

УДК 004.82+007.52

РАЗВИТИЕ МИВАРНЫХ СЕТЕЙ И ПРОДУКЦИЙ НА ОСНОВЕ ТРЕХДОЛЬНЫХ И МНОГОДОЛЬНЫХ ГРАФОВ ДЛЯ РЕАЛИЗАЦИИ ПРАВИЛ ВЫБОРА "ЕСЛИ..., ТО..., ИНАЧЕ..."

Варламов О.О., Санду Р.А., Владимиров А.Н.

МАДИ (ГТУ), ФГУП НИИР, Москва

Аннотация. Миварные сети и продукции используются для создания различных экспертных систем и Многомерных эволюционных прикладных автоматизированных информационных систем (МЭПАИС) поддержки принятия решений. Для расширения возможностей по применению этих формализмов проведен анализ существующих подходов и предложены новые пути развития миварных сетей на основе трехдольных и многодольных графов для реализации правил выбора вида "Если..., то..., иначе..." для эволюционной обработки логических правил и вычислительных процедур в реальном времени.

Ключевые слова: мивар, экспертные системы, продукции, логические правила, вычислительные процедуры, многодольные графы.

Введение

Методы создания интеллектуальных систем нашли широкое практическое применение в современных автоматизированных системах. Особую роль играет моделирование рассуждений [1], применение методов дискретной математики [2], миварных баз данных и правил [3], формирование пространства знаний в сети Интернет и Semantic Web [4]. Известная многомерная эволюционная прикладная автоматизированная информационная система (МЭПАИС) поддержки принятия решений применяется в различных предметных областях и основана на миварных логических сетях и использовании линейного матричного метода определения маршрута логического вывода на адаптивной сети правил [5]. Например, она используется для управления инновационными ресурсами (УИР) химической и нефтехимической промышленности (ХНП) России. МЭПАИС УИР ХНП - это экспертная система реального времени для управления инновациями в ХНП России.

Проведенный анализ возможностей миварного подхода и продукции показал, что существуют определенные ограничения для их применения в современных информационных системах [1-5]. Например, существовало такое ограничение в миварных логических сетях, что нельзя было использовать правила выбора, когда только часть выходных переменных получает значения. Формально это можно записать в виде: "если..., то ..., иначе...". Конечно, и в формате традиционных правил "если ..., то ..." можно составить описание предметной области, хотя оно будет более

сложным и потребует значительно большего количества "элементарных правил". При проведении вычислительных экспериментов на многопроцессорных вычислительных комплексах ФГУП НИИР определено, что превышение количества свыше десяти тысяч правил или переменных становится критичным для решения задач в реальном времени и требует разработки новых решений [6-8].

Таким образом, развитие миварного подхода и продукций является актуальной и практически важной задачей.

Логико-вычислительная обработка информации и базы данных

Для решения многих практических задач применения информационных систем и процессов, требуется проводить как логическую, так и вычислительную обработку данных. Исторически так сложилось, что области логического вывода и вычислительной обработки развивались самостоятельно и успешно решали различные классы задач. В некотором смысле, даже существовало противоречие между этими подходами [1-9]. Кроме того, разделяли проблемы обработки и хранения различных данных.

Базы данных преимущественно использовались только для хранения и поиска требуемых данных, а системы логического вывода и вычислений применялись для обработки информации, поиска решений и т.п. Получалось, что эти области относительно слабо пересекались, хотя в плане перспектив развития в каждой из них регулярно провозглашались цели объединения всех функций по накоплению и обработке информации в одной системе [1-15].

Если проводить аналогию с человеком, то наш разум одновременно накапливает и хранит данные, комплексно решает и логические, и вычислительные, и логико-вычислительные задачи. Миварный подход позволяет в едином формализме проводить и эволюционное накопление данных в миварном информационном пространстве, и выполнять совмещенную логико-вычислительную обработку в миварных логических сетях.

Миварный метод логико-вычислительной обработки позволяет решать большой класс сложных научных и практических задач. Прежде всего, необходимо провести анализ существовавших ранее подходов к решению различных классов задач и оценить их ограничения. Затем перейдем к анализу проблем, достижений и перспектив в области баз данных и миварном информационном пространстве унифицированного представления данных и правил.

Возможности и ограничения продукционного подхода

Миварный подход и создание логико-вычислительных сетей позволил эффективно реализовать предложенные для продукций Пospelовым Д.А. *методы прямой и обратной волны* [1, стр. 85]. Отметим, что эти работы не нашли широкого практического применения, т.к.

исходили из предпосылки, что любой вывод носит переборный характер и имеет место факториальный рост вычислительной сложности при увеличении количества переменных. Даже современные суперкомпьютеры при таком подходе позволяют обрабатывать в реальном режиме времени лишь не более 20 правил или переменных. Фактически, это позволяет решать только "игрушечные" модельные задачи, т.к. современные требования начинаются с сотен и достигают десятков тысяч переменных даже для относительно простых предметных областей. Для анализа проблем метода логико-вычислительной обработки данных и его возможных применениях очень важным является следующее заключение Пospelова Д.А.: "Мы хотим отметить, что ядром всех основных типов рассмотренных интеллектуальных систем являются база знаний и блок, осуществляющий вывод с помощью знаний (решатель, планировщик или логический блок). Этот вывод составляет основную процедуру, реализуемую в интеллектуальных системах" [1, стр. 129].

"Продукции в общем виде можно записать в форме "Если..., то...". Сама по себе эта форма оказывается весьма характерной для фиксации знаний в различных областях человеческой деятельности ... **запись знаний в виде систем продукций носит универсальный характер - любые знания можно записать в такой форме**" [1, стр. 129]. Отметим, что Кузнецов О.П. в [2, стр. 282-283] под продуктами понимает множество правил вывода в канонических системах (системах продукций Поста), в которых есть посылки и следствия. Следовательно, применение продукционного подхода для логико-вычислительной обработки разнообразных данных является целесообразным. Существуют различные конструкции продукций. В наиболее общем виде "вместо продукций типа $A \Rightarrow B$ рассматривают более сложные конструкции. В общей форме продукции имеют вид:

$$i, P, A \Rightarrow B, Q.$$

Здесь $A \Rightarrow B$ - обычная продукция "Если ..., то ...", которая носит название *ядра продукции*. Элемент P характеризует внешние условия или *условия применимости* продукции, определяемые факторами, не входящими непосредственно в A , например целями, которые стоят перед рассуждающей системой. Условия P позволяют из всех продукций, у которых в левой части ядра стоит A , отбирать нужную часть продукций. Элемент P характеризует *сферу проблемной области* базы знаний или *предусловия применимости* продукции. Эти предусловия ничем не отличаются от P , но выделяют подсистемы продукций на ранг выше тех, которые выделяют условия. Предусловия задают формальную систему, в рамках которой будут проводиться логические рассуждения... Наконец, Q характеризует *постусловия* продукции, указывающие на те изменения, которые необходимо внести в базу знаний и в систему продукций после реализации данной продукции" [1, стр. 134-135].

Отметим, что существует аналогичный подход, основанный на гиперправилах с мультиактивизаторами [3]. Хорошевский В.Ф. в

[4, стр. 82-83] при описании "слоенного пирога" Semantic Web выделяет промежуточный "слой правил", для которого ведутся исследования различных систем вывода на правилах. Перейдем к базам данных.

Обзор и анализ существующих моделей данных баз данных

Рассмотрим принципы организации и структурирования данных в существующих моделях данных баз данных (БД). В работе [3] проведен анализ практически всех моделей представления данных: реляционных, сетевых, иерархических, семантических сетей, инфологических, онтологий, "сущность-связь" и миварного информационного пространства (всего более 25 моделей данных). В работе [10] Когаловский М.Р. описывает эволюцию технологий баз данных и систематизирует сведения о важнейших технологиях, состоянии и перспективах развития базовых стандартов моделей данных. Одним из крупнейших достижений является создание реляционной модели и, базирующейся на ней, теории реляционных баз данных [10, стр. 73]. Среди перспективных исследований выделим следующие: создание новых технологий для крупномасштабных систем; поддержка логики приложений в среде базы данных; интеграция структурированных и слабоструктурированных данных [10, стр. 74-75].

Марков А.С. и Лисовский К.Ю. в своей работе [11] рассмотрели основные математические модели представления об отношении: теоретико-множественные, конструктивные и предикатные. Также они обосновали выбор методов вычислительной логики для повышения уровня интеллектуальности реляционных баз данных. Кроме того, представлены еще сетевые и иерархические модели данных, слабоструктурированные данные, описан объектно-ориентированный подход к проектированию баз данных [11]. Марков А.С. вводит понятие "логическое программирование реляционных баз данных" в качестве одного из названий направления **"дедуктивные базы данных"**. Под этим подразумевается следующее: "...системе реляционных баз данных не только известна концептуальная модель данных, но и ... система обладает средствами *логического вывода*, т.е. она способна выводить теоремы из аксиом, логические следствия из гипотез и ... порождать потенциально бесконечное множество новых реляционных таблиц на основе исходно заданного базисного набора таблиц" [11, стр. 47]. Как видим, не только в работе [3] предложено совместить хранение данных с логическим выводом, но работа [3] опубликована раньше работы [11].

Саймон А.Р. в своей работе [12] дает многоаспектный анализ ключевых областей современных массовых технологий баз данных и оценивает перспективы их развития. Особый интерес представляют принципы активных систем управления баз данных (СУБД): "среды активных баз данных представляют собой некоторую комбинацию возможностей искусственного интеллекта (**ИИ**) и управления данными" [12, стр. 319]. В качестве способа представления бизнес-правил в контексте активных БД, Саймон А.Р. описывает "используемые в области

искусственного интеллекта *правила продукций* ... во многих отношениях языки правил в этих областях (искусственный интеллект и базы данных) концептуально подобны, и ... в будущем эти две области достигнут тесной интеграции" [12, стр. 319]. Более того, "активные базы данных открывают двери на пути к пока еще неуловимому будущему интеллектуальных баз данных с высоким уровнем технологии искусственного интеллекта" [12, стр. 320-321]. Подчеркнем, что "это будущее" уже наступило и реализовано оно именно в миварном подходе и программе "УДАВ" [6-8, 15]. Напомним, что у Поспелова Д.А. в [1] описан российский подход к *продукциям* и их обработке.

В книге [13] подробно рассмотрены структуры данных и алгоритмы, которые являются фундаментом современной методологии разработки программ. Для нас особый интерес представляет определение абстрактного типа данных как математической модели с совокупностью операторов, определенных в рамках этой модели. "Две характерные особенности процедур - обобщение и инкапсуляция - ... отлично характеризуют абстрактные типы данных" [13, стр. 23]. Для создания новых принципов организации и структурирования данных для реально работающих МЭПАИС важным является оценка времени выполнения различных программ, трактовка которых в [13, стр. 28-36] и в [2, стр. 227-239] значительно отличаются.

Кузнецов С.Д. пишет: "наиболее близко находятся современные объектно-ориентированные СУБД, модели данных которых по многим параметрам близки к семантическим моделям (хотя в некоторых аспектах они более мощны, а в некоторых - более слабы)" [14, стр. 144]. Там же рассмотрены "некоторые черты одной из наиболее популярных семантических моделей данных - модель "Сущность-Связь" (ER-модель). На использовании ER-модели основано большинство современных подходов к проектированию баз данных... Моделирование предметной области базируется на использовании графических диаграмм..." [14, стр. 144]. В [3] показаны формальные представления структур представления данных ER-модели и миварного информационного пространства.

Таким образом, существовали теоретические разработки по объединению логической и вычислительной обработки. Из всех моделей данных наиболее перспективным в этом плане является миварное информационное пространство (миварная модель данных) [3], основанное на многомерном эволюционном представлении данных, взаимозаменяемому универсальному представлению информации и возможности унифицированного логического вывода и вычислительной обработки для реализации всех отношений концептуальной модели данных различных областей, например, для МЭПАИС.

Представление продукций в виде двудольных графов и сетей Петри

Перейдем к формализованному представлению продукций и сетей, которые могут быть сформированы из них. Выше было отмечено, что

Поспелов Д.А. [1, стр. 83-84] представляет сеть правил в таком виде, что она отображается в виде однодольного графа. В работах [3, 5-8] предложен подход по взаимосвязи сетей правил и графов. Более того, показано, что некоторые задачи логического вывода можно решать на основе подходов теории графов. Сети правил и процедур целесообразнее представлять в виде двудольных графов, получая нечто аналогичное сетям Петри, но с соответствующим развитием до миварных логических сетей [3, 5-8]. Итак, напомним определения двудольных графов, которые необходимы для продолжения анализа. "Граф $G=(V, E)$ называется двудольным, если существует разбиение $V=\{V_1, V_2\}$ такое, что никакие две вершины из V_1 или из V_2 не являются смежными" [9, стр. 223].

"Двудольным графом $G=(X, Y, E)$ называется неориентированный граф, вершины которого можно разбить на два класса X и Y так, что концы каждого ребра принадлежат разным классам" [2, стр. 125]. Приведем следующее важное замечание: "введенные понятия допускают естественное обобщение. Неориентированный граф называется k -дольным, если его вершины можно разбить на k классов так, что концы каждого ребра принадлежат разным классам" [2, стр. 125]. Таким образом, можно использовать и двудольные, и трехдольные и многодольные (k -дольные) графы для разных предметных областей.

Переход от продукций к Миварным логическим сетям

В России активно развивается новый подход к единому представлению знаний и данных, который получил название "миварный". Как показано в [3, 8, 15] миварный подход является обобщением и развитием продукционного подхода, сетей Петри и других формализмов, применяемых для логической обработки данных. Миварные логические сети могут быть представлены в виде двудольного графа, состоящего из "объектов-переменных" и "правил-процедур". Прежде всего, составляются два списка, которые и образуют "две непересекающиеся доли графа": "объекты" и "правила". Каждое правило в миварной логической сети является развитием продукций, гиперправил с мультиактивизаторами или вычислительных процедур. Доказано, что с точки зрения дальнейшей их обработки все эти формализмы идентичны и представляют собой по сути вершины двудольного графа. Покажем, что для Многомерной эволюционной прикладной автоматизированной информационной системы поддержки принятия решений наиболее адекватным является формализм "миварные логические сети".

В статье [5] представлены теоретические основы создания линейного матричного метода определения маршрута логического вывода на адаптивной сети правил. Суть метода в том, что для сети правил, представленной в списочной форме, строится матрица. Затем, путем анализа этой матрицы определяется факт наличия маршрута вывода, определяются возможные маршруты логического вывода и из этих маршрутов выбирают по заданным критериям оптимальности наиболее

лучший, "кратчайший" маршрут. Пусть известны m - правил и n - переменных. Тогда в матрице $V (n \times m)$ представлены все взаимосвязи между правилами и переменными. При этом, входные переменные правила помечаются символом x , выходные - y , "выводимые" переменные - z , а все искомые переменные - w . В матрицу V добавим одну строку и один столбец для хранения служебной информации. Тогда, получаем матрицу V размерности $(n+1) \times (m+1)$, в которой отражена вся структура исходной сети правил, которая может изменяться. Сразу отметим, что структура этой логической сети может изменяться в любое время, т.е. это сеть правил с изменяемой (эволюционной) структурой. Пример такой матрицы показан на рис. 1. На рис. 2. эта же матрица показана в виде исходного задания двудольного графа миварной логической сети.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1										

Рис. 1. Пример исходного задания матрицы V размерности $(n+1) \times (m+1)$.

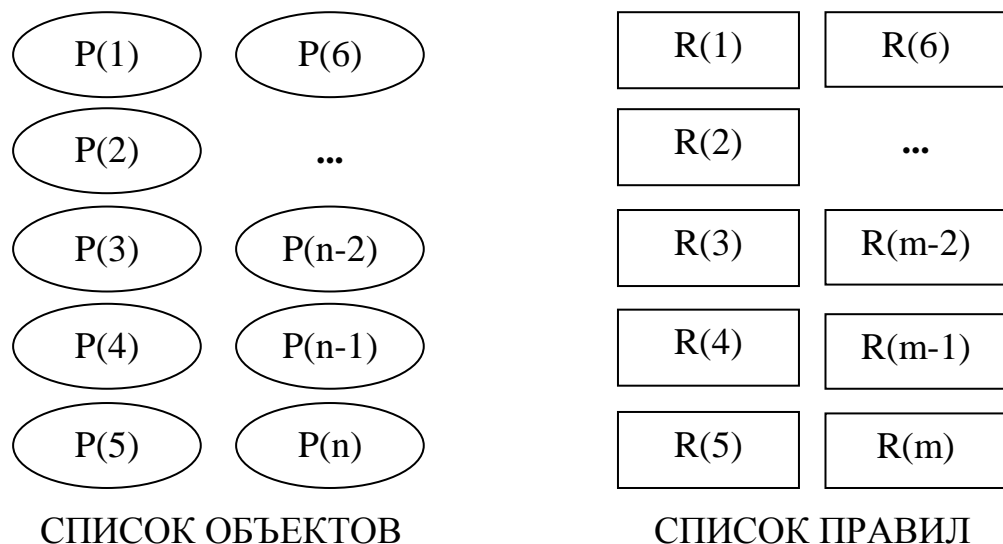


Рис. 2. Пример исходного задания двудольного графа миварной логической сети.

Опишем пример работы метода. Для поиска маршрута логического вывода на полученной матрице производят действия по следующим (пронумерованным) этапам.

1. В строке $(m+1)$ помечают известные z и искомые w переменные, например, как на рис. 3 в строке на позициях: z - 1, 2, 3 и w - $(n-2)$. На рис. 4 это действие показано на примере построения двудольного графа миварной логической сети. В связи с ограничением по объему публикации приведем рисунки только начала и завершения вывода (рис. 5 и рис. 6). В принципе, по рисункам 5 и 6 "в обратном порядке" можно восстановить все этапы работы метода.

2. Осуществляют последовательно, например, сверху вниз, поиск таких правил, которые могут быть активизированы, т.е. у которых известны все входные переменные. Если таких правил нет, то маршрута логического вывода нет и необходимо выдать запрос на уточнение (добавление) входных данных. Если такие правила, которые могут быть активизированы, есть, то у каждого из них в соответствующем месте служебной строки делается пометка, что правило может быть запущено. Например, можно ставить в этих клетках таблицы цифру 1.

3. Если таких правил несколько, то осуществляется выбор по заранее определенным критериям такого или таких правил, которые должны быть активизированы в первую очередь. При наличии достаточных ресурсов, одновременно могут запускаться сразу несколько правил.

4. Имитация запуска правила (процедуры) осуществляется путем присваивания выводимым в этом правиле переменным значений "известно", т.е. в нашем примере - z . Запущенное правило, для удобства дальнейшей работы помечается дополнительно, например, цифрой 2 (это не обязательно). Например, как показано на рис. 5 в клетках $(n-1, m+1)$, $(n, m+1)$ и $(n+1, 1)$ произведены соответствующие изменения. На рис. 6 это действие показано на примере построения двудольного графа миварной логической сети.

5. После имитации запуска правил проводят анализ достижения цели, т.е. анализируют получение требуемых значений путем сравнения служебных символов в служебной строке. Если в служебной строке $(m+1)$ осталось хоть одно значение "искомая" (т.е. w), то осуществляют дальнейший поиск маршрута логического вывода. В противном случае, задача считается успешно решенной, а все задействованные правила в соответствующем порядке их запуска и образуют искомый маршрут логического вывода.

6. Прежде всего, определяют наличие таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе. Если таких правил нет, то маршрута вывода нет и поступают аналогично пункту 2 этого метода. Если такие правила есть, то продолжают поиск маршрута вывода.

7. На следующем этапе опять аналогично этапу 4 запускают правила (имитируют запуск), далее аналогично этапам 5 и 6 выполняют

необходимые действия столько раз, сколько требуется для получения результата. При необходимости, повторяют все этапы со 2 по 7 до достижения результата. При этом результат может быть как положительный - маршрут вывода существует, так и отрицательный - вывода нет из-за неопределенности входных данных. Для наглядности продолжим пошаговое выполнение нашего примера. Итак, в нашем примере необходимо провести имитацию запуска правила 2, что и показано на рис. 5. На рис. 6 это действие показано на примере построения двудольного графа миварной логической сети.

8. В клетках $(4, m+1)$ и $(5, m+1)$ получаем признак выводимости переменных 4 и 5, а в клетке $(n+1, 2)$ формируем признак того, что правило уже было запущено, т.е. ставим цифру 2. После этого, проводим анализ служебной строки и видим, что не все искомые переменные известны. Значит необходимо продолжить обработку матрицы V размерности $(n+1) \times (m+1)$. Анализ этой матрицы показывает возможность запуска правила m .

9. Продолжим дальнейшую обработку примера. При запуске правила m появляются новые значения, в том числе и для искомых переменных, что и показано на рисунке 5. На рис. 6 это же действие показано на примере построения двудольного графа миварной логической сети.

10. Итак, в нашем примере в служебной строке больше не осталось искомых правил, а в клетках таблицы появились новые значения: в клетке $(n+1, m) - 2$, а в клетке $(n-2, m+1)$ вместо значения w появилось значение z . Таким образом, получен положительный результат, следовательно, маршрут логического вывода при данных исходных значениях существует.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w			

Рис. 3. Действие № 1 примера обработки матрицы **V**.

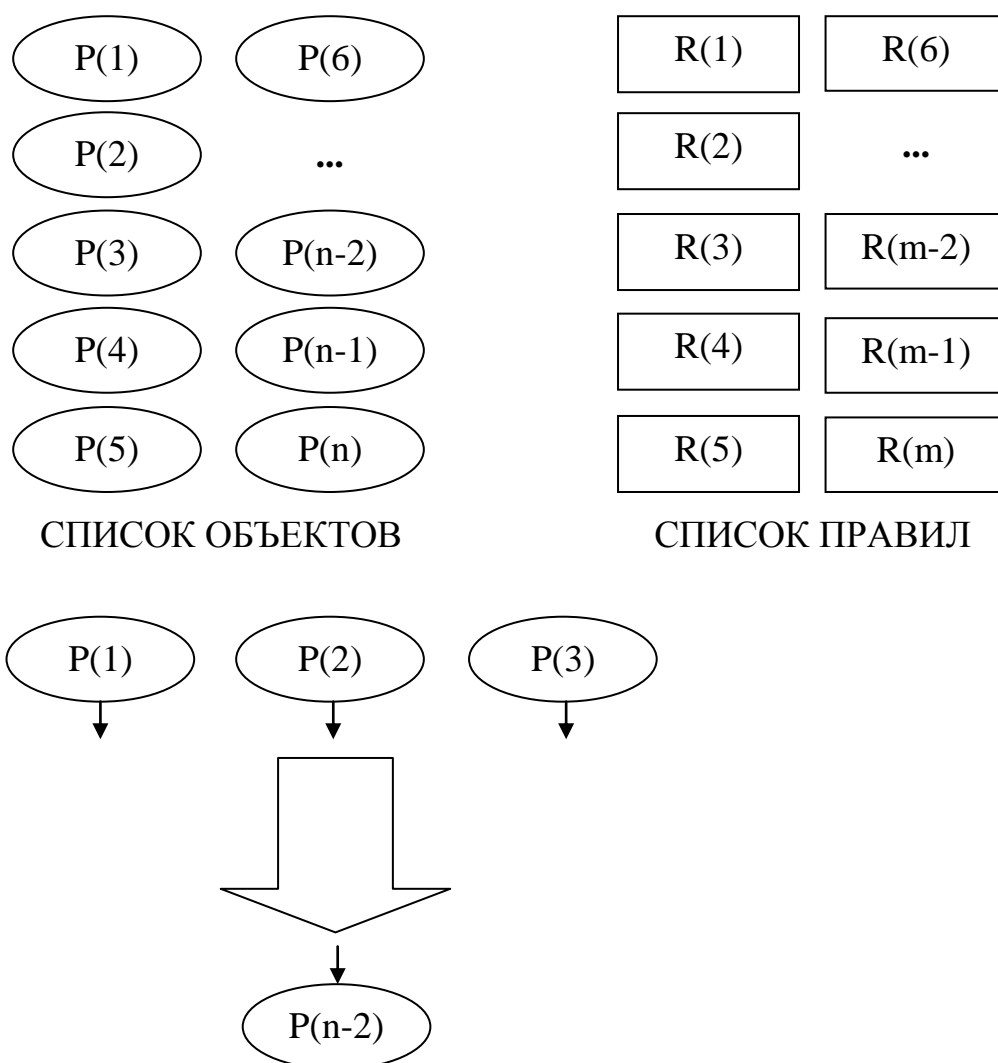


Рис. 4. Шаг № 1 примера поиска логического вывода на миварной сети.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	2
...						...				
m		x		x	x		y			2
m+1	z	z	z	z	z		z(w)	z	z	

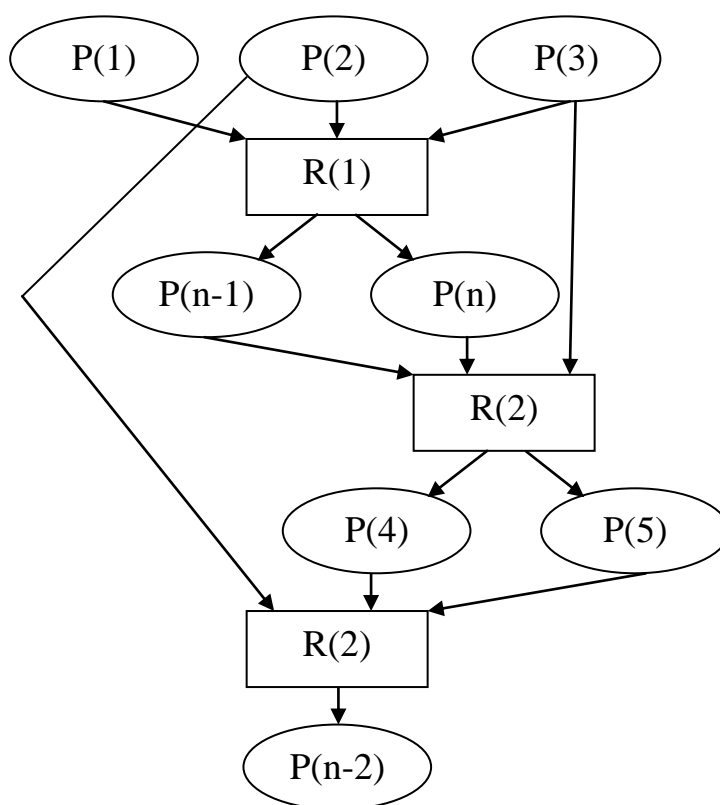
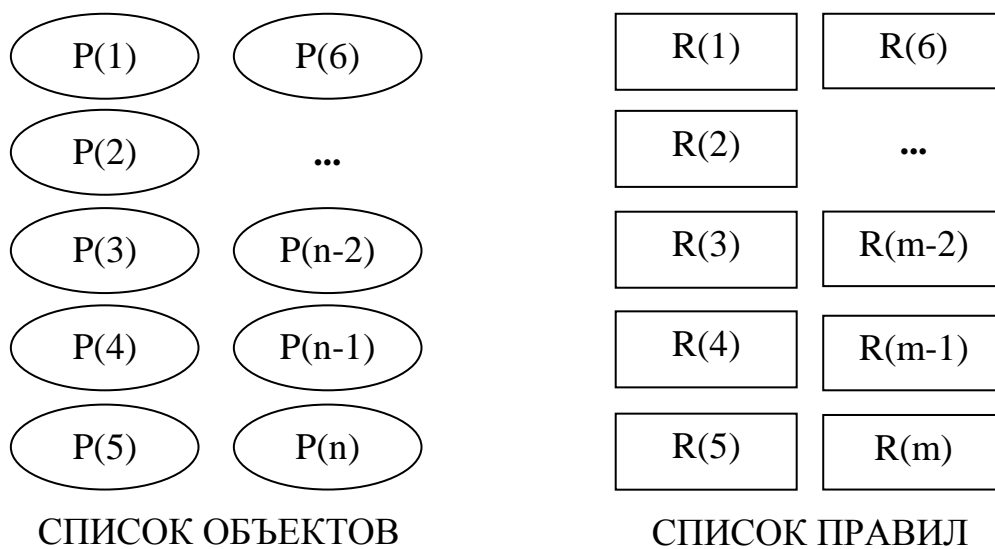
Рис. 5. Действие № 7 примера обработки матрицы **V**.

Рис. 6. Шаг № 7 примера поиска логического вывода на миварной сети.

Линейная вычислительная сложность миварного метода

Общее количество действий при матричном методе определяется суммой действий на каждом этапе: присваивание известных **z** и искомых **w** значений клеткам служебной строки (**m+1**), общее количество таких действий **не более n**; присваивание признака обработки правил в служебном столбце (**n+1**), количество действий не более **2m**, но может быть **не более числа m**; присваивание известных **z** значений клеткам служебной строки (**m+1**), общее количество таких действий **не более n**; определение новых значений клеткам строки (**m+1**), количество таких действий **не более n**. Отметим, что фактически на 1, 3 и 4 этапе производится обработка одного массива данных, т.е. клеток служебной строки (**m+1**). При этом общее, суммарное количество действий на всех этих этапах (1, 3 и 4) не должно превышать общее количество клеток в этой строке, т.к. обработанные значения "вычеркиваются" и более не обрабатываются. Получаем, что общее количество действий (**KD**) при линейном матричном методе поиска маршрута логического вывода (вычислительная сложность) не должно превышать количества клеток в служебных частях матрицы:

$$O(n+m), \text{ т.е. } KD \leq (n+m).$$

В том случае, когда нельзя реализовать предложенные сокращения вычислений, этот метод решает задачу поиска маршрута логического вывода с вычислительной сложностью: **O(nm)**, т.е. имеет место линейная зависимость либо от количества правил, либо от количества переменных. В худшем случае по каждой клетке "проходят" один раз.

В работах [6-8] изложены результаты практической реализации миварных логических сетей на основе создания программного комплекса "УДАВ". Этот комплекс реализует линейной вычислительной сложности матричный метод поиска маршрута логического вывода на миварной сети правил. Таким образом, в работах [3, 5-8] доказано, что существует достаточно универсальный миварный подход для решения многих практических задач применения информационных систем и процессов с одновременным проведением как логической, так и вычислительной обработки данных. Следовательно, существовавшее ранее противоречие между логическим выводом и вычислительной обработкой успешно преодолено с помощью миварных логических сетей. Проведенные вычислительные эксперименты подтверждают выводы, полученные в работах [6-7].

Отметим, что данный метод может применяться для прогнозирования развития предприятий в различных отраслях промышленности и для решения аналогичных задач в масштабах всей экономики России. Универсальность данного метода обусловлена применением продукций и миваров. Как было указано выше, "запись знаний в виде систем продукций носит универсальный характер - любые знания можно записать в такой форме" [1, стр. 129]. Универсальные возможности миварного подхода обусловлены тем, что он обобщает все

известные модели данных, включая семантические сети, "сущность-связь" и онтологии [3, 8]. В свою очередь миварные логические сети являются развитием продукционного подхода с переходом от однодольных графов к двудольным и многодольным и многомерным эволюционным графам. Следовательно, в формализме миварной сети правил можно описать любые данные, а также зависимости, функции и отношения в виде многомерного эволюционного пространства унифицированного представления данных и правил.

Развитие миварных сетей: реализация правил "ВЫБОРА" и переход к многодольным графам

Рассмотрим, как можно сделать так, чтобы после "выполнения" правила получали значение не все переменные (объекты). Существующие до этого правила будем называть "ПРОЦЕДУРЫ", т.к. после них все выходные переменные получают значения. Как было показано на рис. 6, такие правила можно представлять в виде двудольных графов:

- 1) объекты-переменные (овалы на рисунке 6) и
- 2) процедуры (прямоугольники на рисунке 6).

Логично предположить, что возможно использование многодольных графов в целях повышения адекватности моделирования для отдельных задач и различных предметных областей.

Для реализации выбора по значениям надо ввести новый тип правил: "ВЫБОР". Например, если существует два выхода: ДА или НЕТ. В формализме многодольных графов это означает введение третьего типа объектов графа:

- 1) переменные;
- 2) процедуры (без выбора) и
- 3) правила ВЫБОР, когда часть переменных получает значение, а другая часть - нет.

На рисунке 7 первый тип объектов показан "кругами", второй тип - "прямоугольниками", а третий тип - "треугольниками" с двумя типами "исходящих стрелок", например: "сплошные линии" - это "да", а "пунктирные линии" будут соответствовать значению "нет".

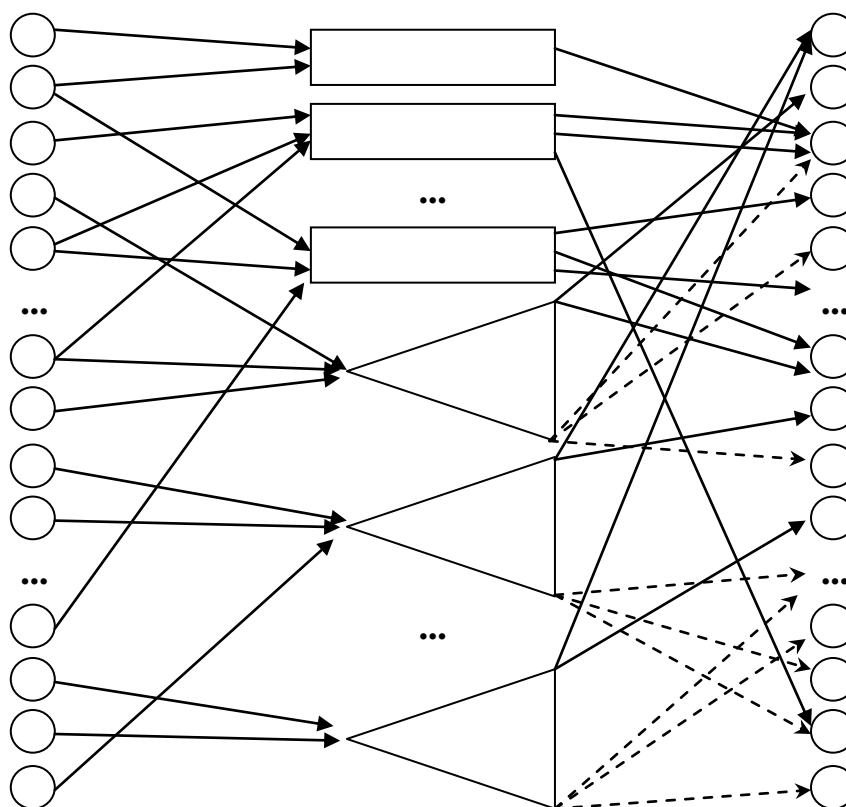


Рис 7. Трехдольный граф миварной сети правил.

Впрочем, можно сразу делать более сложный выбор из нескольких выходных переменных выбирать часть переменных "ДА", а оставшиеся переменные этого же правила - будут "НЕТ", и создать для них отдельную матрицу. По этим же принципам можно реализовывать не только двучный выбор "да/нет", но и более сложные правила выбора из нескольких альтернатив: от двух и больше, например: 3, 4, ... 10 и так далее. Аналогично можно переходить к более многодольным графам.

Для многодольных графов [2, стр. 125] предлагается следующий вариант их представления в миварных сетях. Пусть всего будет $(K+1)$ типов вершин графа, т.е. $(K+1)$ -дольный граф. Нумерацию "долей" такого многодольного графа рекомендуется начать (по математической традиции) с "нулевой" доли и обозначать ее на рисунках в виде кружочка или овала. Все остальные вершины других долей графа предлагаем обозначать по номеру типа от "единицы" до "K", а на рисунке отображать их в виде прямоугольников. Пример представления $(K+1)$ -дольного графа показан на рис. 8. В каждом овале и прямоугольнике есть описание узла графа, которое означает следующее: буква - это выделение объектов (P) и правил (R), далее в скобках первый символ соответствует номеру типа узла (доли) графа, а после запятой набор символов (буквы и цифры) обозначают номер узла в каждом типе многодольного графа: $P(0,n-2)$ - означает "узел нулевого типа ("объект") с номером $(n-2)$ ". Запись: $R(i,j)$ - означает "узел "правило" типа i с номером j ". Запись $R(K,mK-1)$ - это "узел "правило" типа K с номером в этом типе правил $(mK-1)$ ".

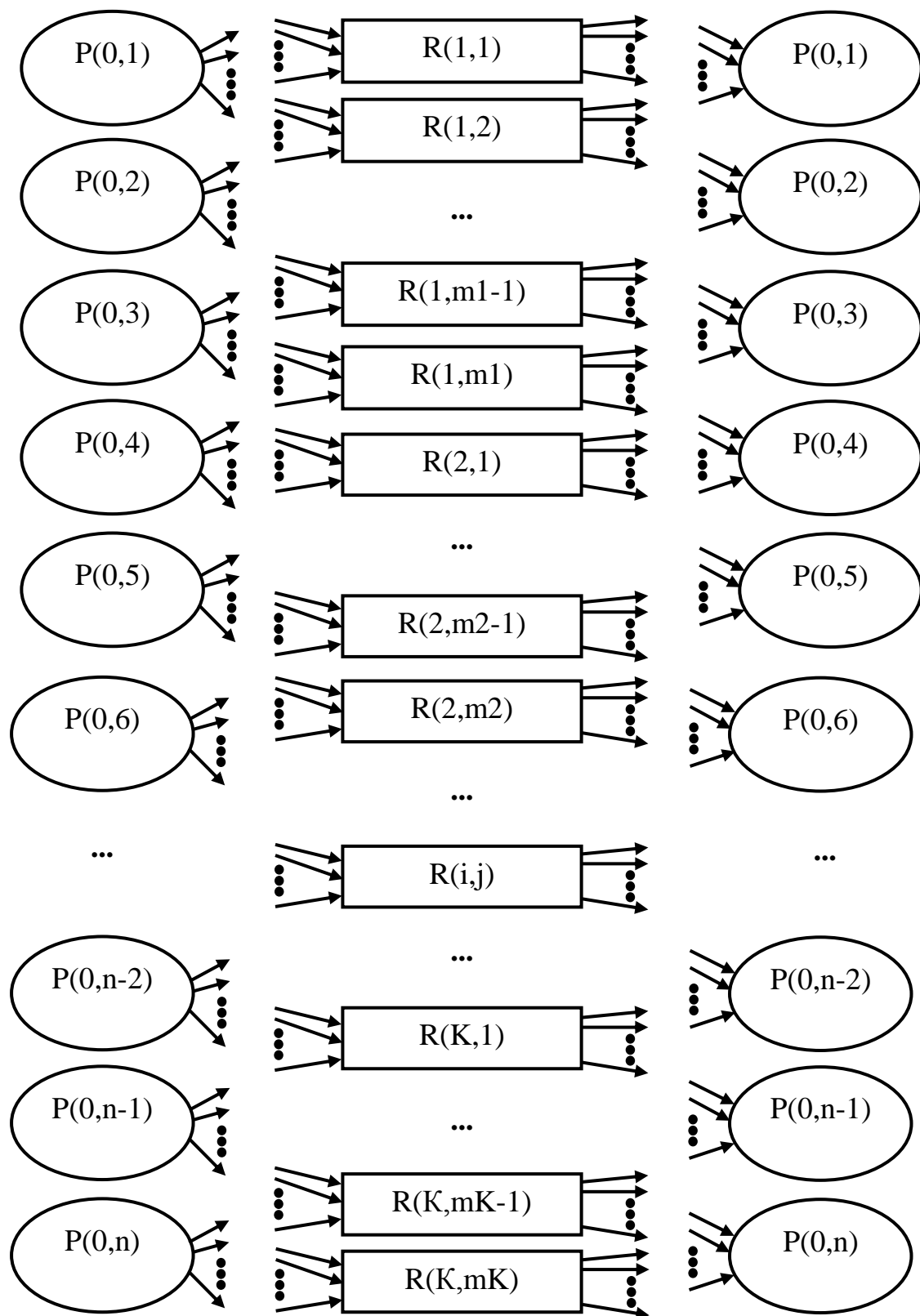


Рис. 8. Пример $(K+1)$ -дольного графа миварной логической сети.

Описание правил выбора из двух альтернатив

Прежде всего, рассмотрим выбор из двух альтернатив. Для этих "правил выбора" определяем все входные переменные, а после их "запуска", в зависимости от самого правила, значения будут получать только одна или несколько "однотипных" переменных из всего множества выходных переменных этого правила, т.е. определенная заранее часть переменных, что будет зависеть от того, как будет устроено каждое конкретное правило.

Затем, часть переменных получает значение (которые "ДА") и "вычеркиваются", а все остальные (которые "НЕТ") получают специальное значение "отрицание" (или что-то подобное) и также вычеркиваются из общей матрицы описания подобных правил выбора. Такое решение требует развития миварных сетей в следующем направлении. Необходимо ввести специальное новое значение. Если раньше у нас было два значения:

- 1) **известно** и есть число, а
- 2) **"пока не известно"**,

то теперь надо добавить в описания правил выбора третье значение:

- 3) **"не могут иметь значения"** - т.е. ложная ветвь поиска.

Предложен следующий вариант реализации такого подхода: в едином трехмерном пространстве в двумерных плоскостях у нас будут описываться конкретные предметные области, а по третьему измерению мы будем вводить новые двумерные плоскости. Тогда получим, что для описания каждой предметной области будут использоваться плоскости или двумерные матрицы - массивы:

- 1) плоскость **входных** значений правил "процедур" с бинарными значениями "да/не известно";
- 2) плоскость **выходных** значений правил "процедур" с бинарными значениями "да/не известно";
- 3) плоскость **входных** значений правил "выбор" с бинарными значениями "да/не известно";
- 4) плоскость **выходных** значений правил "Выбор" с **тремя значениями**: "да / не известно / не будут иметь значение".

В ходе дальнейших исследований по развитию миварного подхода запланировано проведение экспериментов по нескольким вариантам практической реализации хранения правил с тремя или более значениями правил "ВЫБОРА". Кроме того, отдельно будем вводить специальные переменные для правил "ВЫБОРА" в общем списке переменных. В этом списке тоже теперь должно быть три значения: 1) да; 2) не известно; 3) не будут иметь значение, что порождает необходимость проведения новых исследований.

Отметим, что в результате полученных модификаций появляется новое ограничение, т.к. правила "ВЫБОР" могут запускаться только в том случае, если уже вычислены все конкретные переменные, чтобы можно было определить "ветвь вычислений". Впрочем, если у нас много вычислительных ресурсов, то иногда можно "просчитывать" сразу по

нескольким веткам алгоритма, а потом подставлять конкретные значения. Это аналогично известному подходу "недоопределенные переменные", когда учитывают такие переменные при расчетах в качестве констант, значения которых подставляют в самом конце вычислений, когда все переменные определены и вычислены.

Итак, снято еще одно ограничение с миварных сетей и теперь можно решать задачи с "выбором", когда только часть переменных получает значение после выполнения правила, т.е. аналог выбора ДА или НЕТ. Разные предметные области или части этих областей можно представлять в виде трехмерной матрицы, наращивая "вверх" описания частей предметных областей или новых предметных областей.

Выводы

Для прикладных автоматизированных информационных систем поддержки принятия решений и экспертных систем разработан быстрый метод логико-вычислительной обработки данных на основе продукций и миварной логической сети правил. Данный метод позволяет в едином формализме проводить и логическую обработку - вывод, и выполнять различные вычислительные процедуры с возможностью выработки управляющих воздействий или рекомендаций управленцам (экспертам).

Для расширения возможностей баз данных и экспертных систем предложены новые пути развития миварных сетей на основе трехдольных и многодольных графов для реализации правил выбора вида "Если..., то..., иначе..." в целях эволюционного моделирования, накопления данных и унифицированной интеллектуальной обработке логических правил и выполнения вычислительных процедур в реальном времени.

Важным фактором является то, что вычислительная сложность решения любых задач на миварных сетях пропорциональна произведению количества правил-процедур на количество переменных, т.е. является линейной по отношению к переменным, либо по отношению к правилам. Это позволяет обрабатывать сложные практические задачи с несколькими десятками тысяч переменных в реальном масштабе времени, что нашло свое практическое подтверждение в реальных условиях эксплуатации.

ЛИТЕРАТУРА

1. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. - М.: Радио и связь, 1989. - 184 с.
2. Кузнецов О.П. Дискретная математика для инженера. 6-е изд., стер. - СПб: Издательство "Лань", 2009. - 400 с.
3. Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. - М.: Радио и связь, 2002. - 288 с.

4. Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) // Искусственный интеллект и принятие решений, № 1, 2008. С. 80-97.
5. Варламов О.О. Разработка линейного матричного метода определения маршрута логического вывода на адаптивной сети правил // Известия вузов. Электроника, № 6, 2002. С. 43-51.
6. Владимиров А.Н., Варламов О.О., Носов А.В., Потапова Т.С. Применение многопроцессорного вычислительного кластера НИИР для распараллеливания алгоритмов в научно-технических и вычислительных задачах // Труды НИИР: Сб. ст.- М.: НИИР. 2009. № 3. С. 120-124.
7. Носов А.В., Владимиров А.Н., Варламов О.О., Потапова Т.С. Программный комплекс "УДАВ": реализация логического вывода на миварной сети правил // Компьютерные науки и технологии (КНиТ-2009). Ч.1: сборник трудов первой Международной научно-технической конференции. - Белгород: ГиК, 2009. - 275 с. С. 232 - 236.
8. Веб-сайт д.т.н. Варламов О.О. // <http://www.ovar.narod.ru>. 2010.
9. Кук Д., Бейз Г. Компьютерная математика: Пер. с англ. - М.: Наука, Гл. ред. физ.-мат. лит., 1990. - 384 с.
10. Когаловский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2005. - 800 с.
11. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: Учебник. - М.: Финансы и статистика, 2006. - 512 с.
12. Саймон А.Р. Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ./ Под ред. и с предисл. М.Р. Когаловского. - М.: Финансы и статистика, 1999. - 479 с.
13. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. - М.: Издательский дом "Вильямс", 2007. - 400с.
14. Кузнецов С.Д. Базы данных: языки и модели. - М.: ООО "Бином-Пресс", 2008. - 720 с.
15. Веб-сайт компании МИВАР // <http://www.mivar.ru>. 2010.