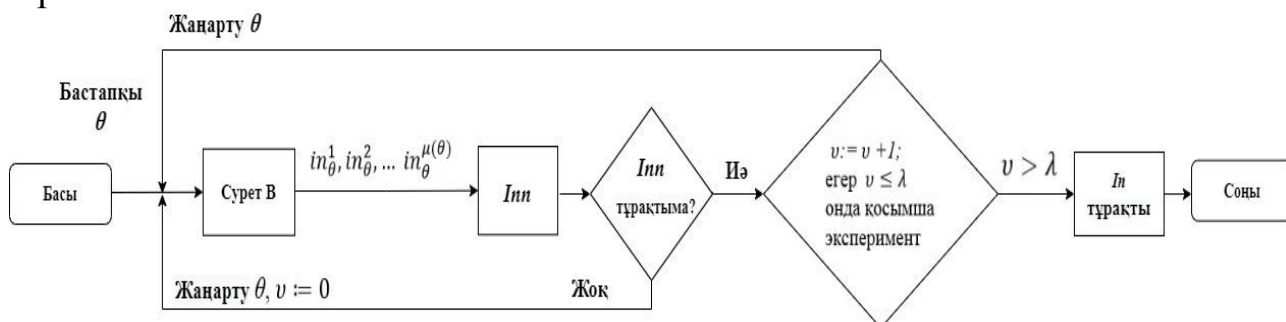
Сурет 2.4 – t уақытында ұғымдарға сілтемелер жолы

Мұнда, $i(t_1)$ ұғымына сілтеме сүзгісіне $i(t_1)$ жету нүктесінен бұрын пайда болады. Осылайша, сүзгінің ішіндегі ұғымдардың нөмірлері қайталанбайды және t сәтіндегі сөздіктің бақылау күйіне сәйкес келеді, сонымен бірге $in = (i_1, i_2, \dots, i_{m(in)})$ ішінде қаталанатын ұғымдар болуы мүмкін, яғни $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$. Сонымен қатар, q_t мәнінен t индексін шығарып тастай аламыз, өйткені біз үшін $\{i_1, i_2, i_3\}$ немесе $\{i_3, i_1, i_2\}$, т.б. арасында ешқандай айырмашылық жоқ, біз олардың барлығын рет бойынша жазамыз $\{i_1, i_2, i_3\}$, мұнда $i_1 < i_2 < i_3$. Және $in = (i_1, i_2, \dots, i_{m(in)})$ бақылау күйлері үшін де дәл солай корреляция бар: $i_1 < i_2 < \dots < i_{m(in)}$.

Сүзгіні ұғымдарға сілтеме жол бойымен солдан оңға қарай жылжытсақ, біз бақылаудың бірнеше күйін аламыз, бірақ олардың кейбіреулері көпжиынды болуы мүмкін, сондықтан біз бірдей жиын ретінде қайталанбайтын көпжиынды және сәйкес жиынды да қарастырдық. Осыған қарамастан, q_t үшін, біз басқа белгілену енгіздік, атап айтқанда $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$ және q_t -ті көпжиын түрінде өңдедік. Табылған q_t ішінен қайталанатын ұғымдарды жоямыз, in_t және in өзара салыстыру арқылы ағымдағы t уақытында пайда болған жаңа ұғымды жиынға жаңа ретінде енгіземіз. Осылайша, біртіндеп сөздіктің жаңа эксперименті арқылы Inn жиыны өзгермейтін жағдайға жетеміз. Сонымен, $Inn = \{in\}$ жиыны әр түрлі in -тен тұрады, мұндағы in мәнінде ұғымдардың қайталануы болмайды.

Сүзгіні t осі бойымен жылжыту арқылы біз жаңа инновацияларға үміткерлерді аламыз, өйткені олардың кейбіреулері сөздіктің алдыңғы жүгірісінде кездесен болуы мүмкін және бұл кез-келген жағдайда сүзгідегі қайталанатын үміткерлерді жою үшін өндеуді қажет етеді. Қайталанатын үміткерлерді алып тастағаннан кейін, мысалы, q_t түрінде, сәйкестігін тексеру қажет, яғни сәйкес in инновациясын Inn инновациялар жиынында бар ма әлде жоқ па. Егер in жаңа болса, яғни, егер Inn жиынында in инновациясы жоқ болса, онда оны Inn жиынына жаңа инновация ретінде қосамыз. Бірнеше жүгірістен кейін Inn мәндері тұрақты шамаға жақындайды және сөздіктің жаңа жүгірістеріне қарамастан, инновациялар жиыны өзгермейтін жағдайға жетеді. Inn инновациялар жиыны бірінші рет тұрақты болады, содан кейін сөздіктегі $\lambda \geq 1$ қосымша жүгірістен кейін нақты тұрақтылыққа жетеді. Мұнда λ - сыртқы процедураның параметрі. Мұнда λ параметрі әр кімнің өз қалауынша, мысалы

кейбіреулер қосымша жүгірістер санын 10, 100 немесе тіпті 1000 деп белгілеуі мүмкін.



Сурет 2.5 – In қалыптасқан инновациялар жиынын анықтау

Сурет 2.5 сәйкес көрсетілгендей $\{in_{\theta}^1, in_{\theta}^2, \dots, in_{\theta}^{\mu(\theta)}\}$ жиыны бұл сөздіктің бір экспериментінде алынған инновациялар мультижиыны, ол $\theta \in D$ бөлек шамасына сәйкес келеді және әр бір қайталанбайтын инновация Inn жиынына қосылады. Мұндағы:

θ – жеке эксперимент үшін есептеу процесін бастайтын кездейсоқ деректер;

v – айнымалы (эксперимент есептеуіші);

Inn – қалыптасу процессіндегі инновациялар жиыны;

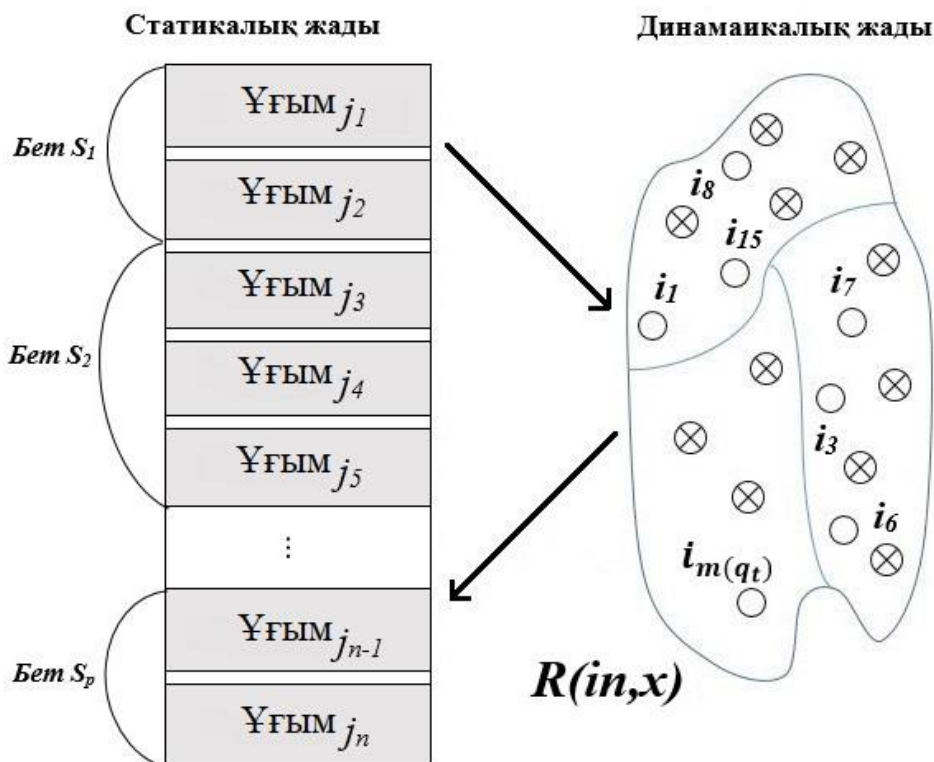
In – қалыптасқан инновациялар жиыны.

Inn тұрақтыма сұрағына келер болсақ, алғашқы ие деген жауап келесідей мағынаны білдіреді: осыған дейін Inn кез-келген итерация үшін тұрақсыз болған және ағымдағы итерация үшін Inn тұрақты болды және алдыңғы итерациядағы Inn мәнінен айырмашылығы жоқ. Содан кейін, егер $v \leq \lambda$ онда үрдіс жоғарғы сызық бойымен қозғала отырып θ мәнің жаңартады және ары қарай схема бойынша жұмыс жасайды. Соңында Inn инновациялар жиыны $\{in\}$ ішіндегі түрлі инновациялардан тұрады, айалық мұнда $Inn = \{in\}$ болсын, және әр бір in ішінде қайталанатын ұғымдар болмайды.

Сонымен, $\lambda \geq 1$ қосымша жүгірістен кейін қосымша эксперименттерді қайта санап, оларға 1,2, ден n -ге дейін жаңа сандар тағайындай отырып, біз бет бойынша қайта орналасқан және әрі қарай қарастыруға ыңғайлы көптеген ұғымдарды аламыз. Алайда мұнда келесідей сұрақ туындауы мүмкін: егер $\lambda \geq 1$ қосымша жүгірістен кейін де жаңа басқару күйі табылса жағдай қалай болады? Демек, λ үшін таңдау тым оптимистік болды, сондықтан λ мәнін қайта қарап, жұмысты басынан бастау керек.

Сурет 2.6 сәйкес көрсетілгендей суреттің сол жағында беттері бар сөздіктің статикалық деңгейін көреміз, беттерде ұғымдар орналасқан, барлығы n ұғым. Суреттің оң жағында беттердің жұмысшы жиынын көреміз, олар ұғымдардың ішжиынан тұрады, олардың барлығы \circ немесе \otimes деп белгіленген, \circ ретінде белгіленгендер инновацияны құрайды, ал \otimes ретінде белгіленгендер жұмысшы жиынға кіретін ұғымдардан тұрады және олардың ешбірі инновацияға кірмейді. Әлбетте, \circ немесе \otimes деп белгіленген ұғымға кез-келген сілтеме бет қателігін

көрсетпеуі қажет. Керісінше, $R(q_t, x)$ жиыннан тыс $R(q_t, x)$ жиыннан шыққан кез-келген сілтеме бет қателігін береді.



Сурет 2.6 – Сөздіктің статикалық және динамикалық жадысы

2.2.3 Траекторияның бақылау күйі

Әдетте жұмысшы жиын ұғымын қарастырғанда екі байланысты ескертулер қолданылады, олардың біреуі $W(t-\tau, t)$, терезе өлшемі τ , ал келесісі $W(k, t)$, екеуі де П. Деннингтің жұмысына оралады [128, 129]. Екіншіге, $k \geq 1$ бүтін саны терезе өлшемі ретінде де қарастырылуы мүмкін. Бірінші жағдайда жұмысшы жиын ретінде статикалық уақыт интервалында $(t-\tau, t)$ сілтеме жасалған сөздік беттері қарастырылады. Біздің жағдайда, жұмысшы жиын ретінде $R(q_t, x)$ нұсқасын қарастырдық. Траекторияның q_t бақылау күйлерін еске отырып, t уақытына дейін ұғымдарға жасалған соңғы сілтемелер жиынын көрсетейік. Осылайша, іс жүзінде t уақытындағы сөздіктің бақылау күйі t уақытындағы жұмысшы жиыны ұғымының аналогы болып табылады.

Бульдік матрица $x = (x_{ri})_{p \times n}$ маңызды рөл атқарады, ол жоғарыда айтылғандай, сөздіктің құрылымын анықтайды, яғни сөздіктің S_1, S_2, \dots, S_p беттерінде орналасқан i_1, i_2, \dots, i_n ұғымдары. Сонымен қатар, x шектеулері үшін (a) - (c) шектеулері сақталуы керек және осы типтегі барлық матрицалар X жиынын құрайды, ал соңында X матрицалары арасында жоғарыда аталған функцияларға сәйкес сөздіктің оңтайлы құрылымын белгілейтін матрица анықталуы керек.

Траекторияның q_{t+1} бақылау күйі $\{i\}$ мәнімен байланысқан q_t жиыны ретінде қалыптасады, содан кейін келесі қатынастар орындалады: $q_{t+1} \subset q_t$ немесе $q_t \subset q_{t+1}$ немесе $q_{t+1} = q_t$.

Осылайша $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$ және $q_{t+1} = \{i_2, i_3, \dots, i_{m(q_t)}, i\}$ бақылау күйі, содан кейін:

I. Егер $i_1 \in q_{t+1} = \{i_2, i_3, \dots, i_{m(q_t)}, i\}$ және $i \notin \{i_2, i_3, \dots, i_{m(q_t)}\}$, онда $q_t \subset q_{t+1}$, (сурет 2.11а);

Келесі шешім

II. Егер $i_1 \in q_{t+1}$ және $i \in \{i_2, i_3, \dots, i_{m(q_t)}\}$ онда $q_t = q_{t+1}$ (сурет 2.11ә);

III. Егер $i_1 \notin \{i_2, i_3, \dots, i_{m(q_t)}\}$ және $i \notin \{i_2, i_3, \dots, i_{m(q_t)}\}$ онда $q_t \not\subset q_{t+1}$ (сурет 2.11б), мұндағы $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$ және $q_{t+1} = \{i_2, i_3, \dots, i_{m(q_t)}, i\}$;

IV. Егер $i_1 \notin \{i_2, i_3, \dots, i_{m(q_t)}\}$ және $i \in \{i_2, i_3, \dots, i_{m(q_t)}\}$ онда $q_{t+1} \subset q_t$ (сурет 2.11в), мұндағы $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$, $q_{t+1} = \{i_2, i_3, \dots, i_{m(q_t)}\}$.

I, II, IV кадамдар стандартты, бірақ III кезең үшін *in* аралық түйіннің астынан $t + 1/2$ сәтіне тең уақыт өтеді деп болжауға болады.

Әр түрлі $\theta \in D$ деңгейінде сөздікте бірнеше эксперимент жасап және q_t бақылау күйін бірнеше рет сәйкестендіріп, келесі іске қосу кезінде *Inn* жиыны өзгермейтін сәтке жеткенде кезде *Inn* жиыны анықталды деп есептейміз. *Inn* жиынына бастапқыда процестің басталу нүктесіне сәйкес келетін $q_0 = \emptyset$, арнайы бақылау күйін қосайық. q_0 бақылау күйі кейінірек, яғни, біздің сөздік негізгі жадыдан күтпеген жерден шығарылған кезде де пайда болуы мүмкін және біраз уақыттан кейін сөздік басынан (*суық іске қосу*) басталғандай іске қосылады.

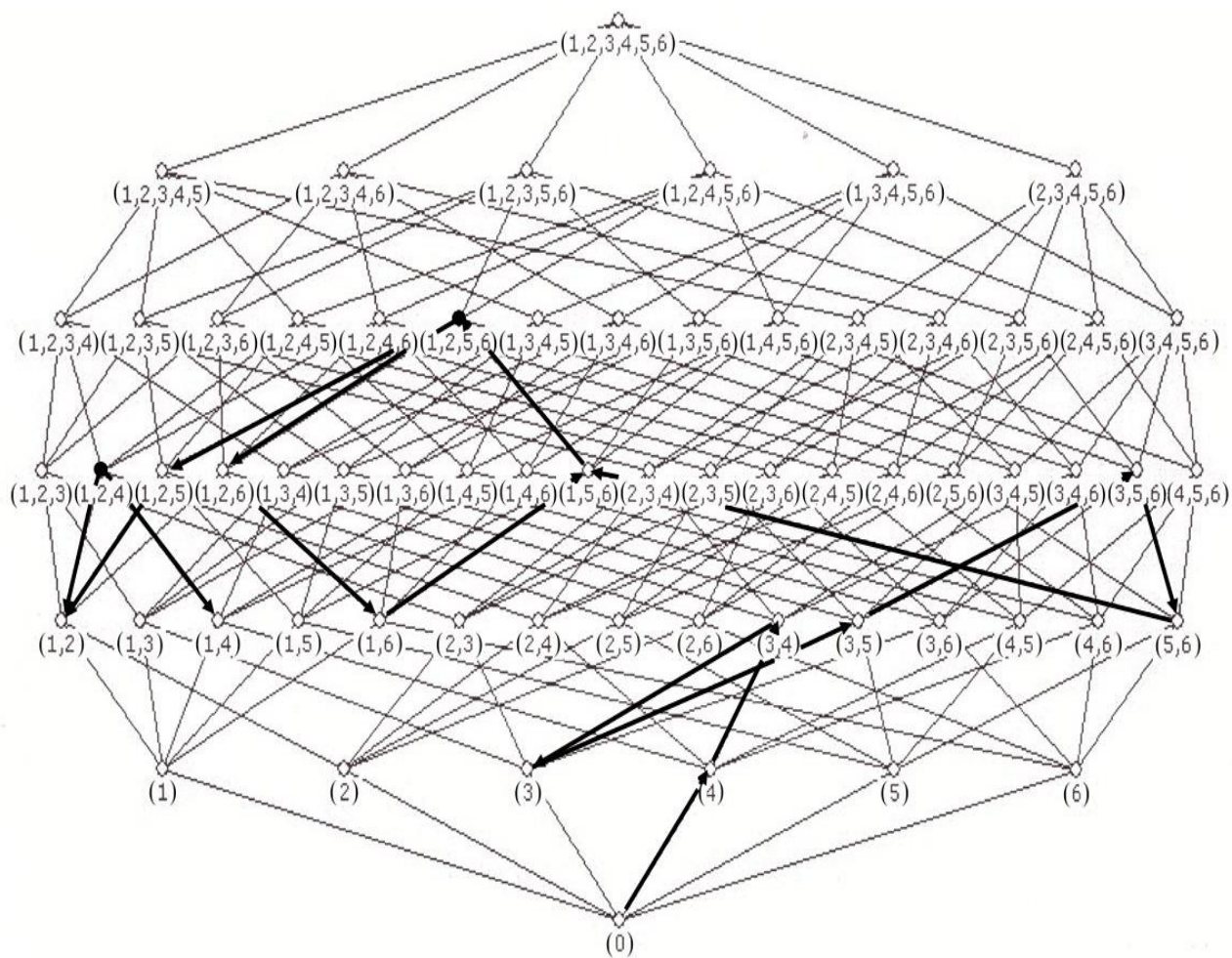
Тағы бір іске қосу – *жылы іске қосу*, яғни, жүйе есептеу процесін қалпына келтіруге мүмкіндік жасай отырып процесті қатесіз жалғастыруға мүмкіндік береді. Әрі қарай, кез-келген осындай оқиғаны «жылы іске қосу» (қайта қосу) ретінде қалпына келтіруді ұсынамыз және біз оны (1), (2) өрнектерінде негізгі және қосымша тапсыраманың функционалына арналған қосымша бет қатесі ретінде қарастырамыз.

2.3 Есептеу процесінің геометриялық интерпретациясы

Инновациялар сөздігін оңтайландыру тапсырмасының өлшемін азайту мақсатында есептеу процесінің геометриялық интерпретациясы қолданылды. Әрі қарай, біз есептеу процесінің геометриялық интерпретациясын береміз. Моделді құру және негізгі мәселені шешу үшін біз бірегей комбинаторлық кеңістікті қолданамыз (Булеан). Бұл екі полюсті және n -өлшемді екілік кубтың жазықтық аналогы. Төменгі полюсте элементтері бос шың орналасқан, осы екі полюстің бірінші қабатында бір элементті жиындар, екіншісінде екі элементтік жиындар орналасқан және т.с.с. Ортасынан бастап қуат азая бастайды және жоғарғы полюсте концепцияның барлық жиынтығына сәйкес келетін шың орналасқан, біздің жағдайда ол 6 санына тең. Процестің немесе эксперименттің траекториясы Булеан шыңдары бойымен кездейсоқ кезуге сәйкес келеді. Мұнда төменгі бөлікте басқылау күйлер пайда болады. Ал t осі бойымен қозғалу траекториясының өзі Булеан шыңдар тізбегіне сәйкес келеді.

Геометриялық интерпретацияның көмегімен есептің өлшемін азайтуға болады, яғни (с) жағдайындағы шектеулер санын. *Inn* жиыны қалыптасқан кезде,

біз Inn ішкі жиынын табу керекпіз және ол \hat{In} деп белгіленеді, бұл (c) шектеуі сақталған жағдайда пайда болады. Кез келген $\hat{in} \in \hat{In}$ элементі қасиетке ие, дәлірек айтқанда Inn элементінде $\hat{in} \not\subseteq in$ сияқты in элементі жоқ. Тізбек булеан түйіндерін кездейсоқ айналып өтуіне сәйкес келеді (сурет 2.7). Кеңістікке келесідей атаулар береміз: кез-келген \hat{in} элементі – булеан түйіні, бұл булеан бойымен кез-келген кездейсоқ кезу барысындағы шыңдар болып табылады (2.7 суреттегі қара түйіндер). Бұл қара түйіндер кез-келген кездейсоқ кезу барсында болады, олардың мәні, яғни түйіндері ауысып отыруы мүмкін. Олардың бірден бір ерекшеліктері, бұл түйіндер басқа ешқандай түйіннің құрамына кірмейді, яғни бұл түйінен булеанның жоғарғы бөлігіне сілтемелер жоқ.

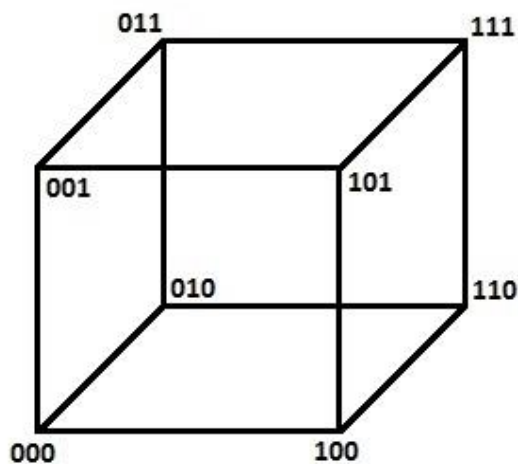


Сурет 2.7 – Булеан және түйіндер арасындағы кездейсоқ кезу

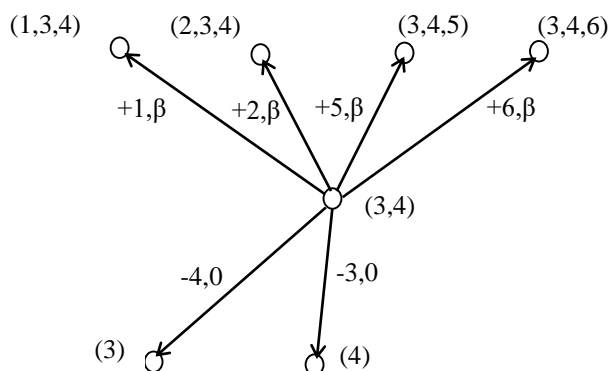
Сурет 2.7 сәйкес көрсетілген кездейсоқ жүру келесі түйіндерден өтеді: $(0) \rightarrow (4) \rightarrow (3,4) \rightarrow (3) \rightarrow (3,5) \rightarrow (3,5,6) \rightarrow (5,6) \rightarrow (1,5,6) \rightarrow (1,2,5,6) \rightarrow (1,2,6) \rightarrow (1,6) \rightarrow (1,5,6) \rightarrow (1,2,5,6) \rightarrow (1,2,5) \rightarrow (1,2) \rightarrow (1,2,4) \rightarrow (1,4)$ және бір эксперименттегі қара түйіндер: $(3,4), (3,5,6), (1,2,5,6), (1,2,4)$.

Булеан комбинаторлық кеңістіктің екі полюсі $n=6$ ұғымдар санына сәйкес келеді және $n+1$ деңгейге ие. Сөздіктің бірнеше рет іске қосылуының нәтижесі бойынша мультиграф енгізу қажет, ол булеанда көршілес түйіндер арасындағы қабырғаларды көшіру арқылы жасалады. Процесс іске қосылған кезде көршілес

түйіндер арасында қанша рет қозғалатындығын, мысалы, in және $in \cup \{i\}$ үшін, жоғары қозғалыс кезінде қанша жүретінін есептеу жеткілікті деп санаймыз. Біздің есептеуде бұл мән $\xi_{in,i}$, $in \in Inn$, $i = 1, 2, \dots, n$ кездейсоқ шама болсын. Содан кейін ξ_{ini} мәндері қосалқы функционал мәндердің орташа коэффициенттерін есептеу үшін пайдалы болады және булеанды түрлендірудің қажеті жоқ.



а



ә

а – Тіктөртбұрышта түйіндердің орналасуы; ә – Булеанының (3,4) түйіні және іргелес түйіндері.

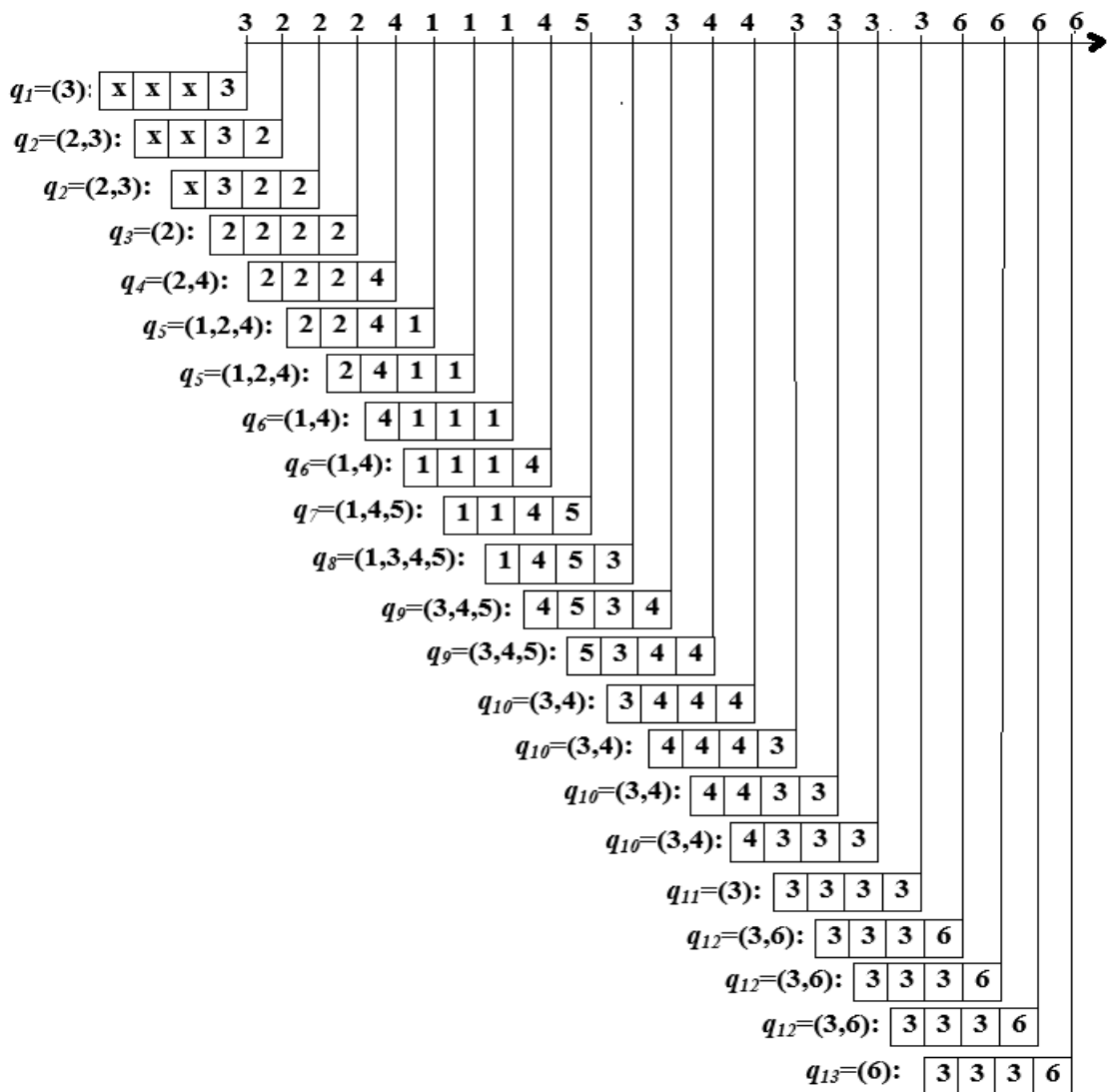
Сурет 2.8 - Булеан, екіполусты комбинаторлық кеңістік $n=6$ кезінде есептеу процесі түйіндердің кездейсоқ жүруі түрінде

Нөлдік деңгейде орналасқан төменгі полюс суретте (0) түрінде берілген, $q_0 = \emptyset$ бос жиынға сәйкес келеді және кез-келген есептеу процесі осы жерден басталады (сурет 2.7). булеан қарапайым конфигурацияға ие және $n \geq 3$ үшін n -өлшемді кеңістіктік бульдік кубқа сәйкес келеді (сурет 2.9а), онда есептеу процесіне сәйкес келетін түйіндер арқылы кездейсоқ жүруді де көрсетуге болады. Бұл үшін булеанның қарапайым құрылымын қолдану ыңғайлы. Сонымен қатар, біздің тапсырманы шешу барысында булеанға жүгіну ыңғайлы, өйткені бақылау күйінің кез-келген $in \in Inn$ булеанның in түйініне тікелей байланысты.

Жоғарғы полюсте сөздік ұғымдар санына сәйкес келеді, яғни, $n=6$. Кез-келген $in = (i_1, i_2, \dots, i_m) \in Inn$ бақылау күйі булеаның (i_1, i_2, \dots, i_m) аралық түйінге сәйкес келеді. Олар l ($1 \leq l < n$) деңгейіне және $l - 1$ деңгейінің m түйінімен және $l + 1$ деңгейінің $n - m$ түйінімен байланысқан. Мысалы, 2 деңгейдегі (3,4) түйіннің 1 деңгейдегі екі түйінмен қосылады, атап айтқанда (3) және (4), сонымен қатар 3 деңгейдегі төрт түйінмен қосылады, атап айтқанда (1, 3, 4), (2, 3, 4), (3, 4, 5), (3, 4, 6):

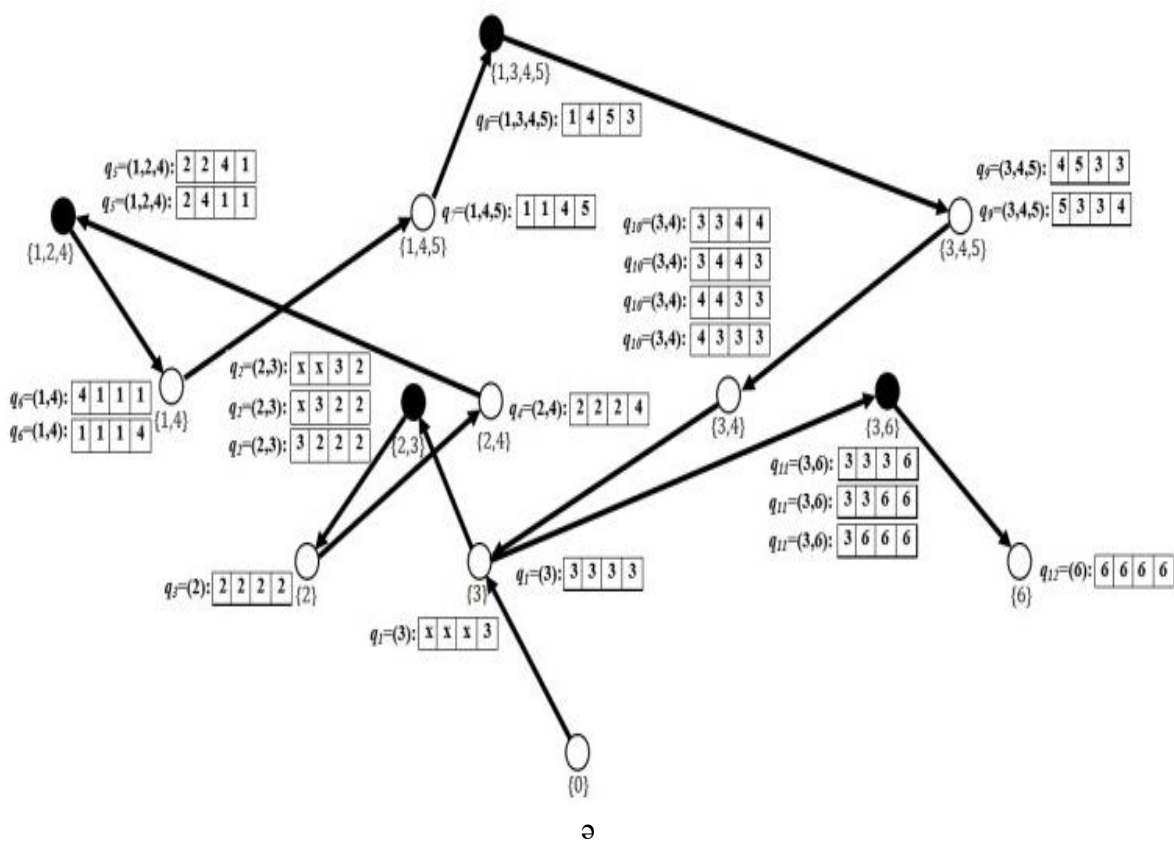
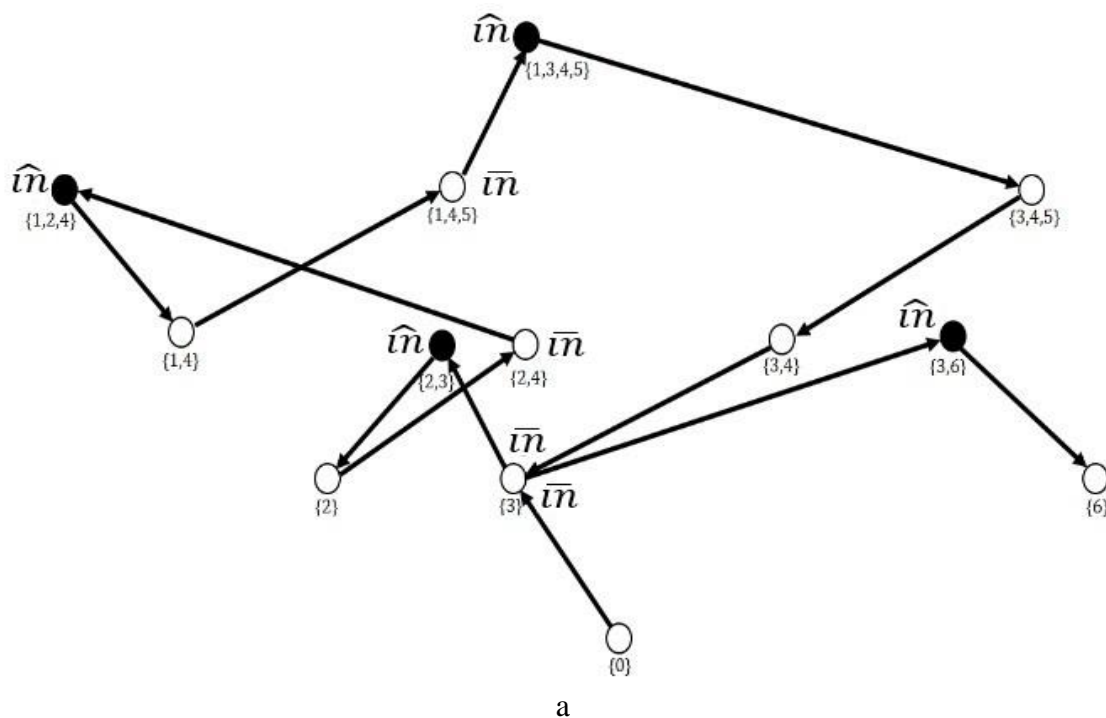
Сурет 2.9ә сәйкес көрсетілгендей 1 немесе 0-ге тең болатын β мәні (3, 4) түйіннен іргелес түйінге өту кезінде бет қателігі болатынын немесе болмайтындығын көрсетеді. Сурет 2.7 сәйкес көрсетілгендей ерекшелінген жолдағы түйіндер арасынан қара түйіндер: (2, 3), (1, 2, 4), (1, 3, 4, 5), (3, 6) және бізге қажет $i\bar{n}$ түйіндердің қабырғалары ($i\bar{n}$, $i\hat{n}$) түйіндері болып табылады, яғни: (2, 4), (1, 4, 5) (3). $l \geq 0$ деңгейіндегі түйіндер санына келетін болсақ, ол $l=0$ -ден $n=6$ -ға дейін C_n^l тең.

Ендеше, қатардың басында орналасқан ұғымдарға сілтемелерді қарастырайық (сурет 2.9).



Сурет 2.9 – Ұғымдарға сілтемелер жолы

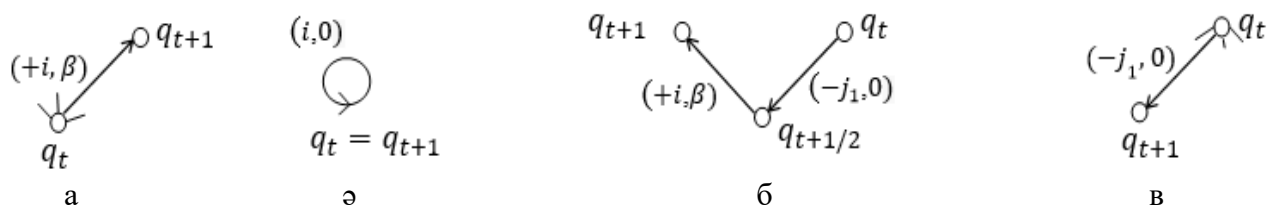
Сілтеме сызығына және жұмыс параметрінің стратегиясына сәйкес $W(k,t)$ үшін жұмысшы жиын болады (сурет 2.10).



a – \hat{in} және in түйіндері; ∂ – q_t сәтіндегі \hat{in} және in түйіндері

Сурет 2.10 – $W=(4,t)$ жұмысшы жиынға сәйкес келетін, \hat{in} және in түйіндері бар ұғымдарға сілтемелер жолы

Мұнда $k=4$, сондықтан сүзгіде $W(4,t)$ сәйкес келетін 4 ұяшық болуы керек. Соңында, бізде булеан шындарымен кездейсоқ \hat{in} және in түйіндері белгіленген (сурет 2.10а). Бірнеше эксперименттен кейін \hat{in} және \bar{in} түйіндері ауысуы мүмкін, бірақ кез-келген жағдайда кездеседі. Сурет 2.10ә сәйкес сүзгінің мазмұны көрсетілген, мұндағы x символы – сүзгінің бос ұяшығы. Сонымен қатар, геометриялық интерпретация бойынша табылған инновациялардың деңгейін, салмағын анықтауға болады. Мұнда сурет 2.11 сәйкес кез-келген қабыраның, немесе доға деп те атауға болады, салмағы 0 немесе 1-ге тең, яғни бұл сәйкесінше бет қателігі жоқ немесе бар екендігін білдіреді. Төменгі суреттер 2.11б және 2.11в көрсетілгендей ішінара немесе сурет 2.11ә сәйкес шеңбер түрінде қабырғаның салмағы “0”-ге тең. Сондықтан, функционалды есептеу кезінде, төмен қарай бағытталған қабырғаларды ескеру қажет емес, керісінше, жоғары қарай жылжу кезінде (сурет 2.11а, 2.11б), қабырғалар салмағын анықтау қажет және мұндай есептеулер үшін біз $x = (x_{ri})_{p \times n}$ матрицасын және $\delta_{in,i}(x)$ функциясын қолданамыз.



а – q_t ; ә – $q_t = q_{t+1}$; б – $q_{t+1/2}$; в – q_{t+1}

Сурет 2.11 – Қабырғалар салмағы

Көрсетілген суреттің кез-келген бөлігінде біз (q_t, q_{t+1}) және $(q_t, q_{t+1/2})$ немесе $(q_{t+1/2}, q_t)$ қабырғалар салмағын $(\pm i, \beta)$ деп белгілейміз. Мұнда $\pm i$ дегеніміз q_t бақылау күйін байытатын жоғары немесе төмен қозғалысты білдіреді (сурет 2.11а, яғни i (жоғары қарай жылжу арқылы) немесе q_t бақылау күйінің салмағын азайтады (сурет 2.11в), i_1 (төменге қарай жылжу) немесе кезекпен екеуі де (сурет 2.11б). Белгілеусіз i саны (сурет 2.11ә) $(t+1) \quad q_t = q_{t+1}$ жағдайын білдіреді. β параметрі 1 немесе 0 тең және оқиғаға байланысты, соған сәйкес q_t бақылау күйінен q_t бақылау күйіне дейін бет қателігі болады немесе болмайды.

Осылайша In жиынын анықтағаннан кейін \hat{In} анықтау алгоритмінің сипаттамасы өте қарапайым болмақ. Ол $in \subseteq \bar{in}$ фактісін анықтау үшін Inn элементтерін сұрыптаудан және ағымдағы in элементін басқа $\bar{in} \in Inn$ салыстырудан тұрады. Алдымен in элементін Inn жиынынан алып тастау керек. Егер in болса, яғни $in \subseteq \bar{in}$ қосымшасы болса, онда оны in қарауынан \hat{in} элементіне үміткер ретінде алып тастау керек. Егер жоқ болса, онда Inn жиынынан келесі \bar{in} элементін $in \subseteq \bar{in}$ құрамына кіретіндігін тексеруді жалғастыруымыз керек. Егер Inn жиынынан \bar{in} таба алмасақ және In жиыны таусылса, онда in элементі \hat{in} болады және осы \hat{in} элементін \hat{In} жиынына

қосамыз. Процесті келесі Inn элементтермен in ретінде қайталап, Inn жиыны таусылғанша \hat{In} жиынын құрамыз. Inn жиынын компьютерде қалай сақтау керектігін $In = [\Delta_{in,i}]_{|Inn| \times n}$ ретінде ұсынамыз, өйткені оны бульдік матрица ретінде ұсынуға болады. Егер бақылау күйі $in = (i_1, i_2, \dots, i_m) \in Inn$ болса, онда \hat{In} матрицасының in қатарында тек $\Delta_{in,i_j} = 1$ болады, мұндағы $j=1,2,\dots,m$ дегеніміз in қатардағы басқа элементтер нөлге тең, яғни $\Delta_{in,i} = 0, i \neq i_j$ үшін, $i \neq i_j, j = 1,2,\dots,m$. Мұндағы тағы бір маңызды ескерту, кез-келген сөздіктегі эксперимент шекті уақытқа ие.

2.3.1 Инновациялар сөздігін оңтайландыру тапсырмасының функционалын тұрғызу

Инновациялар сөздігін оңтайландыру тапсырмасының функционалын тұрғызуды $x = (x_{ri})_{p \times n}$ матрицасы үшін тиісті шектеулердің өрнектерін іздеуден бастаймыз. Сонымен қатар, бастапқы тапсырманың геометриялық аспектісін талқылау да пайдалы. Бұдан әрі Inn және \hat{In} жиындарының анықталған деп қарастырамыз.

Жоғарыда келтірілген ақпаратты пайдалана отырып, кездейсоқ $\xi_{in,i}$ шамасын енгіземіз, яғни бұл шама сөздектегі бір экспериментті орындау үшін in бақылау күйін орындаған кездегі i ұғымдарына сілтемелер саны (сурет 2.3). Яғни, мұндағы $\xi_{in,i}$ - бұл in инновациясы белсенді болғанда i концепциясына жасалған сұраныстар саны. $E_{in,i}$ - бұл in инновациясы белсенді болғанда i концепциясына жасалған сұраныстар санының математикалық күтімі. Кездейсоқ $\xi_{in,i}^j$ шама $\xi_{in,i}$ сияқты болсын, бірақ сөздіктің j -ші экспериментінде $j = 1,2,\dots,h$. Сонымен қатар $\xi_{in,i}$ және $\xi_{in,i}^j$ күтілім мәндері:

$$E(\xi_{in,i}) = E(\xi_{in,i}^j) = E_{in,i}, \quad j = 1,2,\dots,h$$

және кез келген $in \in Inn, i = 1,2,\dots,n$ үшін $E_{in,i}^{(h)} = (1/h) \cdot \sum_{j=1}^h \xi_{in,i}^{(j)}$ орташа мәніне тең болсын. Сонымен бірге, біздің сөздікпен $h \geq 1$ эксперимент өткізілді және кез-келген эксперименттен кейін j санына ие болды, $\xi_{in,i}^j$ ($in \in Inn, i = 1,2,\dots,n$) айнымалылар есептелді. Еске салайық, жиында $E_{in,i}^{(h)}$ орташа мәнін есептеу керек, ол үшін қайталау формуласы ыңғайлы

$$E_{in,i}^{(j)} := \frac{k-1}{k} \cdot E_{in,i}^{(j-1)} + \frac{1}{k} \cdot \xi_{in,i}^j, \quad in \in Inn, i = 1,2,\dots,n; \quad i \notin in; \quad j = 1,2,\dots,h$$

басында $E_{in,i}^{(0)} = 0$ барлық $in \in Inn$ үшін, $in \in Inn, i \in \{1,2,\dots,n\}; \quad i \notin in$.

Осылайша, $E_{in,i}^{(h)}$ ($in \in Inn, i \in \{1,2,\dots,n\}; \quad i \notin in$) мәндері сөздіктің h жұмысынан кейін есептеледі. Мұндағы қызықты сұрақтың бірі in инновациясы белсенді болған жағдайда және i ұғымына сілтеме жасалғанда бет қателігі

болама? Мұндай жағдай t осінің кез-келген жерінде болуы мүмкін. Бұл сұрақтың жауабын in бақылау күйіне және $x = (x_{ri})_{p \times n}$ матрицасына тәуелді функция береді. Атап айтқанда, $\delta_{in,i}(x)$ функциясы:

$$\delta_{in,i}(x) = \begin{cases} 0, & \text{егер ұғым } i \in S \in R(in, x) \\ 1, & \text{өзге жағдайда} \end{cases}$$

мұнда S дегеніміз $R(in, x)$ бетінің белгіленуі; $\delta_{in,i}(x) = 0$ мәні $in \rightarrow i$ оқиғасы кезінде бет қателігі жоқ екенін білдіреді, яғни $\delta_{in,i}(x) = 0$ мәніне ие болады егер i ұғымы $R(in, x)$ ішіндегі қандайда бір S бетінде қамтылса. Кері жағдайда, $\delta_{in,i}(x) = 1$ мәні бет қателігіне сәйкес келеді. Әрине, егер ұғымы $i \in in$ болса, онда кез-келген $x \in X$ үшін $\delta_{in,i}(x) = 0$ болуы керек. Сонымен қатар, егер $x = (x_{ri})_{p \times n}$ матрицасы бекітілген болса, онда $in \rightarrow i$ оқиғасы қай сәтте және сөздіктің қай экспериментінде болатыны маңызды емес. Қалай болғанда да, екі жауаптың жалғыз нұсқасы бар: барлық $in \rightarrow i$ оқиғалары үшін бет қателігі болады немесе болмайды, яғни берілген in және i және $x \in X$ үшін барлық $in \rightarrow i$ оқиғаларына бірдей жауап. Басқаша айтқанда, t осінде қай жерде болатынына қарамастан, барлық $in \rightarrow i$ оқиғалары үшін бірдей жауап болады. Бұл жерде кез-келген $x \in X$ матрицасы жоғарыда аталған (a)-(c) шектеулерді қанағаттандыратынын ескеру қажет және $\delta_{in,i}(x)$ есептеуін келесідей анықтауға болады:

$$\delta_{in,i}(x) = \begin{cases} 0, & \text{егер } \sum_{j=1}^{m(in)} \sum_{r=1}^p x_{ri} * x_{rj} \geq 1 \quad in \neq in_0 \\ 1, & \text{өзге жағдайда} \end{cases}$$

Осыдан бастап in_0 бақылау күйін Inn жиынан алып тастайық және сөздіктегі бір экспериментте беттер сәтсіздігінің жалпы саны кездейсоқ мәні ξ тең болсын:

$$\xi(x) = \sum_{in \in In} \sum_{i=1}^n \xi_{in,i} \cdot \delta_{in,i}(x) + \sum_{i=1}^n \xi_{in_0,i} \quad (1)$$

онда минимизациялауды қажет ететін негізгі тапсырманың функционалы үшін келесі қарастырылады:

$$F^0(x) = \sum_{in \in In} \sum_{i=1}^n E_{in,i} \cdot \delta_{in,i}(x) + \sum_{i=1}^n E_{in_0,i} \rightarrow \min_{x \in X} \quad (2)$$

Айта кету керек, ξ өрнегінде кез-келген $\xi_{in,i}$ мәні $x \in X$ матрицасына тәуелді емес, керісінше $\delta_{in,i}(x)$ функциясы берілген $in \in Inn$ мен i және $x \in X$

тәуелді және кездейсоқ $in \rightarrow i$ мен бұл орындалатын t оське байланысты емес. Қосымша тапсырманың $F^{(h)}(x)$ функционалы үшін:

$$F^{(h)}(x) = \sum_{in \in In} \sum_{i=1}^n E_{in,i}^{(h)} \cdot \delta_{in,i}(x) + \sum_{i=1}^n E_{in,i}^{(h)} \rightarrow \min_{x \in X} \quad (3)$$

(3) өрнегіндегі $E_{in,i}^{(h)}$ шамасын булеанның in және $in \cup i$ түйіндерін байланыстыратын қабырғаға (доғасына) тиістіруге болады, мұндағы функция $\delta_{in,i}(x) = 1$. Өзге жағдайда, яғни егер функция $\delta_{in,i}(x) = 0$ болса, онда бұл қабырғаны нөлге теңестіру қажет. Бұл белгіленген $x \in X$ үшін $F^{(h)}(x)$ функционалының мәнін есептеуге көмектеседі. Егер қарастырылатын қабырғаның (жоғары қараған) салмағы 1 немесе 0-ге тең болса, жоғарыда айтылғандай, көрші түйіндерден жылжу кезінде бет қателігінің болуы немесе болмауына $\delta_{in,i}(x)$ байланысты болуы жеткілікті болады. Сонда $E_{in,i}^{(h)}$ шамасы тек тиісті жағдайда ескерілуі керек және оны (3) көбейткіші ретінде қолданады.

Шектеулер (a) - (c). Негізгі тапсырма (2) үшін де, қосымша тапсырма үшін де (3) рұқсат етілген X шешімдер жиынын анықтайтын шектеулер жүйесі ((a) - (c) шарттары) келесідей жазылады:

$$\sum_{i=1}^n l_i \cdot x_{ri} \leq v_r, \quad r = 1, 2, \dots, p; \quad (a)$$

$$\sum_{r=1}^p x_{ri} = 1, \quad i = 1, 2, \dots, n; \quad (b)$$

$$\sum_{r=1}^p v_r \cdot h_{inr}(x) \leq H_{in}, \quad in \in Inn; \quad (c)$$

$$x_{ri} \in \{0, 1\}, r = 1, 2, \dots, p; i = 1, 2, \dots, n \quad (d)$$

мұнда (c) v_r мәні - $r, r = 1, 2, \dots, p$ бет ұзындығы. Сонымен қатар $h_{in,r}(x)$ функциясы: $h_{in,r}(x) = 1$, егер $S_r \in R(in, x)$ және $h_{inr}(x) = 0$ болса, кері жағдайда, яғни, $h_{in,r}(x)$ функциясы $R(in, x)$ жиынның характеристикалық функциясы болып табылса. $h_{in,r}(x)$ оңай есептеуге болады егер in және r мәндері берілген болса x матрицасының элементтері арқылы, атап айтқанда егер $in = (i_1, i_2, \dots, i_{m(in)}) \in Inn$ болса, онда:

$$h_{in,r}(x) = \max_{1 \leq j \leq m(in)} x_{ri_j} \quad (4)$$

бұл жерде $x = (x_{ri})_{p \times n}$ матрицасы тек 1 мәніне кез-келген бағанда ие, оның ішінде $i_1, i_2, \dots, i_{m(in)}$. Енді (3) қосымша есептің $F^{(h)}(x)$ функционалын есептеу барысында пайдалы болатын келесідей қадамдарға назар аударайық:

Қадам - 0. Мұнда түзетілуі керек: бастапқы матрица $x = (x_{ri})_{p \times n}$. Мұндағы x матрицасы ретінде сөздікті қайта құрылымдаудан бұрын қолданатын $x^0 = (x_{ri}^0)_{p \times n}$ матрицаны алуға болады; l_1, l_2, \dots, l_n (ұғым ұзындығы) және v_1, v_2, \dots, v_p (бет ұзындығы) мәндері және k, h берілген. Мұндағы k дегеніміз $WS(k, t)$ стратегиясының параметрі және сүзгі өлшемі, ал $h \geq 1$ мәні $F^{(h)}(x)$ функционалын құру үшін сөздіктегі эксперимент саны.

Қадам - 1. Біз $\lambda \geq 1$ бүтін сан параметрі берілген деп санаймыз. Мұнда $\lambda \geq 1$ мәні дегеніміз Inn жиынын анықтауға болатын жалпы процедураның параметрі болып табылады. Бұл дегеніміз егер λ сөздігінің қосымша экспериментінен кейін жаңа бақылау күйін табуға тырысса, онда Inn жиыны өзгермейді және содан кейін Inn жиыны анықталды дегенді білдіреді.

Қадам - 2. Біз $H > 0$ бүтін сан параметрі берілген деп санаймыз. H дегеніміз алдын-ала белгілі жүйелік константа, ол жоғарыдағы (c) шектеу кезінде H_{in} константасының орнына қолданылады. Мұнда H константасы кез келген $R(in, x)$ жұмыс жиынының өлшемін шектеуі керек, содан кейін (c) кез-келген $in \in Inn$ үшін $H_{in} = H$ қоямыз. $H = \max_{in \in Inn} H_{in}$ қою мүмкіндігі кез-келген $H_{in}, in \in Inn$ константа үшін ұсынылады.

Қадам - 3. $E_{in,i}^{(h)}$ коэффициенттері есептелінеді. h шамасы - бұл (3) ішіндегі $E_{in,i}^{(h)}$ коэффициенттерін есептеу үшін эксперимент саны. Мұнда бізге ыңғайлы формуланы қолдана аламыз:

$$E_{in,i}^{(j)} := \frac{k-1}{k} \cdot E_{in,i}^{(j-1)} + \frac{1}{k} \cdot \xi_{in,i}^j, \quad in \in Inn, i = 1, 2, \dots, n; \quad i \notin in; \quad j = 1, 2, \dots, h$$

мұнда j шамасы – сөздіктегі эксперименттің j -ші саны. Барлық $in \in Inn, i \in \{1, 2, \dots, n\}; i \notin in$ үшін бастапқы мән $E_{in,i}^{(0)} = 0$. Мұнда кез келген эксперимент үшін $\xi_{in,i}^j, in \in Inn, i = 1, 2, \dots, n; i \notin in; j = 1, 2, \dots, h$ мәні қол жетімде. Кез-келген бетті іске қосу кезінде, ұғымдарға сілтеме қол жетімді болды және $\xi_{in,i}^j$ мәндерін есептеу қиын емес, нақты айтқанда, сүзгі t осінің бойымен солдан оңға жылжу барысында, $in \in Inn, i = 1, 2, \dots, n; i \notin in; j = 1, 2, \dots, h$. Мұнда кез-келген $\xi_{in,i}^j$ арнайы есептеуіш болсын және оны η_{qi}^{jt} деп көрсетеміз, мұндағы $t = 1, 2, \dots, \gamma$ сәті үшін t осінің бойымен қозғалыс.

$$\eta_{in,i}^{jt} = \begin{cases} \eta_{in,i}^{j(t-1)} + 1, & \text{егер ағымдағы } in \text{ және ұғымдарға келесі сілтем } i \text{ болса} \\ \eta_{in,i}^{j(t-1)}, & \text{өзге жағдайды} \end{cases}$$

және ағымдағы эксперимент аяқталғаннан кейін $\xi_{in,i}^j = \eta_{in,i}^{j\gamma}$ қою керек, мұнда γ дегеніміз j -ші. Кез келген $in \in Inn, i = 1, 2, \dots, n; i \notin in$ үшін бастапқы мән

$\xi_{in,i}^j = 0, \eta_{in,i}^{j0} = 0$. Жалпы, мұнда $\eta_{in,i}^{j0}, \eta_{in,i}^{j1}, \dots, \eta_{in,i}^{j\gamma}$ айнымалы массивтерді енгізу қажет емес. Программист стилінде тек бір айнымалыны қолдану жеткілікті, мысалы, $\eta_{in,i}^j$ және жоғарыдағы формуланы қайта жазамыз: $\eta_{in,i}^j := \eta_{in,i}^j + 1$, егер ағымдағы бақылау күйі in болса және ұғымға келесі сілтеме i болса, өзге жағдайда $\eta_{in,i}^j := \eta_{in,i}^j$. Мұнда $D = \{\theta\}$ бастапқы мәліметтерінің кездейсоқ тізбегіне және кез келген эксперименттен кейін ұғымдарға сәйкес сілтемелер қол жетімді деп саналады.

Қадам - 4. (a)-(d) шектеулері арқылы $F^{(h)}(x)$ функционалды (3) мәселесін шешу. Мұнда классикалық алгоритмдерге енгізілген идеяларды, яғни анық емес санау алгоритмін; тармақталған және байланыстырылған алгоритмді; булеандік бағдарламалау әдісін; реттік есептеулер әдісін [130-132]; жуықтау комбинаторлық әдісін [133, 134]; генетикалық алгоритмді [135, 136] және жоғарыда аталған әдістермен жергілікті оңтайландырудың үйлескен әдісін қолданып көруге болады.

2.3.2 Моделді оңтайландыру

Мұнда $R(in, x)$ жұмысшы жиынның өлшемін бақылау үшін теңсіздіктер қатарын қысқарту қарастырылады. Айта кету керек, (c) шектеу құрамында $|Inn|$ теңсіздігі бар және олар көп. Мұнда (c) теңсіздік қатарын айтарлықтай қысқартуға мүмкіндік бар. Жоғарыда айтылғандай, практикалық тұрғыдан біз жүйелік константаның болуын ұсына аламыз, айталық N , кезкелген $R(in, x)$ жұмысшы жиын шектеулі және ол (константа) алдын ала белгілі деп саналады. Сондай-ақ, \hat{In} жиынын ескере отырып біз (c) жүйесін келесіге ауыстыра аламыз:

$$\sum_{r=1}^p v_r \cdot h_{\hat{in},r}(x) \leq H, \quad \hat{in} \in \hat{Inn}; \quad (5)$$

бірақ формальды түрде, бұған дейін барлық $h_{in} = H, in \in Inn$ үшін (c) қою керек. Айталық $|R(in, x)|$ дегеніміз $R(in, x)$ жұмысшы жиынның ұзындығы болсын, мұндағы $in \in Inn$ және $x \in X$. Алмастырудың себебін анықтау үшін, 2.7, 2.10a суреттеріне назар аударайық, мұнда $\hat{in} \in \hat{Inn}$ түйіндеріне сәйкес келетін қара түйіндер және $in \subsetneq \hat{in}$ ұқсас $in \in In$ түйіндері суреттелген. Сонда келесі қатынастар орындалады:

$$\text{егер } in \subseteq \hat{in} \text{ болса, онда } R(in, x) \subseteq R(\hat{in}, x) \text{ және } |R(in, x)| \leq |R(\hat{in}, x)|$$

мұнда $in \in Inn, \hat{in} \in \hat{In}$ және егер теңсіздік (5) кейбір $\hat{in} \in \hat{In}$ үшін орындалса, онда ол $in \subseteq \hat{in}$ болатын кез келген $in \in Inn$ үшін де орындалады. Мұнда кез келген $in \in Inn$ шамасы \hat{In} жиынның кемінде бір \hat{in} жиынына тиесілі болады (сурет. 2.10 a), түйін (3)).

Әрине мүмкін болатын X шешімдер жиыны бос емес екені анық, өйткені $S_{g_1}, S_{g_2}, \dots, S_{g_p}$ беттеріндегі i_1, i_2, \dots, i_n бастапқы тарату ұғымдары (a)-(d) шектеулерін қанағаттандырады [137].

2.3.3 Функционалды бағалау

Мұнда (2), (3) функционалдағы қосындыларды қысқарту қарастырылған. Жоғарыда келтірілген сөздікті қайта құрудың сызықтық емес моделі сызықтық емес (2) және/немесе (3) функционалды, сонымен қатар (a)-(b) және (c) көрсетілген сызықты емес шектеулерді қамтиды. (5) көрсетілген \hat{Inn} жиынның қуаты (c) көрсетілген Inn жиынның қуатынан көп айырмашылығы жоқ. (2) немесе (3) функционалына келетін болсақ, егер ұғым $i \in in$ болса, онда кез келген $x \in X$ үшін $\delta_{in,i}(x) \equiv 0$ және (2) немесе (3)-дегі екінші қосындыда $i = 1, 2, \dots, n$ орнына тек $i \in I(in)$ индекстері бар, мұндағы $I(in)$ жиыны - құрамында in -ға тиесілі i ескере отырып (2) немесе (3) қосылымдардың санын азайту идея негізінде орындауға болады. Геометриялық тұрғыдан, 2.7 суретке сүйене отырып, $i \in in$ оқиғасы 2.11 суреттегі ϑ , ν жағдайларына сәйкес келеді, яғни төменге қарай қозғалу немесе түйіннің айналасында айналмалы қозғалыс. Сонымен қатар, $in = (i_1, i_2, \dots, i_{m(in)})$ және i берілген болса және оқиға кезінде $in \rightarrow i$, егер 2.11 суреттегі ν жағдайға сәйкес келетін $i_1 \notin (i_2, i_3, \dots, i_{m(in)})$ және $i \in (i_2, i_3, \dots, i_{m(in)})$ оқиғалары болса, онда бет қателігі болмайды. Егер 2.11 суреттегі ϑ жағдайға сәйкес $i_1 \in (i_2, i_3, \dots, i_{m(in)})$ және $i \in (i_2, i_3, \dots, i_{m(in)})$ оқиғалары болса, онда да беттің қателігі болмайды.

Функционалды бағалауды құруға негізделген дискретті оңтайландырудың кейбір әдістері [130, с. 187; 131, р. 118; 132, р. 1099; 133, с. 255], біздің жағдайда (3) мәселе үшін, бұл төменгі функционалды бағалау болып табылатынды:

$$\sum_{\hat{in} \in \hat{Inn}} \sum_{\bar{in} \in Inn_{\hat{in}}} E_{\bar{in}i_{\bar{in} \setminus \bar{in}}}^{(h)} \cdot \delta_{\bar{in}i_{\bar{in} \setminus \bar{in}}}(x) \leq \sum_{in \in Inn} \sum_{i=1}^n E_{in,i}^{(h)} \cdot \delta_{in,i}(x), \quad (6)$$

және [130, с. 203; 131, р. 119; 132, р. 1098; 133, с. 258] жұмыстарынан қарай отырып бағалау функционалын шешу қиын емес, біз үшін бұл:

$$\sum_{\hat{in} \in \hat{Inn}} \sum_{\bar{in} \in Inn_{\hat{in}}} E_{\bar{in}i_{\bar{in} \setminus \bar{in}}}^{(h)} \cdot \delta_{\bar{in}i_{\bar{in} \setminus \bar{in}}}(x) \rightarrow \min_{x \in X} \quad (7)$$

содан кейін ол тиісті күрделілікпен (3) мәселенің нақты (оңтайлы) шешімін табуға мүмкіндік береді. (6) теңдеудің сол жағындағы $Inn_{\hat{in}}$ жиыны Inn ішкі жиыны және ол \bar{in} санынан тұрады, мұндағы кез келген \bar{in} дегеніміз $Inn_{\hat{in}}$ жиынынан тұрады. Егер булеанға қарайтын болсақ (сурет 2.7), \bar{in} түйіні \hat{in} түйіндерімен қабырға арқылы байланысқан, яғни (\bar{in}, \hat{in}) доғасы \bar{in} және \hat{in} түйіндерін байланыстырушы қабырғасы. Бұл жағдайда (6) сол жақтағындағы және (7) қабырғаларды (\hat{in}, \bar{in}) есепке алудың қажеті жоқ, себебі (\hat{in}, \bar{in}) қабырғаларының салмағы 0-ге тең. Мұнда (6) сол жағында, кез келген $\hat{in} \in \hat{Inn}$ үшін \bar{in} түйіні $Inn_{\hat{in}}$ жиыны таусылғанша орындалады. Егер h сөздікті іске қосу кезінде \bar{in} түйінінен \hat{in} түйініне дейін тым болмаса бір сілтеме болса, біз $Inn_{\hat{in}}$ жиынына \bar{in} түйінін қосамыз. (6) сол жақтағы i саны $\hat{in} \setminus \bar{in}$ деп анықталады, яғни

$i = \hat{m} \setminus \bar{m}$, бұл $i_{\hat{m} \setminus \bar{m}}$ белгісі болсын. (6) өрнегінің сол жағында (6) оң жағына карағанда аз қосындыдан тұрады. Тура осы жағдайды (2) мәселенің функционалы туралы да айтуға болады, яғни, $F^0(x)$. (2) оңтайлы шешіміне келер болсақ, мәселенің оңтайлы шешімі (3), x_h^* матрицасы болатын (3) мәселенің оңтайлы шешімі (2) мәселенің ε оңтайлы шешімі болатын бастапқы мәліметтер үшін жағдайларды көрсету мәні болады

$$Pr\{|F^0(x^*) - F^0(x_h^*)| \leq \varepsilon\} \geq 1 - \eta, \quad \varepsilon > 0, \quad \eta \in (0,1) \quad (8)$$

мұнда x^* матрицасы (2) мәселенің белгісіз оңтайлы шешімі.

Бұл шарттар, ең алдымен $\xi_{in,i}, i = 1, 2, \dots, n; in \in Inn$ және сөздіктегі эксперимент h санының төменгі шектері үшін (8) теңсіздікті сақтай отырып үлестіру заңдарының ортақ қасиеттерін анықтайды. (2) теңдеуіндегі $E_{in,i}$ белгілі мәндері үшін, яғни әр кездейсоқ шаманың үлестірім заңы $\xi_{in,i}, i = 1, 2, \dots, n; in \in Inn$ болса, бастапқы (2) есептің де белгілі, (3) есептің де шешу алгоритмі бағалау функциясының (6) және $\delta_{in,i}(x)$ функциясының қасиетіне негізделуі мүмкін. Егер $E_{in,i}$ мәні белгісіз болған жағдайда (егер кездейсоқ шаманың $\xi_{in,i}, i = 1, 2, \dots, n; in \in Inn$ үлестірім заңы белгісіз болса) онда (2) есепті шешу күрделі болады. Бұл жағдайда (3) есебі (2) есепті шешудің көмекшісі ретінде қолдануға болады, ал (3) есептің оңтайлы шешімі, яғни бұл x_h^* матрицасы болсын, (2) есепті шешудің теңсіздігі ретінде болуы мүмкін.

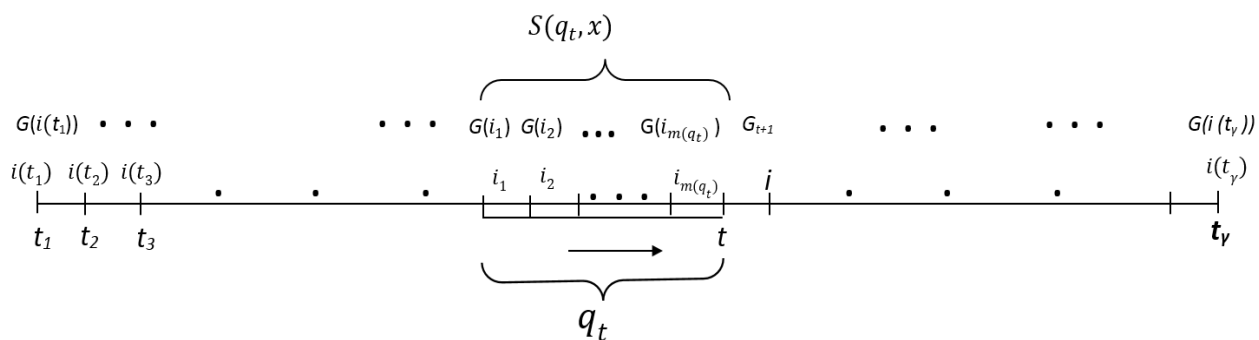
3 ЭПИДЕМИЯ ҚАУІПІН ТӨМЕНДЕТУ МАҚСАТЫНДА МУЛЬТИАГЕНТТІ ЖҮЙЕ ТОПТАРЫН ҚАЙТА ҚҰРЫЛЫМДАУ МОДЕЛІ

Бұл бөлімде мультиагентті жүйе қарастырылған. Жүйенің агенттері топтарға бөлінген және бір-бірімен байланыс барысында инфекция жұқтыру қаупі бар. Эпидемияның өрбу ықтималдығын азайту мақсатында агенттерді топтарға қайта бөлу арқылы мультиагентті жүйенің жұмысын оңтайландыру тапсырмасы тұжырымдалған. Комбинаторлық мәселе ретінде бұл тапсырма *NP* толық есептер класына жатады, ол осы зерттеудің іргелі аспектісін көрсетеді. Бұл бөлімде қолданылатын стационарлық жүйе тұжырымдамасы алдыңғы, яғни екінші бөлімде қолданылған жұмысшы жиын тұжырымына қайта оралады, ол есептеу процестері теориясындағы маңызды ұғым болып табылады және басқа процестерді зерттеуге негіз бола алады. Сонымен қатар, бұл бөлімде әлеуметтік жүйеде инновацияның таралу үлгісі қарастырылған.

3.1 Агенттердің байланыс кезінде эпидемияны жұқтыруы мүмкіндігі бар мультиагентті жүйенің топтарын қайта құру моделін тұрғызу

Басқару жүйелерінің кең тараған класстарының бірі биокибернетикалық жүйелер болып табылады, олар техникалық жүйелер сияқты, теңгерімсіздік жағдайында жүйе тұрасыздық танытады. Бұл тұрақсыздық жүйенің қалыпты жұмыс режимін бұзуы мүмкін және біздің жағдайымыз агенттердің топтарға сәтсіз бөлінуіне байланысты болуы мүмкін. Осыған байланысты, осы бөлімде ғылыми қызығушылық тудыратын және қолданбалы мәнге ие мультиагентті жүйе қарастырылған.

Сонымен, n агенттерден тұратын, p топтарға бөлінген және әр бір агент тек бір топқа жататын (a шарты) мультиагентті жүйе болсын. Жүйенің әрбір агентіне "салмақ" тағайындалады, оның өмір сүру кеңістігінің сызықтық өлшемі, соның шеңберінде агент өзіне жүктелген операциялар жиынын орындай алады. Сонымен бірге, бір-бірімен әрекеттесетін агенттер арасында инфекция жұқтыру немесе тарату қаупіне ұшырайды. Әр топқа «салмақ» тағайындалады – ол топтағы агенттердің тіршілік кеңістігі. Кез-келген топтағы агенттердің жалпы «салмағы» топтың жалпы салмағынан аспауы керек (b шарты)). Айталық, l_i айнымалысы i агентінің салмағы болсын, $i = 1, 2, \dots, n$, және v_r айнымалысы r нөмерлі топтың салмағы, $r = 1, 2, \dots, p$. Жұқтырылған топтардың нөмерлері орналасқан «стационар» бар, яғни, біз бұл топтардың өздерін де, осы топтардың жұқтырған агенттерін де қарастырамыз, іс жүзінде олар агенттер арасында байланыс орнатады. Стационардағы агенттердің басқа агенттерге байланысы уақыт бойынша тізбекті түрде эксперимент кезінде жүзеге асырылады, әр бір эксперименттің мақсаты эксперимент кезінде пайда болған байланыстар траекториясын алу және өңдеу болып табылады, олар t уақыт шкаласында бекітілген, яғни t_0 сәттен бастап траекторияның белгілі бір максималды байланыс сәтіне дейін. Бір эксперименттегі агенттерге байланыс траекториясы сурет 3.1 сәйкес көрсетілді.



Сурет 3.1 – Бір эксперименттегі агенттерге байланыс траекториясы

Мұнда,

$S(q_t, x)$ - агенттер тобының жұмысшы жиыны

$q_t - t$ сәтіндегі траекторияның бақылау күйі

$i(t_1) i(t_2) i(t_3) \dots i_1 i_2 \dots i_{m(q_t)} i \dots i(t_\gamma)$ – агенттерге сілтемелер жолы

$i_1 i_2 \dots i_{m(q_t)}$ - t сәтіндегі агенттер траекторияның бақылау күйі

$G(i(t_1)) \dots G(i_1) G(i_2) \dots G(i_{m(q_t)}) G_{t+1} \dots \dots G(t_\gamma)$ - топтарға тілтемелер жолы

$G(t_1)$ – сілтеме жасалған топтың нөмірі

$G(i_j)$ - q_t бақылау күйінде i_j агенті бар топтың нөмірі

Біз мультиагентті жүйенің жұмыс істеуі үнемі эксперименттер жүргізуден тұрады деп санаймыз, олардың әрқайсысы $\theta \in D$ кездейсоқ мәліметтермен анықталады (мұндағы D – бұл потенциалды шексіз жиын) және бастапқы кезеңде, қайта құру алдында, байланыс құратын агенттердің өзара әрекеттесуінің траекториясын зерттеу қажет. Оның үстіне кез-келген экспериментке жұмсалатын уақыт шектеулі. Бастапқыда мультиагентті жүйенің агенттерді топтарға бөлінген болсын және агенттер арасында эпидемия тудыруы мүмкін фонда жұмыс істейтін етіп берілсін. Мұндағы мақсат – эпидемияның туындау ықтималдығын азайту, біздің жағдайымызда, агенттерді топтарға бөлу арқылы, жұқтырған агенттердің санын азайту. Агенттердің топтарға бөлінуін, сондай-ақ әр агенттің топтық нөмірін $x = (x_{ri})_{p \times n}$ бульдік матрицасы арқылы оңай анықтауға болады, мұндағы, егер i нөмерлі агент r нөмерлі топта орналасқан болса және $x_{ri} = 0$ кері жағдайда. Жоғарыда айтылғандай x матрицасы a) және b) шарттарын қанағаттандыруы керек:

$$\sum_{r=1}^p x_{ri} = 1, \quad i = 1, 2, \dots, n; \quad (a)$$

$$\sum_{i=1}^n l_i \cdot x_{ri} \leq v_r, \quad r = 1, 2, \dots, p; \quad (b)$$

және тағы бір қосымша шарт c) болсын. a) - c) шарттарын қанағаттандыратын барлық осындай матрицаларды X арқылы белгілейміз. Берілген $x \in X$ матрицасымен нөмерді $r_i(x)$ белгілеу арқылы және (a) , (b) шарттарын ескере отырып i агенті бар топты анықтау үшін қолданамыз:

$$r_i(x) = \sum_{r=1}^p x_{ri} \cdot r, \quad i = 1, 2, \dots, n$$

Агенттерді $x^0 = (x_{ri})_{p \times n}$ матрицалы топтарға алдын-ала бөлінген деп есептейміз, бұл кезде мультиагентті жүйе қалыпты жағдайда жұмыс істейді және қайта құруды қажет етеді. Осындай тапсырмалардың, оның ішінде менім экстремалды тапсырмам үшін де соңғы мақсаты $x^* = (x_{ri})_{p \times n}$ матрицасын анықтау, онда берілген функционалды минимумға қол жеткізіледі. Бұл бөлімде бір эксперименттегі жұқтырылған агенттер санының функционалды-математикалық күтімін және $h \geq 1$ эксперименті арқылы есептелген эмпирикалық функционалды анықтау, яғни h экспериментінде жұқтырған агенттердің орташа саны. Мұндай зерттеудің маңызды қадамы - модельді құру және оған сәйкес есепті шығару, біздің жағдайда, экстремалды есеп және оның ерекшеліктерін анықтау. Бұл қадам есепті шешудің сәйкес алгоритмін синтездеуде пайдалы болуы мүмкін.

$\delta_{qi}(x)$ функциясын q мәні, i нөмірі және $x = (x_{ri})_{p \times n}$ матрицасы бойынша келесідей анықтауға болады:

$$\delta_{qi}(x) = \begin{cases} 0, \text{ егер } \sum_{j=1}^{m(q)} \sum_{r=1}^p x_{ri} * x_{rj} \geq 1, & \text{мұндағы } q \neq q_0 \\ \sum_{j=1}^n x_{r_i(x),j}, & \text{өзге жағдайда} \end{cases}$$

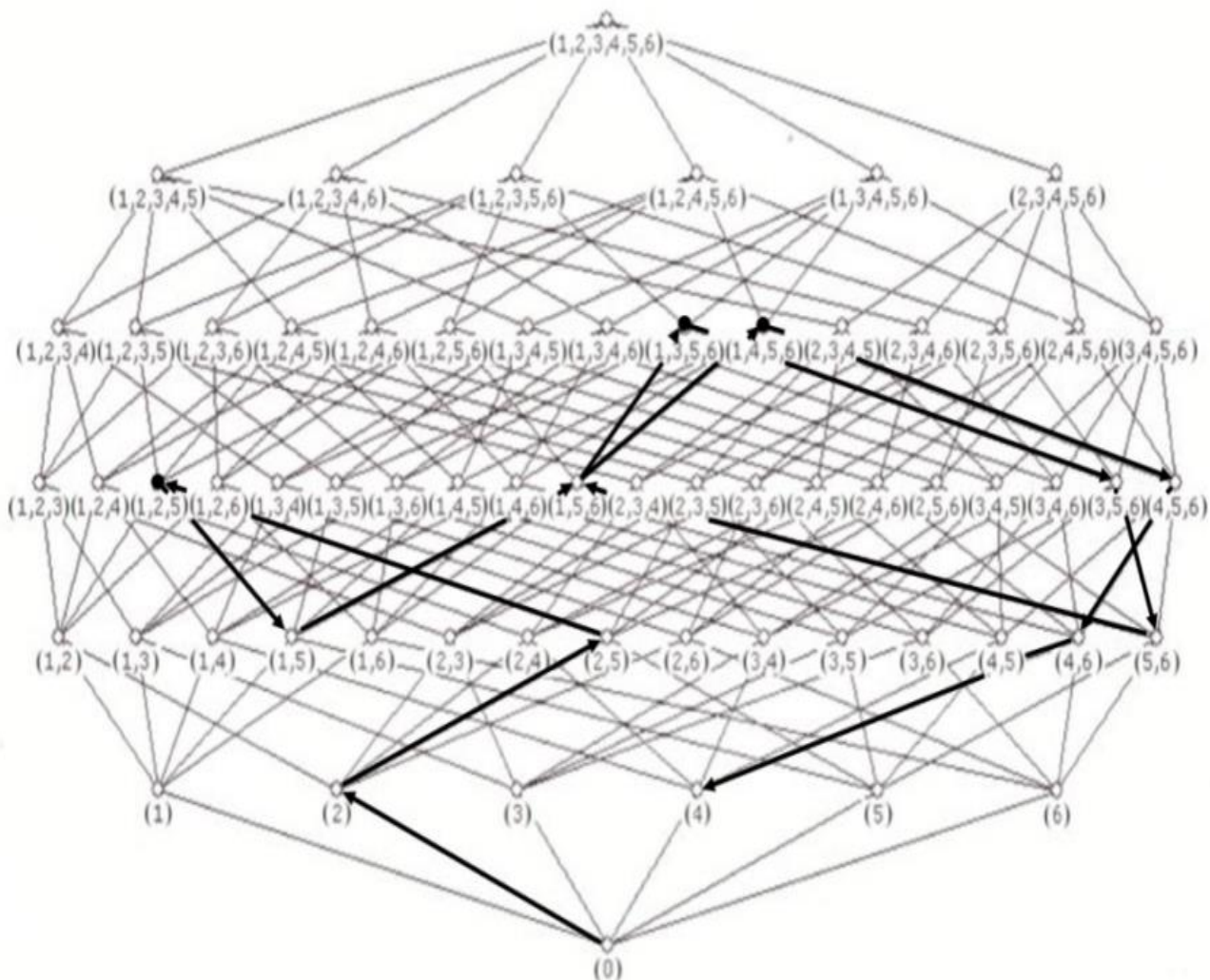
Белгілі бір жағдайларда, атап айтқанда, стационардағы агенттің стационарда жоқ агентпен кезекті байланысы кезінде соңғысына эпидемия жұғады және ол осы агент орналасқан топтағы барлық агенттерге жұғады. Осыдан кейін барлық топ стационарға орналастырылады. Сонымен қатар, стационардағы агенттер арасындағы байланыс жұқтыруға әкелмейді. Мұндай агенттердің өздері, t сәтіндегі байланыстар траекториясының басқару күйін қалыптастырады, бұны q_t арқылы белгілейміз. Жағдайды нақтылау үшін q_t мультижиын екенін ескереміз, оны $q_t = \{i_1, i_2, \dots, i_{m(q_t)}\}$ деп жазамыз, мұндағы $i_j, j = 1, 2, \dots, m(q_t)$, t сәтіне дейінгі соңғы k байланыс жасалған стационардағы агенттер саны. Стационарды жылжыту арқылы, біздің жағдайымызда k ұшықтары бар рамканы t осі бойымен оң жаққа, мұнда оське агенттер нөмірі жазылған (Сурет 3.1), біз стационар ішінде жұқтырған топтардың нөмірімен бірге траекторияның бақылау күйлерін алатын боламыз. Сурет 3.1 көрсетілгендей стационардың әр бір ұшығына топтың нөмірі жазылады $G(i_j)$,

олар агент нөмерінен тұрады $i_j, j = 1, 2, \dots, m(q_t)$, мұндағы $\{i_1, i_2, \dots, i_{m(q_t)}\}$ жиыны t сәтіндегі траекторияның бақылау күйін құрайды. Сурет 3.1 көрсетілгендей t осі бойында топтар нөмері және олармен байланыстағы агенттер жазылған. Стационардың құрамы эксперименттің кез-келген сәтінде, мысалы t , жоғарғы бөлігінде инфекцияланған агенттер топтарының санынан: $G(i_1), G(i_2), \dots, G(i_{m(q_t)})$, ал төменгі бөлігінде агенттердің өздерінен тұрады. Сонымен қатар t сәтінде стационардағы барлық топтар жиыны $S(q_t, x)$ арқылы белгіленеді және осы мультижиын q_t бақылау күйі және x матрицасынан туындайды. Ал бақылау күйіндегі қайталанбайтын агенттердің топтық нөмері жиынын сәйкесінше $S(q, x)$ деп белгілейміз. Мұнда бірнеше эксперимент жүргізу мүмкіндігі мен тапсырманың ерекшеліктерін ескере отырып, q_t жазбасындағы t индексін алып тастаймыз және сәйкесінше агенттердің нөмірі қайталанбайтын жиынды q деп белгілей аламыз және оны $q = (i_1, i_2, \dots, i_{m(q)})$ ретінде жазамыз. Мұндағы q жиындағы элементтердің реті маңызды емес деп есептейміз. Бақылау күйінің қайталанбайтын барлық жиынын $Q = \{q\}$ арқылы белгілейміз және Q жиыны ақырғы санды эксперименттер нәтижесінде және байланыс траекторияларын өңдеу нәтижесінде табуға болатындығын ескертеміз. Сонымен қатар, кез-келген q бақылау күйінің элементтері өсу бойынша реттелген, содан кейін әрбір $q \in Q$ үшін комбинаторлық кеңістіктің q шыңымен байланыстырылуы мүмкін.

Комбинаторлық кеңістік шыңдарының арасындағы Q жиынның бақылау күйі болатындарды белгілейміз. Бұл кез-келген бақылау күйлері басқа ешқандай бақылау күйлерінің құрамына кірмейді және оларды \hat{q} деп белгілейміз. Басқаша айтқанда, Q жиынының элементтері арасындағы кез-келген \hat{q} үшін q бақылау күйі болмайды, $q \neq \hat{q}$ және $\hat{q} \subset q$. Сурет 3.2 көрсетілгендей бір траекториядағы 14 бақылау күйі арасында тек 3 мұндай бақылау күйі кездеседі, яғни \hat{q} : (1, 2, 5), (1, 3, 5, 6), (1, 4, 5, 6). Мұндай траекторияның шыңдары әрбір эксперимент кезінде өзгеріп отыру мүмкін, бірақ әрдайым кездеседі. Мұндай \hat{q} жиындарды \hat{Q} деп белгілейміз. Сонымен қатар, бақылау күйлерін \bar{q} деп белгілейміз, олар комбинаторлық кеңістіктегі $\hat{q} \in \hat{Q}$ бақылау күйлерінің аталары болып табылады. Бұл \bar{q} бақылау күйлері \hat{q} бақылау күйлерімен (\bar{q}, \hat{q}) доғасын құрайды. Бекітілген \hat{q} үшін мұндай \bar{q} жиынын $\bar{Q}_{\hat{q}}$ арқылы белгілейміз. Комбинаторлық кеңістіктің төменгі полюсіне сәйкес келетін $q_0 = \{\emptyset\}$ формальді бақылау күйін енгіземіз және оны комбинаторлық кеңістіктің (0) шыңы деп кодтаймыз және осы жерден бірінші эксперимент арналған процесс басталады, сондықтан $t_0=0$. Егер келесі эксперименттің процесі комбинаторлық кеңістіктің $q = (j_1, j_2, \dots, j_{m(q)}) \neq q_0$ шыңында аяқталса, онда біраз уақыт ішінде стационардағы агенттердің ауруы созылмалы болады және келесі траектория $q_0 = \{\emptyset\}$ емес q шыңынан басталады деп болжауға болады. Келесі траектория үшін t_0 сәтіндегі $S(q, x)$ стационарының құрамы келесідей болады:

$$S(q, x) = \{G(j_1), G(j_2), \dots, G(j_{m(q)})\}.$$

Келесі 3.2-суретте булеан шындырымен кездейсоқ кезу мысалы көрсетілген:
 $(0) \rightarrow (2) \rightarrow (2,5) \rightarrow (1,2,5) \rightarrow (1,5) \rightarrow (1,5,6) \rightarrow (1,3,5,6) \rightarrow (3,5,6) \rightarrow (5,6) \rightarrow (1,5,6) \rightarrow (1,4,5,6) \rightarrow (4,5,6) \rightarrow (4,6) \rightarrow (6)$.



Сурет 3.2 – $n=6$ агенттер саны үшін комбинаторлық кеңістік

Егер жаңа траекторияға алғашқы байланыс кезінде агент жұқтырса, онда жұқтыру стационардағы $j_1, j_2, \dots, j_{m(q)}$ агенттердің кейбіреуінен болады. Басқа жағдайларда, $q = q_0$ болуы мүмкін, егер стационардағы барлық агенттер жазылып және келесі траектория кез-келген n агенттің біреуінің кездейсоқ жұқтырғанынан басталса, ал сәйкес бақылау күйі комбинаторлық кеңістіктің 0-ші деңгейінде орналасса. Ресурстардың шектеулігіне байланысты, $S(q, x)$ жиынның күші p шамасынан кіші және кез-келген $S(q, x)$ жиынның жалпы салмағын шектейтін (c шарты) константа белгілі деп санаймыз. Мұндай константаны $L_q > 0$ деп белгілейміз. c шарты келесідей жазылады:

$$\sum_{r=1}^p v_r \cdot h_{qr}(x) \leq L_q, \quad q \in Q; \quad (c)$$

(с) формуласындағы $h_{qr}(x)$ функциясы r нөмірлі топ $S(q, x)$ жиынына кірме әлде кірмейме соны анықтайды және $q = (i_1, i_2, \dots, i_{m(q)}) \in Q$ берілгені бойынша i нөмірі және $x = (x_{ri})_{p \times n}$ матрицасы арқылы келесідей есептеледі: $h_{qr}(x) = \max_{1 \leq j \leq m(q)} x_{ri_j}$, $q \neq q_0$. (а)-(с) шектеулері $p+n+|Q|$ қарапайым емес шектеулерден тұрады, мұндағы $|Q|$ сызықты емес. (с) шектеуіндегі L_q константа $S(q, x)$ ішіне кіретін жиынның жалпы салмағын шектейді. Практикалық және геометрикалық түсініктемелерге (сурет 3.2) негіздене отырып, (с) шектеуіндегі теңсіздіктер санын азайтуға болады. Егер константа L алдын ала белгілі болса, онда (с) шектеуінің оң жағындағы L_q ескере отырып, келесідей шектеу жазуға болады:

$$\sum_{r=1}^p v_r \cdot h_{\hat{q}r}(x) \leq L, \quad \hat{q} \in \hat{Q}; \quad (d)$$

Барлық $q \in Q$ үшін формальды $L_q = L$ қойып және барлық $q \in Q$ арасынан тек \hat{q} болатын q аламыз. (d) жүйесі бастапқы (с) шектеулер жиынына қарағанда азырақ теңсіздіктерге ие. Айталық $|R(q, x)|$ дегеніміз $R(q, x)$ жұмысшы жиынның ұзындығы болсын, мұндағы $q \in Q$ және $x \in X$. Егер (d) теңсіздігі кейбір $\hat{q} \in \hat{Q}$ үшін рас болса, онда ол $q \subseteq \hat{q}$ болатын барлық $q \in Q$ рас болады:

$$\text{яғни егер } q \subseteq \hat{q} \text{ онда } R(q, x) \subseteq R(\hat{q}, x) \text{ және } |R(q, x)| \leq |R(\hat{q}, x)|$$

мұнда кез-келген $q \in Q$ шыңы \hat{Q} ішіне кіретін кем дегенде бір \hat{q} шыңымен жабылатындығын ескеру қажет [138, 139].

3.2 Әлеуметтік қоғамда инновацияның таралу ақпараттық жүйесін сипаттау

Қоғамдық пікір – бұл қоғамда болып жатқан жайыттарға, болмыстарға және мәселелерге қоғамның немесе оның бір бөлгінің жасырын немесе анық қатынасын қамтитын жалпы ахуалдың жағдайы. Қоғамдық пікір бірқатар функцияларды атқарады. Олар: ақпараттандыру, кеңес беру, бақылау және қоғамның қызығушылығын тудыратын өзекті мәселелер бойынша шешім қабылдауға талап ету. Қоғамдық пікір тек ғана түрлі әлеуметтік факторларға тәуелді емес, сонымен қатар белгілі бір жағдайларда басқа әлеуметтік құбылыстарға әсер етуі мүмкін. Қоғамдық пікірді әлеуметтік жүйелердің түрлі элементтеріне әсер етуін ескере отырып қарастыру керек, нақты айтқанда экономикаға, саясатқа, мәдениет саласына, экологияға және т.б.

Қазірге кезде кез келген мемлекеттің қор нарығы және қаржы жүйесі бағасының өсу немесе құлдырау оқиғаларына жиі әсер етеді [140-142], (Қосымша А). Бір мемлекеттің өзге мемлекетке, компанияларларға және ресми тұлғаларға қолданылған санкциялары жарияланғаннан кейін БАҚ-да «нарықтың құлдырауы» туралы хабар жиі таратылады, бұл өз кезегінде қор нарығының және

ұлттық валютаның әлсіреуіне ықпал етеді [143]. Осы жайыттарды ескере отырып, көптеген ғалымдар ақпарат немесе инновация диффузиясын терең зерттеді және сол зерттеулер медиаэсерді түсінуге үлкен септігін тигізді. Мысалы, П.Дойчманн және У.Даниельсон жаңалықтар диффузиясын зерттеу барысында олардың басқа инновация диффузияларына қарағанда біршама жылдам таралатынын байқаған. Бұл зерттеулер үлкен маңыздылыққа ие болды себебі, сол арқылы көптеген ғалымдар жаңалықтар диффузиясына жергілікті, ұлттық және халықаралық деңгейде қызығушылық танытты.

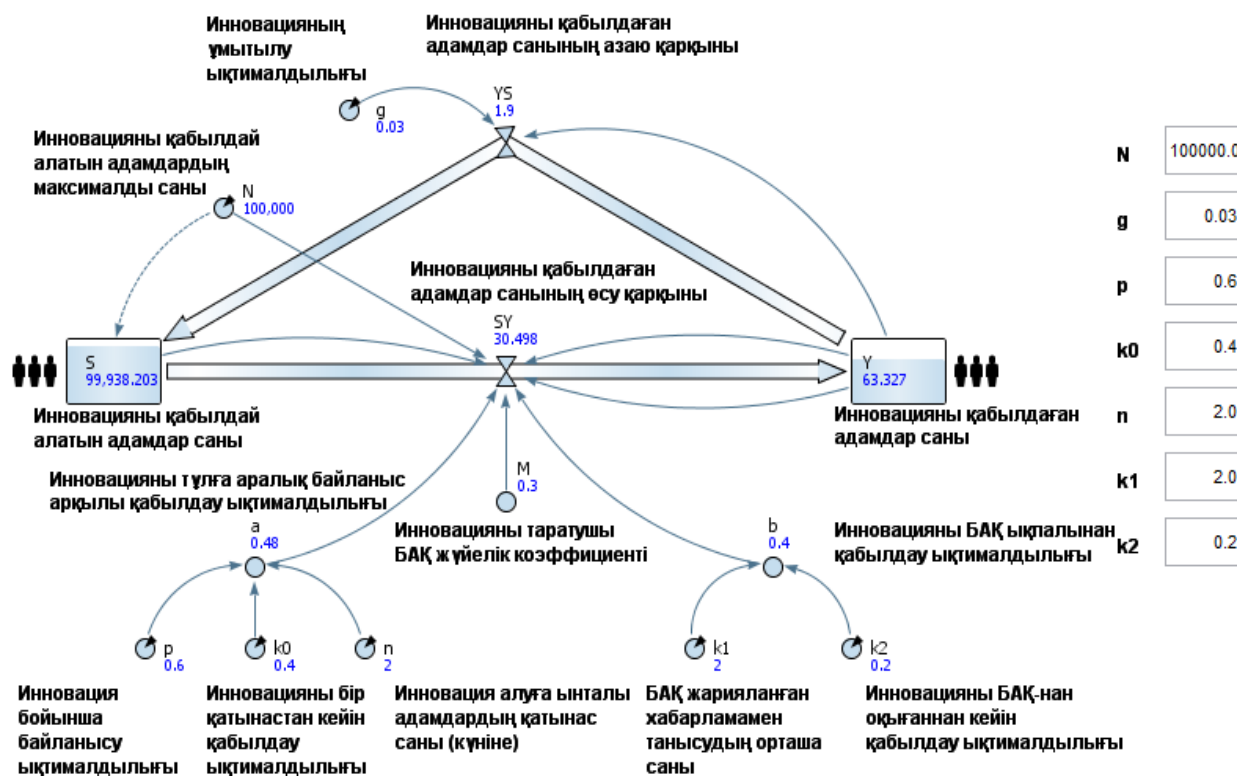
Қоғамдағы инновацияның беталысын болжау, олардың тұтынушы топтардың арасында таралу тапсырмалары ақпараттық коммуникациялық технологиялар сервисі дамымай тұрып пайда болған. XX ғасырдың 60-шы жылдары әлеуметтік жүйелерге инновациялардың ену үрдістерін және қоғамдағы өзгерістерді түсіндіретін математикалық модельдер және ұғымдар ұсынылды. Олардың ішінде Э. Роджерс және Ф. Бассты айтуға болады, бұл ғалымдардың зерттеу нәтижелері қазіргі кезге дейін қолданылып және дамып келеді.

Идеялардың таралу аясын кеңейтуге арналған модельдердің бірі – жасушалы автомат моделі [144-147]. Әлеуметтік және ақпараттық желі ретінде кездейсоқ графты қолдануға болады: N шыңдары бар желі үшін екі шың p ықтималдығымен қосылады. Кездейсоқ графтарда қысқа жолдың орташа ұзындығы және шағын кластерлеу коэффициенті бар. Кездейсоқ графтардың желілік құрылымы көбінесе шыңдардың қосылу ықтималдығына байланысты. Егер қосылу ықтималдығы төмен болса, онда граф кішкентай оқшауланған компоненттерге бөлінеді, ал жоғары болса, компоненттер түпнұсқалық графтың мөлшеріне сәйкес бір желіге біріктіріледі. Жолдың қысқа орташа ұзындығын және кластердің жоғары дәрежесін алу үшін жақын әлемдік әсерге ие желілік модель ұсынылады. Желі төмен габаритті тормен құрыла бастайды. Жоғарыда қарастырылған барлық модельдер нақты желілерге тән кейбір қасиеттерге ие. Желі бастапқы графтан басталады және бастапқы желідегі әр түйіннің дәрежесі кемінде 1 болуы керек, әйтпесе ол әрқашан желінің қалған бөліктерінен бөлініп тұрады. Уақыттың әр нүктесінде желіге жаңа түйін қосылады. Әрбір жаңа түйін қолданыстағы түйіндерге осы түйіндердің қосылу санына пропорционалды түрде қосылады. Ең көп қосылған түйіндер, әдетте, одан да көп сілтемелер жинақтайды, ал олардың саны аз түйіндер жаңа түйіндерді біріктіру үшін таңдалуы екіталай. Бұл модельдер коммуникациялық гипотезаға негізделген, гипотеза бойынша қоғамда ақпараттар алдымен кішігірім топты қамтиды, содан кейін өзге тұлғаларға әсер етеді. Көптеген жүйелерді қарапайым тізбек процестерімен сипаттауға болады. Ең айқын мысал - инфекцияның таралуы. Эпидемия жағдайында ауру бір адамнан екінші адамға жақын байланыс арқылы өтеді [148, 149]. Әлеуметтік және биологиялық жүйелердегі жағдай осыған ұқсас [150]. Талдаудың қарапайымдылығы үшін біз екілік нұсқаларды ғана қарастырамыз. Мысалы, бастапқыда барлық агенттер белсенді емес болып саналады, бірақ бірнеше агент белсенділік танытқан кезде тізбектік реакция басталады да олар желідегі қалған агенттерді белсендіреді. Идеялардың

таралуын талдау және модельдеу үшін желі ретінде кездейсоқ геометриялық графты қарастырамыз. Осылайша, $L \times L$ өлшемді кеңістікте кездейсоқ орналасқан N шыңдары бар граф болсын. Екі түйін бір-бірімен байланысқан болып саналады егер олар бір бірінен R қашықтығынан алыс емес орналасса. Кездейсоқ геометриялық графтың маңызды параметрі – бұл, бір түйіндегі көршілердің орташа саны ретінде анықталатын шыңның орташа дәрежесі. Бұл желілер үшін ортақ тапсырма - түйіндердің оңтайлы байланысын табу. Бұл мәселені шешу үшін келесі үлгіні қарастырамыз. Бастапқыда барлық агенттер белсенді емес күйде болады және белгілі уақыт аралығында деректерді тексеріп тұрады. Оқиғалар туралы деректер пайда болғанан кейін агенттер іске қосып, нақты уақытта режимінде деректер жинайды. Ереже: Агент бірнеше көршісі оянған кезде ғана оянады. Идеялардың таралуын талдау және модельдеу үшін желі ретінде кездейсоқ геометриялық графты қарастырамыз. Осылайша, $L \times L$ өлшемді кеңістікте кездейсоқ орналасқан N шыңдары бар граф болсын. Екі түйін бір-бірімен байланысқан болып саналады егер олар бір бірінен R қашықтығынан алыс емес орналасса. Кездейсоқ геометриялық графтың маңызды параметрі – бұл, бір түйіндегі көршілердің орташа саны ретінде анықталатын шыңның орташа дәрежесі.

3.2.1 Әлеуметтік қоғамда инновацияның таралуын имитациялық модельдеу

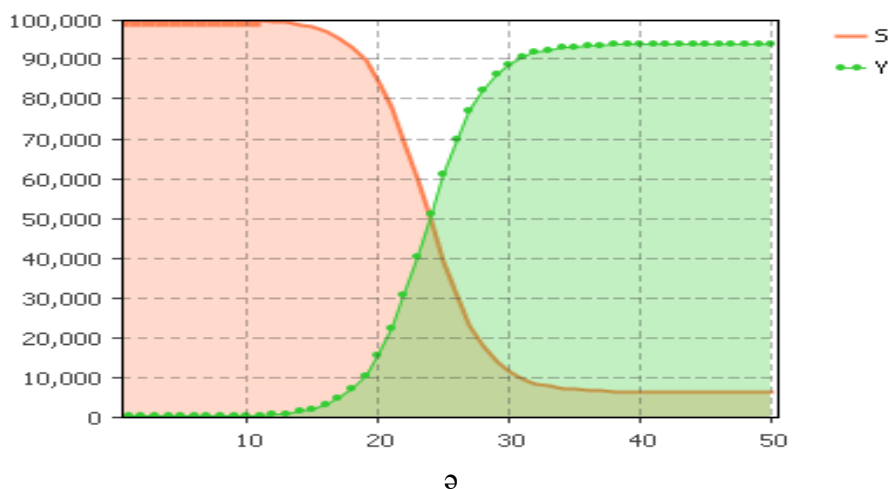
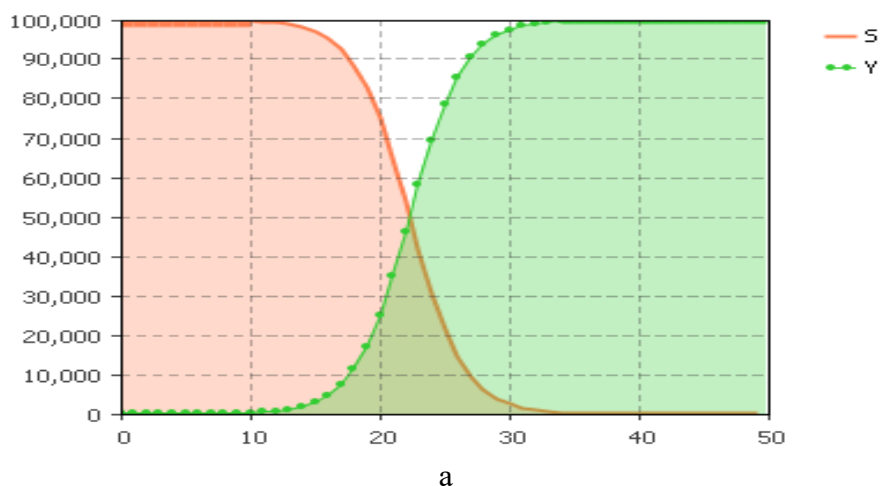
Инновацияның таралуын тәжірибе жүзінде модельдеу Anylogic [151] имитациялық модельдеу пакетінде орындалған. Бұл модельде агенттердің тәртібін түрлі әсер етуші параметрлердің есебімен зерттеуге болады.



Сурет 3.3 – Инновацияның таралу моделінің диаграммалар классы

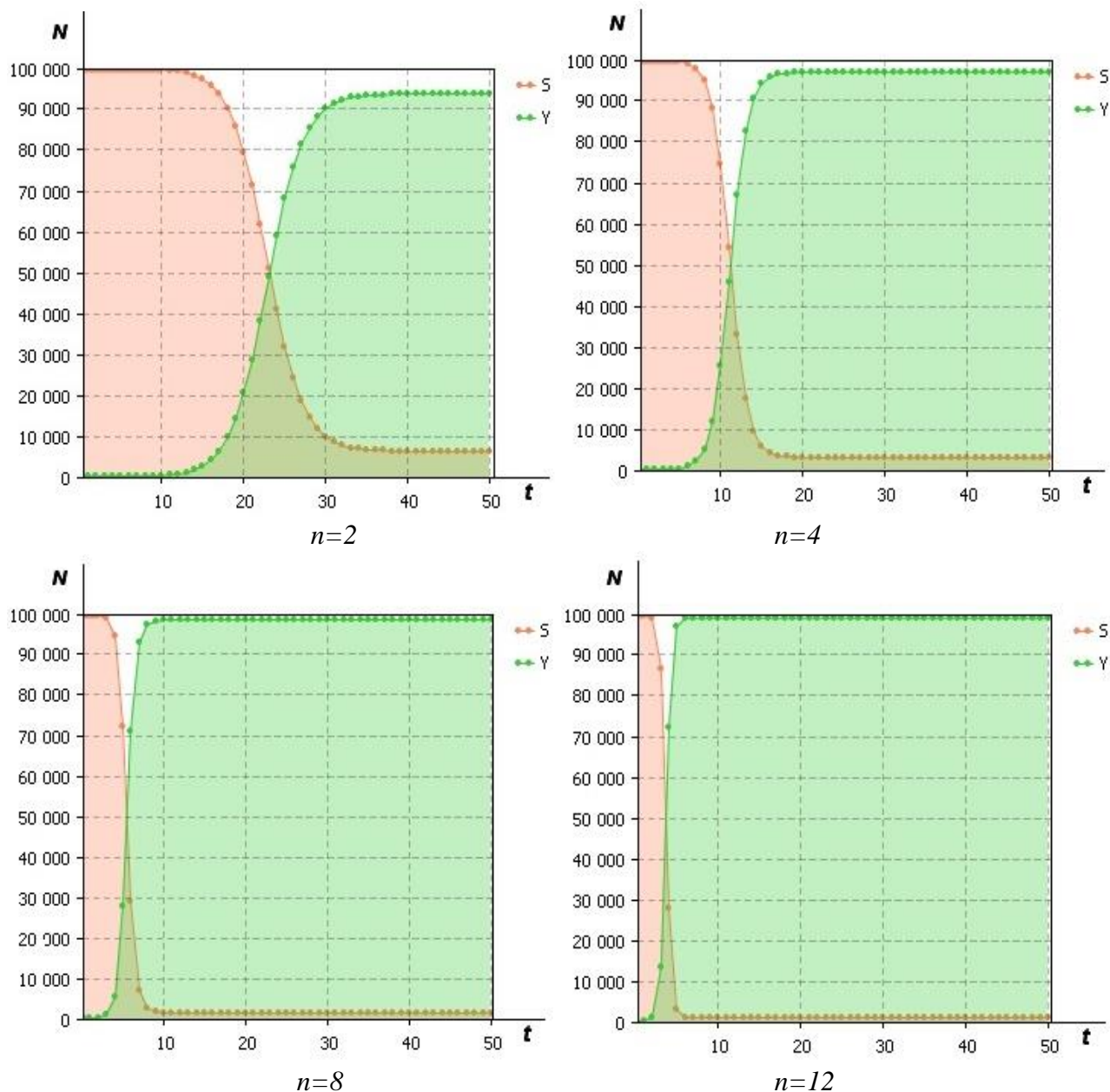
Өмірлік циклдың бастапқы деңгейінде инновация жарнама әсері ұшырайды, инновация ешкімге белгісіз, және агенттер оны қабылдау үшін инновация жарнамаланады. Нәтижесінде, жарнаманың тиімділігі, агенттер саны сияқты параметрлерге байланысты агенттердің белгілі бір бөлігі инновацияны сыртқы жайыттардың әсерінен қабылдайды (жарнама, БАҚ) [152, 153]. Инновацияны қабылдаған агенттер санының артуымен жарнаманың тиімділігі төмендейді, бірақ қандайда бір сәтте тұлғааралық коммуникация әсері артады. Бұл эффектiнiң себебi, агенттер инновацияны ресурс көлемi, сендiру күшi, байланыс жиiлiгi параметрлерiнiң тәуелдiлiгiнен қабылдайды және идеяны ендi қабылдаушы агенттер идеяны бiрiншi болып қабылдаған агенттермен сұхбаттасу нәтижесiнде қабылдайды (Қосымша Ө).

Сонымен қатар, белгiлi уақыт аралығында белгiсi бiр ықтималдықпен қолданушылардың бiр күйден екiншi күйге ауысуын модельдеуден бөлек, тәжiрибелiк маңыздылығы және iске асырылуы бойынша қолданушылардың қоғамға тiкелей әсерi, олардың әртүрлi әлеуметтiк байланыстардан ауысу процесстерi де үлкен қызығушылыққа ие. Сурет 3.4 сәйкес көрсетiлгендей, ақпаратты қабылдаған адамдардың динамикасы S -тәрiздi қисық түрiнде, Э.Роджерс тұжырымында ұсынылғандығын көрсетедi [3, б. 99-112].



Сурет 3.4 - S ақпаратты қабылдауға қабілетті және Y ақпарат қабылдаған адамдардың санының динамикасы: a - ақпаратты ұмыту коэффициентінсіз, b - ақпаратты ұмыту коэффициентімен

S ақпаратты алуға қабілетті және Y ақпаратты алған адамдар санының динамикасын көрсететін уақыт кестелерін келтірейік: (Сурет 3.5), келесі бастапқы шарттарда: $p=0.6$; $k_0=0.4$; $k_1=2$; $k_2=0.2$; $M=0.3$; $g=0.03$; $S(0)=100000$; $Y(0)=3$.



Сурет 3.5 - Адамдардың қатынас жиілігіне байланысты ақпаратты қабылдау жылдамдығы

Тәжірибе қоғамдастықтың жаңа мүшелерінің ақпаратты қабылдау жылдамдығы мен ақпаратқа (идеяларға) құмар адамдардың байланысының жиілігі арасындағы маңызды байланысты анықтады. Мұндағы уақыт өлшемі - тәулік (n). Ақпараттың ұмытылу коэффициентін ($g=0.03$) ескере отырып, тәулігіне екі байланыста ($n=2$), ақпаратты қабылдау жылдамдығы 32 тәулікте максимумға жетсе, тәулігіне он екі байланыста ($n=12$) ақпаратты қабылдау жылдамдығы 6 тәулікте ішінде максимумға жетеді. Біздің эксперименттік модельдеу нәтижелеріміз ақпараттың ұмытылу оның таралуына қаншалықты

әсер ететіндігін және тіпті белгілі бір жағдайларда таратуды тоқтатуға әкелуі мүмкін екендігін көрсетеді. Бұл нәтижелер түрлі ғылыми зерттеулерде және індеттің таралуын бақылауда қолдануға қызығушылық тудыруы мүмкін. Зерттелген модельді интернет арқылы жаңалықтардың таралуы мен олардың агенттердің әрекетіне әсерін сипаттау үшін пайдалануға болады. Сонымен қатар қаржылық жаңалықтарды зерттеу де ерекше қызығушылық тудырады, себебі олар ақша ағындарының қозғалысына және қаржы нарығына қатысушылардың мінез-құлқына тікелей әсер етеді. Белгілі бір қаржылық жаңалықтарының таралуы және ақпараттық факторлардың әсері қаржы активтерінің бағасы арасындағы корреляцияны сипаттамаларын талдауға бағытталған модель құруға мүмкіндік береді.

Бірінші кезеңде ақпараттық таратуға қатысатын агенттердің қимылын бағалау қажет. Ол үшін жаңалықтар мен сұраулардағы (сілтемелер индексі) кез-келген тақырып бойынша сілтемелерін санын өлшеуге мүмкіндік беретін Google Trends құралын қолданған жөн. Бұл шама 0-ден 100-ге дейінгі мәндерді қамтиды.

Осы сілтемелер индексінің уақыттық қатарына сырғымалы орта мәнін қолдана отырып, біз бөлімде зерттелген нәтижелерге сәйкес келетін, осы ақпаратқа қызығушылық және оның ұмытылу (сары сызық) кезеңдерін көре аламыз. Сонымен қатар, аталған модель қор нарығындағы инвесторлардың беталысын сипаттауға қолданылуы мүмкін. Бұл ретте, "x" идеясы ретінде инвесторлардың қандайда бір қаржы құралының баға трендісі бойынша көтерілу немесе құлдыруы туралы пікірі болуы мүмкін. Инвесторлар өз бағаларын белгілеу, ұсыну және келісімге тұру арқылы "x" идеясын таратады. Қаржы құралы бойынша жасалған келісімнің бағасын көру арқылы басқа инвесторлар "x" идеясы туралы біледі, және осы трендті қолдай отырып идеяны қабылдайды немесе қабылдамайды. α және n параметрлерінің косвенный оценки дегеніміз тренд кезіндегі сатылған қаржы активтерінің көлемі болып табылады. β коэффициенті трендтің әсер етуі бойынша корреляцияланған. R коэффициенті қоғам арасында қаржы құралының айтылу жиілігі бойынша бағаланады (Қосымша Б).

Тұлғааралық коммуникация жаңа ақпаратты қабылдауға ықпал етуші негізгі факторлардың бірі болып табылады. Ақпараттың таралу үлгісі технологиялардың, тауарлардың, оқытудың жаңа үлгілерінің таралу динамикасын және қылмыстың процесстерді дұрыс сипаттау мүмкіндігі бар. Сондықтан бұл саланы зерттеу әртүрлі жағдайларды болжау және бәсекелес нарықтық жағдайда тауарлар мен қызметтерді бақылау, сонымен қатар, қазіргі жағдайда интернет пен әлеуметтік желілердің дамуын ескере отырып, мемлекеттің, қоғамның және оның әрбір мүшесінің қауіпсіздігін қамтамасыз ету мәселелерін зерттеу үшін.

3.2.2 Жасушалы автомат кеңістігінде инновацияның таралуын талдау

Әлеуметтік желілердің негізгі міндеттерінің бірі - ақпарат тарату. Яғни, мақалалар, хабарламалар аудио, бейне және басқа да ақпарат түрлері әлеуметтік желілердің негізі болып табылады. Ақпараттар әлеуметтік желілерге байланыс

құралдары немесе бұқаралық ақпараттар құралдары (БАҚ) арқылы жүзеге асады. Веб-блогтар мен басқа да онлайн бұқаралық ақпарат құралдарының пайда болуымен (жаңалықтар порталы, форумдар және т.б.) ақпараттардың таралу үрдісін талдау жеңілдей түсті. Барлық ақпараттар жалпыға қолжетімді доменде сақталады, пайдаланушылар ақпаратты мәтін бойынша да, рейтингтік жүйе арқылы да (мысалы, «лүпіл», «бөлісу» белгілері) өз пікірін ашық түрде бөліседі. Мұның бәрі ақпараттарды таратудың үдерістерін бақылауға және таралу әсерін бағалауға мүмкіндік береді. Дегенімен әлеуметтік желілерде ақпараттың таратылуын модельдеу қиын тапсырмалардың бірі болып қала береді. Мұның себебі, үлкен көлемді ақпараттардың түрлі дереккөздерден жинаумен байланысты. Бұдан бөлек, зерттеу үшін әлеуметтік желілердің ұсыныстар, сілтемелер, тегтер, хабарламалар сияқты элементтерін өңдеу және қадағалау қажет. Әсер ету - бұл өзара ықпалдасу барысында басқа субъектінің, яғни жеке адамның немесе ұжымның, мінез-құлқын, көзқарасын, ниетін, идеяларын және талдауын өзгерту процесі және нәтижесі болып табылады [154, 155].

Психологтардың зерттеуі көрсеткендей [145, б. 39; 146, б. 550], көбінесе әлеуметтік желілердегі агенттерге шешім қабылдау үшін қажетті деңгейдегі ақпарат жетіспейді немесе оны өздігінен өңдей алмайды, сондықтан олардың қабылдаған шешімдері басқа агенттердің шешімдеріне немесе идеяларына негізделуі мүмкін (әлеуметтік ықпал). Әлеуметтік ықпал екі процесте жүзеге асырылады: қарым-қатынас (қарым-қатынас барысында, тәжірибе алмасу және ақпарат алмасу, белгілі бір мәселелерді және идеяларды талқылау) барысында және салыстыру (әлеуметтік идентификациялау мен басқа агенттермен өзін салыстыру, стратегиялық артықшылығын табу, басқа агент «менің жағдайымда болса, не істемек?» деген сұраққа жауап беру барысында инновацияларды қабылдай алады. Әлеуметтік желілер ақпараттың, идеялардың таралуына және оның мүшелері арасында ықпал етуде үлкен рөл атқарады. Әлеуметтік желілердегі ақпараттың таралуы «инновацияның таралуы» (*diffusion of innovations*) терминімен тығыз байланысты [156], сондықтан төменде қарастырылған кейбір ұғымдар осындай диффузияның модельдерін қарастырылады.

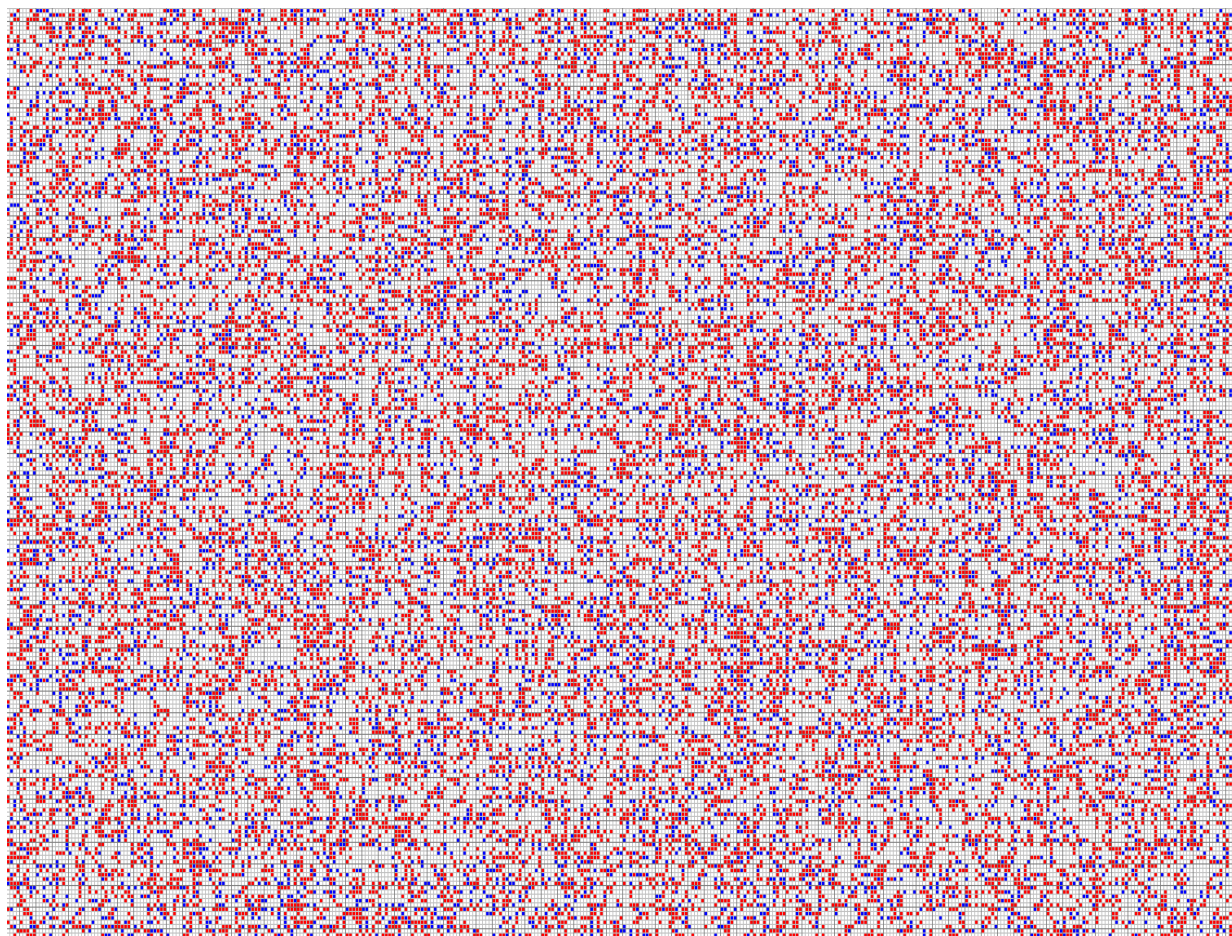
Зерттеу барысында ақпараттың таратылу процесін модельдеу үшін жасушалы автомат таңдалынды. Жасушалы автомат - бұл бір-бірімен байланысқан біртекті жасушаларды қамтитын дискретті динамикалық жүйе. Бұл кез-келген жасуша кез-келген сәтте шекті күйлердің біреуін қабылдай алатын ерікті өлшемдер торы және де торлардың бір ұяшықтан басқа ұяшықтарға ауысуы белгісі ережесі бойынша орындалады. Әрбір жасушаның күйі осы жасушаның маңында орналасқан жасушалармен анықталады [157]. Сондықтан, мұндай анықтама әлеуметтік желілер анықтамасына қайшы келмейді. Осыған сүйене отырып, біз бұл модель әлеуметтік диаграмманы құруға және әлеуметтік процестерді модельдеу үшін қолайлы деп есептей аламыз. Жасушалы автомат келесі ережелермен анықталады:

– әрбір жасушадағы мәндердің өзгеруі бір мезгілде жүреді (уақыт бірлігінің өзгеруі қадам болып табылады);

- жасушалы автомат желісі біртекті болып табылады, яғни ереже барлық жасушаларға бірдей;
- жасуша тек көрші жасушаға ықпал ете алады;
- жасуша күйінің саны.

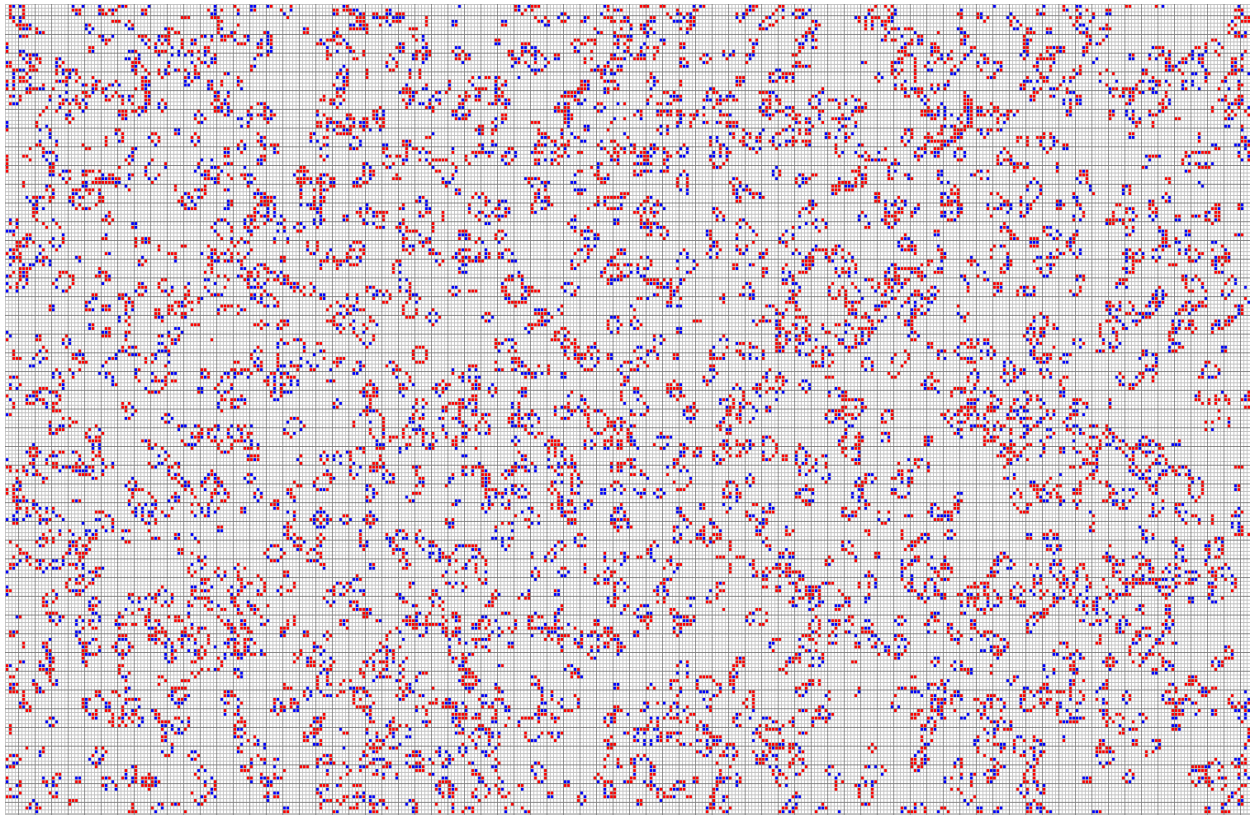
Жасушалы автоматтың теориясы инновация диффузиясын талдау үшін қолданылады. Бұл процесс интернет желісіндегі ақпараттардың таралуына өте ұқсас. Яғни, модельді біздің тапсырмаға сәйкес түрлендірсек: жеке адам екі жағдайды бірін қабылдайтын ұяшыққа сәйкес келеді: 1 - ақпарат қабылданды, 0 - ақпарат қабылданбады. Айтарлық, ақпаратты бір рет қабылдағаннан кейін жасуша өзгеріссіз қалады. Автомат жақын көршілердің пікірін ескере отырып, ақпаратты қабылдау немесе қабылдамау туралы шешім шығарады: егер ақпарат t көршілермен қолдас, p - ақпараттарды қабылдау ықтималдығы болса, және егер $pt > R$, мұндағы R - тіркелген шекті мән, онда жасуша ақпаратты қабылдайды. Сонымен қатар, ақпарат түріне қосымша талаптар қойылуы мүмкін: жасуша соңғы ақпаратқа ие (қызыл түсті), жасушадағы ақпарат ескірген (көк түсті), жасушада ақпарат жоқ немесе ол туралы ұмытылып кеткен (ақ түсті).

Сурет 3.6 сәйкес көрсетілгендей осы үдерістердің әртүрлі уақытын аралығында сандық модельдеу нәтижесі көрсетіледі. Уақыт өте келе, қауымдастықтағы агенттер инновацияны қабылдаған және қабылдамаған болып түрлі топтарға бөлінгені көрінеді.



а

а – бастапқы күй

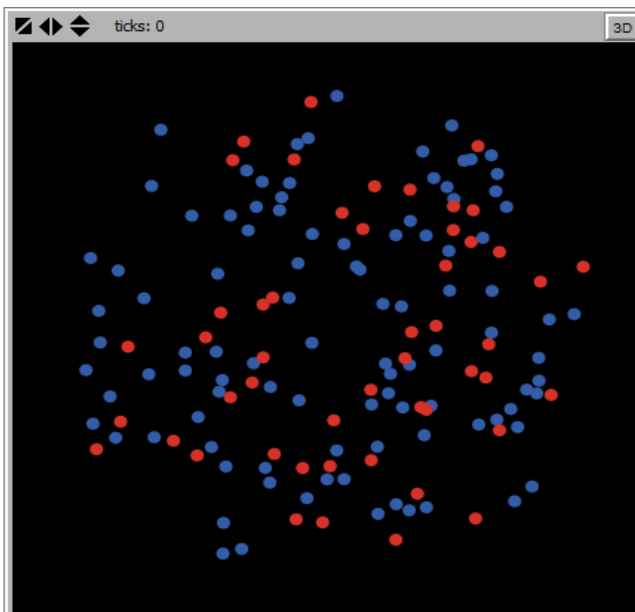


ә

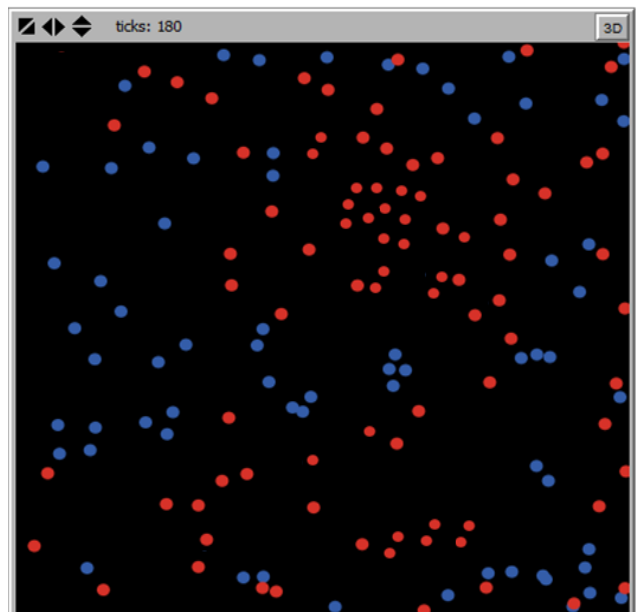
ә –соңғы күй

Сурет 3.6 – Инновцияның таралуын кеңістікте модельдеу нәтижелері

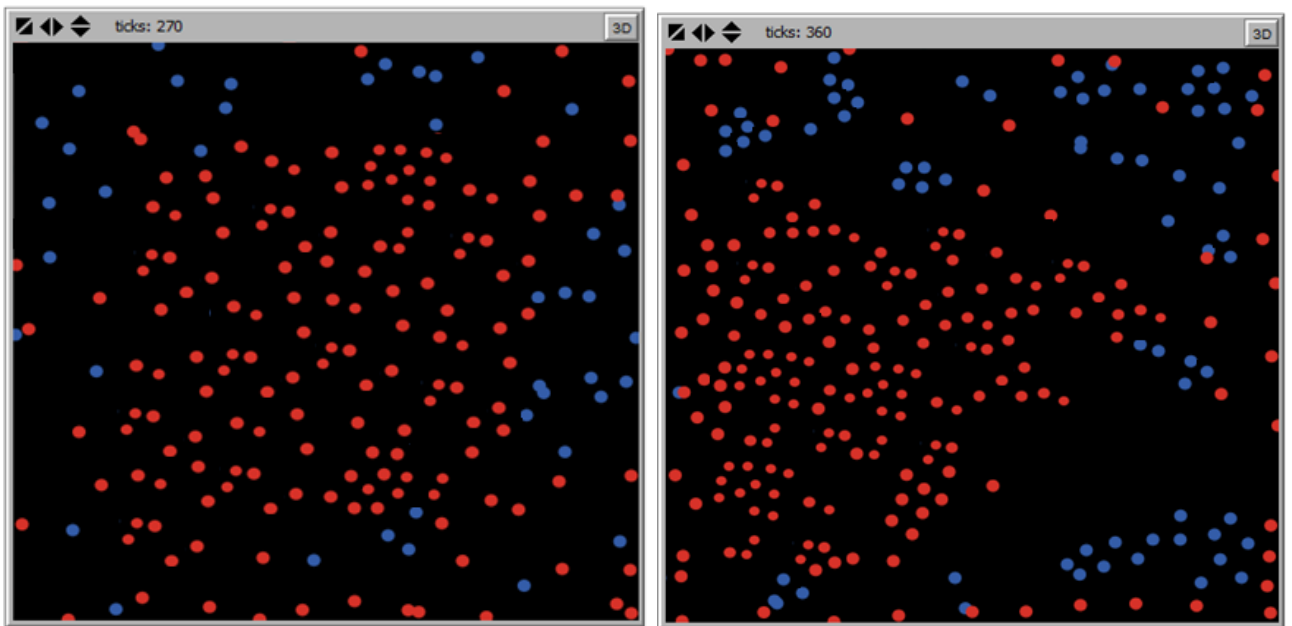
Сурет 3.7 сәйкес көрсетілгендей әртүрлі уақыт аралығында осы процестерді жазықтықта сандық модельдеу (қызыл нүктелер - инноваторлар, көк - агенттер) нәтижелері көрсетілген. Зерттеу нәтижесі бойынша уақыт өте келе қауымдастық жазықтық бойымен қозғалатын және саны жағынан шамалас топтарға бөлінетіндігін көрсетеді.



а



ә



б

в

Сурет 3.7 - Жазықтықта модельдеудің нәтижелері: *a* – бастапқы қалып; *b* - 180 уақыттық қадам; *c* - 270 қадам; *d* - тұрақталған күй.

Әлеуметтік жүйеде инновацияның таралуын зерттеу нәтижелері жаңа идея пайда болған кезде қоғам оны бірдей қабылдамай, жаңа идеяны қабылдайтын және қабылдамайтын топтарға бөлінетінін көрсетті. Тәжірибе нәтижелері көрсеткендей, идеяны қабылдаған қауымдастық мүшелерінің үлесі белгілі бір деңгейге жеткенде бірден күрт артады және, сәйкесінше, инновацияның таралу динамикасы басқаша бағыт алады. Уақыт өте келе келіспеушіліктер жойылады, өйткені қауымдастық мүшелері бір пікірді ұстанатын болады. Әртүрлі уақыт аралығында осы процестерді жазықтықта сандық модельдеу (қызыл нүктелер - инноваторлар, көк - агенттер) нәтижелері көрсетілген. Зерттеу нәтижесі бойынша уақыт өте келе қауымдастық жазықтық бойымен қозғалатын және саны жағынан шамалас топтарға бөлінетіндігін көрсетеді [158].

Әлеуметтік жүйеде инновацияның таралуын зерттеу нәтижелері жаңа идея пайда болған кезде қоғам оны бірдей қабылдамай, жаңа идеяны қабылдайтын және қабылдамайтын топтарға бөлінетінін көрсетті. Тәжірибе нәтижелері көрсеткендей, идеяны қабылдаған қауымдастық мүшелерінің үлесі белгілі бір деңгейге жеткенде бірден күрт артады және, сәйкесінше, инновацияның таралу динамикасы басқаша бағыт алады [159].

ҚОРЫТЫНДЫ

Диссертациялық жұмыста инновациялар сөздігінің жұмысын оңтайландыру тапсырмасы қойылды. Инновациялар сөздігінің жұмысын оңтайландыру тапсырмасы үшін жұмысшы жиын стратегиясы (WS стратегия) қолданылды. Қазіргі уақытта WS стратегиясы әдетте теориялық зерттеу базасы ретінде, салыстырмалы зерттеулер немесе қосалқы мақсаттар үшін қолданылады, ал алмасу стратегиясы ретінде оны жүзеге асыру қымбат болып саналады. Алайда, біздің жағдайда, сөздік ұғымдарын қолдана отырып алынған нәтижелерден WS тәрізді жылдам әрі арзан алмастыру алгоритмін құруға болады. Соңғысы алынған нәтижелердің маңыздылығын күшейтеді. Жұмыста WS стратегия негізінде инвариант қасиетін қанағаттандыратын инновация ұғымы енгізілді. Дәл осы инварианттардың болуы булеан шыңдары бойынша кездейсоқ кезу ретінде есептеу процесінің геометрикалық интерпретациясын құруға мүмкіндік берді. Бұл, өз кезегінде, бір эксперимент кезінде сөздіктегі бет қателіктерінің математикалық күтімін азайту тапсырмасы үшін және h эксперименті бойынша бет қателіктерінің орташа санын азайту тапсырмасы үшін қолданылды. Есептеу процесінің геометриялық интерпретациясы инновациялар сөздігімен байланысты есептеу процесін нақтылауға көмектеседі. Сонымен қатар, ол инновациялар сөздігі үшін оңтайландыру моделін түсінуді жеңілдетеді. Онда инновациялар сөздігін оңтайландыру тапсырмасын шешу үшін шешімдер жиынының анықтайтын функционал және шектеулер жүйесі бар. Есептеу процесінің геометриялық интерпретациясы сонымен қатар *in* бақылау күйі және x матрицасымен туындайтын жұмысшы жиын шектеулерін нақтылауға мүмкіндік берді. Сонымен қатар, есептеу процесінің геометриялық интерпретациясы сызықтық емес теңсіздіктер санын едәуір азайтуға мүмкіндік береді, бұл оңтайландыру тапсырмасының көлемін едәуір азайтады және есептеу алгоритмінің тиімділігіне оң әсер етті. Диссертациялық жұмысты орындау барысында алынған нәтижелер:

– алғаш рет жұмысшы жиын стратегиясы негізінде инновациялар сөздігінің жұмысын оңтайландыру тапсырмасы қойылды;

– біздің зерттеулер аясында инновация ұғымы енгізілді және инновацияның қалыптасу механизмі сипатталынды;

– инновациялар сөздігін оңтайландырудың математикалық моделі құрылды:

а) инновациялар сөздігін оңтайландыру тапсырмасының функционалы (соның ішінде $h \geq 1$ эксперименті бойынша эмпирикалық функционал) және шектеулер жүйесі табылды;

б) бірегей комбинаторлық кеңістік (Булеан) шыңдары бойынша кездейсоқ кезу ретінде есептеу процесінің геометриялық интерпретациясы табылды;

с) есептеу процесінің геометриялық интерпретациясы негізінде инновациялар сөздігін оңтайландыру тапсырмасының өлшемін азайту мүмкіндігі көрсетілді;

- оңтайландыру тапсырмасының бағалау функционалы құрылды;
- эпидемия қауіпін төмендету мақсатындағы мультиагентті жүйе топтарын қайта құрылымдау тапсырмасы үшін қолданылуы көрсетілді.

Диссертациялық жұмысты орындау барысында алынған нәтижелер инновацияның қалыптасу механизмін зерттеуге, қоғамдағы инновацияның қалыптасуын және таралуын, саяси пікірлердің, идеялардың туындауын және таралуын және олардың қоғамға әсерін сараптау барысында қолдануға мүмкіндік береді. Сонымен қатар, алынған нәтижелер құрылған модельдің ерекшеліктерін ескеретін практикалық оңтайландыру алгоритмін құруға негіз береді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Друкер П.Ф. Бизнес и инновации / пер. с англ. – М.; СПб.; Киев: Вильямс, 2007. – 432 с.
- 2 Клаус Ш. Төртінші индустриялық революция. – Алматы: Ұлттық аударма бюросы, 2018. – 200 б.
- 3 Rogers E.M. Diffusion of Innovations. – Ed. 5th. – London: Free Press, 2003. – 576 p.
- 4 Bass F. A new product growth model for consumer durables // Management Science. – 1969. – Vol. 15(5). – P. 215-227.
- 5 Deutschmann P.J., Danielson W.A. Diffusion of knowledge of the major news story // Journalism Quarterly. – 1960. – Vol. 37. – P. 345-355.
- 6 Baronchelli A. A gentle introduction to the minimal Naming Game // Belgian Journal of Linguistics. – 2016. – Vol. 30(1). – P. 171-192.
- 7 Ломакин С.Г., Федотов А.М. Модель самоорганизации в агентных системах с передачей информации // Системный анализ и информационные технологии (САИТ-2013): тр. 5-й междунар. конф. – Красноярск, 2013. – С. 243-247.
- 8 Мурзин Ф.А. и др. О новых инструментах поиска информации на основе грамматики связей // Интеллектуальный анализ сигналов, данных и знаний: методы и средства: сб. ст. всеросс. науч.-практ. конф. с междунар. уч. – Новосибирск: НГТУ, 2017. – С. 16-166.
- 9 Барахнин В.Б. Обработка информации – это процесс, который происходит прежде всего при участии человека // Теория и практика общественно-научной информации. – 2013. – №21. – С. 211-218.
- 10 Дюсембаев А.Е. Математические модели сегментации программ. –М.: ФИЗМАТЛИТ, 2001. – 208 с.
- 11 Tussupov D.A. et al. Application of Cellular Automata for Modeling and Review of Methods of Movement of a Group of People // Journal of Theoretical and Applied Information Technology. – 2019. – Vol. 97, №15. – P. 4000-4010.
- 12 Щепина И.Н. Модели диффузии и замещения нововведений (обзор литературы). – М.: ЦЭМИ АН СССР, 1990. - 48 с.
- 13 Tarde G. The laws of imitation (E.C. Parsons, Trans.). – NY.: Holt, 1903. – 434 p.
- 14 Rogers E.M., Singhal A. Diffusion of innovations // In book: An integrated approach to communication theory and research. – Mahwah, NJ: Lawrence Erlbaum Associates, 1996. – P. 409-420.
- 15 Rogers E., Shoemaker F. Communication of Innovations: A Crosscultural Approach. – Ed. 2nd. – London:Free Press, 1971. – 476 p.
- 16 Bandura A. Social foundations of thought and action: A social cognitive theory. – New Jersey: Prentice-Hall, 1986. – 617 p.
- 17 Mansfield E. Industrial research and Technological Innovation // The Economic Journal. – 1968. – Vol. 78, Issue 311. – P. 676-679.
- 18 Мэнсфилд Е. Экономика научно-технического прогресса / пер. с англ. – М.: Прогресс, 1970. – 237 с.

- 19 Bass F. Comments on “A New Product Growth for Model Consumer Durables The Bass Model” // *Management Science*. -2004.-Vol.50, No.12. -P.1833-1840.
- 20 Fedotov A.M., Murzakhmetov A.N., Dyusembaev A.E. Expansion of ideas and processes in social and biological communities // *Eurasian Journal of Mathematical and Computer Applications*. – 2018. – Vol. 6, Issue 4. – P. 17-28.
- 21 Baronchelli A., Gong T., Puglisi A. et al. Modeling the emergence of universality in color naming patterns // *Proc. Natl. Acad. Sci.* – 2010. – Vol. 107. – P. 2403-2407.
- 22 Мурзахметов А.Н. Құнды қағаздар қоржынның тиімді аумағын есептеу жүйесі // *ҚазККА Хабаршысы*. – 2017. – №2(101). – Б. 241-248.
- 23 Baronchelli A., Loreto V., Steels L. In-depth analysis of the naming game dynamics: The homogeneous mixing case // *Int. J. Mod. Phys.* – 2008. – Vol. C19. – P. 785-812.
- 24 Puglisi A., Baronchelli A., Loreto V. Cultural route to the emergence of linguistic categories // *Proc. Natl. Acad. Sci.* – 2008. – Vol. 105. – P. 7936-7940.
- 25 Murzakhmetov A.N., Infante Moro A., Dyusembayev A.E. Social media and networks as a tool to analysis of tourists preferences and requirements // *Vestnik KazNRTU*. – 2020. – Vol. 2(138). – P. 227-231.
- 26 Бапанов А.А., Мурзахметов А.Н. Современные классификаций онтологий // *Сб. матер. 10-й междунар. науч. конф. студентов и молодых ученых "Наука и образование-2015"*. – Астана, 2015. – С. 425-427.
- 27 Гришко М.В., Мурзахметов А.Н. Принятие инвестиционных решений на основе эконометрического анализа в условиях Казахстанского фондового рынка // *Вестник НАН РК*. – 2015. – Т. 2. – С. 250-256.
- 28 Steels L. A self-organizing spatial vocabulary // *Artificial Life*. – 1995. – Vol. 2, Issue 3. – P. 319-332.
- 29 Baronchelli A., Felici M., Loreto V. et al. Sharp transition towards shared vocabularies in multi-agent systems // *J. Stat. Mech.* – 2006. – Vol. 2006. – P. 06014.
- 30 Lu Q., Korniss G., Szymanski B.K. The naming game in social networks: community formation and consensus engineering // *J. Economic Interaction Coordination*. – 2009. – Vol. 4, 2009. – P. 221-235.
- 31 Xie J., Sreenivasan S., Korniss G. et al. Social consensus through the influence of committed minorities // *Phys. Rev.* – 2011. – Vol. E84. – P. 011130.
- 32 Zhang W., Lim C.C. Noise in naming games, partial synchronization and community detection in social networks // <https://arxiv.org/pdf/1008.4115.pdf>. 22.02.2019.
- 33 Dall'Asta L., Baronchelli A., Barrat A. et al. Agreement dynamics on small-world networks // *Euro. Phys. Lett.* – 2006. – Vol. 73. – P. 969.
- 34 Liu R.R., Jia C.X., Yang H.X. et al. Naming game on small-world networks with geographical effects // *Physica A*. – 2009. – Vol. 388. – P. 3615-3620.
- 35 Barrat A., Baronchelli A., Dall'Asta L. et al. Agreement dynamics on interaction networks with diverse topologies // *Chaos*. – 2007 – Vol. 17. – P. 026111.

- 36 Dall'Asta L., Baronchelli A., Barrat A. et al. Nonequilibrium dynamics of language games on complex networks // *Phys. Rev.* – 2006. – Vol. E74. – P.036105.
- 37 Baronchelli A., Dall'Asta L., Barrat A. et al. The role of topology on the dynamics of the Naming Game // *Eur. Phys. J. Special Topics.* – 2007. – Vol. 143. – P. 233-235.
- 38 Wang W.X., Lin B.Y., Tang C.L. et al. Agreement dynamics of finite-memory language games on networks // *Eur. Phys. J.* – 2007. – Vol. B60. – P. 529-536.
- 39 Li B., Chen G., Chow T.W.S. Naming game with multiple hearers // *Commun. Nonlinear Sci.* – 2013. – Vol. 18. – P. 1214-1228.
- 40 Fedotov A.M., Tusupov J.A., Sambetbayeva M.A. et al. Classification model and morphological analysis in multilingual scientific and educational information systems // *Journal of Theoretical and Applied Information Technology.* – 2016. – Vol. 86, Issue 1. – P. 96-111.
- 41 Revnivykh A.V., Fedotov A.M. Main Reasons of Information Systems Vulnerability // *Global Journal of Pure and Applied Mathematics.* – 2016. – Vol. 12, Issue 3. – P. 2133-2142.
- 42 Fedotov A.M., Tussupov J., Sambetbayeva M. et al. Development and implementation of a morphological model of kazakh language // *Eurasian Journal of Mathematical and Computer Applications.* – 2015. – Vol. 3, Issue 3. – P. 69-79.
- 43 Fedotov A.M., Fedotova O.A., Sambetbayeva M.A. Multilingual Thesaurus in Information System for Scientific and Educational Activity Support // *Data Analytics and Management in Data Intensive Domains: proced. 20th internat. conf. (DAMDID/RCDL'2018).* – M.: FRC CSC RAS, 2018. – P. 360-361.
- 44 Федотов А.М., Ломакин С. Математическое моделирование распространения мнений в обществе // *Актуальные проблемы вычислительной и прикладной математики 2014: матер. междунар. конф.* – Новосибирск, 2014. – С. 78-79.
- 45 Leonova Y.V., Fedotov A.M., Fedotova O.A. Thematic classification of the thesis abstracts // *Eurasian Journal of Mathematical and Computer Applications.* – 2017. – Vol. 5, Issue 1. – P. 74-86.
- 46 Abdikerimova G.B., Murzin F.A., Bychkov A.L. et al. Software tools for cell walls segmentation in microphotography // *Journal of Theoretical and Applied Information Technology.* – 2018. – Vol. 96(15). – P. 4783-4793.
- 47 Kantureyeva M.A., Tussupov J.A., Murzin F.A. et al. Application of cellular automata for modeling and review of methods of movement of a group of people // *Journal of Theoretical and Applied Information Technology.* – 2019. – Vol. 97(15). – P. 4000-4010.
- 48 Yerimbetova A.S., Sagnayeva S.K., Murzin F.A. et al. Creation of tools and algorithms for assessing the relevance of documents // *Proceed. of the 3rd rus.-pac. conf. on Computer Technology and Applications (RPC-2018).* - Vladivostok, 2018. – P. 140611.
- 49 Murzin F.A., Sluev V.A. A memory organization for parallel computers // *New Generation Computing.* – 1988. – Vol. 6(1). – P. 3-18.

- 50 Мурзин Ф.А., Полетаев С.А. История развития суперкомпьютерной вычислительной техники // Конструирование и оптимизация параллельных программ: сб. ст. – Новосибирск, 2008. – С. 174-215.
- 51 Baronchelli A., Caglioti E., Loreto V. et al. Dictionary-based methods for information extraction // *Physica A: Statistical Mechanics and its Applications*. – 2004. – Vol. 342, Issue 1-2. – P. 294-300.
- 52 Loreto V., Baronchelli A., Puglisi A. Mathematical modeling of language games // In book: *Evolution of Communication and Language in Embodied Agents*. – Berlin: Springer, 2009. – P. 263-281.
- 53 Alessandretti L., Sun K., Baronchelli A. et al. Random walks on activity-driven networks with attractiveness // *Physical review*. – 2017. – Vol. E95(5). – P. 052318.
- 54 Baronchelli A., Loreto V., Puglisi A. Individual biases, cultural evolution, and the statistical nature of language universals: the case of color naming systems // *PloS one*. – 2015. – Vol. 10(5). – P. e0125019.
- 55 Ляпунов А.А. О математических проблемах кибернетики // *Известия высших учебных заведений*. – 1958. – №5. – С. 166-174.
- 56 Янов Ю.И. О логических схемах алгоритмов // *Проблемы кибернетики: сб. ст.* – М.: Наука, 1958. – Вып. 1. – С. 75-128.
- 57 Лавров С.С., Гончаров Л.И. Автоматическая обработка данных, хранение информации в памяти ЭВМ. – М.: Наука, 1971. – 160 с.
- 58 Dearnley F.H., Hewell G.B. Automatic segmentation of programs for a two level store computer // *Computer journal*. – 1964. – Vol. 7, Issue 3. – P. 185-187.
- 59 Dennis J. Segmentation and the design of multiprogrammed computer systems // *Journal of ACM*. – 1965. – Vol. 12, Issue 4. – P. 750-759.
- 60 Schurman A. The Application of graphs, of analysis, of distribution, of loops in a program // *Information and Control*. – 1964. – Vol. 7, Issue 3. – P. 275-282.
- 61 Schurman A. Rozklad konturow w grafje skonczonej oraz zastosowanie teorii grafow do automaticznego programowania // *Algoritmy*. – 1965. – Vol. 2, Issue 4. – P. 28-35.
- 62 Ramamoorthy C.V. The analytic design of look ahead and program segmenting systems for multiprogrammed compute // *Proc. 21st nat. dep. conf. ACM*. – NY, 1966. – P. 66-75.
- 63 Achard M.S. Segmentation automatique des programmes independamment des langages de programmation. – Paris, 1975. – 350 p.
- 64 Ferrari D., Kobayashi M. Program restructuring for global LRU Environment // *Proc. of the conf. intern. comput. symp.* – Liege, Belgium, 1977. – P. 277-283.
- 65 Файнштейн И.А. Некоторые методы оптимальной сегментации программ // В кн.: *Современные проблемы управления*. – М.: Наука, 1974. – С. 172-176.

- 66 Kral J. The Formulation of the Problem of Program Segmentation in the Terms of Pseudoboolean Programming // *Kybernetika*. – 1968. – Vol 1, Issue 4. – P. 1-6.
- 67 Fowler M., Beck K. Refactoring: Improving the Design of Existing Code. – Ed. 2nd. – Boston: Addison-Wesley Professional, 2018. – 432 p.
- 68 Rachow P. Refactoring Decision Support for Developers and Architects based on Architectural Impact // *Proceed. internat. conf. on Software Architecture Companion (ICSA-C)*. – Hamburg, 2019. – P. 262-266.
- 69 Bobde S., Phalnikar R. Restructuring of Object-Oriented Software System Using Clustering Techniques // *Proceed. of internat. conf. on Computational Science and Applications (ICCSA 2019)*. – Pune: MIT World Peace University, 2019. – P. 419-425.
- 70 Rathee A., Chhabra J.K. Restructuring of Object-Oriented Software through Cohesion Improvement Using Frequent Usage Patterns // *ACM SIGSOFT Software Engineering Notes*. – 2017. – Vol. 42, №3. – P. 1-8.
- 71 Феррари Д. Оценка производительности вычислительных систем. – М.: Мир, 1981. – 545 с.
- 72 Brian W.K. Optimal Sequential Partitions of Graphs // *Journal of the ACM*. – 1971. – Vol. 18, №7. – P. 34-40.
- 73 Черенин В.П., Хачатуров В.Р. Решение методом последовательных расчетов одного класса задач о размещении производства // *Экономико-математические методы*. – 1965. – №2. – С. 279-290.
- 74 Dusembaev A.E. Correctness of models of algorithms of segmentation of programs // *Int. Journal Pattern Recognition and Image Analysis*. – 1993. – Vol. 3, №2. – P. 167-184.
- 75 Dusembaev A.E. Statistical models of optimization for virtual memory systems with WS strategy // *Proc. COMPSTAT 84, Int. Symp.* - Prague, 1984.
- 76 Lau E. J., Ferrari D. Program restructuring in multilevel virtual memory // *IEEE Tr. Software Eng.* – 1983. – Vol. SE-9, №1. – P. 69-78.
- 77 Горинштейн Л.Л. О разрезании графов // *Известия АН СССР*. – 1969. – №1. – С. 79-85.
- 78 Каспшицкая М.Ф., Сергиенко И.В. Модели и методы решения на ЭВМ комбинаторных задач оптимизации. – Киев: Наукова думка, 1981. – 288 с.
- 79 Ляпунов А.А. О логических схемах программ // *Проблемы кибернетики: сб. ст.* – М.: Наука, 1958. – Вып. 1. – С. 46-74.
- 80 Янов Ю.И. О логических схемах алгоритмов // *Проблемы кибернетики: сб. ст.* – М.: Наука, 1958. – Вып. 1. – С. 75-120.
- 81 Калужнин Л.А. Об алгоритмизации математических задач // *Проблемы кибернетики: сб. ст.* – М.: Наука, 1959. – Вып. 2. – С. 51-69.
- 82 Karp R. A note on the application of graph theory to digital computer programming // *Inform. and Control*. – 1960. – Vol. 3, №2. – P. 179-190.
- 83 Ferrari D. Improving locality by critical working sets // *Comm. of ACM*. – 1974. – Vol. 17, №11. – P. 614-620.

- 84 Hatfield D.J., Gerald J. Program restructuring for virtual memory // IBM Syst. Journal. – 1971. – Vol. 10, №3. – P. 168-192.
- 85 Horspool R.N., Laks J.M. An improved block sequencing method for program restructuring // J. Syst. and Software. – 1983. – №3. – P. 245-250.
- 86 Achard M.S., Babonneau J.Y., Carpentier M. et al. The clustering algorithms in the OPALE restructuring system // In book: Performance of Computer Installations. – North Holland, 1978. – P. 137-153.
- 87 Мурзахметов А.Н. Интернет жаңалықтарының мәтіндерін автоматты талдау моделі // Д. Серікбаев атындағы ШҚМТУ хабаршысы. – 2017. – №2. – Б. 112-116.
- 88 Babonneau J.Y., Achard M.S., Morriset G. et al. Automatic and general solution to the adaptation of programs in paging environment // Procceed. of 6th ACM. symp. on Operating Systems Principles. – NY., 1977. – P. 109-116.
- 89 Peachey J.B., Bunt R.B., Colbourn C.J. Some empirical observations on program behavior with applications to program restructuring // IEEE Trans. Software Eng. – 1985. – Vol. SE-11, №2. – P. 188-193.
- 90 Kobayashi M. A set of strategy-independent restructuring algorithms // Software Practice and Experience. – 1977. – Vol. 7. – P. 109-116.
- 91 Masuda T., Shiota H., Noguchi K. et al. Optimization of program organization by cluster analysis // Procceed. of 6th IFIP congress. – Stockholm, 1974. – P. 261-266.
- 92 Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. – Киев: Наукова думка, 1985. – 381 с.
- 93 Berztiss A.T. A note on segmentation of computer programs // Information and Control. – 1968. – Vol. 12, №1. – P. 21-22.
- 94 Comeau L.W. A study of effect of user program optimization system // ACM symp. Operating Syst. Principles. – Cutlinburg, 1967. – P. 229-239.
- 95 Дюсембаев А.Е. Информатика. Структуры данных, сортировка, поиск.: курс лекций и лаборат. работ. – Алматы, 2010. – 153 с.
- 96 Таненбаум Э.С. Архитектура компьютера. – Изд. 6-е. – СПб.: Питер, 2019. – 816 с.
- 97 Dusembaev A.E. An operator approach to discrete programming with application to Neural Network Modeling // Procceed. of 2nd intern. conf. Ion Fuzzy Systems and Soft Computing (CAFS'96). – Siegen: Germ, 1996. – P. 181-190.
- 98 Авен О.И., Кимельфельд Б.Н., Коган Я.А. Управление многоуровневой памятью в вычислительных системах // Автоматика и телекоммуникация. – 1972. – №11. – С.15-28.
- 99 Поспелов Д.А. Введение в теорию вычислительных систем. – М.: Сов. радио, 1972. – 280 с.
- 100 Ferrari D. A tool for automatic program restructuring // Procceed. ACM nat. conf. – Atlanta, G, 1973. – P. 228-231
- 101 Ngetich M., Otieno C., Kimwele M. A Model for Code Restructuring, A Tool for Improving Systems Quality in Compliance with Object Oriented Coding

Practice // International Journal of Computer Science Issues. – 2019. – Vol. 16, Issue 3. – P. 32-36.

102 Newman C.D., Bartman B., Collard M.L. et al. Simplifying the construction of source code transformations via automatic syntactic restructurings // Journal of Software: Evolution and Process. – 2017. – Vol. 29, Issue 4. – P. 1-18.

103 Sinha M., Jain S. A Survey Paper on Program Restructuring and Component Reuse with Data Mining Technique // International Journal of Engineering Research and General Science. – 2016. – Vol. 4, Issue 3. – P. 578-581.

104 Pallavi A.K., Rane P., Patil S. Code restructuring: Tool for quality Improvement // IJCST. – 2011. – Vol. 2, SP 1. – P. 122-126.

105 Kaur S., Singh P. How does Object-Oriented Code Refactoring Influence Software Quality? Research Landscape and Challenges // Journal of Systems and Software. – 2019. – Vol. 157. – P. 1-42.

106 Oliveira D., Bibiano A.C., Garcia A. On the Customization of Batch Refactoring // Proc. 3rd internat. Workshop on Refactoring (IEEE/ACM). – Montreal, 2019. – P. 13-16.

107 Lung C., Xu X., Zaman M. et al. Program restructuring using clustering techniques // Journal of Systems and Software. – 2006. – Vol. 79, Issue 9. – P. 1261-1279.

108 Alanko T.O., Haikala I.J., Kutvonen P.H. Program restructuring in segmented virtual memory // Performance Evaluation. – 1981. – Vol. 1. – P. 153-169.

109 Zhuravlev Yu.I. Algebras over sets of incorrect (heuristic) algorithms // Kibernetika. – 1978. – Vol. 2. – P. 35-43.

110 Zhuravlev Yu.I. Principles of construction of justification of algorithms for the solution of badly formalized problems // Math. Zametki. – 1978. – Vol. 23, Issue 6. – P. 899-501.

111 Zhuravlev Yu.I. The way to generate recognition algorithms being correct for the given reference sample // Zh. Vychisl. Mat. Mat. Fiz. – 1979. – Vol. 19(3). – P. 200-208.

112 Таненбаум Э.С., Бос Х. Современные операционные системы / пер. с англ. – Изд. 4-е. – СПб.: Питер, 2018. – 1119 с.

113 Denning, P.J. The working set model for program behavior // Communications of the ACM. - 1968. — Vol. 11, Issue. 5. — P. 323-333.

114 Таненбаум Э.С. и др. Операционные системы: разработка и реализация / пер. с англ. – СПб.: Питер, 2007. – 702 с.

115 Hartley S.J. Compile-time program restructuring in multiprogrammed virtual memory systems // IEEE Trans. Software Eng. – 1988. – Vol. 14, №11. – P. 1640-1645.

116 Horspool R.N., Laks J.M. An improved block sequencing method for program restructuring // J. Syst. and Software. – 1983. – №3. – P. 245-250.

117 Kobayashi M. A set of strategy-independent restructuring algorithms // Software Practice and Experience. – 1977. – Vol. 7. – P. 109-116.

- 118 Paris J.F. Program Restructuring in Segmenting Environments // In book: Experimental Computer Performance Evaluation. – Amsterdam: North Holland, 1981. – P. 249-264.
- 119 Lau E.J., Ferrari D. Program restructuring in multilevel virtual memory // IEEE Tr. Software Eng. – 1983. – Vol. SE-9, №1. – P. 69-78.
- 120 Paris J.F., Ferrari D. An analytical study of strategy-oriented restructuring algorithms // Performance Evaluation. – 1984. – №4. – P. 117-132.
- 121 Dyusembaev A.E. Mathematical Models of Program Segmentation. – М.: Nauka, 2001. – 208 p.
- 122 Ferrari D., Lau E. An experiment in program restructuring for performance enhancement // Proceed. 2nd intern. conf. Software Eng. – San Francisco, California, USA, 1976. – P. 146-150.
- 123 Ferrari D. Computer Systems Performance Evaluation. – Ed. 1st. – Hoboken, New Jersey: Prentice Hall, 1978. – 554 p.
- 124 Foulds L. R. Combinatorial Optimization. – Berlin; NY.: Springer Verlag, 1984. – 280 p.
- 125 The Traveling Salesman Problem and its variations / ed. by A. Punnen, G. Gutin. – Dordrecht; Boston; London: Kluwer Academic Publishers, 2002. – 848 p.
- 126 Dyusembaev A.E. Correct models of program segmenting // Journal of Pattern Recognition and Image. – 1993. – Vol. 3, №6. – P. 187-204.
- 127 А.С. 17094. Особенности модели оптимизации работы словаря инноваций при стратегии рабочего множества / А.Е. Дюсембаев, А.Н. Мурзахметов; опубли. 29.04.2021, Бюл. №. 17094.
- 128 Denning P. Working set: Past and Present // IEEE Trans. Software Eng. – 1980. – Vol. Se-6, №1. – P. 64-84.
- 129 Denning P., Sluts D.R. Generalized working sets for segment reference strings // Commun. of ACM. – 1978. – Vol. 21. – P. 750-759.
- 130 Емеличев В.А., Комлик В.Н. Метод построения последовательных планов для решения задач дискретной оптимизации. – М: Наука, 1981. – 208 с.
- 131 Emelichev V.A. Diskrete Optimierung. Sequentielle Lösungsschemata // I. Kibernetika. – 1971. – №6. – P. 109-121.
- 132 Emelichev V.A., Komlik V.I. The solution of discrete programming problems by the construction of a sequence of schemes // Sov. Math. Dokl. – 1969. – Vol. 10. – P. 1098-1100.
- 133 Хачатуров В.Р. Комбинаторные методы решения задач большой размерности. – М.: Наука, 2000. – 360 с.
- 134 Черенин В.П., Хачатуров В.Р. Решение методом последовательных расчетов одного класса задач о размещении производства // Экономико-математические методы. – 1965. – №2. – С. 279-290.
- 135 Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации. – Нижний Новгород: НИНГУ им. Н.И. Лобачевского, 2006. – 136 с.

136 Kochetov Yu.A., Pljasunov A.V. Genetic local search for the graph partitioning problem under cardinality constraints // Zh. Vychisl. Mat. Mat. Fiz. – 2012. – Vol. 52, №1. – P. 164-176.

137 А.С. 14230. Построение модели оптимизации работы словаря инновации на основе стратегий WS / А.Н. Мурзахметов, А.Е. Дюсембаев; опубл. 30.12.2020, Бюл. №14230.

138 А.С. 24565. Информационная система формирования и оптимизации словаря инноваций/ А.Е. Дюсембаев, А.Н. Мурзахметов; опубл. 28.03.2022, Бюл. №24565.

139 А.С. 23780. Построение модели реорганизации групп мультиагентной системы с возможным инфицированием агентов при контактах/ А.Е. Дюсембаев, А.Н. Мурзахметов; опубл. 21.02.2022, Бюл. №23780.

140 А.С. 002334. Модель принятия решений для Казахстанского фондового рынка на основе эконометрического анализа / М.В. Гришко, А.Н. Мурзахметов, А.Е. Дюсембаев; опубл. 29.04.2015.

141 Мурзахметов А.Н., Бапанов А.А. Прогнозирования на основе эконометрических модели // Интеллектуалдық ақпараттық және коммуникациялық технологиялар - «Қазақстан – 2050» стратегиясы аясында үшінші индустриалды революцияны жүзеге асырудың құралы: матер. 3-ші халықар. ғыл.-практ. конфер. – Астана, 2016. – Б. 240-243.

142 Umbetov U.U., Murzakhmetov A.N. Modeling the influence of internet news on the decision-making process of investors // Proceed. of internat. scient.-prac. conf. “Auezov Readings”. – Shymkent, 2017. – P. 318-321.

143 Murzakhmetov A.N. Research Imitation Multiagent Models of the Securities Market. Materials of the II International scientific-practical conference "Integration of the Scientific Community to the Global Challenges of Our Time". – Osaka, 2017. – P. 46-52.

144 Лобанов А.И. Модели клеточных автоматов // Компьютерные исследования и моделирование. – 2010. – Т. 2, №3. – С. 273-293.

145 Тоффоли Т., Марголюс Н. Машины клеточных автоматов / пер. с англ. – М.: Мир, 1991. – 280 с.

146 Ломакин С.Г., Федотов А.М. Анализ модели передачи информации в сети клеточных автоматов // Вестник Новосибирского государственного университета. – 2014. – Т. 12, №3. – С. 86-99.

147 Gorkovenko D.K. Comparison of epidemic models and cellular automata in modeling of diffusion of information in social // St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems. – 2017. – Vol. 10, Issue 3. – P. 103-113.

148 Bratus A.S., Novozhilov A.S., Platonov A.P., Dynamic systems and models of biology. – М.: FizMatlit, 2010. – 400 p.

149 Ляпунов А.А. Биogeоценозы и математическое моделирование // Природа. – 1971. – №10. – С. 38-41.

150 Ghaderi J., Srikant R. Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate // Automatica. – 2014. – Vol. 50, Issue 12. – P. 3209-3215.

151 Borshchev A. The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6. – Chicago: Anylogic north america, 2013. – 612 p.

152 Fedotov A.M., Murzakhmetov A.N., Dyusembaev A.E. et al. To Analysis of the Model of Innovation Diffusion in the Social Systems under the Influence of the Media and Interpersonal Communication // Information. – 2018. – Vol. 21, Issue 3. – P. 1187-1196.

153 Федотов А.М., Ломакин С. Математическое моделирование распространения мнений в обществе // Proceed. internat. conf. «Advanced mathematics, computations and applications-2014». – Novosibirsk, 2014. – P. 78-79.

154 Минаев В.А., Овчинский А.С., Скрыль С.В. и др. Как управлять массовым сознанием: современные модели: монография. – М.: РосНОУ, 2013. – 200 с.

155 Murzakhmetov A.N., Dyusembaev A.E., Umbetov U.U. et al. Study of the innovations diffusion on the base of naming game mathematical model // Compusoft. – 2020. – Vol. 9, Issue 1. – P. 3547-3551.

156 Федотов А.М., Мурзахметов А.Н., Гришко М.В. и др. Моделирование распространения инновации в социально-экономических системах // Известия Национальной академии наук Республики Казахстан. – 2017. – №6(316). – С. 39-44.

157 Мурзахметов А.Н., Бейсов Н.К. Жасушалы автомат кеңістігінде ақпараттың таралуын талдау // ҚазҰТЗУ хабаршысы. – 2019. – №6(136). – Б. 462-465.

158 Fedotov A.M., Murzakhmetov A.N., Dyusembayev A.E. et al. Model of innovation diffusion in the social systems under the influence of the media and interpersonal communication // Proceed. 18th internat. sympos. on Advanced Intelligent Systems (ISIS 2017). – Daegu, 2017. – P. 916-922.

159 Fedotov A.M., Barakhnin V.B., Murzakhmetov A.N. et al. Modelling of process information dissemination and its impact dynamics to mass consciousness // Journal of Theoretical and Applied Information Technology. – 2020. – Vol. 98, №23. – P. 3691-3702.

ҚОСЫМША А

Авторлық құқық куәліктері

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК
2020 жылғы «30» желтоқсан № 14230

Автордың (лардың) жөні, аты, әжесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
МҰРЗАХМЕТОВ АСЛАНБЕК НҮРБЕКОВИЧ, ДЮСЕМБАЕВ АНУАР ЕРМУҚАНОВИЧ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Построение модели оптимизации работы словаря инновации на основе стратегий WS

Объектіні жасаған күні: 29.12.2020





Құжат түпнұсқасының тілдігі <http://www.kazpatent.kz/> сайтының
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)
в разделе «Авторское право» <https://copyright.kazpatent.kz>

ЭЦҚ қол қойылды Оспанов Е.К.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ



РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2021 жылғы «29» сәуір № 17094

Автордың (пардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
ДЮСЕМБАЕВ АНУАР ЕРМУКАНОВИЧ, МУРЗАХМЕТОВ АСЛАНБЕК НУРБЕКОВИЧ

Авторлық құқық объектісі: ЗЕМ-ге арналған бағдарлама

Объектінің атауы: Особенности модели оптимизации работы словаря инноваций при стратегии рабочего множества

Объектіні жасаған күні: 28.04.2021



Құжат түсіндірмесін <http://www.kazpatent.kz/> сайтының
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте kazpatent.kz
в разделе «Авторское право» <https://copyright.kazpatent.kz>

ЭЦҚ қол қойылды

Осламов Е.К.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ



РЕСПУБЛИКА КАЗАХСТАН

**АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ**

КУӘЛІК

2022 жылғы «21» ақпан № 23780

Автордың (лардың) жеңі, аты, әкесінің аты (егер ел және басын қуандыратын құжатта көрсетілсе):
ДЮСЕМБАЕВ АНУАР ЕРМУКАНОВИЧ, Мурзахметов Асламбек Нурбекович

Авторлық құқық объектісі: **ЭЕМ-ге арналған бағдарлама**

Объектінің атауы: **Построение модели реорганизации групп мультиагентной системы с возможным
идентифицированием агентов при контактах**

Объектіні жасаған күні: **18.02.2022**



Құжат тексеруіне қатысты <http://www.kazpatent.kz/ru> сайтының
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)
в разделе «Авторское право» <https://copyright.kazpatent.kz>

ЭЦҚ қол қойылды

Е. Қуантыров

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ



РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

ҚУӘЛІК

2022 жылғы «28» наурыз № 24565

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
ДЮСЕМБАЕВ АНУАР ЕРМУКАНОВИЧ, Мурзахметов Асланбек Нурбекович

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Информационная система формирования и оптимизации словаря инноваций

Объектіні жасаған күні: 25.03.2022



Құжат түпнұсқасының <http://www.kazpatent.kz> сайтының
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)
в разделе «Авторское право» <https://copyright.kazpatent.kz>

ЭЦҚ қол қойылды

А.Естаев



Aston Alliance

ЕВРО-АЗИАТСКИЙ
ДЕПОЗИТАРИЙ
АВТОРСКИХ
ПРОИЗВЕДЕНИЙ
www.avtor-web.com

РОССИЙСКАЯ
НАЦИОНАЛЬНАЯ
БИБЛИОТЕКА

Все авторские произведения депонируются
в Российской Национальной Библиотеке
(г.Санкт-Петербург, Российская Федерация)

СВИДЕТЕЛЬСТВО

о депонировании объектов
интеллектуальной собственности

Регистрационный № 002334

Вид объекта:	Авторское произведение
Название материала:	МОДЕЛЬ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ КАЗАХСТАНСКОГО ФОНДОВОГО РЫНКА НА ОСНОВЕ ЭКОНОМЕТРИЧЕСКОГО АНАЛИЗА
Авторы:	Гришко М.В., Мурзахметов А.Н., Дюсембаев А.Е.
Правообладатель:	Гришко М.В., Мурзахметов А.Н., Дюсембаев А.Е.
Адрес публикации в Интернете:	http://avtor-web.com/index.php?option=com_desposition &task=display_desp_det&id=2478&lang=ru
Дата депонирования (создания)	29 апреля 2015 г.



Уполномоченный оператор
в странах СНГ и Евразии
Евро-азиатского электронного
депозитария "AVTOR"

С.Муминов

ООО "Астон Альянс"
(Санкт-Петербург, РФ)

Aston Alliance
www.aston-alliance.com

Юридическая информация: Владельцем Евро-Азиатского Электронного Депозитария "AVTOR" является Общество с ограниченной ответственностью "Астон Альянс" (Санкт-Петербург, ОГРН 1137847065344, Российская Федерация), оказывающее услуги по СНГ, Европе и Азии.

Деятельность депозитария и подтверждение авторских прав осуществляется в соответствии с нормами Бернской конвенции об охране литературных и художественных произведений.

В соответствии со статьей 6 Закона Республики Казахстан от 10 июня 1996 года № 6-І "Об авторском праве и смежных правах" Евро-азиатский депозитарий "AVTOR" обеспечивает подтверждение факта создания авторских произведений по следующим критериям:

1. Являющихся результатом творческой деятельности.
2. Выраженных (существующих) в объективной форме.

ҚОСЫМША Ә

Енгізу актісі



«УТВЕРЖДАЮ»

Врио директора Федерального исследовательского центра
информационных и вычислительных технологий

к.ф.-м.п.  А.В. Юрченко

« 8 » 02 2021 г.

АКТ

о внедрении результатов диссертационного исследования

Настоящий акт подтверждает, что научные и практические результаты, полученные в ходе выполнения диссертации на соискание ученой степени доктора PhD по специальности 6D070300-«Информационные системы» докторанта Казахского национального университета им. аль-Фараби (Алматы, Казахстан) внедрены в Федеральном исследовательском центре информационных и вычислительных технологий (Новосибирск, Россия). Разработанная А.Н. Мурзахметовым (при участии А.М.Федотова, В.Б.Барахнина, А.Е.Дюсембаева) информационная модель дала возможность анализировать процесс формирования и распространения инновации в сообществе и её влияние на массовое сознание. Внедрение результатов диссертационного исследования, также могут быть применены в биологических, экономических и социально-психологических исследованиях ФИЦ ИВТ и других организаций.

Ведущий научный сотрудник ФИЦ ИВТ
д.т.н.



В.Б. Барахнин

ДБ АО «Сбер Банк»
Тараз, 080000, проспект Абая, 120
Телефон: 595910, +77787549600
E-mail: taraz@sberbank.kz
Адрес в интернете:
<http://www.sberbank.kz>

АКТ
о внедрении результатов диссертационного исследования

Настоящим подтверждаю, что результаты диссертационного исследования доктора PhD по специальности 6D070300-«Информационные системы» докторанта Казахского национального университета им. аль-Фараби внедрены в банковскую систему АО «Сбер Банк» для сбора и анализа распространения информации и дезинформации в сообществе. Полученные результаты позволили анализировать факторы внешней среды, влияющих на конъюнктуру, экономических, политических, технологических и т.д. для поддержки принятия решения и в систему маркетинга.

АО «Сбер Банк» выражает свою признательность Мурзахметову Асланбеку Нурбековичу за предоставленную возможность практического применения результатов диссертационного исследования и надеется на дальнейшее активное сотрудничества с нашим банком.

Директор
ДБ АО «Сбер Банк»



_____ Д.Б. Акшалов

«06» _____ 21 _____ 2021 г.

ҚОСЫМША Б

Бағдарлама листингі

```
//Фрагмент исходного кода программы в Java
package diffusion_innovation;

import java.io.Serializable;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayDeque;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.Currency;
import java.util.Date;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.LinkedHashSet;
import java.util.LinkedList;
import java.util.List;
import java.util.ListIterator;
import java.util.Locale;
import java.util.Map;
import java.util.PriorityQueue;
import java.util.Random;
import java.util.Set;
import java.util.SortedMap;
import java.util.SortedSet;
import java.util.Stack;
import java.util.Timer;
import java.util.TreeMap;
import java.util.TreeSet;
import java.util.Vector;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics2D;
import java.awt.geom.AffineTransform;
import static java.lang.Math.*;

import java.awt.geom.Arc2D;

public class Main extends Agent
{
    // Параметры

    public
    double p;

    /**
     * Возвращает значение по умолчанию параметра <code>p</code>.
     * <i>Пользователь не должен вызывать этот метод</i>

```



```

    */

    public double _p_DefaultValue_xjal() {
        final Main self = this;
        return
0.6
;
    }

    public void set_p( double p ) {
        if (p == this.p) {
            return;
        }
        double _oldValue_xjal = this.p;
        this.p = p;
        onChange_p_xjal( _oldValue_xjal );
        onChange();
    }

    /**
     * Calls "On change" action for parameter p.<br>
     * Note that 'oldValue' in that action will be unavailable if this method is called
by user
     * (current parameter value will be passed as 'oldValue').<br>
     * Please call <code>set_p()</code> method instead.
    */
    protected void onChange_p() {
        onChange_p_xjal( p );
    }

    @AnyLogicInternalCodegenAPI
    protected void onChange_p_xjal( double oldValue ) {
    }

    public
double k0;

    /**
     * Возвращает значение по умолчанию параметра <code>k0</code>.
     * <i>Пользователь не должен вызывать этот метод</i>
    */
    @AnyLogicInternalCodegenAPI
    public double _k0_DefaultValue_xjal() {
        final Main self = this;
        return
0.4
;
    }

    public void set_k0( double k0 ) {
        if (k0 == this.k0) {
            return;
        }
        double _oldValue_xjal = this.k0;
        this.k0 = k0;
        onChange_k0_xjal( _oldValue_xjal );
        onChange();
    }

    /**

```

```

    * Calls "On change" action for parameter k0.<br>
    * Note that 'oldValue' in that action will be unavailable if this method is called
by user
    * (current parameter value will be passed as 'oldValue').<br>
    * Please call <code>set_k0()</code> method instead.
    */
protected void onChange_k0() {
    onChange_k0_xjal( k0 );
}

@AnyLogicInternalCodegenAPI
protected void onChange_k0_xjal( double oldValue ) {

public
double n;

/**
 * Возвращает значение по умолчанию параметра <code>n</code>.
 * <i>Пользователь не должен вызывать этот метод</i>
 */
@AnyLogicInternalCodegenAPI
public double _n_DefaultValue_xjal() {
    final Main self = this;
    return
2
;
}

public void set_n( double n ) {
    if (n == this.n) {
        return;
    }
    double _oldValue_xjal = this.n;
    this.n = n;
    onChange_n_xjal( _oldValue_xjal );
    onChange();
}

/**
 * Calls "On change" action for parameter n.<br>
 * Note that 'oldValue' in that action will be unavailable if this method is called
by user
 * (current parameter value will be passed as 'oldValue').<br>
 * Please call <code>set_n()</code> method instead.
 */
protected void onChange_n() {
    onChange_n_xjal( n );
}

@AnyLogicInternalCodegenAPI
protected void onChange_n_xjal( double oldValue ) {

public
double k1;

/**
 * Возвращает значение по умолчанию параметра <code>k1</code>.

```

```

    * <i>Пользователь не должен вызывать ЭТОТ МЕТОД</i>
    */
    @AnyLogicInternalCodegenAPI
    public double _k1_DefaultValue_xjal() {
        final Main self = this;
        return
2
;
    }

    public void set_k1( double k1 ) {
        if (k1 == this.k1) {
            return;
        }
        double _oldValue_xjal = this.k1;
        this.k1 = k1;
        onChange_k1_xjal( _oldValue_xjal );
        onChange();
    }

    /**
     * Calls "On change" action for parameter k1.<br>
     * Note that 'oldValue' in that action will be unavailable if this method is called
by user
     * (current parameter value will be passed as 'oldValue').<br>
     * Please call <code>set_k1()</code> method instead.
     */
    protected void onChange_k1() {
        onChange_k1_xjal( k1 );
    }

    @AnyLogicInternalCodegenAPI
    protected void onChange_k1_xjal( double oldValue ) {
}

    public
double k2;

    /**
     * Возвращает значение по умолчанию параметра <code>k2</code>.
     * <i>Пользователь не должен вызывать ЭТОТ МЕТОД</i>
     */
    @AnyLogicInternalCodegenAPI
    public double _k2_DefaultValue_xjal() {
        final Main self = this;
        return
0.2
;
    }

    public void set_k2( double k2 ) {
        if (k2 == this.k2) {
            return;
        }
        double _oldValue_xjal = this.k2;
        this.k2 = k2;
        onChange_k2_xjal( _oldValue_xjal );
        onChange();
    }
}

```

```

/**
 * Calls "On change" action for parameter k2.<br>
 * Note that 'oldValue' in that action will be unavailable if this method is called
by user
 * (current parameter value will be passed as 'oldValue').<br>
 * Please call <code>set_k2()</code> method instead.
 */
protected void onChange_k2() {
    onChange_k2_xjal( k2 );
}

@AnyLogicInternalCodegenAPI
protected void onChange_k2_xjal( double oldValue ) {

public
double g;

/**
 * Возвращает значение по умолчанию параметра <code>g</code>.
 * <i>Пользователь не должен вызывать этот метод</i>
 */
@AnyLogicInternalCodegenAPI
public double _g_DefaultValue_xjal() {
    final Main self = this;
    return
0.03
;
}

public void set_g( double g ) {
    if (g == this.g) {
        return;
    }
    double _oldValue_xjal = this.g;
    this.g = g;
    onChange_g_xjal( _oldValue_xjal );
    onChange();
}

/**
 * Calls "On change" action for parameter g.<br>
 * Note that 'oldValue' in that action will be unavailable if this method is called
by user
 * (current parameter value will be passed as 'oldValue').<br>
 * Please call <code>set_g()</code> method instead.
 */
protected void onChange_g() {
    onChange_g_xjal( g );
}

@AnyLogicInternalCodegenAPI
protected void onChange_g_xjal( double oldValue ) {

public
double N;

/**

```

```

    * Возвращает значение по умолчанию параметра <code>N</code>.
    * <i>Пользователь не должен вызывать этот метод</i>
    */
    @AnyLogicInternalCodegenAPI
    public double _N_DefaultValue_xjal() {
        final Main self = this;
        return
100000
    ;
    }

    public void set_N( double N ) {
        if ( N == this.N ) {
            return;
        }
        double _oldValue_xjal = this.N;
        this.N = N;
        onChange_N_xjal( _oldValue_xjal );
        onChange();
    }

    /**
    * Calls "On change" action for parameter N.<br>
    * Note that 'oldValue' in that action will be unavailable if this method is called
by user
    * (current parameter value will be passed as 'oldValue').<br>
    * Please call <code>set_N()</code> method instead.
    */
    protected void onChange_N() {
        onChange_N_xjal( N );
    }

    @AnyLogicInternalCodegenAPI
    protected void onChange_N_xjal( double oldValue ) {
    }

    @Override
    public void setParametersToDefaultValues() {
        super.setParametersToDefaultValues();
        p = _p_DefaultValue_xjal();
        k0 = _k0_DefaultValue_xjal();
        n = _n_DefaultValue_xjal();
        k1 = _k1_DefaultValue_xjal();
        k2 = _k2_DefaultValue_xjal();
        g = _g_DefaultValue_xjal();
        N = _N_DefaultValue_xjal();
    }

    @Override
    public boolean setParameter(String _name_xjal, Object _value_xjal, boolean
_callOnChange_xjal) {
        switch ( _name_xjal ) {
            case "p":
                if ( _callOnChange_xjal ) {
                    set_p( ((Number) _value_xjal).doubleValue() );
                } else {
                    p = ((Number) _value_xjal).doubleValue();
                }
                return true;
            case "k0":

```

```

        if ( _callOnChange_xjal ) {
            set_k0( ((Number) _value_xjal).doubleValue() );
        } else {
            k0 = ((Number) _value_xjal).doubleValue();
        }
        return true;
    case "n":
        if ( _callOnChange_xjal ) {
            set_n( ((Number) _value_xjal).doubleValue() );
        } else {
            n = ((Number) _value_xjal).doubleValue();
        }
        return true;
    case "k1":
        if ( _callOnChange_xjal ) {
            set_k1( ((Number) _value_xjal).doubleValue() );
        } else {
            k1 = ((Number) _value_xjal).doubleValue();
        }
        return true;
    case "k2":
        if ( _callOnChange_xjal ) {
            set_k2( ((Number) _value_xjal).doubleValue() );
        } else {
            k2 = ((Number) _value_xjal).doubleValue();
        }
        return true;
    case "g":
        if ( _callOnChange_xjal ) {
            set_g( ((Number) _value_xjal).doubleValue() );
        } else {
            g = ((Number) _value_xjal).doubleValue();
        }
        return true;
    case "N":
        if ( _callOnChange_xjal ) {
            set_N( ((Number) _value_xjal).doubleValue() );
        } else {
            N = ((Number) _value_xjal).doubleValue();
        }
        return true;
    default:
        return super.setParameter( _name_xjal, _value_xjal, _callOnChange_xjal );
    }
}

```

```

@Override
public <T> T getParameter(String _name_xjal) {
    Object _result_xjal;
    switch ( _name_xjal ) {
        case "p": _result_xjal = p; break;
        case "k0": _result_xjal = k0; break;
        case "n": _result_xjal = n; break;
        case "k1": _result_xjal = k1; break;
        case "k2": _result_xjal = k2; break;
        case "g": _result_xjal = g; break;
        case "N": _result_xjal = N; break;
        default: _result_xjal = super.getParameter( _name_xjal ); break;
    }
    return (T) _result_xjal;
}

```

```

@AnyLogicInternalCodegenAPI
private static String[] _parameterNames_xjal;

@Override
public String[] getParameterNames() {
    String[] result = _parameterNames_xjal;
    if (result == null) {
        List<String> list = new ArrayList<>( Arrays.asList( super.getParameterNames() )
);
        list.add( "p" );
        list.add( "k0" );
        list.add( "n" );
        list.add( "k1" );
        list.add( "k2" );
        list.add( "g" );
        list.add( "N" );
        result = list.toArray( new String[ list.size() ] );
        _parameterNames_xjal = result;
    }
    return result;
}

// Динамические переменные

public double SY;

public double YS;

public double a;

public double M;

public double b;

public double S;

public double Y;
/**
 * Записывает переменные модели в заданные массивы
 */
@AnyLogicInternalCodegenAPI
public void getPhaseVector_xjal(double[] _d_xjal, int _d_idx_xjal, double[] _a_xjal,
int _a_idx_xjal) {
    _d_xjal[ _d_idx_xjal + 0 ] = S;
    _d_xjal[ _d_idx_xjal + 1 ] = Y;
    _d_idx_xjal += 2;
    super.getPhaseVector_xjal(_d_xjal, _d_idx_xjal, _a_xjal, _a_idx_xjal);
}
/**
 * Записывает заданные массивы в переменные модели
 */
@AnyLogicInternalCodegenAPI
public void putPhaseVector_xjal(double[] _d_xjal, int _d_idx_xjal, double[] _a_xjal,
int _a_idx_xjal) {
    S = _d_xjal[ _d_idx_xjal + 0 ];
    Y = _d_xjal[ _d_idx_xjal + 1 ];
    _d_idx_xjal += 2;
    super.putPhaseVector_xjal(_d_xjal, _d_idx_xjal, _a_xjal, _a_idx_xjal);
}
/**

```

```

    * Эта функция получает фазовый вектор из начальных условий (используется при
    разрешении алгебраических циклов в начальных условиях)
    */
    @AnyLogicInternalCodegenAPI
    public void getPhaseVectorForInitialConditions_xjal(double[] _a_xjal, int
    _a_idx_xjal) {

        _a_xjal[ _a_idx_xjal + 0 ] = S;
        _a_idx_xjal += 1;
        super.getPhaseVectorForInitialConditions_xjal(_a_xjal, _a_idx_xjal);
    }

    /**
    * Эта функция помещает фазовый вектор в начальные условия (используется при
    разрешении алгебраических циклов в начальных условиях)
    */
    @AnyLogicInternalCodegenAPI
    public void putPhaseVectorForInitialConditions_xjal(double[] _a_xjal, int
    _a_idx_xjal) {
        S = _a_xjal[ _a_idx_xjal + 0 ];
        _a_idx_xjal += 1;
        super.putPhaseVectorForInitialConditions_xjal(_a_xjal, _a_idx_xjal);
    }

    @AnyLogicInternalCodegenAPI
    public void assignInitialConditions_xjal() {
        super.assignInitialConditions_xjal();
        _assign_a_Formula_xjal();
        _assign_M_Formula_xjal();
        _assign_b_Formula_xjal();
        _assign_Y_Formula_xjal();
        _assign_SY_Formula_xjal();
        _assign_YS_Formula_xjal();
    }

    @AnyLogicInternalCodegenAPI
    public void setupInitialConditions_xjal(Class<?> callerClass) {
        if (callerClass != Main.class) {
            return;
        }
        if (getInitialAlgebraicFlatEquationsCount_xjal() > 0) {
            SDIntegrationManager integrationManagerForInitialConditions = new
            SDIntegrationManager( 0, getInitialAlgebraicFlatEquationsCount_xjal(),
            getInitialFormulaFlatEquationsCount_xjal() );
            integrationManagerForInitialConditions.doStep( this, 0, 0.1, true );
        } else {
            assignInitialConditions_xjal();
        }
    }

    @AnyLogicInternalCodegenAPI
    public void _assign_a_Formula_xjal() {
        a =
    p * k0 * n
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void _assign_M_Formula_xjal() {
        M =
    0.3
    ;
    }

```



```

    }
    @AnyLogicInternalCodegenAPI
    public void _assign_b_Formula_xjal() {
        b =
k1 * k2
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void _assign_Y_Formula_xjal() {
        Y =
3
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void _assign_SY_Formula_xjal() {
        SY =
(S * (a * Y + M * b)) / N
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void _assign_YS_Formula_xjal() {
        YS =
g * Y
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void formulasExecute_xjal() {
        super.formulasExecute_xjal();
        _assign_a_Formula_xjal();
        _assign_M_Formula_xjal();
        _assign_b_Formula_xjal();
        _assign_SY_Formula_xjal();
        _assign_YS_Formula_xjal();
    }

    @AnyLogicInternalCodegenAPI
    public void getRightPart_xjal( double[] _d_xjal, int _d_idx_xjal, double[] _a_xjal,
int _a_idx_xjal ) {
        _get_S_Derivative_xjal( _d_xjal, _d_idx_xjal + 0 );
        _get_Y_Derivative_xjal( _d_xjal, _d_idx_xjal + 1 );
        _d_idx_xjal += 2;
        super.getRightPart_xjal( _d_xjal, _d_idx_xjal, _a_xjal, _a_idx_xjal );
    }

    @AnyLogicInternalCodegenAPI
    public void getRightPartForInitialConditions_xjal( double[] _a_xjal, int _a_idx_xjal
) {
        _get_S_Equation_xjal( _a_xjal, _a_idx_xjal + 0 );
        _a_idx_xjal += 1;
        super.getRightPartForInitialConditions_xjal( _a_xjal, _a_idx_xjal );
    }

    @AnyLogicInternalCodegenAPI
    public void _get_S_Derivative_xjal( double[] _d_xjal, int _d_idx_xjal ) {
        _d_xjal[ _d_idx_xjal ] =
+YS - SY
    ;
    }
    @AnyLogicInternalCodegenAPI
    public void _get_Y_Derivative_xjal( double[] _d_xjal, int _d_idx_xjal ) {
        _d_xjal[ _d_idx_xjal ] =

```

```

+SY - YS
;
}

/**
 * Идентификатор группы верхнего уровня
 */
@AnyLogicInternalCodegenAPI
protected static final int _icon = -1;

@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody3_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody3_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody3_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody4_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody4_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody4_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody5_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody5_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody5_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI

```

```

protected static final double[] _shapeBody_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody1_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody1_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody1_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody2_pointsDX_xjal() {
    return new double[] { 0.0, 0.0, 3.0, -1.0, -2.0, -2.0, -2.0, 0.0, -1.0, 0.0, 1.0,
3.0, 4.0, 5.0, 4.0, 6.0, 6.0, 6.0, 5.0, 1.0, 4.0, 4.0, 4.0, 0.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody2_pointsDY_xjal() {
    return new double[] { 0.0, 4.0, 2.0, 3.0, 3.0, 10.0, 11.0, 11.0, 9.0, 17.0, 17.0,
17.0, 17.0, 9.0, 11.0, 11.0, 10.0, 3.0, 3.0, 2.0, 4.0, 0.0, -3.0, -3.0, };
}
@AnyLogicInternalCodegenAPI
protected static final double[] _shapeBody2_pointsDZ_xjal() {
    return new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, };
}
}

```

```

@Override
@AnyLogicInternalCodegenAPI
public void executeShapeControlAction( int _shape, int index, String value ) {
    switch( _shape ) {
        case _editbox:
            set_p( Double.parseDouble(value) );
            break;
        case _editbox1:
            set_g( Double.parseDouble(value) );
            break;
        case _editbox3:
            set_N( Double.parseDouble(value) );
            break;
        case _editbox4:
            set_k2( Double.parseDouble(value) );
            break;
        case _editbox5:
            set_k1( Double.parseDouble(value) );
            break;
        case _editbox6:

```

```

        set_n( Double.parseDouble(value) );
        break;
    case _editbox7:
        set_k0( Double.parseDouble(value) );
        break;
    default:
        super.executeShapeControlAction( _shape, index, value );
        break;
    }
}

    @Override
    @AnyLogicInternalCodegenAPI
    public void action(){
        executeShapeControlAction( _editbox4, 0, value );
    }

    @Override
    public void setValueToDefault() {
        setText( getShapeControlDefaultValueString( _editbox4, 0 ) );
    }
};

editbox5 = new ShapeTextField(
    Main.this, true, -10.0, 60.0,
    50.0, 30.0,
    controlDefault, controlDefault, true, _editbox5_Font,
    ShapeControl.TYPE_DOUBLE, getShapeControlMinimum( editbox5 ),
getShapeControlMaximum( editbox5 ) ) {
    @Override
    public void updateDynamicProperties(boolean publicOnly) {
        _editbox5_SetDynamicParams_xjal( this );
        super.updateDynamicProperties(publicOnly);
    }

    @Override
    @AnyLogicInternalCodegenAPI
    public void action(){
        executeShapeControlAction( _editbox5, 0, value );
    }

    @Override
    public void setValueToDefault() {
        setText( getShapeControlDefaultValueString( _editbox5, 0 ) );
    }
};

editbox6 = new ShapeTextField(
    Main.this, true, -10.0, 20.0,
    50.0, 30.0,
    controlDefault, controlDefault, true, _editbox6_Font,
    ShapeControl.TYPE_DOUBLE, getShapeControlMinimum( editbox6 ),
getShapeControlMaximum( editbox6 ) ) {
    @Override
    public void updateDynamicProperties(boolean publicOnly) {
        _editbox6_SetDynamicParams_xjal( this );
        super.updateDynamicProperties(publicOnly);
    }

    @Override
    @AnyLogicInternalCodegenAPI

```

```

public void action(){
    executeShapeControlAction( _editbox6, 0, value );
}

@Override
public void setValueToDefault() {
    setText( getShapeControlDefaultValueString( _editbox6, 0 ) );
}
};

editbox7 = new ShapeTextField(
    Main.this, true, -10.0, -20.0,
    50.0, 30.0,
    controlDefault, controlDefault, true, _editbox7_Font,
    ShapeControl.TYPE_DOUBLE, getShapeControlMinimum( editbox7 ),
getShapeControlMaximum( editbox7 ) ) {
    @Override
    public void updateDynamicProperties(boolean publicOnly) {
        _editbox7_SetDynamicParams_xjal( this );
        super.updateDynamicProperties(publicOnly);
    }

    @Override
    @AnyLogicInternalCodegenAPI
    public void action(){
        executeShapeControlAction( _editbox7, 0, value );
    }

    @Override
    public void setValueToDefault() {
        setText( getShapeControlDefaultValueString( _editbox7, 0 ) );
    }
};

text1 = new ShapeText(
    SHAPE_DRAW_2D, true, 290.0, 150.0, 0.0, 0.0,
    black, "Темп увеличения \r\nколичества агентов, \r\nппринявших инновацию",
    _text1_Font, ALIGNMENT_LEFT );

text2 = new ShapeText(
    SHAPE_DRAW_2D, true, 30.0, 250.0, 0.0, 0.0,
    black, "Количество агентов, \r\nспособных принять \r\nпинновацию",
    _text2_Font, ALIGNMENT_LEFT );

text3 = new ShapeText(
    SHAPE_DRAW_2D, true, 540.0, 250.0, 0.0, 0.0,
    black, "Количество агентов, \r\nппринявших инновацию",
    _text3_Font, ALIGNMENT_LEFT );

text12 = new ShapeText(
    SHAPE_DRAW_2D, false, 320.0, 20.0, 0.0, 0.0,
    black, "Темп уменьшения количества агентов, \r\nппринявших инновацию",
    _text12_Font, ALIGNMENT_LEFT );

shapeBody3 = new ShapeCurve(
    true, -2.0, -7.0,
    null, black,
    24, true, _shapeBody3_pointsDX_xjal(), _shapeBody3_pointsDY_xjal(),
    true, 1.0, LINE_STYLE_SOLID );

shapeBody4 = new ShapeCurve(

```

```

true,-2.0, -7.0,
null, black,
24, true, _shapeBody4_pointsDX_xjal(), _shapeBody4_pointsDY_xjal(),
true, 1.0, LINE_STYLE_SOLID );

shapeBody5 = new ShapeCurve(
true,-2.0, -7.0,
null, black,
24, true, _shapeBody5_pointsDX_xjal(), _shapeBody5_pointsDY_xjal(),
true, 1.0, LINE_STYLE_SOLID );

shapeBody = new ShapeCurve(
true,-2.0, -7.0,
null, black,
24, true, _shapeBody_pointsDX_xjal(), _shapeBody_pointsDY_xjal(),
true, 1.0, LINE_STYLE_SOLID );

shapeBody1 = new ShapeCurve(
true,-2.0, -7.0,
null, black,
24, true, _shapeBody1_pointsDX_xjal(), _shapeBody1_pointsDY_xjal(),
true, 1.0, LINE_STYLE_SOLID );

shapeBody2 = new ShapeCurve(
true,-2.0, -7.0,
null, black,
24, true, _shapeBody2_pointsDX_xjal(), _shapeBody2_pointsDY_xjal(),
true, 1.0, LINE_STYLE_SOLID );

text13 = new ShapeText(
SHAPE_DRAW_2D, true,40.0, 70.0, 0.0, 0.0,
black,"Максимально возможное \r\nколичество агентов, \r\nспособных принять
инновацию",
_text13_Font, ALIGNMENT_LEFT );

text14 = new ShapeText(
SHAPE_DRAW_2D, true,110.0, 290.0, 0.0, 0.0,
black,"Вероятность принятия инновации \r\nв межличностном контакте",
_text14_Font, ALIGNMENT_LEFT );

text15 = new ShapeText(
SHAPE_DRAW_2D, true,20.0, 420.0, 0.0, 0.0,
black,"Вероятность \r\nкоммуникации по \r\nтеме инновации",
_text15_Font, ALIGNMENT_LEFT );

text16 = new ShapeText(
SHAPE_DRAW_2D, true,140.0, 420.0, 0.0, 0.0,
black,"Вероятность \r\nпринятия \r\nинновации при \r\nодном контакте ",
_text16_Font, ALIGNMENT_LEFT );

text17 = new ShapeText(
SHAPE_DRAW_2D, true,260.0, 420.0, 0.0, 0.0,
black,"Количество контактов \r\nагентов, принявших \r\nинновацию (в день)",
_text17_Font, ALIGNMENT_LEFT );

text18 = new ShapeText(
SHAPE_DRAW_2D, true,260.0, 320.0, 0.0, 0.0,
black,"Коэффициент массовости \r\nи регулярности СМИ ",
_text18_Font, ALIGNMENT_LEFT );

text19 = new ShapeText(

```

```

        SHAPE_DRAW_2D, true,400.0, 420.0, 0.0, 0.0,
        black,"Среднее количество \r\познакомлений с одним \r\псообщением в СМИ",
        _text19_Font, ALIGNMENT_LEFT );

text20 = new ShapeText(
        SHAPE_DRAW_2D, true,550.0, 420.0, 0.0, 0.0,
        black,"Вероятность принятия инноваций\r\ппод воздействием СМИ",
        _text20_Font, ALIGNMENT_LEFT );

text4 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, 110.0, 0.0, 0.0,
        black,"k2",
        _text4_Font, ALIGNMENT_LEFT );

text5 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, 70.0, 0.0, 0.0,
        black,"k1",
        _text5_Font, ALIGNMENT_LEFT );

text6 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, 30.0, 0.0, 0.0,
        black,"n",
        _text6_Font, ALIGNMENT_LEFT );

text7 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, -10.0, 0.0, 0.0,
        black,"k0",
        _text7_Font, ALIGNMENT_LEFT );

text8 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, -50.0, 0.0, 0.0,
        black,"p",
        _text8_Font, ALIGNMENT_LEFT );

text9 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, -90.0, 0.0, 0.0,
        black,"g",
        _text9_Font, ALIGNMENT_LEFT );

text11 = new ShapeText(
        SHAPE_DRAW_2D, true,-40.0, -130.0, 0.0, 0.0,
        black,"N",
        _text11_Font, ALIGNMENT_LEFT );

text = new ShapeText(
        SHAPE_DRAW_2D, true,520.0, 300.0, 0.0, 0.0,
        black,"Вероятность принятия \r\пинновации в результате\r\пвоздействие СМИ",
        _text_Font, ALIGNMENT_LEFT );

text10 = new ShapeText(
        SHAPE_DRAW_2D, true,130.0, 30.0, 0.0, 0.0,
        black,"Вероятность забывания \r\пинновации",
        _text10_Font, ALIGNMENT_LEFT );

}

```