
СЕКЦИЯ 1 | МОДЕЛИРОВАНИЕ РАССУЖДЕНИЙ
И НЕКЛАССИЧЕСКИЕ ЛОГИКИ

УДК 004.832.3

**ОБЗОР МЕТОДОВ НАХОЖДЕНИЯ СТЕПЕНЕЙ
ОБОСНОВАНИЯ В СИСТЕМАХ АРГУМЕНТАЦИИ¹**

В.Н. Вагин (*vagin@appmat.ru*)

О.Л. Моросин (*omorsik@gmail.com*)

Национальный исследовательский университет «МЭИ», Москва

В данной работе приводится краткий обзор подходов к формализации аргументации и рассматриваются различные методы применения степеней обоснования в системах аргументации. Степени обоснования позволяют эффективнее моделировать различные задачи аргументации и позволяют не только отвечать на вопросы о правдоподобности того или иного утверждения, но и давать ей числовую оценку. В заключении приводится пример реализации одного из методов.

Введение

Любая современная интеллектуальная система поддержки принятия решений содержит базу знаний. Данные базы знаний формируются, как правило, на основе экспертных знаний. Однако такие базы знаний часто содержат противоречивую информацию. Причины возникновения противоречий могут быть различными: ошибки формализации знаний, неверная интерпретация, противоречивость мнений нескольких экспертов. Проблема обработки противоречий является очень актуальной, так как классические методы логического вывода неприменимы для баз знаний, содер-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-00862).

жащих противоречия. Одним из способов обнаружения и разрешения внутренних противоречий является применение аргументации. Под аргументацией обычно понимают процесс построения предположений, относительно некоторой анализируемой проблемы. Как правило, этот процесс включает в себя обнаружение конфликтов и поиск путей их решения. В отличие от классической логики, аргументация предполагает, что могут быть доводы как «за», так и «против» некоторого предположения. Недостатком классических систем аргументации является оперирование качественными статусами аргументов – «поражен» и «не поражен». Введение степеней обоснования позволяет оперировать не только терминами «за» и «против», но и давать числовую оценку аргументам и контрагументам. Для подтверждения некоторого предположения необходимо доказать, что существует больше доводов «за» и степень их обоснования выше, чем у доводов «против».

В разделе 1 будут коротко приведены основные определения и идеи, используемые при реализации систем аргументации, в разделе 2 будет приведен обзор подходов к применению степеней обоснования в системах аргументации. В разделе 3 приводится пример применения системы аргументации со степенями обоснования.

1. Теоретические основы аргументации

Существует несколько формализаций теории аргументации. Например, системы абстрактной аргументации, предложенные Дантом (Dung P.M.) [Bondarenko и др., 1997], аргументационная система Лина и Шоэма (Lin F., Shoham Y.) [Lin и др., 1989], система Вресвийка (G.A.W. Vreeswijk [Vreeswijk, 1989], система аргументации Поллока (John L. Pollock) [Pollock, 1992] и некоторые другие.

В данной работе будет подробно рассмотрена система аргументации, основанная на пересматриваемых рассуждениях, предложенных Джоном Поллоком [Pollock, 1992].

1.1. Пересматриваемые рассуждения

Прежде чем переходить к изложению основного материала, дадим необходимые определения и введем обозначения.

Как уже говорилось, аргументация – процесс построения аргументов за и против некоторого предположения. *Аргумент* – пара, состоящая из множества посылок и заключения. Записывать такие

пары будем в следующем виде: p/X , где p – заключение, а X – множество посылок.

Например, аргумент $(p \& q)/\{\sim a, b\}$ означает, что из посылок $\sim a, b$ следует заключение $p \& q$. На всех иллюстрациях будем обозначать аргументы овалами. Для аргументов с пустым множеством посылок (такие аргументы называют фактами) будем писать только заключение. Например, фактом является утверждение, что Земля вращается вокруг Солнца.

Ключевым моментом в любых системах пересматриваемых рассуждений является понятие конфликта. Будем рассматривать два типа конфликтов – опровержение(*rebutting*) и подрыв(*undercut*) [Вагин и др., 2008]. *Опровержение* – ситуация, когда некоторые полученные аргументы опровергают выводы ранее полученных рассуждений. *Подрыв* – несимметричная форма атаки, когда один аргумент отрицает связь между посылками и заключением другого аргумента. На графе вывода подзывающие аргументы и пораженные ими аргументы будем соединять жирной пунктирной стрелкой. Пораженные аргументы будем помечать темно-серым цветом.

2. Степени обоснования в пересматриваемых рассуждениях

2.1. Вероятностный подход к степеням обоснования

В вероятностных моделях для каждого аргумента на графике аргументов задается вероятность, показывающая степень уверенности в достоверности аргумента. Таким образом, каждый аргумент A получает значение $P(A)$ из интервала $[0,1]$, где 0 соответствует пораженному аргументу, 1 наиболее обоснованному аргументу.

Графом вероятностей аргументов называется кортеж (A, R, P) , где A – множество аргументов, R – множество связей между ними и P – функция, которая определяет вероятность для каждого аргумента A ($P:A \rightarrow [0,1]$).

Существует несколько способов задания вероятностной функции P . В простейшем случае ее можно определить как 1 для не вступающих в конфликт аргументов, и как 0 для пораженных. Заданная таким образом модель будет в точности соответствовать классической системе аргументации с качественными статусами «поражен» и «не поражен».

Рассмотрим некоторые способы задания функции P .

2.1.1. Эпистемический подход [Hunter, 2010], который иначе называется «рациональным» распределением вероятностей. Это такие распределения, в которых атакующим аргументам присваивается высокий уровень доверия, в то время как атакуемым аргументам присваивается низкий уровень доверия, и наоборот. При эпистемическом подходе вероятностное распределение над аргументами используется, чтобы напрямую указать, какие аргументы наиболее достоверны. То есть, чем выше вероятность аргумента, тем более обоснованным он считается. Это удобно, т. к. кроме структуры графа, то есть информации о том, какие аргументы мы рассматриваем, мы можем определить «рациональные» степени представления об аргументах. Обычно при эпистемическом подходе используется следующая интерпретация вероятностей:

- $P(A) = 0$ показывает, что существует уверенность, в том, что A ложно;
- $P(A) < 0.5$ показывает, что A скорее всего ложно;
- $P(A) = 0.5$ показывает, что нет уверенности в том, что A ложно или истинно;
- $P(A) > 0.5$ показывает, что A скорее всего истинно;
- $P(A) = 1$ показывает, что существует уверенность, в том, что A истинно.

2.1.2. Групповой подход [Haenni и др., 1999], в котором используется распределение вероятностей над подграфами аргументов исходных графов. Таким образом, определяются допустимые (с непротиворечивым распределением вероятностей) подмножества аргументов.

Для аргумента A в графе G , с функцией вероятности P , $P(A)$ это вероятность того, что A существует в произвольном полном подграфе G , и $1 - P(A)$ вероятность того, что A не содержится в произвольном полном подграфе G .

2.2. Принцип слабейшей связи

В данном подходе для определения степеней обоснования используется числовая шкала $[0,1]$, где 0 соответствует пораженному аргументу, 1 – наиболее обоснованному аргументу.

Степени обоснования могут быть двух типов:

- 1) степени обоснования исходных аргументов;
- 2) степени обоснования пересматриваемых правил.

Первый тип степеней обоснования присваивается каждому

исходному аргументу, и представляет собой некую оценку достоверности источника, из которого получен данный аргумент. Например, по телевизору сказали, что вероятность осадков 70%. Соответственно мы можем построить аргумент $A1: \text{Завтра(дождь)}$ со степенью обоснования 0.7. Степени обоснования будем записывать функцией $Jus(A)$. То есть для приведенного примера $Jus(\text{Завтра (дождь)})=0.7$. Конкретные механизмы получения степеней обоснования зависят прежде всего от предметной области. Например, это могут быть статистические данные (в 90% этот источник дает верные данные) или экспертные оценки (вероятность роста акций 60%).

Второй тип степеней обоснования связан с пересматриваемыми правилами. Как уже говорилось выше, часто пересматриваемые правила появляются в результате формализации знаний эксперта вида «Если А, то чаще всего В». Такие правила также могут нести в себе некоторую числовую оценку.

Например, применение анальгина в 85% приводит к снижению температуры тела пациента (формально $R1: (\forall x) \text{ прием (анальгин, } x) \Rightarrow \text{ понижение_температуры}(x)$).

Одновременное использование обоих типов степеней обоснования является довольно сложной задачей и требует дополнительных исследований. В данной статье ограничимся рассмотрением степеней обоснования первого типа – для изначально заданных аргументов.

Итак, нам необходимо задать функцию $Jus(A)$ для вычисления любого из аргументов в графе вывода. Будем считать, что для изначальных аргументов эта величина является определенной. На значение этой функции будут оказывать влияние два фактора – дерево вывода аргумента (т.е. степень обоснования аргументов, которые использовались в выводе данного аргумента) и конфликты с другими аргументами. Для удобства рассмотрим эти два фактора раздельно: $Jus_{anc}(A)$ – унаследованная степень обоснования и $Jus_{con}(A)$ – на сколько конфликт подрывает обоснование аргумента. Пусть $\text{Anc} = \{\text{Anc}_i\}$, $i \in 1..n$ – множество аргументов Anc_i , участвовавших в выводе аргумента A , n – количество таких аргументов.

$$Jus_{anc}(A)=\min(\{Jus(A1), Jus(A2) \dots Jus(An)\}), \quad (2.1)$$

где A_1, A_2, \dots, A_n – аргументы, использовавшиеся при выводе A .

Формулу (2.1) называют принципом слабейшей связи [Pollock, 2001]. Отметим, что из формулы (2.1) следует, что если производить вычисление степеней обоснования рекурсивно, начиная от исходно заданных аргументов, то можно искать минимум не из всех аргументов в базисе, а только на предыдущем шаге. Таким образом, если у аргумента один предок, то его унаследованная степень обоснования будет равна степени обоснования его предка.

Если при вычислении Jus_{anc} ищутся наиболее слабые аргументы, то при определении того, насколько конфликт уменьшает обоснования, используются наиболее сильные аргументы. Пусть A_{conf} – множество аргументов, вступающих в конфликт с A , $n=|A_{conf}|$, тогда

$$Jus_{con}(A) = \begin{cases} \max_{i \in 1..n} Jus_{anc}(A \cap fl_i), & n > 0; \\ 0, & \text{в противном случае.} \end{cases} \quad (2.2)$$

В формуле (2.2) используется Jus_{anc} для того, чтобы верно обрабатывать случаи, когда между аргументами есть конфликт типа опровержение.

Итак, окончательно:

$$Jus(A) = \begin{cases} Jus_{anc}(A) - Jus_{con}(A), & Jus_{anc}(A) > Jus_{con}(A); \\ 0, & \text{в противном случае.} \end{cases} \quad (2.3)$$

Данная модель степеней обоснования была успешно реализована и протестирована на некоторых задачах. В заключение приведем пример решения тестовой задачи со степенями обоснования.

3. Пример работы системы аргументации

Приведем решение тестовой задачи, полученное в разрабатываемой системе аргументации.

Имеется следующая задача:

Предприниматель хочет вложить некоторую сумму в развивающийся бизнес и получить от этого доход. Есть компания СМР, которая занимается ИТ бизнесом. По статистике ИТ рынок находится на подъеме и 75% ИТ компаний показывают положительную динамику развития. Компания СМР занимается разработкой программного обеспечения под операционные системы семейства Linux. В России менее 10% пользователей Linux и, следовательно, спрос на

продукцию будет не велик. Однако 80 процентов производства компании направлено на международный рынок программного обеспечения для веб-серверов, которые в основном работают на Linux.

Запишем условия формально на языке ЛППП.

A1. $IT(CMP)$ justification: 0,75

A2. $под_Linux(CMP)$ justification: 1

A3. $рынок_сбыта(для_серверов)$ justification: 0,8

A4. $(\forall x)(инвестировать(x) \Rightarrow доход)$

R1. $(\forall x)(IT(x)) \Rightarrow (\forall x)рост_акций(x)$

R2. $(\forall x)(рост_акций(x)) \Rightarrow инвестировать(x)$

R3. $(\forall x)(под_Linux(x)) \Rightarrow плохой_спрос(x)$

R4. $(\forall x)(плохой_спрос(x)) \Rightarrow \sim инвестировать(x)$

R5. $(\forall x)рынок_сбыта(для_серверов) ==> ((под_Linux(x)) @ плохой_спрос(x))$

На рисунке 1 приведен граф вывода для данной задачи.

Рассмотрим решение данной задачи поэтапно.

1) Из аргумента *A1: $IT(CMP)$* и правила вывода *R1* получается пересматриваемый аргумент *A6 $Jus(A6)=Jus(A1)=0.75$* .

2) Из аргумента *A2: $под_Linux(CMP)$* и правила вывода *R3* получается пересматриваемый аргумент *A8: $плохой_спрос(CMP)$* . $Jus(A8)=Jus(A2)=1$.

3) Аргумент *A8: $плохой_спрос(CMP)$* с помощью правила *R5* вступает в конфликт типа подрыв с аргументом *A3: $рынок_сбыта(для_серверов)$* . Согласно формуле 1.3 $Jus(A3) = Jus_{anc}(A3) - Jus(A8) = 0.2$

4) Из аргумента *A6* и правила *R2* получается аргумент *A7: $Инвестировать(CMP)$* . $Jus(A7)=Jus(A6)=0.75$.

5) Из аргумента *A8* и правила вывода *R4* получается аргумент *A9 : $\sim Инвестировать(CMP)$* .

6) Аргументы *A9* и *A7* вступают в конфликт типа опровержение, степень обоснования аргумента *A7* выше, и следовательно, $Jus(A7)=0.55$, а аргумент *A9* становится полностью пораженным с $Jus(A9)=0$.

7) Используя правила обратного вывода из интереса *In5*, получается интерес *In11*, из которого получается интерес *In14*.

8) Из аргумента *A10: $Инвестировать(x) \rightarrow доход$* , аргумента *A7: $Инвестировать(CMP)$* , применяя унификацию и правило Modus Ponens, получаем аргумент *A12*. Согласно принципу слагаемости получаем интерес *In14*.

бейшей связи $Jus(A12) = 0.55$. Этот аргумент подтверждает интерес $In14$.

9) Окончательно получаем, что интересы подтверждены со степенью обоснования 55%. То есть инвестиция в компанию *CMP* принесет доход с вероятностью 55%.

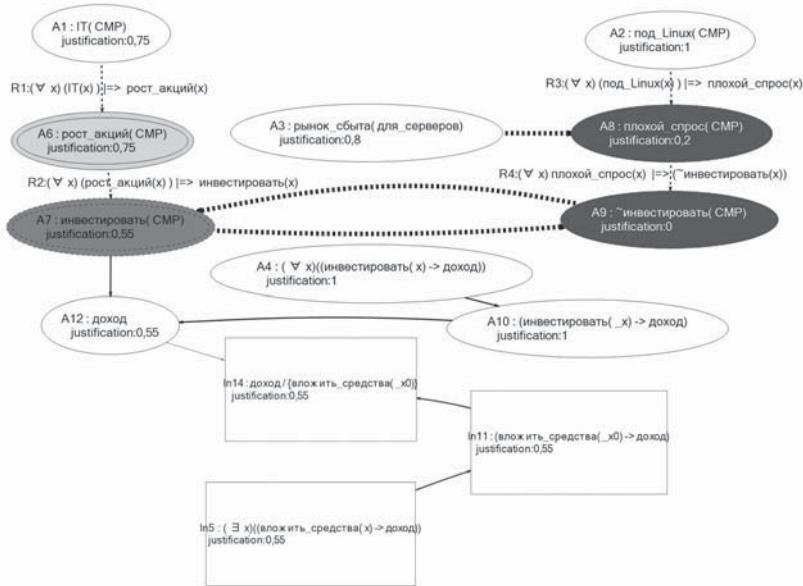


Рис. 1. Граф вывода

В приведенном примере для вычисления степеней обоснования применялся принцип слабейшей связи. Исходя из заданных степеней обоснования для начальных аргументов были вычислены степени обоснования всех выведенных аргументов, что позволило дать количественную оценку достоверности полученных выводов.

Заключение

Конфликты в базах знаний являются серьезной проблемой при построении любых интеллектуальных систем. Поиск, анализ и, возможно, разрешение этих конфликтов – важная и интересная задача. В данной статье был рассмотрен один из подходов к данной проблеме – использование аргументации, а именно теории пере-

сматриваемых рассуждений. Были рассмотрены теоретические основы пересматриваемых рассуждений и дан обзор системы, основанной на них. Кроме того, применение механизма степеней обоснования позволило решить важную задачу – давать не только качественные оценки («поражен» или «не поражен»), но и количественные. Разработанная программная система справилась со многими задачами, не решаемыми с точки зрения классической логики. Она была протестирована и отлажена на многих тестовых задачах из статьи Врейсвика «Interpolation of Benchmark Problems in Defeasible Reasoning» [Vreeswijk, 1995].

СПИСОК ЛИТЕРАТУРЫ

- [Bondarenko и др., 1997] Bondarenko A., Dung P.M., Kowalski R.A., Toni F. An Abstract Argumentation-Theoretic Framework for Defeasible Reasoning // Artificial Intelligence. – 1997. – Vol. 93, № 1–2. – P. 63–101.
- [Lin и др., 1989] Lin F., Shoham Y. Argument Systems. A Uniform Basis for Nonmonotonic Reasoning // Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. – San Mateo CA: Morgan Kaufmann Publishers Inc, 1989. – P. 245–355.
- [Vreeswijk, 1989] Vreeswijk G.A.W. Abstract Argumentation Systems // Artificial Intelligence. – 1997. – Vol. 90. – P. 225–279.
- [Pollock, 1992] Pollock J.L. How to Reason Defeasibly // Artificial Intelligence. – 1992. – Vol. 57.
- [Besnard, 2008] Besnard P., Hunter A. Elements of Argumentation. – Cambridge MA: MIT Press, 2008. – 298 p.
- [Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. 2-е издание дополненное и исправленное. – М.: Физматлит, 2008. – 712 с.
- [Pollock, 2001] Pollock J.L. Defeasible reasoning with variable degrees of justification // Artificial Intelligence. Artificial Intelligence. – 2001. – Vol. 133.
- [Haenni и др., 1999] Haenni R., Kohlas J., Lehmann N. Probabilistic Argumentation Systems, Handbook of Defeasible Reasoning and Uncertainty Management Systems // Algorithms for Uncertainty and Defeasible Reasoning. – 1999. – Vol. 5. – P. 221–287.
- [Vreeswijk, 1995] Vreeswijk G.A.W. Interpolation of Benchmark Problems in Defeasible Reasoning // Proceedings of the 2nd World Conference on the Fundamentals of Artificial Intelligence (WOCFAI 95). – Paris, France, 1995. – P. 453–468.

УДК 007:681.518.2.

**МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ИНТЕЛЛЕКТУАЛЬНОГО
АГЕНТА ПРИ НЕПРЕДВИДЕННОМ УВЕЛИЧЕНИИ
ДЛИТЕЛЬНОСТИ ЕГО ДЕДУКТИВНОГО ЦИКЛА
В УСЛОВИЯХ ЖЕСТКИХ ВРЕМЕННЫХ ОГРАНИЧЕНИЙ**

М.М. Виньков (*vinkovmm@mail.ru*)

РосНИИ ИТ и АП, Москва

И.Б. Фоминых (*igborfomin@mail.ru*)

МЭИ, Москва

В работе обсуждается подход к повышению устойчивости рационального агента, функционирующего в жестком реальном времени, к возникновению непредвиденных ситуаций, связанных с его функционированием. Данный подход основан на концепции метапознания и реализуется средствами т.н. активных логик.

Введение

Одной из наиболее важных и сложных проблем теории интеллектуальных систем является обеспечение их устойчивости («неуязвимости») к непредвиденным заранее ситуациям («аномалиям» [Anderson et al., 2005]). Аномалии возникают из-за несовершенства имеющихся у агента знаний о внешней среде и о собственных познавательных процессах и отрицательно влияют на его функционирование. Особенно тяжелыми последствия аномалий могут быть в интеллектуальных системах т.н. жесткого реального времени. В работе [Виньков и др., 2010] был предложен подход к моделированию поведения интеллектуального агента, направленного на повышение устойчивости к аномалиям в условиях жестких временных ограничений. Этот подход основан на использовании концепции метапознания, реализованной средствами специально разработанного для этой цели семейства т.н. активных логик. Однако существует важный класс аномалий, рассмотренный ниже, при возникновении которых указанный подход оказывается недейственным. В работе предлагается способ преодоления данной проблемы, в основе которого лежит специализированный вариант грануляции времени.

1. Режим жесткого реального времени

Системы реального времени можно подразделить на два класса – системы мягкого и системы жесткого реального времени. Для систем мягкого реального времени характерно плавное ухудшение качества их работы при увеличении времени отклика на изменения, происходящие во внешней среде. По-другому обстоит дело для систем жесткого реального времени, для которых характерно существование критического временного порога (deadline) решения стоящей перед системой задачи. Превышение этого порога чревато катастрофическими последствиями. Типичным примером аномалии для интеллектуальных систем такого рода является ситуация, когда событие, ожидаемое в назначенное время, тем не менее не наступило и можно говорить о возникновении угрозы превышения времени, отпущенного на осуществление познавательного процесса, целью которого является решение задачи, т.е. о катастрофическом ухудшении качества функционирования как самого агента, так и всей многоагентной системы в целом. Ясно, что и в многоагентных системах жесткого реального времени временной ресурс каждого агента также строго ограничен, раз это имеет место для всей системы в целом.

2. Концепция метапознания

Так как для интеллектуального агента, функционирующего в режиме жесткого реального времени, временной ресурс отводится, в первую очередь, на осуществление некоего познавательного процесса, то и аномалии, с которыми может столкнуться такой агент, проявляются в том, что познавательный процесс начинает идти «как-то не так», как ожидает агент. Ясно, что для того, чтобы вовремя выявить аномальное течение познавательного процесса, агенту потребуется, в некотором смысле, метапознавательная деятельность.

Термин «метапознание» (metacognition) был введен Д. Флэйвеллом [Flavell, 1979], [Flavell, 1987] и определен им как осведомленность индивидуума о своих когнитивных процессах и связанных с ними стратегиях, или, говоря его словами, как «знание и познание относительно познавательных явлений». В других источниках ме-

тапознание часто определяют просто как рассуждения о рассуждениях (например, в [Stefanuk, 1990]), имея в виду при этом «познание второго порядка».

3. Метакогнитивный цикл

Понятие «метакогнитивный» цикл было предложено в [Anderson et al., 2006] в контексте использования принципов метапознания для улучшения устойчивости к аномалиям рационального агента с ограниченным временным ресурсом. Он определяется как циклическое выполнение следующих трех этапов:

- самонаблюдение (мониторинг);
- самооценка (анализ выявленной аномалии) ;
- самосовершенствование (регуляция познавательного процесса).

Этап самонаблюдения, на котором выявляется наличие аномалий познавательного процесса, строится с привязкой возможных действий агента, влияющих на внешнюю среду, к ожидаемым последствиям этих действий. Признаком наличия аномалии при этом является несоответствие ожиданий агента с поступающей информацией о внешней среде. На этапе самооценки устанавливается степень угрозы для качества функционирования агента, которую таит в себе выявленная аномалия, а на этапе самосовершенствования, если угроза реальна, происходит выбор новой стратегии решения задачи, стоящей перед агентом. Таким образом, логическая система, формализующая рассуждения агента с жестко ограниченными ресурсами, должна давать ему возможность оценивать имеющийся у агента временной ресурс в каждый момент времени таким образом, чтобы в зависимости от результатов оценки агент мог менять ход своих рассуждений (темперальная чувствительность [Elgot-Drapkin, 1998]). Кроме того, агент должен быть толерантен к противоречиям в своих знаниях и быть способен их выявлять. Необходимым условием также является способность агента оценивать в каждый момент времени полноту имеющихся у него знаний и осознавать не только то, что он знает, но и то, чего именно он не знает. Логические системы, дающие агентам такие возможности, будут рассмотрены ниже.

4. Рассуждения во времени и метарассуждения на основе активной логики

Для того, чтобы дать возможность наблюдать за процессом рассуждения агента *во время* его осуществления, была предложена логическая система, получившая название «шаговая логика» [Elgot-Drapkin, 1998], ставшая исторически первой реализацией более общей концепции, получившей название «активная логика» [Anderson et al., 2004],[Anderson et al., 2005],[Anderson et al., 2006]. Сохраняя возможность рассуждать об агентах, как бы «глядя на них со стороны», в то же время активная логика позволяет самому агенту соотносить процесс своего рассуждения с событиями, происходящими во внешней среде в результате его деятельности или помимо нее. Как модель дедукции, активная логика характеризуется языком, множеством дедуктивных правил, а также множеством «наблюдений». Рассуждение во времени характеризуется выполнением циклов дедукции, называемых *шагами*. Так как в основе активной логики лежит дискретная модель времени, то эти шаги играют роль временного эталона – время измеряется в шагах. Знания агента ассоциируются с индексом шага, на котором они были впервые получены.

То, что узнал агент на шаге t (t -знания), используется для вывода новых знаний на шаге $(t + 1)$. Наиболее важные дедуктивные правила вывода в активной логике представлены ниже:

$$\frac{t : \text{now}(t)}{t + 1 : \text{now}(t + 1)} \quad \text{отсчет времени,}$$

где $\text{now}(\cdot)$ – одноместный метапредикат, выражающий текущее время;

$$\frac{\begin{array}{l} t : \phi, \phi \rightarrow \psi \\ t + 1 : \psi \end{array}}{t + 1 : \neg K(t, \phi)} \text{ modus ponens ;}$$
$$\frac{t : \phi, \text{sub}(|\phi|, |\phi|).[\phi]}{t + 1 : \neg K(t, \phi)} \text{ негативная интроспекция,}$$

где ϕ – любая формула, не известная агенту i на шаге t , но являющаяся подформулой некоторой известной ему формулы φ , $\text{sub}(\cdot, \cdot)$ – двухместный метапредикат, выражающий отношение «быть подформулой», $|\phi|$, $|\varphi|$ – константы, обозначающие имена формул

ϕ и φ , $[\phi]$ – нотация, означающая, что формула ϕ отсутствует в текущих знаниях агента на шаге t , $K(.,.)$ – двухместный метапредикат, используемый для выражения того факта, что агенту известна некоторая формула в некоторый момент времени;

$t : \phi, \neg \phi$ - выявление «прямых» противоречий,
 $t + 1 : contra(t, |\phi|, |\neg \phi|)$

где $contra(.,.)$ – двухместный метапредикат, используемый для выражения того факта, что в текущих знаниях агента присутствуют формула ϕ и ее отрицание $\neg \phi$ в некоторый момент времени t .

Наблюдения могут осуществляться на любом шаге дедуктивного процесса. Результатом наблюдения является формула, выражающая некоторое утверждение и ассоциированная с соответствующим шагом.

5. Непредвиденное увеличение длительности дедуктивного цикла

Как уже отмечалось выше, рассуждение во времени осуществляется посредством выполнения циклов дедукции, называемых шагами. При этом делается допущение, что шаги рассуждений имеют примерно одинаковую длительность: если длительность различных шагов также различна, то этой разницей можно пренебречь. Однако на практике это не всегда так – на длительность дедуктивных циклов могут оказывать влияние различные факторы, иногда не поддающиеся учету. Нетрудно видеть, что в таких случаях свойство темпоральной чувствительности агента утрачивается: измерение времени в дедуктивных шагах оказывается слишком неточным. Это означает, что аномалии данного типа могут быть не распознаны агентом, чье поведение описывается средствами активной логики.

6. Множественная грануляция времени

Как представляется, проблему, описанную в предыдущем разделе, можно преодолеть, если каждому конкретному дедуктивному циклу сопоставить его индивидуальную длительность, отказавшись при этом от постулата о неразличимости различных дедук-

тивных циклов по времени их осуществления. Если при использовании длительности дедуктивного цикла в качестве временного эталона (как это принято в активной логике) мы имеем пример грануляции временной информации [Тарасов, 2011], то переход к индивидуальной длительности каждого цикла естественно назвать множественной грануляцией времени: теперь каждому дедуктивному циклу соответствует индивидуальная временная гранула. Представленный ниже подход к решению проблемы непредвиденного увеличения длительности дедуктивного цикла базируется на построениях, приведенных в [Виньков, 2008], и основан на следующем.

Время рассматривается как бесконечная последовательность натуральных чисел из множества N . Будем обозначать ее Ck (глобальные часы). Так как моделирование поведения многоагентной системы осуществляется в различных условиях (=прогонах), каждому такому прогону ставится в соответствие т.н. часы прогона модели Ck , отражающие его специфику (принцип грануляции времени). Часы прогона модели – это конечная или бесконечная строго возрастающая подпоследовательность глобальных часов, члены которой интерпретируются как моменты времени (на глобальных часах) завершения дедуктивных циклов. Множество всех таких моментов времени обозначим Ck^* . Каждый «тик» часов прогона модели соответствует одному выполнению конкретного дедуктивного цикла. При этом порядковый номер этого цикла совпадает не с моментом времени его завершения, а только с порядковым номером этого момента времени на часах прогона модели. Это позволяет, меняя часы прогона модели, имитировать различные условия работы многоагентной системы. Используются также две функции – $clock(.)$ и $rank(.)$. Первая отображает множество N во множество Ck^* . Терм $clock(n)$ интерпретируется как момент времени, имеющий порядковый номер n на часах прогона модели. Вторая функция является обратной по отношению к первой, т.е. для всех $t \in Ck^* rank(t) = clock^{-1}(t)$, ее значением является порядковый номер момента времени t на часах прогона модели Ck . Тогда, если время завершения предыдущего дедуктивного цикла было t , следующий дедуктивный цикл завершится в момент времени $clock(rank(t) + 1)$. Обозначим $(clock(rank(t) + 1))$ как $next(t)$, т.е. момент

времени, следующий после момента t на часах ск. Тогда, в условиях грануляции времени дедуктивные правила вывода изменяются следующим образом (на примере правила отсчета времени):

$$\frac{t: \text{now}(t)}{\text{next}(t): \text{now}(\text{next}(t))}$$

7. Пример обнаружения аномалии познавательного процесса

Ниже приведен пример процесса метарассуждений, когда факт, что вопреки ожиданиям агента некоторое событие A не стало во время ему известно (момент времени 2), проявляется в виде прямого противоречия.

Пример 7.1.

Пусть ск = $(0, 1, 13, 15, 17, \dots)$

0: ... now(0), (now(t) \wedge $t \geq 5$) $\rightarrow K(\text{next}(t), A)$, ...

1: ... now(1), (now(t) \wedge $t \geq 5$) $\rightarrow K(\text{next}(t), A)$, $\neg K(1, A)$, ...

13: ... now(13), (now(t) \wedge $t \geq 5$) $\rightarrow K(\text{next}(t), A)$, $\neg K(1, A)$, $\neg K(13, A)$, $\neg K(15, A)$, ...

15: ... now(15), (now(t) \wedge $t \geq 5$) $\rightarrow K(\text{next}(t), A)$, $\neg K(1, A)$, $\neg K(13, A)$, $\neg K(15, A)$, $K(13, A)$, ...

17: ... now(17), (now(t) \wedge $t \geq 5$) $\rightarrow K(\text{next}(t), A)$, $\neg K(1, A)$, $\neg K(13, A)$, $\neg K(15, A)$, $K(13, A)$, *contra*(15, $K(13, A)$, $\neg K(13, A)$) ...

В моменты времени 0, 1, 13, 15 сработало правило негативной интроспекции, в результате чего были последовательно выведены формулы $\neg K(1, A)$, $\neg K(13, A)$, $\neg K(15, A)$. В момент времени 15 сработало правило $(\text{now}(t) \wedge t \geq 5) \rightarrow K(\text{next}(t), A)$, выражающее ожидание, что по прошествии 5 единиц времени агент будет знать, что в результате осуществляемого им познавательного процесса произошло событие A . В момент времени 17 сработало правило обнаружения противоречия, в данном случае между ожиданием агента, что он узнает о наступлении события A по истечении 5 единиц времени, и фактом, что к указанному времени информации о наступлении события A агенту не поступило. Заметим, что без множественной грануляции времени данная аномалия не была бы выявлена вовремя.

Заключение

Рассмотренная в данном докладе множественная грануляция времени ориентирована на применение в многоагентных системах жесткого реального времени. Опыт практического применения метапознавательных стратегий в сочетании с множественной грануляцией времени существенно расширяет класс непредвиденных ситуаций, которые могут быть успешно разрешены при моделировании поведения интеллектуальных агентов. Тем не менее рассмотренный подход не следует трактовать как панацею, решающую все проблемы, связанные с обеспечением устойчивости к аномалиям интеллектуальных систем. Не вызывает сомнений также, что на пути решения этих проблем метакогнитивный подход является необходимым звеном.

Благодарности. Работа выполнена при финансовой поддержке РФФИ (проекты 09-01-00877, 12-07-00083, 13-07-00396).

СПИСОК ЛИТЕРАТУРЫ

[Виньков и др., 2010] Виньков М.М., Фоминых И.Б. Повышение устойчивости к аномалиям интеллектуального агента с ограниченным временным ресурсом: метакогнитивный подход //Труды XII национальной конференции по искусственному интеллекту с международным участием КИИ-2010. – М.: Физматлит, 2010.

[Виньков, 2008] Виньков М.М. Время как внешняя сущность при моделировании рассуждений рационального агента с ограниченными ресурсами //Труды XI национальной конференции по искусственному интеллекту с международным участием КИИ-2008. – М.: Физматлит, 2008.

[Тарасов, 2011] Тарасов В.Б. Грануляция информации, нестандартные и гибридные нечеткие множества/ В.Б.Тарасов// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов VI Международной научно-практической конференции (Коломна, 16–19 мая 2011 г.). – М.: Физматлит, 2011. – Т.1. – С. 35–49.

[Anderson et al., 2006] Michael L. Anderson and Tim Oates and Waiyian Chong and Don Perlis. The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. Journal of Experimental and Theoretical Artificial Intelligence. 2006. 18. 3. 387–411.

[Anderson et al., 2005] Michael L. Anderson, Don Perlis. Logic, self-awareness and self-improvement: The metacognitive loop and the problem of brittleness, in Journal of Logic and Computation. 15. 1. 2005.

- [Anderson et al., 2004] Michael L. Anderson, Bryant Lee. Empirical results for the use of meta-language in dialog management. Proceedings of the 26th Annual Conference of the Cognitive Science Society. 2004.
- [Brown, 1987] Brown, A. (1987). Metacognition, executive control, self control, and other mysterious mechanisms. In F. Weinert and R. Kluwe (Eds.), *Metacognition, Motivation, and Understanding* (pp. 65–116). Hillsdale, NJ: Erlbaum.
- [Cox et al., 2007] Cox, Raja. Metareasoning: Manifesto, in BBN Technical Memo TM-2028, 2007.
- [Elgot-Drapkin, 1998] J. Elgot-Drapkin. Step Logic: Reasoning situated in time. PhD thesis. Department of computer science, University of Maryland, Colledge-Park, Maryland, 1988.
- [Fagin et al., 1988] R. Fagin and J. Y. Halpern, Belief, awareness and limited easoning, *Artificial Intelligence* 34 (1988), 39–76.
- [Flavell, 1979] Flavell, J. H. (1979). Speculations about the nature and development of metacognition. In F. Weinert & R. Kluwe, eds., *Metacognition and Motivation*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [Flavell, 1987] Flavell, J. H. (1987) Metacognition and cognitive monitoring: A new era in cognitive- developmental inquiry. *American Psychologist*, 34(10), 1987. – p.p. 906–911.
- [Metcalfe et al., 1994] Metcalfe, J., & Shimamura, A. P. (1994). Metacognition: knowing about knowing. Cambridge, MA: MIT Press.
- [Newell, 1990] F. Newell. Unified Theories of Cognition. Cambridge, MA: Cambridge University Press, 1990.
- [Purang et al., 1999] K. Purang, D. Purushothaman, D. Traum, C. Andersen, D. Traum, D. Perlis . Practical Reasoning and Plan Executing with Active Logic. 1999. Proceedings of the IJCAI'99 Workshop on Practical Reasoning and Rationality.
- [Stefanuk, 1990] Stefanuk V.L. The meta expert system// Proceedings of the seminar «Expert Systems on Personal Computers». – Moscow: MDNTP, 1990.
- [Raja et al., 2007] Raja, Lesse A framework for meta-level control in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 15(2):147–196.

УДК 007:519.816

ОСОБЕННОСТИ ПОСТРОЕНИЯ НЕЧЕТКИХ БАЙЕСОВСКИХ СЕТЕЙ ДОВЕРИЯ ДЛЯ МОДЕЛИРОВАНИЯ ТЕМПОРАЛЬНЫХ РАССУЖДЕНИЙ

А.С. Захаров (*auth1989@yandex.ru*)

Филиал МЭИ в г.Смоленске, Смоленск

В работе предлагается механизм учета темпоральных высказываний в рамках аппарата нечетких байесовских сетей доверия, что позволяет проводить моделирование темпоральных рассуждений на их основе. Описываются особенности построения предлагаемой гибридной модели.

Введение

Байесовские сети доверия (БСД) [Pearl, 1987] являются эффективным средством, позволяющим моделировать правдоподобные рассуждения, и потому имеют перспективы использования в системах поддержки принятия решений (ППР). Однако область применимости БСД существенно ограничивается невозможностью эффективного представления ряда существенных в области моделирования правдоподобных рассуждений аспектов, среди которых одним из ключевых является возможность представления различных видов неопределенности (помимо стохастической). Данная проблема отчасти решается аппаратом нечетких байесовских сетей доверия (НБСД) [Борисов, 2009], которые, обладая всеми возможностями БСД, позволяют моделировать рассуждения в условиях наличия лингвистической и эпистемологической неопределенности. Нечеткие байесовские сети доверия принято рассматривать как частный вид гибридных байесовских сетей [Pan, 2000], основанных на взаимодействии аппарата теории нечетких множеств и байесовских сетей доверия. В работе [Ren, 2005] приводится пример практического применения нечетких байесовских сетей для анализа рисков.

Однако, хотя НБС и устраняют часть ограничений классических БСД, этого в подавляющем большинстве случаев недостаточно, поскольку практически каждая реальная задача ППР так или иначе затрагивает временной аспект рассуждений. Поэтому на первый

план выходят так называемые темпоральные рассуждения – рассуждения, в которых учитываются временные зависимости между посылками и заключениями.

1. Нечеткие байесовские сети и темпоральная логика

Темпоральная логика является разновидностью модальной логики и может быть использована для решения задач моделирования рассуждений, когда необходим учет их временного аспекта. Тем не менее исследования, касающиеся темпоральной логики [Еремеев, 2006], [Еремеев, 2011], [Вагин и др., 2008], не затрагивают вопросы использования при моделировании рассуждений мер истинности высказываний, а в частности, вероятностной и нечеткой вероятностной мер истинности.

Создание нечеткой байесовской сети с поддержкой темпоральных зависимостей является задачей, решение которой актуально и способно повысить выразительность классической байесовской сети доверия, а значит и эффективность ППР с их использованием.

Простым темпоральным высказыванием называется высказывание, характеризующее динамику истинности классического высказывания, путем применения некоторого *темпорального оператора* [Еремеев, 2006] [Еремеев, 2011].

Для моделирования рассуждений с использованием аппарата нечетких байесовских сетей (НБС) особый интерес представляют операторы прошлого времени. В качестве меры истинности высказываний о состояниях переменных модели в НБС предлагается использовать аппарат нечетких вероятностных мер, поэтому для эффективного моделирования темпоральных рассуждений необходимо разработать принципы вычисления нечеткой вероятностной меры для утверждений, имеющих темпоральную природу.

Ключевыми темпоральными операторами прошлого времени являются «когда-либо в прошлом» (\blacklozenge), «всегда в прошлом» (\bullet), «когда-либо в прошлом на интервале $n..m$ » (\blacklozenge_n^m), «всегда в прошлом на интервале $n..m$ » (\bullet_n^m) [Еремеев, 2011].

Сложными темпоральными высказываниями будем называть высказывания, в которых простые темпоральные высказывания

связаны между собой темпоральными отношениями. Предлагаемый подход основан на результатах, предложенных в рамках темпоральной интервальной логики Алена. На множестве темпоральных высказываний определим множество темпоральных отношений $Rt = \{before, after, during\}$ [Allen, 1991]. По результатам анализа прикладных задач ППР в области медицинской диагностики, при анализе организационно-технических, финансовых и информационных рисков вышеуказанные отношения определены как наиболее значимые.

Грамматика сложных темпоральных высказываний необходима для формального определения принадлежности некоторого высказывания к классу сложных темпоральных высказываний. Для определения данной грамматики воспользуемся формой Бэкуса-Наура:

$$\begin{aligned} S &\rightarrow TMP \mid R(TMP, TMP); \quad TMP \rightarrow Ty; \\ R &\rightarrow Before \mid After \mid During; \quad T \rightarrow \bullet \mid \bullet^m \mid \bullet_n^m \end{aligned}$$

Здесь *Before*, *After*, *During*, \bullet , \bullet^m , \bullet_n^m , \wedge , \vee , y – терминалы грамматики, причем y – некоторое немодальное высказывание. Как следует из вышеописанной грамматики, в выражении условной вероятности само условие может быть либо применением некоторого темпорального оператора T к немодальному высказыванию (Ty), либо темпоральным отношением между высказываниями ($R(TMP, TMP)$).

Сложные темпоральные высказывания, исходя из требования непротиворечивости, не могут быть составлены произвольным образом на основе простых темпоральных высказываний и темпоральных отношений. В классической байесовской сети задаются условные вероятности вида $P(A = x_A \mid B = x_B, C = x_C, \dots)$, в байесовской сети с простыми темпоральными высказываниями – $P(A = x_A \mid T(B) = x_B, T(C) = x_C, \dots)$, где T – применение некоторого темпорального оператора. В байесовской сети со сложными темпоральными высказываниями для простых темпоральных высказываний могут указываться темпоральные отношения, в которых они участвуют, причем любая переменная не может быть одновременно и свободной, и связанной некоторым темпоральным отношением с другой переменной, т.е. допустимы условные вероятности вида

$$P(A = x_A \mid R(T(B) = x_B, T(C) = x_C), T(D) = x_D, \dots).$$

Здесь R – некоторое темпоральное отношение.

Однако использование темпоральных высказываний в НБС имеет ряд существенных особенностей, которые необходимо учитывать для непротиворечивого моделирования правдоподобных рассуждений с их помощью.

2. Особенности построения темпоральных нечетких байесовских сетей

Байесовские сети доверия являются математическим аппаратом для выражения причинно-следственных связей между различными переменными и извлечения информации, нетривиально следующей из описанных зависимостей. Понятие полной группы событий играет важную роль в формулировке байесова правила, лежащего в основе байесовских сетей доверия.

Однако в случае использования темпоральных высказываний понятие полной группы событий весьма нетривиально. В частности, высказывание, противоположное высказыванию «событие A наступало всегда в прошлом» или «событие A наступало когда-либо в прошлом», уже не так просто определить. Решение данной задачи актуально, поскольку позволяет обобщить формулу полной вероятности и формулу Байеса на случай темпоральных высказываний, что необходимо в случае наличия причинно-следственных связей, учитывающих временной аспект.

В работе [Вагин и др., 2008] в качестве основы для построения модальной логики используется исчисление предикатов первого порядка. Данный подход можно использовать для нахождения отрицания темпоральных высказываний.

Пусть t – некоторый момент времени, P – некоторое высказывание, тогда фразу вида «высказывание P истинно в момент времени t » будем обозначать как Pt . Модальность «когда-либо в прошлом P » можно выразить следующим образом: найдется такой момент времени в прошлом, в который истинно P , что на языке исчисления предикатов записывается как $\exists t : Pt$. Модальность «всегда в прошлом P » фактически означает следующее: в каждый момент времени в прошлом истинно P – в терминах исчисления предикатов записывается как $\forall t : Pt$.

Тогда отрицанием модального высказывания «когда-либо в прошлом P » будет формула следующего вида: $\neg \exists t : Pt$, из которой, пользуясь аксиоматикой исчисления предикатов (правило отрицания квантифицированных формул), можно вывести $\forall t : \neg Pt$, что эквивалентно «всегда в прошлом $\neg P$ ». Рассуждая аналогично, отрицание модального высказывания «всегда в прошлом P » можно представить как $\neg \forall t : Pt$, что дает $\exists t : \neg Pt$, т.е. «когда-либо в прошлом $\neg P$ ».

Пусть переменные A и B могут принимать значения « t » (истина) и « f » (ложь). Тогда отрицанием высказывания « $A = t$ » является высказывание « $A = f$ ». Аналогично для B . Также предположим, что между A и B существует причинно-следственная связь вида « $A \rightarrow B$ ».

Байесово правило в этом случае, будучи примененным для нахождения вероятности события « A когда-либо в прошлом равно t », дает следующий результат:

$$P(A_{\bullet t} | B_t) = \frac{P(B_t | A_{\bullet t}) * P(A_{\bullet t})}{P(A_{\bullet t}) * P(B_t | A_{\bullet t}) + P(A_{\bullet f}) * P(B_t | A_{\bullet f})}.$$

Модальные операторы «когда-либо в прошлом на интервале от m до n » (Φ_m^n) и «всегда в прошлом на интервале от m до n » (\bullet_m^n) также являются дуальными.

Так, высказывание «когда-либо в прошлом на интервале от m до n истинно P » ($P_{\bullet_m^n}$) может быть выражено в терминах логики предикатов первого порядка аналогично высказыванию P_\bullet с учетом следующего ограничения: $t \in [m, n]$ и вспомогательных определений:

$$\forall(x : A(x))B(x) \leftrightarrow \forall x[A(x) \rightarrow B(x)] \quad (1.1)$$

$$\exists(x : A(x))B(x) \leftrightarrow \exists x[A(x) \wedge B(x)] \quad (1.2)$$

Утверждение $t \in [m, n]$ обозначим как $X_t^{m,n}$, тогда рассматриваемое высказывание в терминах логики предикатов, согласно определению (1.2), выглядит как: $\exists t : X_t^{m,n} \wedge P$. Аналогично, высказывание «всегда в прошлом на интервале от m до n истинно P » ($P_{\bullet_m^n}$), с учетом введенных обозначений, согласно определению (1.1), может быть выражено в следующем виде: $\forall t : X_t^{m,n} \rightarrow P$.

Для нахождения отрицаний рассматриваемых модальных высказываний, как и ранее, будем считать, что они эквивалентны отрицанию их формулировок в терминах логики предикатов. Так, отрицанием модального высказывания $P_{\bullet_m^n}$ является формула: $\neg \exists t : X_t^{m,n} \wedge P = \forall t : \neg X_t^{m,n} \vee \neg P = \forall t : X_t^{m,n} \rightarrow \neg P$, что эквивалентно $(\neg P)_{\bullet_m^n}$, т.е. «всегда в прошлом на интервале от m до n ложно P ». Отрицание модального высказывания $P_{\bullet_m^n}$ может быть выражено формулой $\neg \forall t : X_t^{m,n} \rightarrow P = \exists t : \neg(\neg X_t^{m,n} \vee P) = \exists t : X_t^{m,n} \wedge \neg P$, что эквивалентно $(\neg P)_{\bullet_m^n}$, т.е. «всегда в прошлом на интервале от m до n ложно P ».

Как следствие из вышеприведенных результатов, для нахождения вероятности события « A когда-либо на интервале от m до n равно t » дает следующий результат:

$$P(A_{\bullet_m^n t} | B_t) = \frac{P(B_t | A_{\bullet_m^n t}) * P(A_{\bullet_m^n t})}{P(A_{\bullet_m^n t}) * P(B_t | A_{\bullet_m^n t}) + P(A_{\bullet_m^n f}) * P(B_t | A_{\bullet_m^n f})}.$$

Таким образом, для непротиворечивого использования байесового правила при работе с вероятностными оценками истинности высказываний, учитывающих темпоральный аспект, необходимо задавать условные вероятности с учетом описанной выше дуальной природы темпоральных операторов прошлого времени.

Еще одна особенность построения НБС с темпоральными высказываниями заключается в осуществлении прямого и обратного вывода в условиях наличия сложных темпоральных высказываний. Здесь применяется подход, суть которого заключается в замене сложного темпорального высказывания совокупностью простых с дополнительными ограничениями.

Рассмотрим варианты темпоральных отношений между простыми темпоральными высказываниями, и для каждого варианта укажем способ его представления без применения отношения. Примем следующие обозначения: $I_A = [x_A^-, x_A^+]$ – интервал времени, на котором определяется истинность высказывания A , $I_B = [x_B^-, x_B^+]$ – интервал времени, на котором определяется истинность высказывания B .

« A истинно всегда в прошлом и до этого B ». « A истинно всегда в прошлом и после этого B ». Данные высказывания являются бессмысленными и их истинность не может быть установлена. Интервал I_A – это все множество моментов времени от начала мо-

делирования до текущего момента. Поскольку не существует момента времени, следующего за каким-либо моментом из I_A или же предшествующего ему, то истинность зависимого высказывания не может быть установлена.

«A истинно всегда в прошлом и в течение этого B». Данное высказывание без потери семантики можно преобразовать к виду «*A истинно всегда в прошлом и B*». Вообще, отношение «*в течение*» при использовании совместно с темпоральным оператором «*всегда в прошлом*» может быть опущено.

«A истинно всегда в прошлом на интервале от m до n и до этого B». В случае если *интервалы времени для A и B не пересекаются*, возможны два случая: I_B предшествует интервалу I_A – в данном случае высказывание может быть преобразовано к «*A истинно всегда в прошлом на интервале от m до n и B*». Если I_A предшествует I_B , то высказывание бессмысленно и степень его истинности не может быть установлена. Если же *интервалы I_A и I_B пересекаются*, то возможны две ситуации. Если $x_B^- > x_A^-$, то степень истинности высказывания не может быть установлена. Если $x_B^- < x_A^-$, то интервалом времени, на котором определяется истинность B, удовлетворяющим ограничению отношения «*before*», является интервал $[x_B^-, x_A^-]$.

«A истинно всегда в прошлом на интервале от m до n и после этого B». Данный случай аналогичен рассмотренному ранее. В случае, когда I_B предшествует I_A , степень истинности высказывания не может быть установлена. Если I_B следует за I_A , то высказывание может быть преобразовано к виду «*A истинно всегда в прошлом на интервале от m до n и B*». Если же интервалы I_B и I_A пересекаются, то при условии $x_B^+ < x_A^+$ степень истинности высказывания не может быть установлена. Иначе, в случае выполнения условия $x_B^+ > x_A^+$ интервалом времени, на котором определяется истинность B, удовлетворяющим ограничению отношения «*after*», является интервал $[x_A^+, x_B^+]$.

«A истинно всегда в прошлом на интервале от m до n и в течение этого B». В случае выполнения одного из двух условий $x_B^- < x_A^-, x_B^+ < x_A^-$ или $x_B^- > x_A^+, x_B^+ > x_A^+$ степень истинности данного высказывания не может быть установлена. В противном случае истинность высказывания B может быть определена на интервале $[\max(x_A^-, x_B^-), \min(x_A^+, x_B^+)]$.

Исходя из того, как при прямом выводе вычисляется истинность высказывания с темпоральным оператором «когда-либо в прошлом» [Захаров, 2013], можно сказать, что она полностью определяется каким-то одним моментом времени, истинность подоператорного высказывания в котором максимальна. Поэтому ограничения, накладываемые темпоральным отношением, также зависят от данного момента времени и формулируются относительно него. Примем следующее обозначение: t_n – момент времени, который определяет истинность высказывания «*A* истинно когда-либо в прошлом».

«*A* истинно когда-либо в прошлом и до этого *B*». Если $t_n \geq x_B^-$, то истинность высказывания *B* определяется на интервале $[x_B^-, \min(x_B^+, t_n)]$, в противном случае все сложное темпоральное высказывание считается ложным.

«*A* истинно когда-либо в прошлом и в течение этого *B*». Если $t_n \in [x_B^-, x_B^+]$, то истинность *B* определяется в момент времени t_n , в противном случае высказывание ложно.

«*A* истинно когда-либо в прошлом и после этого *B*». Если $t_n \leq x_B^+$, то истинность высказывания *B* определяется на интервале $[\max(x_B^-, t_n), x_B^+]$, в противном случае все сложное темпоральное высказывание считается ложным.

Заключение

Таким образом, в работе предложен способ совместного использования темпоральных высказываний и нечетких байесовских сетей, показаны особенности построения предлагаемого класса моделей. Для предлагаемой модели созданы алгоритмы прямого и обратного вывода, имеющие асимптотическую верхнюю границу сложности $O(n^2)$, где n – число переменных модели. Всё это позволяет непротиворечивым образом проводить интерпретацию решаемой задачи поддержки принятия решений в терминах модели и в дальнейшем использовать полученную модель в процессе решения указанной задачи.

СПИСОК ЛИТЕРАТУРЫ

- [Борисов, 2009]** Борисов В.В., Белозерский А.Ю. Анализ рисков на основе нечетких байесовых сетей // Нейрокомпьютеры, № 8, 2009. С. 23–30.

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах. 2-е изд. // Под редакцией В.Н. Вагина, Д.А. Поспелова. – М.: ФИЗМАТЛИТ, 2008.

[Еремеев, 2006] Еремеев А.П. Логика ветвящегося времени и ее применение в интеллектуальных системах поддержки принятия решений // Сб. тр. 10-й нац. конф. по искусственному интеллекту с междунар. уч. КИИ – 2006. В 3-х т. Т.3. М.: Физматлит, 2006. С. 746–754.

[Еремеев, 2011] Еремеев А.П., Куриленко И.Е. Темпоральные модели на основе логики ветвящегося времени в интеллектуальных системах // Искусственный интеллект и принятие решений, 2011, № 1. С. 14–26.

[Захаров, 2013] Захаров А.С. Моделирование темпоральных рассуждений с использованием динамических нечетких байесовских сетей, научное издание «Информационные технологии, энергетика и экономика», сборник трудов 10-й международной научно-технической конференции студентов и аспирантов, 18-19 апреля 2013. С. 54–57.

[Allen, 1991] Allen J.F. Time and Time Again: The Many Ways to Represent Time // The International Journal of Intelligent Systems, 6(4), July 1991. P. 341–355.

[Pan, 2000] Pan, H., Liu, L.: Fuzzy Bayesian networks: A general formalism for representation, inference and learning with hybrid Bayesian networks. Int. J. Pattern Recogn. 14, 2000. P. 941–962.

[Pearl, 1987] Pearl J. Probabilistic reasoning using graphs // Uncertainty in Knowledge-Based Systems / Ed. By B. Bouchon, R. R. Yager. Springer-Verlag, 1987. P. 201–202.

[Ren, 2005] Ren J., Wang J., Jenkinson I. Fuzzy Bayesian modelling in maritime risk analysis // GERI Annual Research Symposium, 2005. P. 1–6.

УДК 004.832

**РАСПРОСТРАНЕНИЕ ОГРАНИЧЕНИЙ
И ЭВРИСТИЧЕСКИЙ ПОИСК
С ПРИМЕНЕНИЕМ МАТРИЧНОГО ПРЕДСТАВЛЕНИЯ
КОНЕЧНЫХ ПРЕДИКАТОВ¹**

А.А. Зуенко (*zuenko@iimm.ru*)

ФГБУН Институт информатики и математического моделирования
технологических процессов КНЦ РАН, Апатиты

В работе предлагаются методы распространения ограничений и эвристического поиска, направленные на решение задач удовлетворения ограничений, где переменные определены на конечных доменах. Разработанные методы используют матричное представление конечных предикатов, что позволяет обойтись без преобразования ограничений в совокупность бинарных отношений и дает возможность эффективно редуцировать пространство поиска.

Введение

Согласно [Russel, 2003] задача удовлетворения ограничений (Constraint Satisfaction Problem – CSP) определена множеством переменных x_1, x_2, \dots, x_n и множеством ограничений C_1, C_2, \dots, C_m . Каждая переменная x_i имеет непустую область определения D_i (область возможных значений, домен). Каждое ограничение C_i включает некоторое подмножество переменных и задает допустимые комбинации значений для этого подмножества. Состояние задачи описывается как присваивание значений некоторым (частичное присваивание) или всем переменным (полное присваивание): $\{x_i = v_i, x_j = v_j, \dots\}$. Присваивание, не нарушающее никаких ограничений, называется допустимым присваиванием. Решением задачи CSP является полное присваивание, которое удовлетворяет всем ограничениям.

Далее рассматриваются только задачи CSP, где переменные имеют конечные области определения. По оценке Bartak [Bartak,

¹ Работа выполнена при финансовой поддержке РФФИ (проекты №№ 12-07-00550-а, 12-07-00689-а, 13-07-00318-а, 14-07-00205-а, 14-07-00256-а, 14-07-00257-а), ОНИТ РАН (проект 2.3 в рамках текущей Программы фундаментальных научных исследований) и Президиума РАН (проект 4.3 Программы № 16).

1999], более 95% приложений задач CSP используют конечные домены.

Методы решения задач удовлетворения ограничений могут быть разбиты на три класса [Ruttkay, 1998]. Первый класс содержит различные варианты алгоритмов *поиска в глубину с возвратами*, которые строят решение путем расширения частичного присваивания шаг за шагом, используя различные эвристики и применяя разумные стратегии возврата из тупиковых вершин. Каждое решение должно представлять собой полное присваивание и поэтому, в худшем случае, должно находиться на глубине n , если имеется n переменных. Другими словами, дерево поиска имеет конечную глубину n (по количеству переменных). По этим причинам для решения задач CSP широко применяются алгоритмы поиска в глубину. В таких задачах сам путь, по которому достигается некоторое решение, не представляет интереса.

Ко второму классу относятся алгоритмы *распространения ограничений*, которые исключают из пространства поиска некоторые элементы, не входящие в решение, обеспечивая снижение размерности задачи. Эти алгоритмы не строят сами по себе решение, поскольку исключают не все элементы, не входящие в решение. Они применяются или для препроцессинга задачи до использования алгоритмов другого типа, или перемежаются с шагами алгоритма другого типа (например, поиска с возвратами) для повышения производительности последнего.

Наконец, *структурные алгоритмы* используют информацию о структуре первичного или двойственного графа ограничений задачи. Алгоритмы этого класса производят декомпозицию исходной задачи CSP на слабо связанные подзадачи, которые могут быть решены с помощью методов предыдущих двух классов. Таким образом, структурные алгоритмы также используются в сочетании с алгоритмами других типов.

Задачи CSP, как правило, решаются с использованием *графов* или *гиперграфов ограничений*. Традиционно при решении задач CSP ограничения с конечной областью определения сводятся к множеству бинарных отношений посредством введения достаточного количества вспомогательных переменных.

В настоящей работе для эффективного решения задач в рамках концепции программирования в ограничениях предлагается при-

менять подход, основанный на использовании матричного представления ограничений. Предлагаемые далее методы принадлежат к первым двум классам методов комбинаторного поиска и позволяют обойтись без предварительной «бинаризации» ограничений.

В качестве математического аппарата для моделирования ограничений используется алгебра кортежей (АК) [Кулик и др., 2010], [Зуенко, 2013].

1. Матричное представление ограничений

В [Кулик и др., 2010] приводятся основы АК и демонстрируется ее применение для унификации представления и обработки различных видов данных и знаний, а также решения различных задач логического и логико-вероятностного анализа. Близкий подход применяется также в [Zakrevskij, 2013] для решения задач распознавания образов и упрощения баз знаний.

Конечные предикаты в АК можно сжато представить с помощью двух типов структур: *C*-систем и *D*-систем. Конкретные экземпляры этих структур называются АК-объектами.

С помощью *C*-систем удобно представлять дизъюнктивные нормальные формы (ДНФ) конечных предикатов. Продемонстрируем это на примере. Пусть задан конечный предикат:

$$\varphi(x, y, z) = (x=a, b) \wedge (y=a, c) \vee (z=d).$$

Для простоты все переменные определены на одном и том же множестве $\{a, b, c, d\}$. Здесь и далее будем использовать запись вида $(x=a, b)$ для обозначения выражения $(x=a) \vee (x=b)$. Учитывая, что область истинности одноместного предиката $(x=a, b)$ есть $\{a, b\}$, то область истинности предиката $\varphi(x, y, z)$ может быть представлена в виде следующей *C*-системы:

$$R[XYZ] = \begin{bmatrix} \{a, b\} & \{a, c\} & * \\ * & * & \{d\} \end{bmatrix}.$$

Атрибуты X, Y, Z *C*-системы $R[XYZ]$ соответствуют переменным x, y, z формулы $\varphi(x, y, z)$. Заметим, что «*» – сокращенное обозначение всего диапазона возможных значений (домена) атрибута. *C*-систему $R[XYZ]$ можно преобразовать в многоместное отношение следующим образом:

$$(\{a, b\} \times \{a, c\} \times \{a, b, c, d\}) \cup (\{a, b, c, d\} \times \{a, b, c, d\} \times \{d\}).$$

С помощью D -систем моделируются конъюнктивные нормальные формы (КНФ) конечных предикатов. D -система записывается как матрица компонент-множеств, которая ограничена перевернутыми прямыми скобками.

D -системы позволяют легко вычислять дополнение C -систем: вычисляется дополнение для каждой компоненты-множества. Например, предикат $\neg\varphi = (\neg(x=a,b) \vee \neg(y=a,c)) \wedge \neg(z=d)$, что равносильно $\neg\varphi = ((x=c,d) \vee (y=b,d)) \wedge (z=a,b,c)$, может быть выражен как D -система $\bar{R}[XYZ]$:

$$\bar{R}[XYZ] = \begin{bmatrix} \{c,d\} & \{b,d\} & \emptyset \\ \emptyset & \emptyset & \{a,b,c\} \end{bmatrix}.$$

Пустая компонента « \emptyset » – это фиктивная компонента, не содержащая ни одного значения.

2. Базовые методы вывода на ограничениях

2.1. Концепция усеченного домена

При решении задач CSP дерево поиска можно представить как *дерево частичных присваиваний* – это дерево, где каждая вершина соответствует некоторому частичному присваиванию. Корень дерева отвечает пустому присваиванию. В вершине v выбирается лишь одна переменная, которой еще не было присвоено значение на предыдущих уровнях дерева поиска.

Поскольку переменным задачи CSP в АК соответствуют атрибуты, заданные на определенных доменах, то вполне естественно рассматривать процесс присваивания значения переменной как переход от старого домена атрибута к новому одноЭлементному домену. Поясним эту мысль более детально.

Рассмотрим, как реализуется операция присваивания переменной некоторого значения с использованием матричного представления логических ограничений. Пусть у нас изначально задано ограничение в виде D -системы $C[XYZ] = \begin{bmatrix} \{c\} & \{d,e\} & \{h\} \\ \{a\} & \{e\} & \{g\} \end{bmatrix}$, причем соответствие между атрибутами и их доменами следующее: $X - \{a, b, c\}$, $Y - \{d, e, f\}$, $Z - \{g, h\}$. Текущее присваивание (например, $x=b$) можно задавать C -системой, состоящей из одной строки:

$$E[XYZ] = [\{b\} * *].$$

Тогда процесс присваивания значения переменной и вычисления «остатка» D -системы $C[XYZ]$ выглядит так:

$$\begin{aligned}
 C[XYZ] \cap_G E[XYZ] &= \left[\begin{matrix} \{c\} & \{d,e\} & \{h\} \\ \{a\} & \{e\} & \{g\} \end{matrix} \right] \cap_G \left[\begin{matrix} \{b\} & * & * \end{matrix} \right] = \\
 \left[\begin{matrix} \{c\} & * & * \\ * & \{d,e\} & * \\ * & * & \{h\} \end{matrix} \right] \cap_G \left[\begin{matrix} \{a\} & * & * \\ * & \{e\} & * \\ * & * & \{g\} \end{matrix} \right] \cap_G \left[\begin{matrix} \{b\} & * & * \end{matrix} \right] &= \left[\begin{matrix} \{b\} & \{d,e\} & * \\ \{b\} & * & \{h\} \end{matrix} \right] \\
 \cap_G \left[\begin{matrix} \{b\} & \{e\} & * \\ \{b\} & * & \{g\} \end{matrix} \right].
 \end{aligned}$$

Последнее выражение получено в результате применения операции обобщенного пересечения (обозначается символом « \cap_G ») сначала к множителям 1 и 3, затем к множителям 2 и 3.

Пересечение двух полученных C -систем уже не удается представить в виде единой D -системы в пространстве $X - \{a, b, c\}$, $Y - \{d, e, f\}$, $Z - \{g, h\}$. Однако, если перейти к пространству $X - \{b\}$, $Y - \{d, e, f\}$, $Z - \{g, h\}$ (а такой переход вполне обоснован с учетом того, что результат пересечения в компонентах атрибута X не может иметь значений, отличных от b , поскольку в каждой из пересекаемых систем нет других значений в данном столбце), тогда:

$$\begin{array}{ccc}
 X & Y & Z \\
 \{b\} & \{d,e,f\} & \{g,h\} \\
 \emptyset & \{d,e\} & \{h\} \\
 \emptyset & \{e\} & \{g\}
 \end{array}$$

Для удобства объяснения здесь и далее в верхних двух строках D -систем записываются имена атрибутов и множества допустимых значений этих атрибутов (домены атрибутов).

Если сравнить между собой исходную D -систему и ту D -систему, которая получена после присваивания, то можно сделать важный вывод о том, что операция присваивания сводится к замене старого домена атрибута X на новый домен с вычеркиванием из соответствующего столбца тех значений, которые не принадлежат новому домену.

На следующем шаге поиска мы можем несколько упростить D -систему, элиминировав атрибут X , поскольку он более не связан ограничениями с другими атрибутами:

Y	Z
$\{d, e, f\}$	$\{g, h\}$
$\begin{bmatrix} \{d, e\} \\ \{e\} \end{bmatrix}$	$\begin{bmatrix} \{h\} \\ \{g\} \end{bmatrix}$

Будем считать, что ограничения задачи моделируются в виде некоторой D -системы. Состояние задачи в терминах введенных матрицеподобных структур полностью характеризуется с помощью совокупности одноэлементных доменов атрибутов, которые уже упомянуты из D -системы и самой D -системы, представляющей собой остаток, полученный из исходной D -системы в ходе присваиваний и применения правил редукции, которые представлены ниже.

Одним из основных отличий методов, предлагаемых в данной работе, от ранее разработанных в АК методов, является то, что в новых методах решение строится путем пошагового усечения доменов. Само решение представляется в виде совокупности доменов или установлении того факта, что все вершины дерева поиска являются тупиковыми.

Далее перейдем к описанию предлагаемых методов поиска решения задачи CSP, опишем используемые эвристики и применяемые правила редукции пространства поиска.

2.2. Выбор наилучшего преемника на основе эвристик

Среди методов поиска в глубину, используемых при решении CSP, можно выделить те методы, где переменные при конструировании частичного присваивания просматриваются в некотором заранее выбранном порядке. Однако такое статическое упорядочение переменных редко приводит к эффективному поиску. Следовательно, выбор переменной и присваиваемого ей значения должен осуществляться динамически на каждом шаге поиска. Для осуществления подобного выбора применяются различного рода *эвристики*. Методы, использующие эвристики, называются методами *информированного* или *эвристического поиска*. Эти методы позволяют отложить исследование менее перспективных ветвей дерева поиска, отдав предпочтение более перспективным с точки зрения получения скорейшего решения.

Для выбора «наилучшего» преемника текущей вершины дерева поиска необходимо ответить на вопросы:

1. Какой переменной на данном шаге присваивать значения?
2. Выбор какого из значений переменной быстрее всего может привести к цели?

Для ответа на данные вопросы в настоящей работе предлагаются такие эвристики:

Э1. Выбирать атрибут D -системы с доменом, содержащим наименьшее количество значений, что позволяет проверять меньшее количество преемников.

Э2. В случае неоднозначности выбора, производимого согласно **Э1**, выбирать атрибут, количество непустых компонент которого максимально.

Э3. Для формирования нового одноэлементного домена выбирать наиболее часто встречающееся в кортежах D -системы значение атрибута.

2.3. Редукция пространства поиска

Итак, построение дерева поиска состоит в переходе от некоторой D -системы к одной или нескольким D -системам меньшей размерности, то есть осуществляется «разбор случаев». Этот переход управляется предложенными эвристиками. Помимо применения эвристик, позволяющих более эффективно организовать «разбор случаев», можно существенно ускорить процесс поиска при решении задач CSP, рассматривая внутреннюю структуру текущего состояния задачи. Зачастую пространство поиска может быть значительно редуцировано вообще без организации «ветвлений» с помощью алгоритмов, преобразующих описание текущего состояния в эквивалентное ему более простое описание. Обычно такие алгоритмы имеют полиномиальную оценку временной сложности.

Теперь предстоит ответить на следующие вопросы:

1. Каков признак того, что текущая ветка поиска является тупиковой (соответствующее этой вершине текущее присваивание недопустимо)?

2. Что служит признаком успешного завершения процесса поиска?

3. Как уменьшить размерность пространства поиска, не прибегая к ветвлению?

Для ответа на эти вопросы рассмотрим следующие утверждения, приводимые здесь без доказательств:

Утверждение 1 (У1). Если хотя бы одна строка D -системы пуста (содержит все пустые компоненты), то D -система пуста (соответствующая ей КНФ невыполнима).

Утверждение 2 (У2). Если все компоненты некоторого атрибута пусты, то данный атрибут можно удалить из D -системы (удаляются все компоненты, стоящие в соответствующем столбце).

Утверждение 3 (У3). Если в D -системе есть строка (кортеж), содержащая лишь одну непустую компоненту, то все значения, не входящие в эту компоненту, удаляются из соответствующего домена (компонента становится новым доменом данного атрибута).

Утверждение 4 (У4). Если строка D -системы содержит хотя бы одну полную компоненту, то она удаляется (можно удалить соответствующий дизьюнкт из КНФ).

Утверждение 5 (У5). Если компонента некоторого атрибута D -системы содержит значение, не принадлежащее соответствующему домену, то это значение удаляется из компоненты.

Утверждение 6 (У6). Если одна строка D -системы полностью доминирует (покомпонентно содержит) другую строку, то доминирующая строка удаляется из D -системы.

Ответ на первый из поставленных вопросов дает нам **У1**, то есть *признаком недопустимости* частичного присваивания является пустота D -системы.

Ответом на третий вопрос служат остальные утверждения, часть из которых позволяет исключать значения из доменов атрибутов (**У3, У5**) или даже сами атрибуты (**У2**), а часть – исключать из рассмотрения лишние строки (**У4, У6**).

Признак успешного завершения процесса поиска – вычеркивание из D -системы *всех* строк и столбцов без образования пустых строк. Другими словами, результирующее состояние в этом случае будет характеризоваться только совокупностью непустых усеченных доменов.

Описанные преобразования сохраняют множество решений логического уравнения (системы уравнений), моделируемого D -системой. В некоторых случаях, например, при решении задачи «Выполнимость КНФ», можно предложить дополнительные правила редукции, не сохраняющие множество корней, а сохраняющие лишь свойство выполнимости КНФ [Zakrevskij, 2013]. Однако применять дополнительные правила редукции нужно с большой осторожностью.

рожностью, особенно в случае, когда, кроме прочего, переменные связаны нелогическими (количественными) ограничениями, диктуемыми спецификой предметной области, так как на некоторой итерации можно исключить все допустимые (семантически корректные) решения.

Заключение

Разработаны новые методы решения задач CSP, использующие матричное представление конечных предикатов. Они позволяют обойтись без разложения ограничений в бинарные отношения. Анализ внутренней структуры таких матриц дает возможность эффективно редуцировать пространство поиска. Впервые в АК рассмотрена инкрементная постановка задачи поиска и разработан метод распространения ограничений, основанный на пошаговом усечении доменов атрибутов и получении решения задачи CSP в виде совокупности доменов. В отличие от имевшихся в АК и других прототипах методов «слепого» поиска, в разработанных методах широко применяются эвристики.

СПИСОК ЛИТЕРАТУРЫ

[**Зуенко, 2013**] Зуенко А.А. Матрицеподобные вычисления в задачах удовлетворения ограничений /А.А. Зуенко // Шестая Всероссийская мультиконференция по проблемам управления (30 сентября – 5 октября 2013 г.). Материалы мультиконференции: в 4 т. – Ростов-на-Дону: Издательство Южного федерального университета, 2013. Т.1.

[**Кулик и др., 2010**] Кулик Б.А., Зуенко А.А., Фридман А.Я. Алгебраический подход к интеллектуальной обработке данных и знаний – СПб.: Изд-во Политехнического ун-та, 2010.

[**Bartak, 1999**] Bartak R. Constraint Programming: In Pursuit of the Holy Grail // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, 1999.

[**Russel, 2003**] Russel S. and Norvig P. Artificial Intelligence: A Modern Approach. Second edition. – Prentice Hall, 2003.

[**Ruttkay, 1998**] Ruttkay Zs. Constraint satisfaction a survey // CWI Quarterly. 1998. V. 11.

[**Zakrevskij, 2013**] Zakrevskij A. Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities. In: «Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems», IGI Global, 2013.

УДК 004.832

ОБРАБОТКА СПЕЦИАЛЬНЫХ ВИДОВ ОГРАНИЧЕНИЙ ПРИ РЕШЕНИИ ЗАДАЧ УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ С ПОМОЩЬЮ АЛГЕБРЫ КОРТЕЖЕЙ¹

А.А. Зуенко (zuenko@iimm.ru)

ФГБУН Институт информатики и математического моделирования
технологических процессов КНЦ РАН, Апатиты

В работе предлагается модификация базовых методов вывода на ограничениях, разработанных ранее автором для решения задач удовлетворения ограничений с конечными доменами и использующих матричные структуры алгебры кортежей. Суть модификации состоит в применении дополнительных правил редукции пространства поиска, диктуемых спецификой конкретной предметной области, что позволяет существенно ускорить процесс решения задачи.

Введение

В настоящее время методы решения многих задач комбинаторного поиска, в том числе оптимизационных, разрабатываются в рамках направления программирования в ограничениях (Constraints Programming) [Russel et al., 2003]. Ограничения могут иметь вид уравнений, неравенств, логических выражений и т.д. [Bartak, 1999]. При решении многих реальных задач удовлетворения ограничений (Constraint Satisfaction Problem – CSP) стандартные логические или численные методы оказываются либо недостаточно эффективными, либо неприменимыми, особенно если данные описываются в виде множеств и интервалов, содержащих допустимые значения. Последовательное механическое выполнение подобных методов (даже, если это возможно) часто приводит к ошибочным результатам, то есть исключению всех допустимых вариантов решения задачи CSP. Поэтому возникает необходимость в разработке новых эффективных методов, предполагающих совместную обработку разнородных ограничений.

¹ Работа выполнена при финансовой поддержке РФФИ (проекты №№ 12-07-00550-а, 12-07-00689-а, 13-07-00318-а, 14-07-00205-а, 14-07-00256-а, 14-07-00257-а), ОНИТ РАН (проект 2.3 в рамках текущей Программы фундаментальных научных исследований) и президиума РАН (проект 4.3 Программы № 16).

В [Зуенко, 2013] [Зуенко, 2014] рассматривается применение матрицеподобных структур алгебры кортежей (АК) C -систем и D -систем [Кулик и др., 2010] для решения задач CSP, описаны базовые методы вывода на ограничениях, которые сводятся к логическим выражениям над одноместными предикатами. Эти методы, в отличие от традиционных методов решения задач CSP (например, методов теории графов), позволяют обойтись без добавления вспомогательных переменных и преобразования ограничений с конечной областью определения к множеству бинарных отношений.

При организации поиска некоторые типы ограничений приходится обрабатывать с помощью алгоритмов специального назначения, более эффективных по сравнению с методами общего назначения. В теории удовлетворения ограничений подобные ограничения называются глобальными ограничениями [Rossi et al., 2008], там же приводится типизация глобальных ограничений, в частности, представимых в виде линейных уравнений или неравенств. Для каждого типа предлагаются специализированные алгоритмы. Другими словами, процедура поиска конструируется в зависимости от конкретной предметной области путем задания набора глобальных ограничений (необходимых дополнительных проверок). Однако эти специальные алгоритмы обычно также используют графы или гиперграфы ограничений и обладают теми же недостатками, что и традиционные методы общего назначения.

Если задача CSP представлена посредством матриц конечных предикатов [Зуенко, 2014], то при помощи лишь базовых правил редукции не всегда удается успешно распространить ограничения. Дело в том, что некоторые типы ограничений при формализации в структурах АК порождают громоздкие конструкции и их целесообразно рассматривать в качестве глобальных ограничений. Глобальные ограничения в настоящей работе предлагается задавать в виде дополнительных правил редукции пространства поиска, обеспечивающих исключение некоторых значений из доменов атрибутов, как только начинают выполняться определенные условия.

Далее разбирается, каким образом производится обработка специальных видов ограничений на основе упомянутой модификации предложенных ранее автором базовых методов вывода на ограничениях.

1. Работа с глобальными ограничениями

Ряд исследований по удовлетворению ограничений с конечными доменами использует понятие *глобального ограничения* [Rossi et al., 2008]. Классическим примером глобального ограничения служит ограничение *Alldiff*, которое указывает, что все участвующие в нем переменные должны иметь разные значения. Это ограничение часто используется на практике и в связи с этим предлагается как встроеннное ограничение во многих коммерческих и исследовательских системах программирования в ограничениях [Aggoun et al., 1993].

У глобальных ограничений есть два преимущества [Щербина, 2012]. Во-первых, они облегчают задачу моделирования прикладной проблемы в виде задачи CSP. Вместо записи задачи с помощью многих исходных бинарных ограничений ее удается описать с помощью небольшого числа глобальных ограничений для множеств переменных. Во-вторых, можно разработать алгоритм распространения ограничений специального вида, учитывающий особенности ограничения и поэтому более эффективный.

Так, например, ограничение *Alldiff* может обрабатываться, как и любое другое ограничение, если *Alldiff* записать в виде строк *D*-системы. Этот способ порождает довольно громоздкие конструкции. Альтернативный способ – разработать специальную процедуру, дополняющую изложенные ранее базовые методы, которая бы гарантировала соблюдение ограничения *Alldiff* на уровне доменов переменных.

Другими примерами глобальных ограничений являются глобальное *кардиальное ограничение* (global cardinality constraint (gcc)) [R'egin, 1996] и *кумулятивное ограничение* [Aggoun et al., 1993]. Глобальное кардиальное ограничение определяет условие, что число переменных, которым присвоены значения, должно быть между заданными верхней и нижней границами. Кумулятивное ограничение над множеством переменных, представляющих времена выполнения различных работ, гарантирует, что работы упорядочены с учетом имеющихся в наличии величин ресурсов, используемых в любые периоды времени. Ограничения обоих указанных типов обычно используются при составлении расписаний, упорядочении работ и в календарном планировании.

В АК работа с глобальными ограничениями реализуется путем введения специализированных правил редукции пространства по-

иска, дополняющих те правила, которые используются в базовых предметно-независимых методах вывода на ограничениях. Покажем на конкретном примере, как осуществляется совместная обработка логических и кумулятивных ограничений в рамках предлагаемого подхода.

2. Пример анализа глобальных ограничений

Прежде чем приступить к рассмотрению примера, приведем лишь те базовые правила редукции, которые будут применены в дальнейшем:

Утверждение 3 (У3). Если в D -системе есть строка (кортеж), содержащая лишь одну непустую компоненту, то все значения, не входящие в эту компоненту, удаляются из соответствующего домена (компонента становится новым доменом данного атрибута).

Утверждение 4 (У4). Если строка D -системы содержит хотя бы одну полную компоненту, то она удаляется (можно удалить соответствующий дизьюнкт из КНФ).

Утверждение 5 (У5). Если компонента некоторого атрибута D -системы содержит значение, не принадлежащее соответствующему домену, то это значение удаляется из компоненты.

Полный перечень правил, применяемых в базовых методах вывода на ограничениях в структурах АК, можно найти в [Зуенко, 2014]. Здесь сохранена установленная в упомянутой статье нумерация правил.

Формулировка задачи. Пусть имеются список заданий (работ) и ограничения на то, какие задания могут быть выполнены одновременно, выполнение каких заданий должно предшествовать выполнению других и т. д. Требуется найти назначение времен начал работ заданиям так, чтобы все ограничения удовлетворялись. Рассмотрим задачу составления расписания для пяти заданий T_1, T_2, T_3, T_4, T_5 , каждое из которых может быть выполнено за один час. Задания могут начинаться в 1:00, 2:00 или 3:00. Любые работы могут выполняться одновременно, учитывая ограничения на то, что: 1) T_1 начинается после T_3 , 2) T_3 начинается до T_4 и после T_5 , 3) T_2 не может начинаться в то же время, что T_1 или T_4 , 4) T_4 не может начинаться в 2:00.

Кроме чисто логических ограничений, накладывается следующее кумулятивное ограничение: для выполнения T_1 требуется 5 человек, для задания T_2 – 7 человек, для T_3 – 3 человека, для T_4 – 5

человек, для $T_5 - 4$ человека. Количество работ, которые могут выполняться одновременно, ограничено имеющейся рабочей силой (10 человек).

Решение. Можно построить модель составления графика, введя пять атрибутов X_1, X_2, X_3, X_4, X_5 , соответствующих заданиям, с доменами $\{1, 2, 3\}$, где: 1 – 1:00, 2 – 2:00, 3 – 3:00.

Видно, что формулировка задачи опирается на два типа бинарных отношений: «начинаться до/позже чем» и «выполняться одновременно/не одновременно». Запишем условия задачи в виде рассмотренных ранее матрицеподобных структур.

Введем атрибуты X_1, X_2, X_3, X_4, X_5 , которые соотносятся с заданиями T_1, T_2, T_3, T_4, T_5 , соответственно. Все атрибуты имеют одну и ту же область определения $D = \{1, 2, 3\}$ – времена начала работ. Рассмотрим, как формализовать в виде D -системы отношение « T_1 начинается после T_3 ». Отношение « T_1 начинается до или одновременно с T_3 » на языке АК выглядит так:

$$C1[X_1X_3] = \begin{bmatrix} \{1\} & * \\ \{2\} & \{2,3\} \\ \{3\} & \{3\} \end{bmatrix}.$$

Тогда его дополнение, представляющее собой требуемую формализацию, записывается в виде:

$$D1[X_1X_3] = \begin{bmatrix} \{2,3\} & \emptyset \\ \{1,3\} & \{1\} \\ \{1,2\} & \{1,2\} \end{bmatrix}.$$

Представление всех ограничений в виде D -систем может показаться странным. Здесь необходимо иметь в виду то, что предлагаемые методы ориентированы на определенное представление знаний. Не составляет принципиальной сложности разработать двойственные методы, обрабатывающие C -системы.

По аналогии, сформируем D -системы $D2$ и $D3$, соответствующие условиям « T_3 начинается после T_5 », « T_4 начинается после T_3 » (равносильно утверждению « T_3 начинается до T_4 », заданному в условии задачи):

$$D2[X_3X_5] = \begin{bmatrix} \{2,3\} & \emptyset \\ \{1,3\} & \{1\} \\ \{1,2\} & \{1,2\} \end{bmatrix}, D3[X_4X_3] = \begin{bmatrix} \{2,3\} & \emptyset \\ \{1,3\} & \{1\} \\ \{1,2\} & \{1,2\} \end{bmatrix}.$$

Формализация следующего отношения « T_2 не может начинаться в то же время, что T_1 или T_4 » осуществляется путем пересечения двух представленных ниже D -систем $D4$ и $D5$.

$$D4[X_1X_2] = \begin{bmatrix} \{2,3\} & \{2,3\} \\ \{1,3\} & \{1,3\} \\ \{1,2\} & \{1,2\} \end{bmatrix} \text{ и } D5[X_2X_4] = \begin{bmatrix} \{2,3\} & \{2,3\} \\ \{1,3\} & \{1,3\} \\ \{1,2\} & \{1,2\} \end{bmatrix}.$$

Так $D4[X_1X_2]$ получается вычислением дополнения от $C4 = \begin{bmatrix} \{1\} & \{1\} \\ \{2\} & \{2\} \\ \{3\} & \{3\} \end{bmatrix}$, моделирующей утверждение « T_2 начинается в то же время, что T_1 ».

Наконец, утверждение « T_4 не может начинаться в 2:00» записывается как отрицание утверждения « T_4 начинается в 2:00» (запишем его в полном пространстве признаков):

$$D6[X_1X_2X_3X_4X_5] = \overline{[* * * \{2\} *]} =]\emptyset \ \emptyset \ \emptyset \ \{1,3\} \ \emptyset[.$$

Тогда общая формулировка задачи может быть выражена как следующая D -система:

	X_1	X_2	X_3	X_4	X_5
	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$
1	$\{2,3\}$	\emptyset	\emptyset	\emptyset	\emptyset
2	$\{1,3\}$	\emptyset	$\{1\}$	\emptyset	\emptyset
3	$\{1,2\}$	\emptyset	$\{1,2\}$	\emptyset	\emptyset
4	\emptyset	\emptyset	$\{2,3\}$	\emptyset	\emptyset
5	\emptyset	\emptyset	$\{1,3\}$	\emptyset	$\{1\}$
6	\emptyset	\emptyset	$\{1,2\}$	\emptyset	$\{1,2\}$
7	\emptyset	\emptyset	\emptyset	$\{2,3\}$	\emptyset
8	\emptyset	\emptyset	$\{1\}$	$\{1,3\}$	\emptyset
9	\emptyset	\emptyset	$\{1,2\}$	$\{1,2\}$	\emptyset
10	$\{2,3\}$	$\{2,3\}$	\emptyset	\emptyset	\emptyset
11	$\{1,3\}$	$\{1,3\}$	\emptyset	\emptyset	\emptyset
12	$\{1,2\}$	$\{1,2\}$	\emptyset	\emptyset	\emptyset
13	\emptyset	$\{2,3\}$	\emptyset	$\{2,3\}$	\emptyset
14	\emptyset	$\{1,3\}$	\emptyset	$\{1,3\}$	\emptyset
15	\emptyset	$\{1,2\}$	\emptyset	$\{1,2\}$	\emptyset
16	\emptyset	\emptyset	\emptyset	$\{1,3\}$	\emptyset

Для удобства объяснения здесь в верхних двух строках D -системы записываются имена атрибутов и множества допустимых значений этих атрибутов (домены атрибутов).

Анализируя строки 1, 4, 7, 16, по **У3** можно усечь домены для X_1, X_3, X_4 , получим такие соответствия между атрибутами и их доменами: $X_1 - \{2, 3\}, X_2 - \{1, 2, 3\}, X_3 - \{2, 3\}, X_4 - \{3\}, X_5 - \{1, 2, 3\}$. При этом для одновременного выполнения T_4 и T_1 требуется 10 человек, T_4 и T_2 – 12 человек, T_4 и T_3 – 8 человек, а для выполнения работ T_4 и T_5 – 9 человек. Следовательно, кумулятивное ограничение для момента времени «3:00» нарушено лишь для заданий T_4 и T_2 . Для устранения этой некорректности из домена атрибута X_2 исключается значение «3», имеем:

$$X_1 - \{2, 3\}, X_2 - \{1, 2\}, X_3 - \{2, 3\}, X_4 - \{3\}, X_5 - \{1, 2, 3\}.$$

Затем атрибут X_4 элиминируется из D -исходной системы, поскольку его домен стал одноэлементным множеством {3}.

В результате «настройки» D -системы на новые домены, применяя правило **У4**, исключаются из рассмотрения строки 1, 4, 7, 8, 10, 12, 13, 14, 15, 16.

Далее, используя **У5**, получим (здесь и далее, помимо самой D -системы, будем записывать элиминированные атрибуты с их доменами, то есть давать полное описание состояния CSP):

$X_4 - \{3\}$	X_1	X_2	X_3	X_5
	{2,3}	{1,2}	{2,3}	{1,2,3}
2	{3}	\emptyset	\emptyset	\emptyset
3	{2}	\emptyset	{2}	\emptyset
5	\emptyset	\emptyset	{3}	{1}
6	\emptyset	\emptyset	{2}	{1,2}
9	\emptyset	\emptyset	{2}	\emptyset
11	{3}	\emptyset	\emptyset	\emptyset

На основе анализа строки 2 можно сузить домен атрибута X_1 до единственного значения ($X_1 - \{3\}$). Поскольку ранее было получено $X_4 - \{3\}$, то с учетом ограничений на рабочую силу суммарное количество персонала, задействованного для работ в T_1 и T_4 в 3:00, равно 9. Значит, никакая другая работа уже не может быть выпол-

нена одновременно с T_1 и T_4 . Другими словами, значение «3» можно исключить из областей определения других атрибутов. На текущем шаге имеем: $X_1 - \{3\}$, $X_2 - \{1,2\}$, $X_3 - \{2\}$, $X_5 - \{1,2\}$. Поскольку более не обнаружено нарушения кумулятивного ограничения и нельзя усечь домены, то перейдем к рассмотрению D -системы, оставшейся после «настройки» на новые домены (согласно **У4** и **У5**, уходят все строчки, кроме пятой, а в пятой остается лишь одна непустая компонента). Атрибуты X_1 и X_3 элиминируются из D -системы. Получим:

$$\begin{array}{c} X_1 - \{3\}, X_3 - \{2\}, X_4 - \{3\} \\ X_2 \quad X_5 \\ \{1,2\} \quad \{1,2\} \\]\emptyset \quad \{1\}[\end{array}$$

Откуда $X_5 = \{1\}$, следовательно, $X_2 = \{2\}$, поскольку работы T_2 и T_5 не могут выполняться одновременно. В результате нами получено решение:

$$X_1 - \{3\}, X_2 - \{2\}, X_3 - \{2\}, X_4 - \{3\}, X_5 - \{1\}.$$

Другими словами, в 1:00 выполняется T_5 , в 2:00 выполняется одновременно T_2 и T_3 , в 3:00 одновременно производятся работы T_1 и T_4 .

В данной задаче в качестве дополнительного правила редукции используется кумулятивное ограничение, что позволяет решить задачу за полиномиальное время, то есть без организации ветвления.

Заключение

В работе рассмотрены возможности модификации предложенных ранее автором базовых методов вывода на ограничениях, представимых в матрицеподобных структурах АК. Эти методы по-прежнему опираются на концепцию усеченного домена, но для обработки ограничений конкретной предметной области используются дополнительные правила редукции пространства поиска, что позволяет существенно ускорить решение задачи. В настоящее время имеется классификация видов глобальных ограничений, что дает возможность создавать типовые алгоритмы для работы с ними. Преимущество применения АК при реализации вывода на ограничениях заключается в том, что они позволяют решать задачи, для которых классические методы не могут найти решения.

чениях состоит в том, что не требуется предварительно транслировать ограничения в бинарные отношения (графы и т.п.), поскольку процесс поиска не сводится к выводу на графах, а опирается на матричные свойства структур АК.

СПИСОК ЛИТЕРАТУРЫ

- [Зуенко, 2013] Зуенко А.А. Матрицеподобные вычисления в задачах удовлетворения ограничений / А.А. Зуенко // Шестая Всероссийская мультиконференция по проблемам управления (30 сентября – 5 октября 2013 г.). Материалы мультиконференции: в 4 т. – Ростов-на-Дону: Издательство Южного федерального университета, 2013. Т.1.
- [Зуенко, 2014] Зуенко А.А. Распространение ограничений и эвристический поиск с применением матричного представления конечных предикатов // Труды четырнадцатой национальной конференции по искусственному интеллекту (КИИ-2014).
- [Кулик и др., 2010] Кулик Б.А., Зуенко А.А., Фридман А.Я. Алгебраический подход к интеллектуальной обработке данных и знаний – СПб.: Изд-во Политехнического ун-та, 2010.
- [Щербина, 2012] Щербина О.А. Удовлетворение ограничений и программирование в ограничениях / О. А. Щербина // препринт, 2012.
- [Aggoun et al., 1993] Aggoun A. Beldiceanu N. Extending CHIP in order to solve complex scheduling and placement problems // Mathl. Comput. Modelling. 1993. V. 17.
- [Bartak, 1999] Bartak R. Constraint Programming: In Pursuit of the Holy Grail // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, 1999.
- [R'egin, 1996] R'egin J. Generalized arc consistency for global cardinality constraint // Proceedings of the Thirteenth National Conference on Artificial Intelligence. Portland, 1996.
- [Rossi et al., 2008] Rossi F., van Beek P., Walsh T. Chapter 4 «Constraint Programming» in Foundations of Artificial Intelligence. // Handbook of Knowledge Representation / F. van Harmelen, V. Lifschitz, B. Porter (eds.) 2008. V. 3.
- [Russel et al., 2003] Russel S. and Norvig P. Artificial Intelligence: A Modern Approach. Second edition. – Prentice Hall, 2003.

УДК 007:519.816

**РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ ВРЕМЕННОГО ВЫВОДА
НА БАЗЕ ТОЧЕЧНО-ИНТЕРВАЛЬНОЙ ВРЕМЕННОЙ
ЛОГИКИ ДЛЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ
РЕШЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ**

И.Е. Куриленко (*kurilenkoiy@mpei.ru*)

ФГБОУ ВПО «Национальный исследовательский университет
«МЭИ», Москва

Д.А. Шорникова (*daria-chan@mail.ru*)

ФГБОУ ВПО «Национальный исследовательский университет
«МЭИ», Москва

Работа посвящена исследованию алгоритмов вывода в точечно-интервальной временной логике. Описывается механизм вывода в данной логике, базирующийся на переходе от представления информации о времени в виде ограничений к точечному графу. Приводится архитектура прототипа подсистемы временного вывода на основе точечно-интервальной временной логики для интеллектуальных систем поддержки принятия решений реального времени.

Введение

В процессе разработки современных интеллектуальных приложений, таких как интеллектуальные системы поддержки принятия решений реального времени (ИСППР РВ), часто необходимо представлять и обрабатывать большой объем информации, изменяющейся со временем [Eremeev et.al, 2003]. При решении задач во многих проблемных областях искусственного интеллекта (ИИ), таких как планирование, «понимание» естественного языка, диагностика и объяснение, прогнозирование, требуется иметь развитые средства представления и анализа времени и временных зависимостей, временных ограничений. В ряде флагманских работ по искусственному интеллекту констатируется определяющая роль представления знаний о времени и пространстве [Поспелов и др., 1989]. Выполненные за достаточно длительный промежуток исследования привели к появлению достаточно большого количества временных логик и моделей, отличающихся способами введения фактора времени и выразительными возможностями [Еремеев

и др., 2004]. Однако их область применения часто сдерживается сложностью алгоритмов вывода – ведь для многих систем ИИ, оперирующих большим количеством информации и действующих в условиях жестких временных ограничений, одним из важных критериев качества является приемлемое время реакции на запросы пользователя или изменение информации, поступающей в систему [Еремеев и др., 2010]. В связи с этим весьма актуальной является выработка алгоритмов вывода и поиск моделей представления информации о времени, обеспечивающих баланс между необходимым уровнем выразительности и производительности [Кандрашина и др., 2014].

В рамках данной работы рассматриваются использование точечно-интервальной темпоральной логики и реализация алгоритмов вывода для построения подсистемы временных рассуждений, которая может быть использована для решения задач временного вывода в составе современных интеллектуальных систем.

1. Представление информации о времени

Одним из способов явного моделирования времени является представление информации как зависимостей, или ограничений, между временными примитивами [Ladkin, 1987]. В этом случае определяется структура $Z = (V, D, BTR, C)$ [Еремеев и др., 2004], где:

- $V = \{V_1, V_2, \dots, V_m\}$ – конечное множество временных переменных;
- D – область значений временных переменных;
- $BTR = \{r_1, r_2, \dots, r_n\}$ – конечное множество взаимоисключающих бинарных базовых временных ограничений, полное объединение которых является универсальным ограничением U (не накладывающим каких-либо ограничений);
- $C = \{C_{ij} | C_{ij} = \{r_1, r_2, \dots, r_k\}; k > 0; r_1, r_2, \dots, r_k \in BTR; i, j \leq m\}$, конечное множество ограничений, где C_{ij} – ограничение над временными переменными V_i и V_j , интерпретируемое как $(V_i, r_1, V_j) \dots (V_i, r_k, V_j)$. Если C_{ij} состоит только из одного дизъюнкта, то оно называется *точным*.

Элементы множества V могут интерпретироваться как моменты, интервалы времени или длительности. Область значений переменных D , соответствующих моментам времени и длительностям,

представляет собой множество чисел, а для интервальных переменных – множество упорядоченных пар значений.

Задача согласования временных ограничений (ЗСВО) состоит в *определении согласованности* [Троицкий, 2001]. Для этого необходимо проверить, существует ли такой выбор точного временного ограничения $C_{ij}^* \subset C_{ij}$ для каждой пары временных переменных V_i и V_j , что они не противоречат друг другу.

В некоторых случаях требуется обеспечить поиск всех согласованных сценариев.

Для согласования ЗСВО можно построить минимальное представление. Переход к минимальной ЗСВО предполагает перевод всех ее ограничений в минимальные. Минимальным ограничением C_{ij}^{min} для временных переменных V_i и V_j называется множество, состоящее только из выполнимых ограничений для V_i и V_j .

Достаточно часто после решения ЗСВО требуется иметь возможность *определить выполнимые ограничения* между заданной парой временных переменных.

2. Расширенная точечно-интервальная модель времени

Расширенная точечно-интервальная модель является синтезом точечной и интервальной моделей [Zaidi et al., 2006]. В этой модели временными примитивами являются моменты времени и временные. Момент обозначает мгновенное событие, а интервал – длиющееся во времени. Обозначим моменты времени как pZ , pV , а интервалы – $X=(sX, eX)$, где sX – момент времени, определяющий начало временного интервала, а eX – момент времени, определяющий конец временного интервала. В данной модели возможны следующие типы временных ограничений: «момент-момент», «момент-интервал», «интервал-интервал» (ограничения типа «момент-интервал» и «интервал-интервал» можно представить в виде точечных ограничений между началом и концом интервала).

В набор базовых временных ограничений входят как качественные:

- базовые точечные ограничения: одновременно ($=$), раньше ($<$), позже ($>$), раньше или одновременно (\leq), позже или одновременно (\geq), не одновременно (\neq);

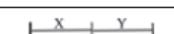
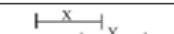
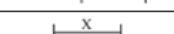
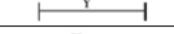
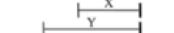
$X b Y$	
$Y a X$	
$X m Y$	
$X o Y$	
$X d Y$	
$X s Y$	
$X f Y$	
$X = Y$	

Рис. 1. Интервальные ограничения

$X b Y$	
$X a Y$	
$X s Y$	
$X f Y$	
$X d Y$	

Рис. 2. Точечно-интервальные ограничения

- базовые интервальные ограничения: *before* (*b*), *after* (*a*), *meets* (*m*), *overlaps* (*o*), *during* (*d*), *starts* (*s*), *finishes* (*f*), *equal* (=) [Allen, 1983] (рис. 1);

- базовые точечно-интервальные ограничения: *before* (*b*, <), *after* (*a*, >), *starts* (*s*), *finishes* (*f*), *during* (*d*) (рис. 2), так и метрические ограничения.

Метрические ограничения определяются на основе следующих функций [Zaidi et al., 2006]:

$Length[sX, eY]=d$, $d \in R^+$ – длина интервала;

$Stamp(pX)=t$, $t \in R$ – точное положение момента времени.

Для решения ЗСВО в точечно-интервальной временной логике выполняется переход к точечному графу [Zaidi et al., 2008].

3. Использование точечных графов в задачах временного вывода

Вершинам точечного графа соответствуют моменты времени, а ребра – ограничения [Zaidi et al., 2006]. Формально точечный граф PG – это орграф (V, E_a, D, T) , где:

V – множество вершин, каждая из которых представляет момент времени на числовой шкале. Моменты p_i, p_j, \dots, p_n могут быть представлены одним моментом $[p_i; p_j; \dots; p_n]$, если обозначают одну и ту же точку на числовой шкале времени.

$$E_a : E_a = E \cup E_{\leq}$$

где E – множество ребер, и каждое ребро e_{12} между вершинами v_1 и v_2 , также обозначаемое (v_1, v_2) , представляет ограничение $v_1 < v_2$. Такие ребра называют *LT-ребрами*;

E_{\leq} – множество ребер, где каждое ребро e_{12} между вершинами v_1 и v_2 , также обозначаемое (v_1, v_2) , представляет ограничение $v_1 \leq v_2$. Такие ребра называют *LE-ребрами*;

D – функция длины ребра: $E \rightarrow R^+$;

T – функция точного положения момента времени вершины: $V \rightarrow R$.

Если между двумя вершинами p и q находится LT-ребро, то точка, представленная вершиной p , находится на шкале времени строго раньше точки, представленной вершиной q . Если же между вершинами p и q находится LE-ребро, точка, соответствующая вершине p , находится на временной шкале раньше или одновременно с точкой, соответствующей вершине q . Если ребро взвешено числовым значением, это означает, что длина соответствующего ему временного интервала равна этому значению.

Алгоритм перехода от ЗСВО к точечному графу достаточно прост. На первом аге все интервальные переменные представляются как пара граничных моментов времени, связанных ограничением $<$. На втором этапе ограничения типа «момент-интервал» и «интервал-интервал» преобразуются в точечные ограничения. Каждый момент v становится вершиной v и каждое точечное ограничение $<$ между моментами v_1, v_2 становится ребром (v_1, v_2) .

После построения точечного графа осуществляется его упрощение. Имеющаяся метрическая информация может быть использована для уменьшения количества вершин в точечном графе и сокращения поискового пространства. Любые вершины, обладающие одним и тем же значением функции T , склеиваются. Для всех вершин v_i, v_j , таких, что $T(v_i) < T(v_j)$ необходимо построить ребро длины $D(v_i, v_j) = T(v_j) - T(v_i)$.

Пусть v^* – множество вершин V таких, что: $\forall v_j \in v^* (v, v_j) \in E_a$.

Пусть $*v$ – множество вершин V таких, что: $\forall v_j \in v^* (v_j, v) \in E_a$.

Для каждой вершины $v_i \in V$ проверяется множество v_i^* : если для вершин v_j, v_k из множества v_i^* выполнено:

1. $D(v_i, v_j) < D(v_i, v_k)$, тогда ребро (v_i, v_k) заменяется ребром (v_j, v_k) , притом $D(v_j, v_k) = D(v_i, v_k) - D(v_i, v_j)$, и вершина v_k удаляется из v_i^* .

2. $D(v_i, v_j) = D(v_i, v_k)$, тогда вершины vj и vk могут быть объединены в ' $v_j;v_k$ ' и $D(v_j;v_k) = D(v_i, v_k) = (v_i, v_j)$.

Для каждой вершины $v_i \in V$ проверяется множество $*v_i$: если для вершин v_j, v_k из множества $*v_i$ выполнено:

1. $D(v_j, v_i) < D(v_k, v_i)$, тогда ребро (v_k, v_i) заменяется ребром (v_k, v_j) , притом $D(v_k, v_j) = D(v_k, v_i) - D(v_j, v_i)$, и вершина v_k удаляется из $*v_i$.

2. $D(v_j, v_i) = D(v_k, v_i)$, тогда вершины vj и vk могут быть объединены в ' $v_j;v_k$ ' и $D(vj;v_k, v_i) = D(v_k, v_i) = (v_j, v_i)$.

На рис. 3 показан пример упрощения графа с помощью операций склейки вершин и ребер. В исходном графе $D(sY, eX) = D(sY, eZ)$, поэтому вершины eX, eZ могут быть склеены в одну $eX;eZ$.

Для проверки согласованности темпоральной информации следует проверить граф на ацикличность. Наличие циклов в графе указывает на несогласованность данных и противоречивость временных ограничений [Gerevini et al., 1995].

Алгоритм вычисления выводимых ограничений опирается на алгоритм определения выполнимых ограничений между моментами времени. Чтобы определить ограничение между двумя моментами p и q , необходимо:

1. Выяснить, существует ли ограничение $(p, \{<\}, q)$.
2. Найти длину интервала (p, q) .
3. Найти точные отметки моментов времени p, q на временной шкале

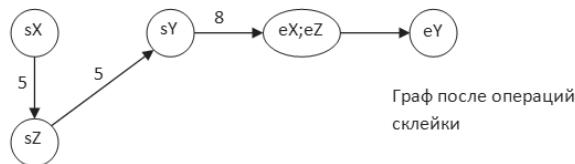
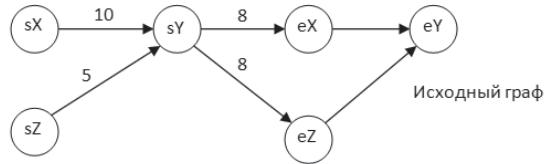


Рис. 3. Упрощение с помощью операций склейки

Для поиска путей в ориентированном графе можно использовать алгоритмы Флойда или Дейкстры, но следует помнить, что при решении задачи возникает необходимость искать путь в инвертированном графе (графе, в котором все ребра имеют противоположное направление).

4. Реализуемый прототип

Рассмотренные алгоритмы реализованы в рамках прототипа системы временных рассуждений (СВР), обеспечивающего:

- представление и хранение временной информации;
- визуализацию точечных графов, в том числе и на промежуточных этапах решения ЗСВО;
- решение задачи согласованности;
- построение минимальной ЗСВО;
- ответы на запросы пользователя о метрической информации.

Также в прототип интегрирован дружественный пользовательский интерфейс, позволяющий вводить временные ограничения и метрическую информацию простым для пользователя образом.

Упрощенная архитектура прототипа изображена на рис. 4.

Основной элемент системы – решатель, который обрабатывает темпоральную информацию, находит решение задачи согласованности и выполняет запросы пользователя.



Рис. 4. Архитектура СВР

Парсер преобразовывает введенную пользователем информацию во внутреннее представление и передает в хранилище темпоральной информации. Из хранилища информация считывается решателем и трансформируется в форму точечного графа для применения к ней алгоритмов вывода.

Система поддерживает интерактивное взаимодействие с пользователем. Пользователь может дополнять информацию, и система автоматически проверяет ее «на лету». В случае возникновения противоречий между временными ограничениями система выдаст пользователю предупреждения.

Заключение

Построение и реализация алгоритмов, реализующих вывод в расширенной точечно-интервальной временной логике, а также создание повторно используемых компонент временного вывода для интеллектуальных систем позволяют расширить качество решения ряда задач искусственного интеллекта. Рассмотренные в рамках данной работы алгоритмы реализованы в составе прототипа подсистемы временных рассуждений. Проведенные с помощью прототипа эксперименты показали, что алгоритмы упрощения точечных графов позволяют снизить объем неточной информации и хорошо сказываются на объеме поискового пространства для последующих алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

[Eremeev et.al, 2003] Eremeev A.P., Vagin V.N. A Real-time Decision Support System Prototype for Management of a Power Block // International Journal «Information Theories & Applications», 2003, Vol. 10, Number 3, pp. 248–255.

[Башлыков и др., 2001] Башлыков А.А., Еремеев А.П. Экспертная диагностическая система как компонент интеллектуальной системы поддержки принятия решений реального времени // Новости искусственного интеллекта. 2001. № 3.

[Еремеев и др., 2010] Еремеев А.П., Куриленко И.Е. Средства темпорального вывода для интеллектуальных систем реального времени // Интеллектуальные системы. Коллективная монография. Выпуск четвертый. М.: Физматлит, 2010.

[Еремеев и др., 2004] Еремеев А.П., Троицкий В.В. Концепции и модели представления времени и их применение в интеллектуальных системах // Новости искусственного интеллекта. 2004. №1.

[Поспелов и др., 1989] Кандрашина Е. Ю., Литвинцева Л.В., Поспелов Д.А. Представление знаний о времени и пространстве в интеллектуальных системах // М.: Наука, 1989.

[Рыбина и др., 2014] Рыбина Г.В., Мозгачев А.В. Реализация темпорального вывода в динамических интегрированных экспертных системах // Искусственный интеллект и принятие решений. 2014. №1.

[Троицкий, 2001] Троицкий В.В. Временные рассуждения, как задача согласования ограничений // Радиоэлектроника, электротехника и энергетика: Седьмая международная научно-техническая конференция студентов и аспирантов МЭИ-2001: Тез. докл. В 3-х т. М.: Изд. МЭИ, 2001.

[Allen, 1983] Allen J.F. Maintaining Knowledge about Temporal Intervals. // Communications of the ACM. 1983. Vol. 26, no. 11.

[Ladkin, 1987] Ladkin P.B. The Logic of Time Representation // Ph.D. thesis. University of California at Berkeley. – Berkeley, CA, US, 1987.

[Zaidi et al., 2006] Zaidi, A. K. and Lee W. Wagenhals, Planning Temporal Events Using Point Interval Logic // Special Issue of Mathematical and Computer Modeling: 43, Elsevier, 2006.

[Zaidi et al., 2008] Mashhood Ishaque, Faisal Mansoor, Abbas K. Zaidi: An Inference Mechanism for Point-Interval Logic. FLAIRS Conference 2008: 600–605.

[Gerevini et al., 1995] Alfonso Gerevini, Lenhart K. Schubert: Efficient Algorithms for Qualitative Reasoning about Time. Artif. Intell. 74(2): 207–248 (1995).

УДК 512.563.6

О РЕШЕНИЯХ ПРОДУКЦИОННО-ЛОГИЧЕСКОГО УРАВНЕНИЯ НА БУЛЕВОЙ РЕШЕТКЕ

С.Д. Махортов (*sd@expert.vrn.ru*)

Воронежский государственный университет

И.Ю. Иванов (*hourIsCorp@gmail.com*)

Воронежский государственный университет

Вводится класс продукционно-логических уравнений на конечной булевой решетке. Рассматриваются вопросы разрешимости таких уравнений. Вводятся понятия канонического отношения на булевой решетке и структурного расслоения канонического отношения, которые позволяют свести поиск приближенного решения уравнения к задаче обхода вершин ориентированного графа. Результаты могут быть применены для оптимизации логического вывода и верификации соответствующих баз знаний.

Введение

Идея представления и исследования логических систем алгебраическими методами была заложена в работах А. Линденбаума, А. Тарского. Согласно их теории, множество формул пропозиционального языка нулевого порядка рассматривается как алгебра, операции которой соответствуют логическим связкам этого языка [Расева и др., 1972]. При этом общая алгебраическая логика, представляя эффективный аппарат для исследования самих логических теорий, в ряде случаев оказывается недостаточно эффективной в силу своей универсальности. К таковым относится область широко распространенных на практике логических систем продукционного типа. Они основываются на правилах (продукциях) вида « A порождает B », где A и B – элементы некоторой иерархии. Семантика продукции обычно импликативна: «если справедливо A , то справедливо B », однако возможны и другие интерпретации подобных правил [Чечкин, 1991], [Махортов, 2009a].

Естественной и эффективной моделью представления знаний служат математические решетки [Тейз и др., 1990]. В монографии [Махортов, 2009b] разработан математический аппарат, позволяющий формально исследовать продукционно-логические системы.

За основу может быть взята булева решетка с введенным на ней дополнительным отношением, представляющим множество продукции. Полученная алгебраическая система названа LP-структурой (Lattice Production Structure). В [Махортов, 2004] был введен и исследован класс производственно-логических уравнений, позволяющих оптимизировать обратный логический вывод. Однако семантика этих уравнений основывается всего на двух логических связках – импликации и конъюнкции. Таким минимальным набором логических связок ограничиваются типовые производственные системы.

В настоящей работе вводится новый класс уравнений в LP-структуре. В отличие от [Махортов, 2004], он обладает полным набором связок пропозиционального языка. Обоснован способ нахождения приближенных решений. Ввиду ограниченности объема статьи результаты даны без доказательств. Уравнения могут применяться для оптимизации логического вывода в производственных системах, правила которых содержат не только конъюнкции, но также дизъюнкции и отрицания.

Логический вывод в LP-структурах перекликается с нахождением функциональных зависимостей в реляционных базах данных. При их выводе используются дедуктивные правила Армстронга ([Armstrong, 1974]), применяемые к элементам булеана. Однако из «решеточных» операций используется лишь объединение, и набор правил Армстронга существенно беднее правил вывода в LP-структурах. Близким к теории LP-структур можно также считать FCA-анализ формальных понятий [Ganter, 1999], применяемый в исследованиях двухмерных данных с семантикой «объекты-признаки». Он также основан на решетках и рассматривает бинарные отношения между множествами. Однако эти две теории существенно различаются. В частности, импликации FCA порождаются решеточным порядком и поэтому не образуют циклов, в то время как отношения в LP-структурах не имеют подобных ограничений (подробнее об отличиях см. в [Махортов, 2009b]).

1. Основные сведения о решетках и бинарных отношениях

Необходимая информация о решетках содержится в [Биркгоф, 1984], [Гретцер, 1981]. Напомним лишь основные понятия.

Решеткой называется множество \mathbb{F} , частично упорядоченное некоторым отношением \leq , причем для любой пары $x, y \in \mathbb{F}$ су-

ществуют их точная нижняя (пересечение $x \wedge y$) и точная верхняя (объединение $x \vee y$) грани. Решетка \mathbb{F} ограничена, если в ней существуют наименьший и наибольший элементы O, I соответственно.

Элемент x решетки \mathbb{F} называется неразложимым, если из $x = x_1 \vee x_2$, $x_1, x_2 \in \mathbb{F}$ следует $x = x_1$ или $x = x_2$; конеразложимым, если из $x = y_1 \wedge y_2$, $y_1, y_2 \in \mathbb{F}$ следует $x = y_1$ или $x = y_2$.

Элемент x ограниченной решетки \mathbb{F} называется атомом, если $O < x$ и не существует такого $x_1 \in \mathbb{F}$, что $O < x_1 < x$ (x покрывает O); коатомом, если $x < I$ и не существует $x_2 \in \mathbb{F}$, что $x < x_2 < I$ (I покрывает x).

Решетка \mathbb{F} называется решеткой с дополнениями, если для любого элемента $x \in \mathbb{F}$ существует элемент $x' \in \mathbb{F}$, называемый дополнением элемента x , такой, что $x \wedge x' = O$, $x \vee x' = I$.

Решетка \mathbb{F} называется дистрибутивной, если для любых элементов $x, y, z \in \mathbb{F}$ выполняются равенства:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \quad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

Булевой решеткой называется дистрибутивная решетка с дополнениями. Любой элемент конечной булевой решетки \mathbb{F} , отличный от O , имеет единственное представление в виде объединения атомов, а любой элемент, отличный от I , имеет единственное представление в виде пересечения коатомов.

Бинарное отношение R (далее – просто отношение) на булевой решетке называется [Махортов, 2009б]: *контрапозиционным*, если из $(x, y) \in R$ следует $(y', x') \in R$; *дистрибутивным*, если из $(x, y_1), (x, y_2) \in R$ следует $(x, y_1 \wedge y_2) \in R$, а из $(x_1, y), (x_2, y) \in R$ следует $(x_1 \vee x_2, y) \in R$; *продукционно-логическим* (далее – просто логическим), если оно содержит отношение \leq , а также контрапозиционно, дистрибутивно и транзитивно.

Логическим замыканием произвольного отношения R называется наименьшее логическое отношение, содержащее R . Два отношения R_1, R_2 эквивалентны ($R_1 \sim R_2$), если они имеют общее логическое замыкание. LP-структурой нулевого порядка называется булева решетка с заданным на ней логическим отношением [Махортов, 2009б].

Упорядоченная пара (x, y) элементов булевой решетки \mathbb{F} ло-

гически связана отношением $R \subseteq \mathbb{F} \times \mathbb{F}$ (обозначается $x \xrightarrow{R} y$, [Махортов, 2009b]), если выполнено одно из условий:

$$(x, y) \in R; \quad (1.1)$$

$$x \leq y; \quad (1.2)$$

$$y' \xrightarrow{R} x'; \quad (1.3)$$

$$\text{существуют такие } y_1, y_2 \in \mathbb{F}, \text{ что} \quad (1.4)$$

$$y_1 \wedge y_2 = y, x \xrightarrow{R} y_1, x \xrightarrow{R} y_2;$$

$$\text{существуют такие } x_1, x_2 \in \mathbb{F}, \text{ что} \quad (1.5)$$

$$x_1 \vee x_2 = x, x_1 \xrightarrow{R} y, x_2 \xrightarrow{R} y;$$

$$\text{существует такой } z \in \mathbb{F}, \text{ что } x \xrightarrow{R} z, z \xrightarrow{R} y. \quad (1.6)$$

В [Махортов, 2009b] доказано, что для любого отношения R на булевой решетке логическое замыкание совпадает с множеством \xrightarrow{R} всех упорядоченных пар, логически связанных отношением R .

2. Каноническое отношение и начальное множество

Будем рассматривать только конечные булевые решетки. Известно [Гретцер, 1981], что такая решетка изоморфна булевой решетке всех подмножеств некоторого конечного универсума.

Итак, далее всюду $\mathbb{F} = 2^F$ – булеан конечного множества F мощности $n > 0$. Воспользуемся обозначениями теории множеств: элементы решетки \mathbb{F} обычно будем обозначать большими латинскими буквами; если $A, B \in \mathbb{F}$, то $A \subseteq B$ – принадлежность пары (A, B) отношению решеточного частичного порядка \leq (оно совпадает с отношением вложения), $A \cap B$ – пересечение A и B , $A \cup B$ – объединение A и B , \bar{A} – дополнение элемента A (т. е. $\bar{A} = F \setminus A$).

Заметим, что наименьший элемент решетки \mathbb{F} есть пустое подмножество \emptyset множества F , наибольший элемент – само множество F , атомы – одноэлементные подмножества множества F , коатомы – $(n-1)$ -элементные подмножества множества F . Таким образом, решетка \mathbb{F} имеет 2^n элементов, n атомов и n коатомов.

Коатомы решетки \mathbb{F} будем обозначать маленькими латинскими буквами. *Коразложением* элемента $A \in \mathbb{F}$, отличного от F , будем называть его представление в виде пересечения коатомов.

Пусть R – отношение на решетке. Далее всюду под логическим отношением будем понимать дистрибутивное, транзитивное отношение, содержащее \subseteq , не требуя при этом свойства контрапозиционности. Рассмотрим отношение $\xrightarrow{R}{}_0$, которое строится аналогично \xrightarrow{R} , но исключая контрапозиционное правило 1.3. Очевидно, $\xrightarrow{R}{}_0 \subseteq \xrightarrow{R}$.

Аналогично [Махортов, 2009б] можно показать, что $\xrightarrow{R}{}_0$ есть наименьшее логическое отношение, содержащее R , т. е. $\xrightarrow{R}{}_0$ – логическое замыкание R . Эквивалентность R_1, R_2 определим равенством $\xrightarrow{R_1}{}_0 = \xrightarrow{R_2}{}_0$ и обозначим $R_1 \sim_0 R_2$.

Определение 2.1. Отношение R называется *каноническим*, если правые части всех его пар являются коатомами.

Теорема 2.1. Пусть R – отношение на решетке \mathbb{F} и $(A, B) \in R$, причем $B = \bigcap_{i=1}^k B_i$. Тогда отношение R' , полученное из R путем замены пары (A, B) множеством пар $\{(A, B_i) | i = 1, \dots, k\}$, эквивалентно R .

Следствие 2.1. Для любого отношения R на решетке \mathbb{F} , пары которого не содержат F в правой части, существует эквивалентное каноническое отношение.

Начальное множество, определение которого дается ниже, с точки зрения обратного логического вывода соответствует множеству фактов, не выводимых в продукционной системе на основании имеющегося множества правил.

Определение 2.2. Конеразложимый элемент $X \in \mathbb{F}$ называется *начальным* при отношении R , если для любой пары $A \xrightarrow{R}{}_0 B$ из $B = B_1 \cap X$ следует $A = A_1 \cap X$, $A_1, B_1 \in \mathbb{F}$.

Замечание 2.1. Непосредственно из определения следует, что элемент F является начальным при любом отношении R , так как для любого $M \in \mathbb{F}$ справедливо $M = M \cap F$.

Определение 2.3. Начальным множеством решетки \mathbb{F} при отношении R называется подмножество $\mathbb{F}_0(R)$ (или просто \mathbb{F}_0), образованное по следующим правилам:

- 1) если X – конеразложимый начальный при отношении R элемент решетки \mathbb{F} , то $X \in \mathbb{F}_0(R)$;
- 2) если $A, B \in \mathbb{F}_0(R)$, то $A \cap B \in \mathbb{F}_0(R)$.

Лемма 2.1. Если $x \in \mathbb{F}$ – коатом и $M \subseteq x$, $M \in \mathbb{F}$, то x участвует в коразложении элемента M .

Теорема 2.2. Конеразложимый элемент $X \in \mathbb{F}$ является начальным при отношении R тогда и только тогда, когда для любой пары $(A, B) \in R$ из $B = B_1 \cap X$ следует $A = A_1 \cap X$.

3. Продукционно-логическое уравнение

Введем понятие продукционно-логического уравнения на булевой решетке. Задача поиска решения такого уравнения соответствует обратному логическому выводу в продукционной системе, логика которой использует полный набор логических связок propositionalного языка.

Если для $A, B \in \mathbb{F}$ выполнено $A \xrightarrow{R} B$, то будем говорить, как это принято в теории отображений, что A является прообразом B при отношении \xrightarrow{R} , B является образом A при отношении \xrightarrow{R} и писать $R^{L_0}(A) = B$ (под R^{L_0} подразумевается отношение \xrightarrow{R}).

Определение 3.1. Продукционно-логическим (далее – просто логическим) уравнением на решетке \mathbb{F} называется уравнение вида

$$R^{L_0}(X) = B, \quad (3.1)$$

где $B \in \mathbb{F}$ – заданный элемент, X – искомый в \mathbb{F} элемент.

Определение 3.2. Решением уравнения 3.1 называется максимальный прообраз B из начального множества \mathbb{F}_0 . Приближенным решением уравнения (3.1) называется любой прообраз B из \mathbb{F}_0 . По определению решение является и приближенным решением.

Рассмотрим уравнение

$$R^{L_0}(X) = B_1 \cap B_2 \quad (3.2)$$

и соответствующие ему уравнения

$$R^{L_0}(X) = B_1, \quad R^{L_0}(X) = B_2. \quad (3.3)$$

Теорема 3.1. Если X_1, X_2 – решения уравнений (3.3), то $X = X_1 \cap X_2$ – решение уравнения (3.2).

Теорема 3.1 позволяет свести поиск (точного или приближенного) решения уравнения (3.1) к поиску (точных или приближенных) решений множества уравнений с правой частью простейшего вида. Предположим, что элемент B отличен от F . Тогда существует ко-разложение $B = b_1 \cap \dots \cap b_k$, где $b_i \in \mathbb{F}$, $i = 1, \dots, k$ – коатомы. Если X_i

- решение уравнения $R^{L_0}(X) = b_i$, то по теореме 3.1 $X = X_1 \cap \dots \cap X_k$
- решение уравнения (3.1). Аналогично, если \tilde{X}_i – приближенное решение уравнения $R^{L_0}(X) = b_i$, то $\tilde{X} = \tilde{X}_1 \cap \dots \cap \tilde{X}_k$ – приближенное решение уравнения (3.1).

Предположим, что левые и правые части пар отношения R не содержат элемент F . Тогда без ограничения общности можно считать, что R – каноническое отношение, не содержащее пар отношения \subseteq . Если это не так, то по следствию 2.1 R можно привести к каноническому виду и исключить из него пары отношения \subseteq , получив в конечном итоге отношение, эквивалентное R .

Рассмотрим вопрос поиска приближенного решения уравнения

$$R^{L_0}(X) = b, \quad (3.4)$$

где $b \in \mathbb{F}$ – заданный коатом.

Введем понятие структурного расслоения отношения R на виртуальные слои аналогично тому, как это делается в [Махортов, 2009б] для канонического отношения на атомно-порожденной решетке. Сначала разобьем отношение на подмножества, в каждое из которых отнесем пары с одинаковыми коатомами в правой части. Это возможно, так как R – каноническое отношение. Пусть $\{R^p \mid p = 1, \dots, P\}$ – описанное разбиение отношения R .

Определение 3.3. Слоем R_i в отношении R называется подмножество, образованное парами, взятыми по одной из каждого множества R^p , $p = 1, \dots, P$.

Пусть $\{R_t | t = 1, \dots, T\}$ – совокупность всех слоев в отношении R .

Так как $R_t \subseteq R$, то $\xrightarrow{R_t} \subseteq \xrightarrow{R}$. Поэтому если для некоторого $\widetilde{X}_t \in \mathbb{F}_0(R)$ выполнено $\widetilde{X}_t \xrightarrow{R_t} b$, будем говорить, что \widetilde{X}_t – приближенное решение уравнения (3.4), порожденное слоем R_t . Таким образом, если $\widetilde{X}_{t_s}, s=1, \dots, q$ – приближенные решения уравнения (3.4), порожденные теми слоями, в которых эти приближенные решения существуют, то $\tilde{X} = \widetilde{X}_{t_1} \cup \dots \cup \widetilde{X}_{t_q}$ – приближенное решение уравнения (3.4) (т. к. приближенные решения образуют подрешетку). При этом \tilde{X} есть наибольший элемент среди приближенных решений уравнения (3.4), которые можно получить с помощью операций \cap, \cup из $\widetilde{X}_{t_s}, s=1, \dots, q$ [Гретцер, 1981].

Итак, задача поиска приближенного решения уравнения (3.1) разбита на множество подзадач, каждая из которых сводится к по-

иску приближенного решения уравнения вида (3.4) в отдельном слое.

Покажем, как можно осуществить поиск приближенного решения уравнения (3.4), порожденного слоем R_t . Для этого построим ориентированный граф G_{R_t} , вершинам которого поставим во взаимно однозначное соответствие коатомы решетки \mathbb{F} , причем в дальнейшем не будем различать вершину графа G_{R_t} и коатом решетки \mathbb{F} , ей соответствующий. Каждой паре $(A, b) \in R_t$ сопоставим множество дуг (a_i, b) , $i = 1, \dots, r$ графа G_{R_t} тогда и только тогда, когда $a_1 \cap \dots \cap a_r$ – коразложение элемента A . Отметим, что построенный граф не содержит петель, так как отношение R , равно как и его подмножество R_t , не содержит пар отношения \subseteq (это следует из леммы 2.1).

Теорема 3.2. Если коатом b не является входной вершиной графа G_{R_t} и граф G_{R_t} ацикличен, то множество всех входных вершин графа G_{R_t} , из которых достижима вершина b , состоит из коатомов коразложения приближенного решения уравнения (3.4), порожденного слоем R_t .

Заключение

В настоящей работе введен новый класс продукционно-логических уравнений на конечных булевых решетках и рассмотрены вопросы их разрешимости. Предложен и обоснован способ поиска приближенного решения уравнения, основывающийся на представлении слоев отношения в виде ориентированных графов и обходе вершин этих графов.

Полученные результаты (подобно [Болотова и др., 2011]) могут быть использованы для оптимизации обратного вывода в продукционно-логических системах, в правилах которых используется полный набор логических связок пропозиционального языка. Таким образом, расширена область применения метода релевантного обратного вывода, существенно снижающего интенсивность обмена с внешней памятью.

СПИСОК ЛИТЕРАТУРЫ

[Расева и др., 1972] Расева Е., Сикорски Р. Математика метаматематики: пер. с англ. – М.: Наука, 1972. – 591 с.

[Чечкин, 1991] Чечкин А. В. Математическая информатика. – М.: Физматлит, 1991. – 416 с.

[Махортов, 2009а] Махортов С.Д. LP-структуры на решетках типов и некоторые задачи рефакторинга // Программирование. – 2009, № 4. – С. 5–14.

[Тейз и др., 1990] Тейз А., Грибомон П. и др. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: пер. с франц. – М.: Мир, 1990. – 432 с.

[Махортов, 2009б] Махортов С. Д. Математические основы искусственного интеллекта: теория LP-структур для построения и исследования моделей знаний продукционного типа. – М.: МЦНМО, 2009. – 304 с.

[Махортов, 2004] Махортов С. Д. Логические уравнения на решетках // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2004, № 2. – С. 170–178.

[Armstrong, 1974] Armstrong W. W. Dependency Structure of Database Relationships / W. W. Armstrong // Proc. IFIP Congress. – Geneva, 1974. – Р. 580–583.

[Ganter, 1999] Ganter B., Wille R Formal Concept Analysis: mathematical foundations. – Heidelberg : Springer, 1999.

[Биркгоф, 1984] Биркгоф Г. Теория решеток: пер. с англ. – М.: Физматлит, 1984. – 568 с.

[Гретцер, 1981] Гретцер Г. Общая теория решеток: пер. с англ. – М.: Мир, 1981. – 456 с.

[Болотова и др., 2011] Болотова С. Ю., Махортов С.Д. Алгоритмы релевантного обратного вывода, основанные на решении продукционно-логических уравнений // Искусственный интеллект и принятие решений. – 2011. – № 2. – С. 40–50.

УДК 004.052.42

**МЕТОД УСКОРЕННОГО ЛОГИЧЕСКОГО ВЫВОДА
ДЛЯ РЕШЕНИЯ ЗАДАЧИ
ФОРМАЛЬНОЙ ВЕРИФИКАЦИИ¹**

В.Ю. Мельцов (*meltsov69@mail.ru*)
Г.А. Чистяков (*gennadiychistyakov@gmail.com*)
Вятский государственный университет, Киров

В работе приводится постановка логической задачи формальной верификации и кратко описывается специализированный метод логического вывода, предназначенный для решения этой задачи. Метод учитывает особенности структур посылок в базе знаний, что позволяет осуществлять поиск возможного решения наиболее эффективным образом.

Введение

В современном мире практически все сферы человеческой деятельности, так или иначе, связаны с вычислительной техникой. Управление огромным числом сложных процессов возлагается на ЭВМ, компьютеры применяются для решения как повседневных задач обеспечения жизнедеятельности, так и специфических задач в конкретных предметных областях. В связи с этим неизбежным становится повышение требований к надежности используемого программного обеспечения.

В последние двадцать лет активно развивается теория формальной верификации, изучающая методы, которые позволяют однозначно определять корректность программ и алгоритмов. Такой результат становится возможным благодаря применению строгих математических техник, способных, в отличие от повсеместно используемого тестирования, одновременно анализировать все возможные траектории выполнения объекта верификации.

Суть подобных методов состоит в исследовании модели программы или алгоритма на предмет соответствия предъявляемым к ним требованиям (спецификации). При этом для описания и модели, и требований применяются формализмы разного рода.

¹ Работа выполнена при финансовой поддержке Совета по грантам Президента Российской Федерации (проект № СП-1891.2013.5).

1. Формальная верификация алгоритмов и программ с помощью методов логического вывода

На сегодняшний день принято выделять три техники формальной верификации: проверка эквивалентности, дедуктивный анализ и проверка моделей (model checking). Деление обусловлено способами представления модели исследуемого объекта и спецификации. Методы на основе техники проверки эквивалентности оперируют исполняемыми моделями (преимущественно различными автоматами), поэтому чаще всего их область применения ограничивается верификацией аппаратных систем. В методах на основе дедуктивного анализа и model checking используются логико-алгебраические модели, и, следовательно, может быть применен аппарат теории логического вывода.

1.1. Дедуктивный анализ

Техника дедуктивного анализа предполагает, что и модель, и спецификация заданы с помощью логико-алгебраических средств. Как правило, для их описания применяются различные исчисления, модальные логики и всевозможные алгебры.

Верификация с помощью дедуктивного анализа заключается в доказательстве теорем, определяющих отношения между моделью и спецификацией. Характерными чертами этого процесса являются его сложность и невозможность полной автоматизации. В ходе верификации нередко возникает необходимость введения вспомогательных лемм. Таким образом, формальная проверка корректности объекта исследования с помощью техники дедуктивного анализа может быть выполнена только высококвалифицированным экспертом в области логического вывода, а сам процесс требует постоянного контроля с его стороны.

1.2. Проверка моделей

Указанного недостатка лишены методы на основе техники model checking, устанавливающие соответствие исполняемой модели объекта исследования (классической формой представления является структура Крипке) предъявляемым требованиям, выраженным в виде формулы модальной логики (как правило, применяются темпоральные расширения логики высказываний).

Подробное описание особенностей техники проверки моделей, формальные определения структуры Крипке и распространенных

модификаций темпоральных логик представлены в работе [Карпов, 2009]. В качестве примера на рисунке 1 изображены структура Кripке и дерево грамматического разбора (ДГР) формулы темпоральной логики линейного времени.

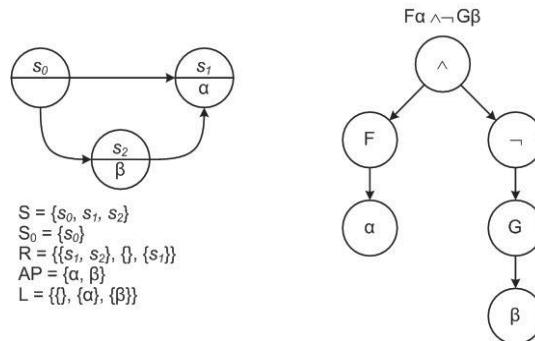


Рис.1. Пример структуры Кripке и ДГР спецификации

Очевидно, что соотношения между состояниями структуры Кripке и свойства состояний могут быть записаны на языке логики предикатов первого порядка. Спецификация в виде формулы темпоральной логики также может быть преобразована в набор утверждений логики предикатов. Для этого достаточно сопоставить каждой вершине ДГР предикатную константу и описать связи между ними. Формальный аспект подобного преобразования рассматривается в работе [Мельцов и др., 2013].

Тогда наличие ошибки в исследуемом объекте может быть подтверждено путем доказательства истинности отрицания предиката, сопоставленного корневой вершине спецификации. В качестве исходных посылок используются выражения, описывающие модель объекта и требования.

1.3. Постановка логической задачи формальной верификации

Данная задача имеет следующую формальную постановку.

Требуется установить выводимость утверждения $1 \rightarrow \neg P(s_0)$ из содержащихся в базе знаний KB посылок. База знаний может быть представлена как объединение посылок из двух множеств $KB = KB_f \cup KB_r$, $KB_f \cap KB_r = \emptyset$. Здесь KB_f – множество фактов,

описывающих свойства модели исследуемого объекта, а KB_r – множество утверждений, соответствующих предъявляемым к объекту требованиям. Факты являются однолитеральными секвенциями вида $1 \rightarrow A(a_1, \dots, a_{|A|})$, где A – некоторая предикатная константа, возможно, с символом отрицания. Утверждения имеют форму

$$1 \rightarrow \bigvee_{i=1}^m A_i(a_{i,1}, \dots, a_{i,|A_i|}).$$

При этом все предикатные константы совпадают с одним из имен *Parent*, *Event*, *End*, *Itself*, P, P_1, P_2, \dots, P_k . Область допустимых значений аргументов всех предикатов, за исключением *Event*, ограничена элементами множества состояний модели S . Первый из аргументов двухместного предиката *Event* принимает значения из множества состояний модели, второй – из множества атомарных предикатов AP .

2. Метод логического вывода делением дизъюнктов на основе определяющего элемента

Для решения сформулированной задачи может быть применен широкий спектр известных методов логического вывода [Вагин и др., 2008]. Тем не менее, наилучшего эффекта удается добиться при использовании средств, способных учитывать специфические особенности базы знаний. Примером такого средства является метод логического вывода делением дизъюнктов на основе определяющего элемента (МЛВДДОЭ), расширяющий обобщенный метод логического вывода делением дизъюнктов (ОМЛВДД).

2.1. Общее описание метода

В связи с применением одной и той же базовой операции – деление дизъюнктов – общий принцип работы МЛВДДОЭ во многом совпадает с принципом работы классического ОМЛВДД. Наиболее существенные различия заключаются во введении новой процедуры (ограниченное образование остатков), введении понятия определяющего элемента, в применении операции разбиения дизъюнкта и использовании предметно-ориентированных оптимизаций. Для наилучшего понимания рассматриваемого метода рекомендуется ознакомиться с полным описанием ОМЛВДД, содержащимся в работе [Страбыкин, 1999].

На верхнем уровне МЛВДДОЭ может быть представлен как совокупность взаимосвязанных процедур: полное образование остатков, ограниченное образование остатков, деление дизьюнктов и шаг логического вывода. Рассмотрим данные процедуры подробнее.

В ходе выполнения процедуры полного образования остатков формируется матрица остатков от деления литералов дизьюнкта-делимого на литералы дизьюнкта-делителя d . По итогам анализа матрицы выносится заключение о выводимости d , принимается решение о построении матрицы остатков следующего уровня или необходимости порождения вспомогательных целей вывода.

Процедура ограниченного образования остатков $\bar{\omega}$ аналогична процедуре ω , однако в правила построения матрицы остатков вводятся некоторые изменения, обусловленные появлением специального литерала – определяющего элемента.

Суть процедуры деления дизьюнктов Ω состоит в делении дизьюнкта D на целевой дизьюнкт d путем построения матрицы остатков нулевого уровня (с помощью процедуры $\bar{\omega}$) и, если это потребуется, последующих уровней (с помощью процедуры ω). По итогам этих действий формируется частный признак решения Q . Результатом деления является заключение о выводимости d через дизьюнкт D ($Q=0$), невозможности доказать истинность d через D ($Q=1$), сигнал о необходимости выполнения операции разбиения дизьюнкта ($Q=F$) или множество дизьюнктов-остатков N , которые следует использовать для доказательства d ($Q=G$).

В ходе выполнения процедуры шага вывода W ко всем утверждениям из множества дизьюнктов исходных секвенций и выводимому дизьюнкту d применяется процедура Ω , после чего формируется признак окончания вывода p . В итоге принимается решение об истинности d ($p=0$), ложности d ($p=1$) или выполняется операция разбиения дизьюнкта ($p=F$), или вывод продолжается для набора вспомогательных целей, полученных в результате конъюнкции всех остатков из множеств N каждой процедуры Ω ($p=G$).

Суть МЛВДДОЭ заключается в последовательном применении процедуры шага вывода W и порождении вспомогательных целей до тех пор, пока не будет доказана истинность или ложность первоначально выводимой секвенции.

2.2. Определяющий элемент

Под определяющим элементом понимается один из литералов каждого утверждения KB_r .

В ходе построения матрицы остатков нулевого уровня для дизъюнктов D и d (процедура ограниченного образования остатков) производная от деления любого отличного от определяющего элемента литерала из \tilde{D} на любой литерал из \tilde{d} определяется как 1.

Кроме того, производная от деления определяющего элемента D на любой литерал из \tilde{d} принимается равной специальному значению θ , если эти литералы не унифицируются только по причине наличия символа отрицания перед одним из них и отсутствия такого перед другим. Такая ситуация является признаком необходимости выполнения операции разбиения дизъюнкта d .

Наличие определяющего элемента позволяет жестким образом задавать для каждого утверждения тот литерал, выводимость которого в большей степени целесообразно доказывать через это утверждение. Специфика базы знаний задачи формальной верификации такова, что в каждом утверждении всегда найдется ровно один такой элемент. Учет этой особенности препятствует разрасланию дерева логического вывода в ширину.

2.3. Операция разбиения дизъюнкта

Операция разбиения дизъюнкта определяется следующим образом.

Пусть в процессе выполнения процедуры шага вывода W при делении некоторого множества дизъюнктов исходных секвенций $D' \subseteq M$ на дизъюнкт выводимой секвенции d для нескольких литералов $\tilde{d}' \subseteq \tilde{d}$ процедура ограниченного образования остатков $\bar{\omega}$ завершилась с получением производной θ .

Тогда процесс логического вывода следует продолжить отдельно для $|\tilde{d}'|$ однолитеральных дизъюнктов, каждый из которых представляет собой инверсию литерала из \tilde{d}' , и для дизъюнкта d' (если он не пуст), содержащего литералы $\tilde{d} \setminus \tilde{d}'$. При этом дизъюнкт d необходимо считать истинным, если вывод дизъюнкта d' будет завершен успешно или вывод какого-либо однолитерального дизъюнкта будет завершен неуспешно. В противном случае дизъюнкт d следует считать ложным.

Структура базы знаний гарантирует, что на следующем после выполнения операции разбиения дизъюнкта шаге вывода, в ходе ограниченного образования остатков, не может быть получена производная q . Таким образом, циклическое повторение разбиения исключено. Следовательно, данная операция обеспечивает полностью МЛВДДОЭ относительно задачи формальной верификации.

3. Пример верификации

Рассмотрим МЛВДДОЭ на примере. Пусть требуется сопоставить алгоритм и спецификацию, изображенные на рисунке 1. Описание фактов и утверждений базы знаний представлено в таблице 1.

Таблица 1

KB_f		KB_r	
№	Факт	№	Утверждение
1	$1 \rightarrow Parent(s_0, s_1)$	1	$1 \rightarrow P_1(\chi) \vee \neg Event(\chi, \alpha)$
2	$1 \rightarrow Parent(s_0, s_2)$	2	$1 \rightarrow P_3(\chi) \vee \neg Event(\chi, \beta)$
3	$1 \rightarrow \neg End(s_0)$	3	$1 \rightarrow \neg P_2(\chi) \vee P_1(\chi) \vee \neg End(\chi)$
4	$1 \rightarrow \neg Event(s_0, \alpha)$	4	$1 \rightarrow \neg P_2(\chi) \vee P_1(\chi) \vee \neg Parent(\chi, \gamma) \vee P_2(\gamma)$
5	$1 \rightarrow \neg Event(s_0, \beta)$	5	$1 \rightarrow \neg P_4(\chi) \vee P_3(\chi)$
6	$1 \rightarrow End(s_1)$	6	$1 \rightarrow \neg P_4(\chi) \vee \neg Parent(\chi, \gamma) \vee P_4(\gamma)$
7	$1 \rightarrow Event(s_1, \alpha)$	7	$1 \rightarrow P_5(\chi) \vee P_4(\chi)$
8	$1 \rightarrow \neg Event(s_1, \beta)$	8	$1 \rightarrow P(\chi) \vee \neg P_2(\chi) \vee \neg P_5(\chi)$
9	$1 \rightarrow Parent(s_2, s_1)$		
10	$1 \rightarrow \neg End(s_2)$		
11	$1 \rightarrow \neg Event(s_2, \alpha)$		
12	$1 \rightarrow Event(s_2, \beta)$		

Результаты процедур шага логического вывода представлены в таблице 2.

Таблица 2

Шаг	Выводимый дизъюнкт	Результат
1	$\neg P(s_0)$	В ходе выполнения процедуры $\bar{\omega}$ получена производная θ . Выполняется операция разбиения дизъюнкта. Получена вспомогательная цель (шаг 2).
2	$P(s_0)$	После выполнения конъюнкции остатков в процедуре W получены две вспомогательные цели (шаг 3 и шаг 9).
3	$P_2(s_0)$	Выполняется операция разбиения дизъюнкта (шаг 4).
4	$\neg P_2(s_0)$	Получены две вспомогательные цели (шаг 5 и шаг 8).
5	$\neg P_2(s_1) \vee \neg P_2(s_2)$	Получена вспомогательная цель (шаг 6).
6	$\neg P_1(s_1)$	Выполняется операция разбиения дизъюнкта (шаг 7).
7	$P_1(s_1)$	Цель успешно доказана с помощью факта $1 \rightarrow Event(s_1, \alpha)$.
8	$\neg P_1(s_0)$	Доказательство не требуется, так как результат получен в параллельной ветви вывода (шаг 5).
9	$P_5(s_0)$	Получена вспомогательная цель (шаг 10).
10	$\neg P_4(s_0)$	Получена вспомогательная цель (шаг 11).
11	$\neg P_3(s_0) \vee \neg P_4(s_1) \vee \neg P_4(s_2)$	Выполняется операция разбиения дизъюнкта (шаг 12 и шаг 13).
12	$P_3(s_0)$	Доказано отрицание цели ввиду наличия факта $1 \rightarrow \neg Event(s_0, \beta)$.
13	$\neg P_4(s_1) \vee \neg P_4(s_2)$	Доказательство не требуется, так как в параллельной ветви вывода (шаг 12) получен отрицательный результат.

Анализ таблицы 2 показывает невозможность вывода секвенции $1 \rightarrow \neg P(s_0)$ из исходной базы знаний. Следовательно, анализируемый алгоритм соответствует спецификации.

Заключение

Предлагаемый в работе метод логического вывода делением дизъюнктов на основе определяющего элемента может применяться для эффективного решения логической задачи формальной верификации. Использование метода совместно с техникой проверки моделей и шаблонами описания типовых ошибок позволяет полностью автоматизировать процесс верификации. Эвристичность, свойственная аппарату логического вывода, обеспечивает получение результата с приемлемыми временными затратами для тех классов моделей, к которым, вследствие их структурных особенностей, не может быть применен символьный подход.

Среди направлений дальнейших исследований необходимо выделить задачу реализации и исследования целесообразности использования дополнительных эвристик на базе принципа концентрации ресурсов.

Предложенный метод является математической основой прототипа программного комплекса для верификации параллельных алгоритмов DDVeifier.

СПИСОК ЛИТЕРАТУРЫ

[**Вагин и др., 2008**] Вагин В.Н., Головина Е.Ю., Загорянская А.А., М.В. Фомина. Достоверный и правдоподобный вывод в интеллектуальных системах. – М.: ФИЗМАТЛИТ, 2008.

[**Карпов, 2010**] Карпов Ю.Г. Model checking. Верификация параллельных и распределенных программных систем. – СПб.: БХВ-Петербург, 2010.

[**Мельцов и др., 2013**] Мельцов В.Ю., Чистяков Г.А. Формирование базы знаний на основе структуры Крипке и формул темпоральной логики// Фундаментальные исследования. 2013. № 8–4.

[**Страбыкин, 1998**] Страбыкин Д.А. Логический вывод в системах обработки знаний. – СПб.: Издательство СПбГЭТУ, 1998.

УДК 004.81

ОЦЕНКИ ГИБРИДНЫХ ВЫВОДОВ В ПРОДУКЦИОННЫХ СИСТЕМАХ

С.Ю. Соловьев (*soloviev@glossary.ru*)
факультет ВМК МГУ, Москва

В работе рассматриваются гибридные логические выводы, построенные одной продукционной системой с привлечением правил из другой продукционной системы. Основное внимание уделяется способу вычисления оценок оригинальности гибридных выводов.

Введение

В некоторых случаях эксперт для выработки заключения вынужден – в порядке цитирования – заимствовать знания своих коллег. Алгоритм, имитирующий способность человека к целенаправленному цитированию, представляется необходимым компонентом сетевых интеллектуальных систем. Понятно, что сам факт существования алгоритма цитирования и его свойства существенно зависят от способа представления знаний и особенностей реализации логического вывода. Однако в любом случае полученное в результате цитирования экспертное заключение неизбежно опирается на некоторое количество «чужих мыслей» и поэтому нуждается в оценке оригинальности.

1. Алгоритм цитирования для продукционных систем

Как показано в [Ginkul et al., 2013], для продукционных систем с монотонным выводом [Осипов, 2013] алгоритм цитирования естественным образом встраивается в процедуру разрешения конфликтов. Алгоритм исходит из существования двух продукционных систем, одна из которых именуется оригинальной, а другая – альтернативной. Основное изложение ведется применительно к оригинальной системе, а альтернативная система рассматривается как потенциальный источник цитат. Обе продукционные системы различаются (1) процедурами разрешения конфликтов, (2) наборами правил O_{PS} и A_{PS} , а также (3) списками потенциально возможных целевых заключений (диагнозов). При этом одинаковые по смыслу

элементы, описывающие проблемные области в обеих системах, представляются одинаковыми фактами. Относительно каждой системы предполагается:

- все продукционные правила имеют вид $\text{IF } f_1 \& \dots \& f_n \text{ THEN } f$;
- для построения заключений используется обратный вывод;
- в рабочую память загружено множество фактов S , описывающих проблемную ситуацию, для которой необходимо найти диагноз;
- для каждой выдвинутой гипотезы процедура разрешения конфликтов вырабатывает либо продукционное правило, либо сообщение об отказе в подтверждении гипотезы;
- подтвержденные гипотезы заносятся в рабочую память, а подтвердившие их правила рассматриваются как материал для построения мотивированного заключения.

Под мотивированным заключением диагноза d понимается минимальное по количеству элементов множество продукции $P(d)$, удовлетворяющее следующим свойствам:

1. $P(d)$ содержит ровно одно правило вида $\text{IF } \dots \text{ THEN } d$;
 2. если $P(d)$ содержит правило $\text{IF } h_1 \& \dots \& h_n \text{ THEN } h$, то для каждого факта $h_i (i = 1 \dots n)$ справедливо одно из двух:
- либо $h_i \in S$;
либо $P(d)$ содержит ровно одно правило вида $\text{IF } \dots \text{ THEN } h_i$.

Каждое мотивированное заключение можно представить в виде ациклического ориентированного графа, как это показано на рис.1 для мотивированного заключения, состоящего из трех продукции: $\text{IF } a \& b \& c \text{ THEN } d$, $\text{IF } e \& b \text{ THEN } a$ и $\text{IF } f \& g \text{ THEN } c$.

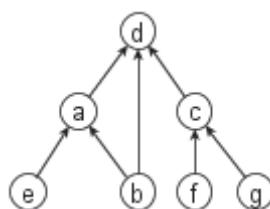


Рис.1. Пример мотивированного заключения

Для описанного класса продукционных систем реализация способности к цитированию сводится к двум модификациям стандартной процедуры разрешения конфликтов. Во-первых, моди-

фицированная процедура в качестве дополнительного аргумента использует альтернативное мотивированное заключение, полученное, соответственно, альтернативной продукционной системой. Альтернативное заключение используется как сборник цитат. Во-вторых, если стандартная процедура разрешения конфликтов впервые сталкивается с необходимостью выработать отказ в подтверждении определенной гипотезы, то модифицированная процедура предлагает использовать правило (если оно есть) из альтернативного заключения. Фактически на каждом шаге логического вывода продукционная система «до последнего» пытается использовать собственные оригинальные правила, и лишь по их исчерпанию обращается к альтернативным знаниям.

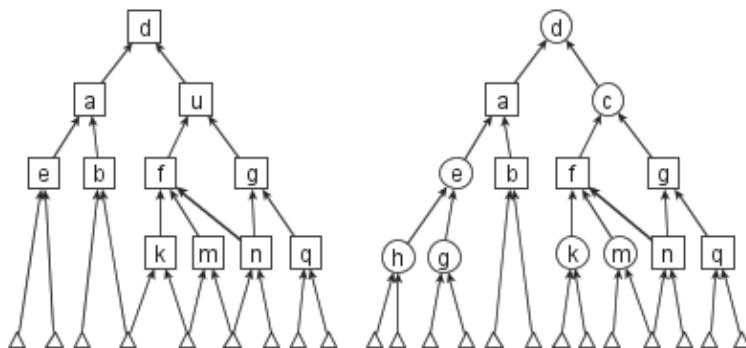


Рис.2. Примеры альтернативного (слева) и гибридного (справа) мотивированных заключений

Если альтернативное заключение диагноза d задано, то любая продукционная система с модифицированной процедурой разрешения конфликтов гарантированно строит мотивированное заключение диагноза d , именуемое гибридным заключением. В худшем случае гибридное заключение совпадает с альтернативным, а в лучшем случае – гибридное заключение является оригинальным, состоящим исключительно из оригинальных правил. В общем случае гибридное заключение содержит и оригинальные, и альтернативные правила. На рис.2 приведены примеры альтернативного заключения и построенного с его помощью гибридного заключения. Прямоугольники на рис.2 соответствуют альтернативным правилам-цитатам.

2. Оценки гибридных заключений

В интересах практического использования гибридных заключений необходимо иметь способ вычисления оценок их оригинальности. Для определенности будем полагать, что оценки оригинальности 0 и 1 отвечают соответственно альтернативным и оригинальным заключениям.

Поскольку альтернативное и гибридное заключения представляют собой множества продукции (обозначим их A и G), то в основу оценки разумно положить известное [Деза и др., 2008] расстояние биотопа (расстояние Танимото):

$$|A \Delta G| / |A \cup G| \equiv 1 - |A \cap G| / |A \cup G|.$$

Однако применительно к оценке оригинальности расстояние биотопа обладает двумя дефектами.

Первый дефект состоит в нечувствительности расстояния биотопа к дубликатам производственных правил из $C = A_{PS} \cap O_{PS}$. Дубликат, попавший в гибридное заключение, неоправданно уменьшает расстояние биотопа. Устранение указанного дефекта приводит к окончательной формуле

$$\text{Оценка } (A, G) = 1 - |(A \cap G) \setminus C| / |A \cup G|.$$

Для заключений, представленных на рис.2 (при $C = \emptyset$), вычисление оценки приводит к следующим результатам: $|A \cup G| = 18$, $|A \cap G| = 6$ и, окончательно, Оценка $(A, G) = 2/3$.

Второй дефект заключается в игнорировании структурных связей в гибридном заключении. Однажды привлеченное к выводу альтернативное правило автоматически инициирует применение других альтернативных правил, которые, тем не менее, следует отличать от инициирующих правил. Компенсировать этот дефект можно использованием вспомогательной оценки оригинальности

$$\text{Оценка}'(A, G) = (|W| + Q) / (|W \cup B| + 1),$$

где $W = \{ (\text{IF } \dots \text{ THEN } Z) \in T \mid (\text{IF } \dots \text{ THEN } Y) \in F \}$,

$B = \{ (\text{IF } \dots \text{ THEN } Z) \in F \mid (\text{IF } \dots \text{ THEN } Y) \in T \}$,

$T = G \cap O_{PS}$, $F = (G \cap A_{PS}) \setminus C$,

$Q = 1$, если множество T содержит корневое правило гибридного заключения G ; в противном случае $Q = 0$.

Множества W и B составляют правила, инициирующие попадание в заключение G соответственно оригинальных и альтернативных правил. Для примера на рис.2 (при $C=\emptyset$) правила множества W отмечены в G символами e, k, m , правила множества B – символами $a, f, g; Q = 1$ и, окончательно, Оценка'(A, G) = 4/7. Вспомогательная оценка позволяет принимать решения в ситуациях, когда основная оценка не дает определенных рекомендаций.

Заключение

Приведенные оценки оригинальности гибридных выводов позволяют в исследованиях по взаимодействию систем, основанных на знаниях, перейти от теоретических изысканий к вычислительным экспериментам.

СПИСОК ЛИТЕРАТУРЫ

- [Деза и др., 2008] Деза Е.И., Деза М.М. Энциклопедический словарь расстояний. – М.: Наука, 2008.
- [Осипов, 2013] Осипов Г.С. Лекции по искусственному интеллекту. – М.: Либроком, 2013.
- [Ginkul at al., 2013] Ginkul G. and Soloviev S. The Quoting-Based Algorithm for Cooperative Decision Making in Production Systems // Proc. Int. Conf. IIS, Chisinau: IMCS. 2013. – http://park.glossary.ru/serios/read_24.php.

СЕКЦИЯ 2 | КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА

УДК 004.912, 004.032.24

**КОНЦЕПЦИЯ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ
АНАЛИЗА ТЕКСТА ДЛЯ ПОПОЛНЕНИЯ ОНТОЛОГИИ¹**

С.А. Анохин (*saanokhin@gmail.com*)

Н.О. Гаранина (*garanina@iis.nsk.su*)

Е.А. Сидорова (*lena@iis.nsk.su*)

Институт систем информатики им. А.П. Ершова

СО РАН, Новосибирск

Рассматривается подход к построению мультиагентной системы анализа текста для пополнения онтологии предметной области. Представлены формальная модель знаний, агентная модель и мультиагентный подход к анализу текста, основанный на взаимодействии информационных агентов, сопоставляемых онтологическим понятиям, и агентов-правил, соответствующих способам представления информации в тексте.

Введение

В настоящее время для хранения большого количества информации и связанных с ней документов используются онтологические базы знаний. Одной из актуальных задач, возникающих при использовании данного подхода, является автоматическое пополнение онтологий (ontology population) на основе поиска релевантных документов, их автоматической обработки и извлечения новой информации.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 13-01-00643; № 13-07-00422) и в рамках проекта ИСИ СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем».

Мультиагентный подход к извлечению информации из разнородных источников данных широко распространен. В частности, он используется для обработки информации из сети Интернет [Santos et al., 2006; Banares-Alcantara et al., 2005], однако обычно в таких работах агенты являются высокоуровневыми сущностями, которые, скорее, направляют потоки данных, используя для их обработки стандартные алгоритмы, чем имеют дело с данными непосредственно. При обработке текстов на естественном языке агенты, скорее, ассоциируются с традиционными лингвистическими уровнями (морфологическим, синтаксическим) либо предназначены распознавать лингвистические феномены, такие как эллипсис, анафора, омонимия и т.п. Эти агенты по существу не используют знания о предметной области и являются вычислительными процессами, которые могут ускорить обработку информации благодаря параллелизму, но никак не влияют на извлечение информации качественно.

В отличие от вышеупомянутых работ в нашем подходе мы используем агентов двух типов, которые коллективно владеют информацией как об рассматриваемых данных, так и о заданной онтологии. Агенты-правила имеют дело с онтологическими и лингвистическими особенностями подъязыка предметной области и обрабатывают данные, получаемые от информационных агентов. Последние наиболее близки агентам из работы [Вольман и др., 2005], которые представляют слова из текста.

1. Модель представления знаний

Агентная модель анализа текста опирается на модель предметной/проблемной области (ПО), представленной онтологией, и лингвистические знания, представленные лексической и семантико-синтаксической моделью подъязыка ПО.

1.1. Онтология предметной области

Опираясь на формальную модель ПО, осуществляется структуризация информации и всего контента системы. В упрощенном виде (достаточном для целей работы) *онтологию ПО* можно представить в виде системы $O = \langle C, R, T, A \rangle$, где (1) $C = \cup C_i$ – конечное непустое множество классов, описывающих понятия ПО; (2) $R = \cup R_i$ – конечное множество бинарных отношений, заданных на

классах (понятиях), и $F_R: C \times C \rightarrow 2^R$ – функция, которая определяет имена бинарных отношений между классами; (3) $T = \cup T_i$ – множество типов данных и $V_i = \{v_1, \dots, v_{Ti}\}$ – множество значений типа T_i ; (4) $A = \{a_1, \dots, a_s\}$ – конечное множество атрибутов, $A_K \subseteq A$ – подмножество ключевых атрибутов для уникальной идентификации экземпляров понятий и отношений, и $F_A: C \cup R \rightarrow 2^{A \times T}$ – функция, которая определяет имена и типы атрибутов классов C и отношений R .

В качестве примера приведем фрагмент онтологии «Научные мероприятия» в контексте задачи анализа информационных сообщений.

```

class Важные_даты (Дата начала: data; Дата окончания: data;
Представление статьи: data; Уведомление о принятии: data)
class Научное_мероприятие (Название: string; Язык: Язык;
Статус: Статус_мероприятия; Частота проведения: string;
Описание мероприятия: string; Тематика: Тематика)
relation Даты мероприятия < Научное_мероприятие, Важные_даты >
relation Научные труды < Научное_мероприятие, Публикация >
relation Программный комитет < Научное_мероприятие, Персона >
```

Информационный контент системы, построенной на онтологии O , представляется как $IC = \langle I, R_I \rangle$, где (1) $I = \cup I_i$ – конечное множество экземпляров классов онтологии O , где I_i класса C_i представляется набором атрибутов a_j со значениями v_j : $I_i = \cup (a_j, v_j)$; (2) $R_I = \cup R I_i$ – конечное множество отношений на множестве I (экземпляров отношений), где экземпляр $R I_i$ отношения R_i включает множество аргументов $Obj_i \subseteq I$ и набор атрибутов: $R_i = (Obj_i, \cup (a_j, v_j))$.

1.2. Модель знаний о лексике предметной области

Модель знаний о лексике ПО или предметный словарь должен включать спецификации терминов трех типов: (1) лексема или однословный термин, (2) устойчивый словокомплекс – словосочетание, характеризующееся высокой частотностью в анализируемом подъязыке (*искусственный интеллект, тезис доклада*), (3) лексическая конструкция – шаблон, описывающий типичное сокращение (*др., т.д., к.ф.-м.н.*), или термин, содержащий индексы (*КИИ'2014, с 24 по 27 сентября 2014 г.*).

Таким образом, лексическая модель знаний определяется тройкой вида $V = \langle L, M, S \rangle$, где (1) $L = W \cup P \cup LT$ – множество предметных терминов, включающих W – множество лексем, P – множество словокомплексов, LT – множество лексических конструкций; (2) M – множество морфологических характеристик лексем W ; (3) S – множество семантических признаков (тематических и семантических классов).

Связь между словарем и онтологией выражается в сопоставлении семантических признаков, приписываемых терминам словаря, с элементами онтологии (атрибутами, понятиями и отношениями).

1.3. Модель представления текста

Задачей автоматической обработки текста является преобразование текста в семантическую сеть информационных объектов – экземпляров классов и отношений онтологии. В процессе обработки текста его представление постепенно изменяется, переходя от менее структурированного к более структурированному. Для описания этого изменения предложена концепция покрытий текста, когда каждое покрытие представляется набором однотипных элементов с заданными текстовыми позициями (интервалами). Выделяются следующие типы покрытий: (1) терминологическое, соответствующее словарным терминам из V , найденным в тексте, (2) сегментное, которое отражает структурное деление и жанр текста, и (3) объектное, которое является упорядоченным множеством лексических объектов и экземпляров классов онтологии.

Терминологическое покрытие ТС текста состоит из множества уровней, где каждый уровень $l_t = (i, pos_i, LO_i)$, где (1) i – порядковый номер уровня; (2) $pos_i = (b_i, e_i)$ – текстовые позиции уровня; (3) LO_i – множество лексических объектов, начинающихся с позиции b_i . Это покрытие упорядочено по начальным позициям терминов b_i , а элементы LO_i упорядочены по длине текстового фрагмента, покрываемого термином.

Каждый лексический объект L формируется на основе словарного термина, найденного в тексте, и снабжается словарной информацией $L=(v, S_v, M_v, A_v, pos)$, где (1) $v \in V$ – термин словаря; (2) S_v – множество словарных семантических классов термина v ; (3) M_v – множество морфологических характеристик термина v ; (4) A_v – множество семантических атрибутов; (5) pos – позиции в тексте.

Сегментное покрытие SC состоит из множества сегментов

$s \in SC$, определяемыми левой и правой текстовыми позициями сегмента $pos_s = (b_s, e_s)$ и его типом t_s (например, *предложение, абзац, заголовок, тематика конференции, программный комитет* и т.п.).

Объектное покрытие ОС текста строится над терминологическим покрытием, расширяя наполнение уровня информационными объектами, построенными в процессе основного анализа текста. Уровень $l \in OC$ – это четверка $l = (i, pos_i, s_i, Obj_i)$, где дополнительные элементы s_i – это множество идентификаторов сегментов, покрывающих уровень, а $Obj_i = LO_i \cup I_i \cup RI_i$ – упорядоченные по длине лексические объекты LO_i и экземпляры классов I_i и отношений RI_i онтологии.

1.4. Агентная модель анализа текста

Агентная модель анализа текста включает агентов четырех типов.

Специальный служебный *агент-контролер* отслеживает момент завершения взаимодействия всех агентов системы.

Агенты-экземпляры и *агенты-отношения* сопоставляются объектам и отношениям онтологии соответственно и образуют *информационные агенты*. Эти агенты порождаются на основе словарной информации и их дальнейшая цель – заполнить свои атрибуты и выявить возможные связи с другими агентами. Агент-экземпляр – это $AI = (C_i; AR_i; AIR_i; SS; Pos)$, где (1) C_i – онтологический класс агента; (2) AR_i – множество агентов-правил, которые могут использовать данного агента в качестве аргумента; (3) AIR_i – множество агентов-отношений, в которых участвует данный агент; (4) SS – структура, отражающая историю создания данного агента и означивания его атрибутов; (5) Pos – позиция в тексте. Агент-отношения – это $AIR = (R_i; RI_i)$, где R_i – онтологический класс отношения агента; RI_i – множество экземпляров отношения. Примером агента-экземпляра может быть агент, описывающий событие КИИ-2014:

agent #1 (C: Научное_мероприятие (1)

Название: «Конференция по искусственному интеллекту»;

Статус: национальный; Тип мероприятия: конференция;

Тематика мероприятия: { компьютерная лингвистика, ... });

AR: { Дата_проведения_мероприятия, Город, ... }; AIR: \emptyset ;

SS:(base:LO₁(v:состояться),struct:{LO₂(v:конференция),...});Pos:[59,71])

Данный агент стремится заполнить свои атрибуты *Язык*, *Частота проведения* и др., а также сформировать отношения *Даты мероприятия*, *Место*, *Программный комитет* и т.д.

Агенты-правила реализуют проверку согласованности информации, получаемой от информационных агентов, и правила пополнения онтологии. В соответствии с получаемыми данными они могут порождать новые значения атрибутов информационных агентов, а также новых агентов-экземпляров и экземпляры агентов-отношений. Способ обработки данных агентом-правилом реализуется на основе заданной схемы, которая явным образом определяет структуру выражений, описывающих однотипные факты, и связывает с элементами онтологии. Сегментное покрытие ограничивает область действия агентов-правил.

Агент-правило – это $AR = (\text{Arg}; F_{\text{res}}(\text{arg}), \text{Res})$, где (1) Arg – вектор аргументов, такой, что каждый аргумент $\text{arg}_i(\text{SC}_i) \in \text{Arg}$ определяется информационным агентом онтологического класса SC_i либо лексическими объектами с семантическим признаком SC_i ; (2) $F_{\text{res}}(\text{arg})$ – функция, проверяющая согласованность вектора значений аргументов arg ; проверяются семантические, синтаксические, сегментные и позиционные ограничения, задаваемые экспертом, накладываемые как на отдельные аргументы, так и на их пары; (3) Res – результат применения правила к arg . Результатом может быть как обновление значений атрибута аргумента, так и порождение нового агента. Приведем пример агента-правила, который порождает связь между *Научным мероприятием* и его датой проведения.

```
rule Дата_проведения_мероприятия
    arg1: Научное_мероприятие (),
    arg2: Важные даты ()
condition segment = Предложение; Pos = prior_post, Synt =
    Упр(arg1,arg2)
⇒ new Relation::Даты мероприятия (first: arg1, second: arg2)
```

2. Мультиагентная система анализа текста

К сожалению, объем статьи не позволяет привести полные формальные описания алгоритма и протоколов, поэтому мы приведем лишь основные посылки. На вход мультиагентной системе анализа MDA поступают правила пополнения онтологии, на основе которых

формируются агенты-правила (см. п.1.4), а также результаты предварительной обработки текста – терминологическое и сегментное покрытие (см. п.1.3). Далее на основе правил согласования семантического словаря и онтологии происходит начальное наполнение объектного покрытия информационными агентами и осуществляется инициация всех агентов.

Система MDA включает множества информационных агентов, агентов-правил и агента-контролера. В процессе взаимодействия информационные агенты обогащаются значениями своих атрибутов и связями с другими агентами. Все агенты действуют параллельно до тех пор, пока не оказывается, что ни один агент-правило не может сработать. Этот момент остановки определяется агентом-контролером. Мы используем оригинальный алгоритм обнаружения завершения, основанный на подсчете активности остальных агентов [Garanina et al., 2014].

Информационные агенты на первой стадии своей деятельности рассылают уже означенные данные всем агентам-правилам, заинтересованным в этих данных. Затем агенты переходят в режим ожидания сообщений. Обработка поступающих сообщений заключается в изменении 1) агентом-экземпляром – своих атрибутов и отношений и 2) агентом-отношением – своих экземпляров. После чего агенты рассылают новые значения агентам-правилам, заинтересованным в этих данных.

У агента-правила есть два параллельных подпроцесса: обработка входных данных от информационных агентов и получение исходящего результата. Обработка входных данных включает (1) формирование вектора аргументов, (2) отсылку вектора аргументов или сообщения об остановке процесса получения результата. Получение исходящего результата включает (1) проверку согласованности аргументов и вектора аргументов, (2) обработку однородных групп, (3) выработку результата и (4) определение агентов, которым нужно высыпать новые значения. Новые информационные агенты начинают работать немедленно с данными, определенными агентом-правилом при их рождении.

Агент-контролер информируется о каждом изменении активности всех агентов. Работа агента-контролера заключается в последовательном подсчете суммарной активности других агентов. Если все агенты, кроме него самого, неактивны, он посыпает всем сообщение СТОП.

Результатом работы MDA является «обогащенное» объектное покрытие текста, из которого впоследствии выбираются все информационные объекты и, после идентификации, помещаются в контент системы. Так, для примера агента-экземпляра, приведенного в п.1.4, результатом будет информационный объект, описывающий научное мероприятие *КИИ-2014*, и все его связи, представленные экземплярами отношений: *Даты мероприятия*, *Место*, *Научные труды* и *Программный комитет*, найденными в тексте.

2.1. Архитектура мультиагентной системы анализа текста

На начальном этапе настройки системы экспертами составляются онтология и словарь ПО, формулируются правила пополнения онтологии, соответствующие паттернам представления иско-мой информации в тексте. Эти знания используются в качестве конфигурации для программного комплекса, осуществляющего обработку документов. Основными модулями комплекса являются словарный компонент и мультиагентное ядро анализа текста. Словарный компонент осуществляет предварительную обработку текста, формируя терминологическое и сегментное покрытия, на основе которых мультиагентное ядро осуществляет сборку экземпляров классов и отношений онтологии. Агенты-правила инициализируются единожды на основе заданных правил пополнения онтологии. Агенты-экземпляры создаются заново для каждого текста. Результаты работы мультиагентного ядра после постобработки могут быть добавлены в информационный контент системы.

На рис.1 приведена концептуальная схема предложенного подхода.

При реализации данного подхода была разработана библиотека для построения мультиагентных систем на языке C++, которая представляет собой набор шаблонных классов и генераторов классов (в терминах [Александреску, 2008]). Библиотека спроектирована с учетом особенностей предложенного алгоритма – большое количество сравнительно простых агентов, интенсивный обмен сообщениями. Агенты являются активными объектами, существующими в рамках некоторого динамического окружения. Любое взаимодействие с окружением осуществляется посредством *аватара* данного агента – сущности, имеющей набор методов восприятия и воздействия (аналогичный подход применен в проекте OpenCog

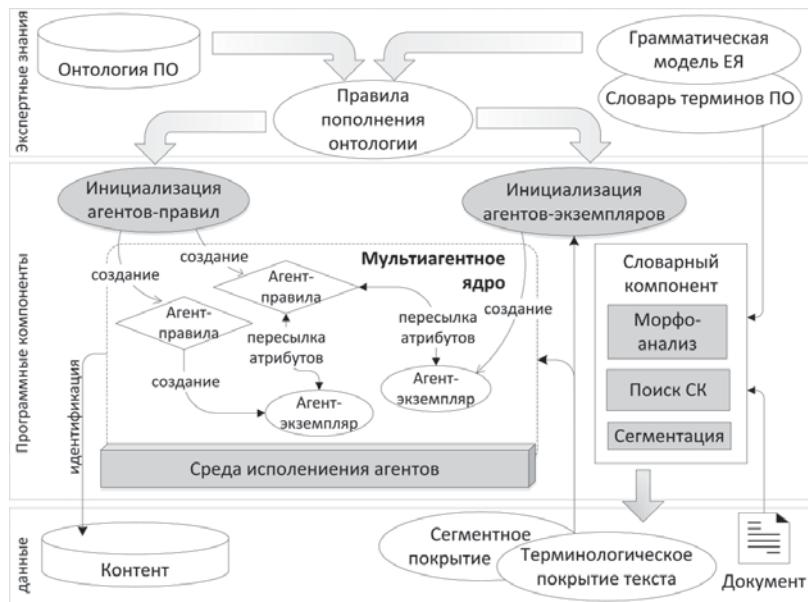


Рис.1. Архитектура мультиагентной системы анализа текста

[OpenCog]). Все многообразие агентов и их ролей задается пользователем с помощью иерархии интерфейсов.

Таким образом, разрабатываемая система является программной технологией, позволяющей пользователям экспертам создавать необходимые ЕЯ-сервисы, обеспечивающие анализ документов и извлечение информации на основе заданных предметно-ориентированных моделей. Качество осуществляющегося анализа зависит от степени компетенции экспертов и уровня подробности моделей, задающих правила извлечения информации.

Заключение

Предложенный мультиагентный подход будет иметь следующие преимущества: (1) агенты ускоряют процесс обработки, поскольку действуют параллельно; (2) они эффективно используют ресурс данных точно тогда и где это необходимо; (3) агенты могут разрешать неоднозначности посредством конкуренции.

На текущий момент разработаны формальные алгоритмы и реализованы базовые модули системы. Также была проведена формальная верификация корректности работы агента-контролера и работоспособности системы с использованием инструмента проверки моделей SPIN. Продолжение работ в данном направлении видится в первую очередь в проведении практических экспериментов и сравнении их результатов с последовательными механизмами анализа текста, основанных на подобных моделях знаний [Кононенко и др., 2009].

СПИСОК ЛИТЕРАТУРЫ

- [Banares-Alcantara et al., 2005] Banares-Alcantara R., Jimenez R., Aldea L. Multi-agent systems for ontology-based information retrieval // European Symposium on Computer-Aided Chemical Engineering (ESCAPE-15), 2005. Barcelona, Espana.
- [Santos et al., 2006] C.T. dos Santos, P. Quaresma, I. Rodrigues, R. Vieira A Multi-Agent Approach to Question Answering // In Computational Processing of the Portuguese Language: 7th International Workshop, PROPOR 2006. LNAI 3960. – Berlin/Heidelberg: Springer Verlag, pp. 131–139.
- [Вольман и др., 2005] Вольман С.И., Минаков И.А., Томин М.С. Мультиагентная система интеллектуального анализа содержимого Интернет-страниц // Тр. VII Международной конференции «Проблемы управления и моделирования в сложных системах». – Самара: СНЦ РАН, 2005. – С. 403–408.
- [Garanina et al., 2014] 82. N. Garanina, E. Bodin, E. Sidorova. Using SPIN for Verification of Multi-agent Data Analysis // Proc. of Fifth Workshop on Program Semantics, Specification and Verification: Theory and Applications (PSSV 2014). –Moscow, 2014. – p. 59–67.
- [Александреску, 2008] Александреску А. Современное проектирование на C++. Серия C++ In-Depth. – Вильямс, 2008.
- [OpenCog] Модуль Embodyment проекта OpenCog. wiki.opencog.org/w/Embodyment
- [Кононенко и др., 2009] Кононенко И.С., Сидорова Е.А. Подход к извлечению фактов из текста на основе онтологии // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог 2009». Вып. 8 (15). М.: РГГУ, 2009. – С. 451–457.

УДК 004.9, 004.89

ПОДХОДЫ К НОРМАЛИЗАЦИИ СЛОВАРЕЙ И УСТАНОВЛЕНИЮ ИДЕНТИЧНОСТИ СУЩНОСТЕЙ ПРИ ОБОГАЩЕНИИ КОНТЕНТА НАУЧНЫХ БАЗ ЗНАНИЙ¹

З.В. Апанович (*aparovich@iis.nsk.su*)
А.Г. Марчук (*mag@iis.nsk.su*)
ИСИ СО РАН, Новосибирск

В данной работе описаны подходы к решению проблем нормализации словарей, установления идентичности сущностей и фильтрации данных, возникающих в процессе использования данных из облака LOD для обогащения контента научных баз данных и знаний. В качестве тестовых примеров использовались данные открытого Архива СО РАН, структурированные при помощи онтологии ОНС, а также различные наборы библиографических данных.

Введение

В связи с бурно развивающимся направлением Semantic Web и его новой ветвью LOD (Связанные Открытые Данные) в Интернете становятся доступными большие объемы информации, посвященной различным научным направлениям. Облако LOD содержит в настоящий момент более 50 миллиардов троек RDF. С одной стороны, эти данные могут быть использованы для обогащения имеющихся семантических баз данных, с другой стороны, имеющиеся базы данные могут быть также полезны для уточнения информации, хранящейся в облаке LOD.

Один из проектов, осуществляемых в Институте систем информатики Сибирского отделения Российской академии наук (ИСИ СО РАН), направлен на обогащение Открытого архива СО РАН (<http://duh.iis.nsk.su/turgunda/Home>, <http://duh.iis.nsk.su/VirtuosoEndpoint/Home/Samples>) [Марчук 2010] данными облака Открытых Связанных Данных (Linked Open Data, LOD) [Bizer 2009]. Фрагмент страницы Открытого Архива, посвященной академику А.П. Ершову, показан на рис. 1.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-00386) и проекта РАН 15/10.

	1976	1988	участник	заместитель заведующего кафедрой вычислительной математики АМФ	Новосибирский государственный университет
	1979	1988	первое лицо	председатель	Комиссия по системному математическому обеспечению Координационного комитета по Вычислительной технике ССР (КОСМО ККВТ АН СССР)
			участник	участник	Собрание "Общим собранием Академии наук избраны п. Сибирь..."
			организатор	организатор	Празднование 10-летия Отдела программирования
			участник		Третий всесоюзный симпозиум "Системное и теоретическое программирование"
	1964	1988	участник	заведующий отделом	Институт вычислительной математики и математической геофизики СО РАН
	1959	1964	участник	заведующий отделом программирования	Институт математики им. С.Л. Соболева СО РАН
отраж. в документе	Отражение				

Рис. 1. Фрагмент страницы Открытого Архива, посвященной академику А.П. Ершову

В работе [Schultz 2012] предложена четырехшаговая стратегия интеграции связанных данных в приложения. Помимо проблем, специфических для конкретного приложения, требуется решить проблему доступа к связанным данным (1), проблему нормализации словарей (2), установления идентичности сущностей (3) и фильтрации данных (4). Способы решения этих проблем варьируют в диапазоне от ручных до автоматизированных [Isele 2010, Ngomo 2011, Shvaiko 2013]. При этом такие проблемы, как проблема установления соответствия между онтологиями, а также проблема объединения данных из разных наборов, «еще находятся в детском состоянии» [Tramp, 2012]. В работах [Apanovich, 2010, Apanovich, 2013] подробно рассмотрены методы исследования онтологий и контента семантических систем при помощи различных методов визуализации, разработанных в ИСИ СО РАН. В данной работе будут рассмотрены проблемы (2), (3) и (4) и продемонстрированы методы решения этих проблем.

В качестве тестовых примеров использовались данные открытого Архива СО РАН, структурированные при помощи онтологии ОНС [Марчук 2010] и различные наборы библиографических данных портала RKBExplorer.com, структурированные при помощи AKT Reference онтологии [AKT 2014].

1. Эксперименты по выравниванию онтологий

Одним из шаблонов построения онтологий в приложениях Semantic Web является то, что сущности, описанные с помощью отношений в одних онтологиях, могут быть описаны как экземпляры классов в другой онтологии. Этот шаблон компенсирует отсутствие атрибутов у предикатов RDF. Такой шаблон используется в онтологии ОНС и позволяет описывать такие факты как, «академик А.П. Ершов был главой отдела в Институте математики СО АН СССР с 1959 по 1964 и руководителем отдела в Вычислительном центре СО АН СССР с 1964 по 1988 год» (см. рис. 1). Для этого существует специальный класс *onc:participation*, соответствующий отношению *akt:has-affiliation* онтологии AKT Reference, экземпляры которого имеют атрибуты *from-date* и *to-date*. По тем же причинам такие классы, как *onc:dating*, *onc:naming*, *onc:authorship*, используются в онтологии ОНС вместо таких предикатов, как *akt:has-author*, *akt:has-date* или *akts:has-pretty-name*. Платой за расширение выразительных возможностей является усложнение проблемы интеграции данных, поскольку систематически возникает необходимость в установлении соответствия между различными группами классов и отношений двух онтологий с различными структурными свойствами. А именно, возникает необходимость установления соответствия между группой вида «Class1-relation1-Class2» одной онтологии и одной или несколькими группами вида «Class3-relation2-Class4-relation3-Class5» другой онтологии. Такая трансляция может быть осуществлена при помощи запроса SPARQL 1.1. Упрощенная версия шаблона этого запроса имеет следующий вид:

```
PREFIX iis:<http://iis.nsk.su#>
PREFIX akt:<http://www.aktors.org/ontology/portal#>
PREFIX akts:<http://www.aktors.org/ontology/support#>
CONSTRUCT {
  _:p a iis:Class4.
  _:p iis:relation2 ?instance1.
  _:p iis:relation3 ?instance2.
}
WHERE {
  ?instance1 akt:relation1 ?instance2.
  ?instance1 a akt:Class1.
  ?instance2 a akt:Class2.
}
```

Для упрощения задачи пользователя по написанию такого рода запросов нами разработана программа, которая позволяет генерировать SPARQL-запросы на основе визуализации онтологии. Пример установления такого соответствия показан на рис. 2. Сначала в интерактивном режиме устанавливается соответствие между двумя наборами классов и отношений, а затем автоматически генерируется шаблон SPARQL-запроса, осуществляющий трансляцию данных.

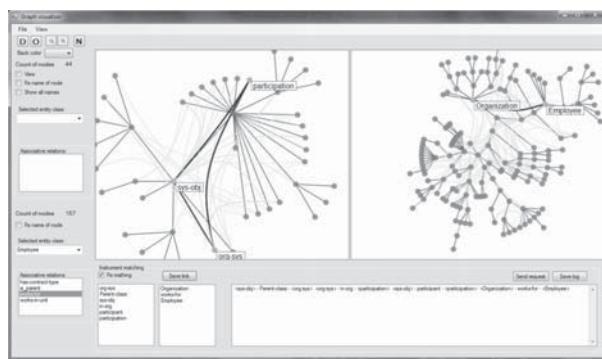


Рис. 2. Интерактивное установление соответствия между классами и отношениями двух онтологий

2. Установление идентичности сущностей и фильтрация результатов

Важным этапом пополнения одной базы знаний при помощи другой является этап установления идентичности сущностей, то есть генерация отношений вида *owl:sameAs*. В частности, необходимо правильно идентифицировать персон, описанных в обеих семантических системах. Хотя эта проблема известна исследователям многие годы, она далека от своего окончательного решения. Подтверждением этого факта является то, что в процессе наших экспериментов мы регулярно обнаруживали в хорошо известных системах поддержки научных исследований ошибочные данные, касающиеся публикаций персон из Открытого архива СО РАН, в том числе публикаций такого известного человека, как академик Андрей Петрович Ершов. Поэтому остановимся на этом вопросе подробнее.

В Открытом архиве все персоны описаны при помощи атрибута *ons:name*. Он имеет формат <Фамилия, Имя, Отчество> и два варианта написания: русскоязычный и (иногда) англоязычный вариант. При этом возможно много вариантов латинского написания русских имен. Русская фамилия Ершов может писаться как Yershov, так и Ershov. Для повышения полноты сравнения русскоязычных вариантов фамилий с их латинскими эквивалентами необходимо генерировать все возможные варианты транслитерации русских букв в латинские. Вторым источником многообразия имен является то, что в латинских информационных системах имена не всегда нормализованы и имеют много вариантов. Это может быть <Имя Фамилия>, <Имя Первая буква Отчества Фамилия> <Первая буква имени Первая буква Отчества Фамилия> и др. В качестве первого шага необходимо по нормализованному русскому имени сгенерировать все возможные варианты транслитерации, а затем соответствующие им варианты атрибута *full-name* и сравнить эти варианты с теми именами, которые содержатся в базе данных RKBExplorer.com.

Конечно, методы сравнения строковых значений не являются новой областью исследования и имеют обширную библиографию [Cohen 2003]. Поэтому достаточно сказать, что для сравнения вариантов имен используется токенизованный вариант метрики Jaro-Winkler [Cohen 2003]. То есть сначала сравниваются фамилии из данных RKBExplorer и Открытого архива с очень высоким пороговым значением, а затем с более низким порогом сравниваются варианты имени и отчества. Неточное совпадение имен на данном этапе является необходимым условием, обеспечивающим полноту поиска, поскольку в латинских вариантах данных постоянно обнаруживаются не предусмотренные правилами варианты транслитерации. Так, в ходе этого эксперимента в базе данных RKBExplorer.com были обнаружены публикации, соответствующие персоне A. P. Yershóv.

После завершения этапа генерации кандидатов программно строится SPARQL-запрос, который по списку фамилий, идентифицированных как совпавшие в обеих базах знаний, выдает список идентификаторов RKBexplorer и атрибутов *akt:full-name*, соответствующих каждому варианту атрибута *ons:name*, и список публика-

ций, связанных с конкретным идентификатором. При этом каждому из этих имен Открытого архива соответствует несколько разных идентификаторов персон, каждый из которых имеет свой собственный атрибут *akt:full-name* и отдельный список публикаций. Так, например, для персоны «Андрей Петрович Ершов» из Открытого архива в базе данных dblp.rkbexplorer.com было обнаружено 18 экземпляров персон с атрибутом *akt:full-name* равным «Andrei P. Ershov», при этом большинству найденных идентификаторов соответствовало по одной публикации. Например, экземпляру с идентификатором <http://dblp.rkbexplorer.com/id/people-e1ac8593dbc7db6ec5766ea313914be4-1211d4d9974a0a977bd166da859d928f> соответствовала одна публикация «Mixed computation in the class of recursive program schemata». А экземпляру с идентификатором <http://dblp.rkbexplorer.com/id/people-e1ac8593dbc7db6ec5766ea313914be4-2-fd1e3b39206345ab05fd9be97bc0d00> соответствовала одна публикация «Time sharing: the need for re-orientation». Помимо этого обнаружились другие персоны с похожими атрибутами *akt:full-name* и своими наборами публикаций. Две публикации соответствовали двум персонам с разными идентификаторами и атрибутом *akt:full-name* равным «A. P. Yershóv», одна публикация – персоне с атрибутом *akt:full-name* равным «A. Ershov», одна публикация – персоне с атрибутом *akt:full-name* равным A. Yershov, две публикации – персоне с атрибутом *akt:full-name* равным Andrew Ershov.

Возникает два вопроса:

- 1) Какие из вышеупомянутых идентификаторов соответствуют одному и тому же физическому объекту и, стало быть, могут быть связаны отношением *owl:sameAs*, а какие из них описывают разные физические объекты?
- 2) Все ли публикации, приписанные персоне с одним идентификатором, принадлежат одному и тому же физическому объекту?

Очевидно, что ответы на эти вопросы имеют существенное влияние, например, на подсчет индекса цитирования. Наши эксперименты показали, что, как правило, публикации персон, имеющихся в Открытом архиве, бывают разбросаны по нескольким разным персонам с разными идентификаторами из базы данных RKBExplorer, и иногда одному и тому же идентификатору приписываются публикации разных физических персон. Стандартным способом иденти-

фикации персон считается их идентификация не только по имени, но и по адресу электронной почты или по персональной странице в Интернете. К сожалению, для многих персон, составляющих наполнение Открытого архива, такой информации не существует. Кроме этого, эти данные могут меняться вместе с изменением места работы. Зато в Открытом архиве поддерживается информация о местах работы персон, с указанием периода работы в конкретной организации, как это показано на примере А.П. Ершова. Кроме этого, в открытом доступе появляются тексты публикаций, которые также могут быть использованы для идентификации персон.

Поэтому в настоящий момент выполнена первая серия экспериментов по идентификации персон на основании их публикаций. Выполняется два вида проверки.

1) Проверка по месту работы. Дата каждой публикации извлекается из RKBExplorer при помощи SPARQL-запроса, по этой информации находится место работы персоны, соответствующее указанному периоду в Открытом архиве, и найденное место работы сравнивается с местом работы, указанным в тексте публикации (если имеется).

2) Проверка по списку цитируемой литературы. Упрощенная процедура состоит в том, что варианты имени тестируемого автора каждой публикации сравниваются с именами авторов публикаций, извлекаемых из списка цитируемой литературы. Если обнаруживается (приблизительное) совпадение имен, то текущая публикация объединяется в одно множество с названиями публикаций одноименного автора из списка цитируемой литературы. Затем та же самая процедура применяется к добавленным публикациям. Эта процедура не всегда позволяет однозначно определить идентичность персон и в дальнейшем планируется ее развитие в нескольких направлениях. Во-первых, предполагается строить более разветвленную сеть цитирования, учитывающую не только автора, нуждающегося в идентификации, но и его соавторов. А во-вторых, включать в эту сеть цитирования ссылки на всех авторов, а не только на авторов с похожими именами.

Заключение

В данной работе рассмотрены проблемы обогащения научных баз знаний при помощи контента библиографических порталов из облака LOD и подходы к их решению. Соответствие между наборами данных, основанных на этих онтологиях, устанавливается при помощи SPARQL-запросов, которые генерируются на основе визуализации онтологий.

Также продемонстрировано, что обычные инструменты, применяемые для установления идентичности сущностей на основе строковых метрик сходства, не позволяют различать авторов публикаций, являющихся тезками или даже однофамильцами. Рассмотрены альтернативные подходы к решению этой проблемы.

СПИСОК ЛИТЕРАТУРЫ

[Марчук, 2010] Марчук А.Г., Марчук П.А. Особенности построения цифровых библиотек со связанным контекстом //Труды RCDL'2010. Двеннадцатая Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» Казань, Казанский университет , 2010. – С. 19–23.

[Apanovich, 2013] Zinaida Apanovich and Alexander Marchuk Experiments on using the LOD cloud datasets to enrich the content of a scientific knowledge base P.Klinov and D.Mouromtsev (Eds.) KESW 2013, CCIS 394 pp. 1–14, Springer Verlag Berlin Heidelberg 2013.

[Apanovich, 2010] Apanovich Z. V., Vinokurov P. S. An extension of a visualization component of ontology based portals with visual analytics facilities. // Bulletin of NCC . – Issue 31. – 2010. – pp. 17–28.

[Bizer, 2009] Bizer, C., Heath, T. , Berners-Lee, T. Linked Data - The Story So Far. //Int. J. Semantic Web Inf. Syst., 5 (3). 2009. P. 1–22.

[Cohen, 2003] William W. Cohen, Pradeep D. Ravikumar, Stephen E. Fienberg: A Comparison of String Distance Metrics for Name-Matching Tasks. IIWeb 2003: 73–78.

[Isele, 2010] Isele R., Jentzsch A., Bizer Ch. Silk Server - Adding missing Links while consuming Linked Data// 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010.

[Schultz, 2012] Schultz A. et al. How to integrate LINKED DATA into your application //Semantic technology & Business Conference, San Francisco, June 5, 2012. http://mes-semantics.com/wp-content/uploads/2012/09/Becker-et-al-LDIF_SemTechSanFrancisco.pdf.

[Shvaiko, 2013] Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges IEEE Transactions on Knowledge and Data Engineering, 25(1) pp. 158–176 (2013).

[Tramp, 2012] Tramp, S., Williams, H., Eck, K.: Creating Knowledge out of Interlinked Data: The LOD2 Tool Stack <http://lod2.eu/Event/ESWC2012-Tutorial.html>.

[Ngomo, 2011] Ngomo A.-C. N., Auer S.: LIMES – A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. //IJCAI 2011: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16–22, 2011 pp. 2312–2317.

[AKT, 2014] Описание AKT Reference ontology: <http://www.aktors.org/ontology>.

[RKBExplorer 2014] Набор данных: RKBExplorer rkbexplorer.com/.

[DBLP 2014] Набор данных DBLP: <http://dblp.rkbexplorer.com/>.

УДК 81'322

ФОРМАЛЬНАЯ МОДЕЛЬ ИМЕННЫХ ЧАСТЕЙ РЕЧИ КАЗАХСКОГО ЯЗЫКА

Б.Ж. Ергеш (*b.yergesh@gmail.com*), **А.С. Муканова,**
Г.Т. Бекманова, А.А. Шарипбай

Евразийский национальный университет им. Л.Н. Гумилева,
Астана, Казахстан

В данной работе описывается построение формальных моделей именных частей речи казахского языка с помощью семантических гиперграфов, на примере имени существительного.

Введение

Казахский язык относится к тюркским языкам, которые в свою очередь относятся к агглютинативным языкам. Казахский язык характеризуется большим числом словоформ для каждого слова. Словоформы образуются при помощи добавления к начальной форме слова (основе) суффиксов и окончаний. Суффиксы и окончания присоединяются в строго определенной последовательности. Слова в казахском языке изменяются по числам, падежам, лицам. Есть притяжательный падеж, как в английском языке.

В настоящее время онтология является мощным и распространенным инструментом моделирования отношений между объектами различных предметных областей. Принято классифицировать онтологии по степени зависимости от задач или прикладной области, по модели представления онтологических знаний и его выражительным возможностям и другим параметрам [Gruber, 1995]. Прикладные онтологии описывают концепты, которые зависят как от онтологии задач, так и от онтологии предметной области. Прикладная онтология разрабатывается на основе общих принципов построения онтологий, но с учетом использования в качестве модели представления знаний семантических гиперграфов. Данный формализм позволяет определить онтологию O в виде тройки: (V, R, K) , где V – множество понятий проблемной среды (вершины гиперграфа), R – множество отношений между понятиями

ми (дуги и ребра гиперграфа), а K – множество имен понятий и отношений в данной предметной области.

Одним из формальных средств представления знаний является язык семантических гиперграфов, в котором можно в зависимости от типов связей реализовывать классифицирующие, функциональные, ситуационные, структурные сети и сценарии. Семантический гиперграф является расширением семантических сетей, где естественным образом представляются n -арные отношения, которые позволяют задавать не только атрибуты объектов, но и представлять их структурные, «целостные» описания [Хахалин, 2009].

В работе [Zhen, etc., 2010] семантический гиперграф описывается как «семантическая сеть на основе гиперграфа (hyper-graph based semantic network (Hy-SN))», они могут представлять более сложные семантические отношения и имеют более эффективную структуру данных для хранения знаний в хранилищах.

Известно [Berge, 1985], [Vizing, 2007], что гиперграф $H(V, E)$ определяется парой (V, E) , где V – множество вершин $V = \{v_i\}$, $i \in I = \{1, 2, \dots, n\}$, а E – множество ребер $E = \{e_j\}$, $j \in J = \{1, 2, \dots, m\}$; каждое ребро представляет собой подмножество V . Вершина v и ребро e называются инцидентными, если $v \in e$. Для $v \in V$ через $d(v)$ обозначается число ребер, инцидентных вершине v ; $d(v)$ называется степенью вершины v . Степень ребра e – число вершин, инцидентных этому ребру, – обозначается через $g(e)$.

Представление модели морфологических правил в виде онтологии позволяет перевести морфологическую модель практически один к одному в объектно-ориентированную модель представления данных, где классы – это части речи казахского языка, а объекты – их семантические категории, например, одушевленность и неодушевленность.

Использование онтологической модели для представления морфологических правил частей речи позволяет полностью описать морфологическую модель и их отношения. Семантический гиперграф используется для описания понятий. Такое представление позволяет перевести в объектно-ориентированную модель данных, где классами являются вершины семантического гиперграфа.

В этой работе подробно описывается существительное казахского языка.

Цель исследования: автоматизация генерации словоформ и новых слов казахского языка и морфологического анализатора казахского языка.

Формализация любого естественного языка является сложной задачей. По мнению авторов, проблема формализации казахского языка хорошо решается через предложенную ниже модель.

Формальная модель имени существительного

Для имен существительных (N) в качестве семантических признаков начальных форм выступают одушевленность (anim) и неодушевленность (inan). В зависимости от этого признака и определяется траектория словоизменения имени существительного. Имя существительное в казахском языке спрягается (pers_end) и изменяется по падежам (cases), а также числам (number) и имеет притяжательную форму (poss_end) [Қазақ ғрамматикасы, 2005].

Для построения онтологии мы использовали редактор онтологии Protégé [protégé, 2014]. Это свободно распространяемый и открытый редактор онтологии и фреймворк для построения баз знаний. Protégé был разработан в Стэнфордском университете совместно с университетом Манчестера.

На рисунке 1 показана онтологическая модель имени существительного с учетом семантических признаков.

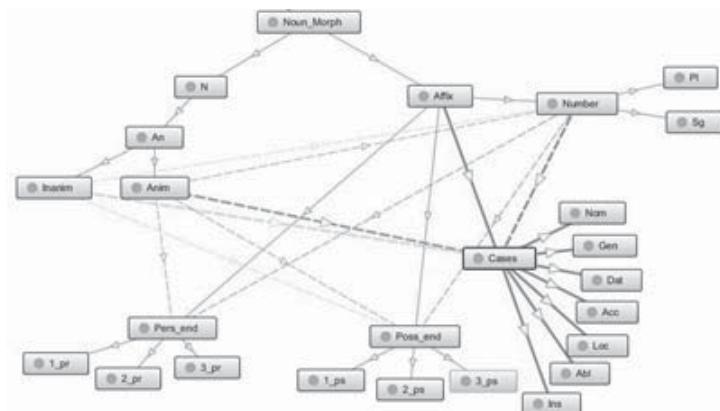


Рис. 1. Онтологическая модель имени существительного

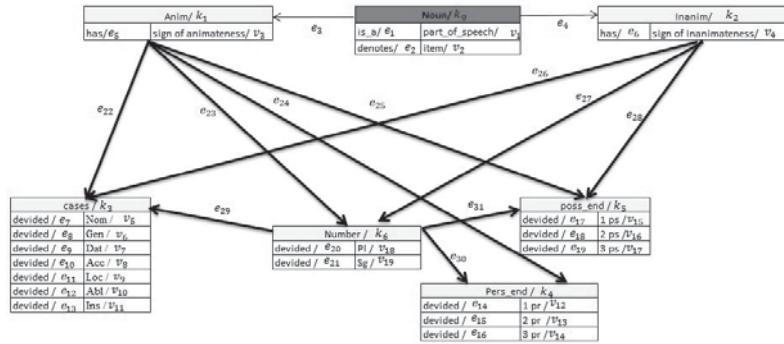


Рис.2. Графическое представление онтологий с использованием семантического гиперграфа

Таблица 1 описывает понятия и отношения онтологической модели имени существительного.

*Таблица 1
Понятия и отношения*

ID	Обозначение	Описание
v0	N	имя существительное
v1	Part_of_speech	Части речи
v2	Item	Предмет
k ₁	Anim	Одушевленный
v3	Sign of animateness	Признак одушевленности
k ₂	Inanim	Неодушевленный
v4	Sign of inanimateness	Признак неодушевленности
k ₃	Cases	Падежи
v5	Nom	именительный падеж
v6	Gen	родительный падеж
v7	Dat	направительный падеж
v8	Acc	винительный падеж
v9	Loc	местно-временной падеж

v10	Abl	исходный падеж
v11	Ins	инструментальный падеж
k ₄	Pers_end	Персональность
v12	1 pr	1-е лицо
v13	2 pr	2-е лицо
v14	3 pr	3-е лицо
k ₅	Poss_end	Посессивность
v15	1 ps	принадлежность 1 лицу
v16	2 ps	принадлежность 2 лицу
v17	3 ps	принадлежность 3 лицу
k ₆	Number	Число
v18	Pl	множественное
v19	Sg	единственное
e1	is_a	является
e2	denotes	обозначает
e3,e4	has_feature	имеет форму
e5,e6	has	имеет
e7 – e21	devided	делится
e22 – e28	change	изменяется
e29 – e31	add	добавляется

Для дифференциации семантических гиперграфов от общего семейства графов звенья и гипердуги будем называть семантическими дугами, а также примем, что в множество вершин семантического гиперграфа входит множество классов $K = \{k_\alpha\}$, $\alpha \in A = \{0, 1, 2, 3, \dots, n\}$, в каждый из которых в свою очередь входит множество экземпляров класса [Починский, 2012]. Вершину-класс можно представить следующей тройкой:

$$k_a = \{V_a, E_a, S_a\},$$

где V_a – множество свойств класса, E_a – множество инцидентных классу семантических дуг, S_a – множество экземпляров класса

Вершина-класс «существительное»:

$$\begin{aligned} k_0 &= \{\{v_1, v_2\}, \{e_1, e_2\}, S_1\} \\ k_1 &= \{\{v_3\}, \{e_5\}, S_2\} \\ k_2 &= \{\{v_4\}, \{e_6\}, S_3\} \\ k_3 &= \{\{v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}, \{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}\}, S_4\} \\ k_4 &= \{\{v_{12}, v_{13}, v_{14}\}, \{e_{14}, e_{15}, e_{16}\}, S_5\} \\ k_5 &= \{\{v_{15}, v_{16}, v_{17}\}, \{e_{17}, e_{18}, e_{19}\}, S_6\} \\ k_6 &= \{\{v_{18}, v_{19}\}, \{e_{20}, e_{21}\}, S_7\} \end{aligned}$$

Мы можем представить морфологическую модель существительного с помощью семантического гиперграфа:

Гиперграф $H(V, E)$,

$$\begin{aligned} \text{где } V &= K = \{k_a\}, E = \{e_a\} \quad V = \{k_0, k_1, k_2, k_3, k_4, k_5, k_6\} \\ E &= \{e_3 = \{k_0, k_1\}, e_4 = \{k_0, k_2\}, e_{22} = \{k_1, k_3\}, e_{23} = \{k_1, k_4\}, \\ &\quad e_{24} = \{k_1, k_5\}, e_{25} = \{k_1, k_6\}, e_{26} = \{k_2, k_3\}, e_{27} = \{k_2, k_5\}, e_{28} \\ &\quad = \{k_2, k_6\}, \\ e_{29} &= \{k_6, k_3\}, e_{30} = \{k_6, k_4\}, e_{31} = \{k_6, k_5\}\} \end{aligned}$$

Из указанного семантического гиперграфа можно получить формальные правила с помощью скобочной записи. В базе данных начальных форм слов объемом 40000 слов с семантическими разметками 25660 слов имена существительные. Количество формальных правил для имени существительного 4552. С помощью этих формальных правил сгенерировано 1 605 725 словоформ имени существительного. Также имя существительное возможно образовать из других частей речи.

Пример словоизменения одушевленного существительного «бала» – «ребенок» содержит все словоформы данного существительного и их морфологическую информацию, которая содержит в сокращенном обозначении информацию о том, в каком числе, падеже находится существительное, от какого лица происходит действие и его принадлежность тому или иному лицу. На примере 1 приведено склонение существительного «бала».

Пример 1. Склонение существительного «бала»

S=бала

$k_0 = \{\{v_1, v_2\}, \{e_1, e_2\}, \{\text{бала}\}\}$

$e_1 = \{\text{бала, anim}\}$,

$e_{22} = \{\text{бала, cases}\}$

$k_3 = \{\{v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}, \{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}\}, \{\text{бала, баланың, балаға, баланы, балада, баладан, баламен}\}\}$

Заключение

Построены онтологические модели морфологических правил казахского языка, что позволило записать формальные правила словоизменения и словообразования каждой части речи. На основе этих правил создан морфологический анализатор казахского языка. Он может быть использован для создания орфографических корректоров казахского языка, которых в данное время не существует. Полученный морфологический анализатор может лежать в основе различных трансляторов, семантических поисковиков, использоваться в речевых технологиях и т.д.

СПИСОК ЛИТЕРАТУРЫ

[Gruber, 1995] Gruber T.R.: Toward Principles for the Design Of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies Vol. 43, Issues 5-6, pp. 907–928, 1995.

[Хахалин, 2009] Г.К. Хахалин. Прикладная онтология на языке гиперграфов // Материалы II Всероссийской конференции «Знания-Онтологии-Теории» (ЗОНТ-09). Новосибирск. – 2009. – С. 223–231.

[Zhen, etc., 2010] Zhen L, Jiang Z. Hy-SN: Hyper-graph based semantic network // Knowledge-Based Systems Vol 23, Issue 8, December 2010, Pages 809–816.

[Berge, 1985] Berge C.C.: Graphs and Hypergraphs. Elsevier Science Ltd, 1985.

[Vizing, 2007] Vizing V.G.: About a coloring of intsdentor in the hypergraph. Ser. 1, 40–45, 2007.

[Қазақ грамматикасы, 2005] Қазақ грамматикасы. Фонетика, сөзжасам, морфология, синтаксис// Астана-2002. С. 20–25.

[protégé, 2014] <http://protege.stanford.edu>

[Починский, 2012] И.А. Починский. Формальное представление семантических гиперграфов и операций над ними // http://rgu-penza.ru/mni/content/files/2012_Pochinskii.pdf.

УДК 81'322.2:81'322.3:004.896

ВОССТАНОВЛЕНИЕ ГРАММАТИЧЕСКИХ ЭЛЛИПСИСОВ ПРИ СИНТАКСИЧЕСКОМ АНАЛИЗЕ¹

Т.Ю. Кобзарева (*t.kobzareva@gmail.com*)

М.Е. Епифанов (*xeme@rambler.ru*)

Д.Г. Лахути (*delir1@yandex.ru*)

Российский государственный гуманитарный университет, Москва

Рассмотрена проблема восстановления грамматических эллипсисов при автоматическом синтаксическом анализе русского предложения. Обсуждаются грамматические условия появления четырех наиболее вероятных типов эллипсисов, позволяющие находить их антецеденты.

Введение

В языке существует несколько механизмов, позволяющих избегать лексических повторов, в частности – грамматический эллипсис – опущение повторяющегося слова или фрагмента предложения (S). Человек, не задумываясь, восстанавливает по контексту опущенные слова: *Он взял пирожок и съел* (пирожок). *Он дал сыну два печенья, а себе взял три* (печенья). *Маша надела красные перчатки, а мне протянула новые синие* (перчатки).

Однако автоматический анализ таких S не может быть реализован стандартным способом, так как эллипсисы нарушают обычную синтаксическую структуру S. Этого не будет происходить, если на ранних этапах анализа мы сможем восстановить опущенные элементы.

До настоящего времени при автоматическом анализе такая задача не ставилась. Ниже описан один из возможных подходов к ее решению в системе автоматического анализа, разрабатываемой в РГГУ. Специфика системы определяется тем, что сегментация S предшествует построению большей части синтагматических связей, что существенно упрощает решение многих задач анализа [Кобзарева, 2007].

¹ Работа частично поддержана грантами РФФИ 12-06-00366-а и 10-01-00532-а.

Система состоит из 6 блоков, работающих в жестком порядке:

1. **Постморфология** – решение несловарных проблем морфологического анализа.
2. **Разрешение омонимии частей речи** [Кобзарева, Афанасьев, 2002].
3. **Предсегментация** – построение проективных фрагментов определительных именных и предложных групп [Кобзарева, 2007], сложного сказуемого и т.д. – связей, формирующих единицы линейной структуры S для этапа сегментации.
4. **Сегментация**: построение сегментов – простых, придаточных S и всякого рода оборотов, границы которых в русском S по правилам пунктуации задаются эксплицитно знаками препинания и сочинительными союзами [Кобзарева, 2007].
5. **Внутрисегментный анализ** – построение связей слов внутри сегментов.
6. **Межсегментный анализ** – построение графа связей сегментов.

Уже сегментация [Кобзарева, 2007] и тем более построение связей слов внутри сегментов в S с эллипсисами порождают множество проблем, поэтому S с эллипсисами до настоящего времени не анализировались.

В настоящей работе кратко рассмотрены элементы грамматики восстановления самых частых типов эллипсисов на базе результатов работы первых трех модулей системы, то есть перед сегментацией.

1. Синтаксические аномалии при грамматических эллипсисах

Каждой «маленькой драме» Л.Теньера [Теньер, 1988] – ситуации, из которых строится S, соответствует простое S или некоторый его трансформ в составе сложного предложения. В таком графе в норме есть предикат-вершина, актанты и сирконстанты. Известны потенциальные роли частей речи в определенных формах в таком графе.

Построение графа связей слов опирается на информацию словаря и грамматики системы. При эллипсисе – элиминировании слов при их потенциальном повторе – оказывается невозможным

строить прогнозируемые словарем и грамматикой связи. Так как возможен эллипсис почти любого узла графа или даже нескольких узлов одновременно, эллипсисы имплицируют разные синтаксические аномалии.

Базой восстановления эллипсисов служит грамматика, описывающая аномальные структуры с эlimинированными потенциальными лексическими повторами: эллипсис возможен только в определенных контекстных ситуациях – при синтаксическом «параллизме» – подобии двух структур, когда без эллипсиса в одном из фрагментов предложения возник бы повтор уже названной лексемы или некоторого фрагмента предложения.

Основой соответствующих алгоритмов является словарь контекстов с потенциальными повторами, в которых, чтобы повторов избежать, опускается одно или нескольких слов. Кроме того, в некоторых случаях удается определить морфологические условия, которым должен удовлетворять антецедент эллипсиса.

2. Наиболее распространенные типы эллипсисов

Рассмотрим основные типы эллипсисов. В примерах на месте эллипсиса стоит Ø, а в круглых скобках около него – одно или несколько опущенных слов.

Тип I. Эллипсис существительных в прилагательных актантах «с сохранением представителя».

Часто встречается опущение существительного (N) в именной группе (ИГ) – эллипсисы «с сохранением представителя», рассмотренные в [Падучева, 1974]: в ИГ эlimинировано N, но сохраняется некоторая ее компонента, отличающая ИГ с эллипсисом от ИГ с антецедентом.

Тип Ia. Ак без N-хозяина: есть согласованное определение (Ак) – прилагательное, местоименное прилагательное, причастие, порядковое числительное, но нет N – хозяина Ак: *Я покупаю книги о войне и исторические Ø* (книги). *Мы включили верхнюю лампу, и нижняя Ø* (лампа) тут же погасла. *Правая рука сильнее левой Ø* (руки). *Моя жизнь* была длиннее, чем *его Ø* (жизнь).

Тип Ib. В ИГ есть количественное или местоименное числительное, но нет полнозначного N: *Петр съел уже две порции мороженого, и я хочу заказать две Ø* (порции мороженого). *Петя нари-*

совал не три акварели, а две Ø (акварели). Сделать 5 упражнений полезнее, чем 3 Ø (упражнения)

Тип II. Элиминирование актанта. Опускается целиком ИГ – актант предиката: *Петр не читал статьи об этой истории, а Иван читал Ø (статью об этой истории). Он открыл сестре дверь, а она забыла потом закрыть Ø (дверь). Они долго любовались этим собором, но не сфотографировали Ø (этот собор).*

Тип III. Эллипсис предикативной вершины. Опускается предикат при его повторе в цепочке сочиненных S, имеющих одинаковую морфо-синтаксическую структуру: *Из пяти их сестер Наталья была за Петерсоном, Вера – Ø (была) за Пыхачевым, Нина – Ø (была) за бароном Раушем фон Траубенберг, Елизавета – Ø (была) за князем... (В. Набоков).*

Тип IV. Эллипсис фрагмента S. В цепочке сочиненных S с одинаковой морфо-синтаксической структурой опускается, начиная со второго из сочиненных, структурно и лексически повторяющийся фрагмент: *Двадцать лет такого танца составляют эпоху, сорок – Ø (лет такого танца составляют) историю. (О.Мандельштам) На одном из листочек написано отец, на другом – Ø (из листочек написано) мать.* (<http://www.folks.ru/mystery.php?art=183>)

Возможны комбинации разных типов эллипсисов: *Мне налили воду, Ø (я) никогда такой вкусной Ø (воды) не пил. Я знаю французский язык, а мой друг Ø (знает) (–) итальянский Ø (язык).*

3. Синтаксические особенности контекстов с эллипсисами

Для восстановления эллипсисов нужно уметь находить контексты, синтаксическая структура которых имплицирует потенциальный повтор. В этих контекстах, соответственно, происходит опущение N и их структура определяет поиск антецедента элиминированного N.

Мы задаем структуру контекстов, в которых происходит опущение каких-то слов или фрагментов S, используя только морфологическую информацию и результаты работы алгоритмов блока предсегментации. То есть при восстановлении грамматических эллипсисов мы опираемся только на последовательность слов и знаков препинания и построенные в предсегментации ИГ, ПГ и сложные именные сказуемые. Линейные структуры, служащие для поиска

эллипсисов и их антецедентов, задаются в алгоритмах линейными конфигурациями [Мельчук, 1964] – последовательностями определенных частей речи, на компоненты которых накладываются определенные морфо-синтаксические условия.

Легко видеть, что, например, эллипсисы типа I появляются в S, где в структурах двух или более фрагментов есть потенциально совпадающие компоненты: при сочинении ИГ с потенциально совпадающими вершинами (ИГСВ), при соподчинении ИГСВ, при подчинении двух ИГСВ двум сочиненным предикатам, при подчинении двух ИГСВ двум сочиненным или соподчиненным предлогам, сочиненным предикатам, вершинам сочиненных S, при подчинении двух ИГСВ предикатам соседних сегментов. При этом ИГ с эллипсисом и ИГ с его антецедентом находятся всегда в разных ветках или деревьях графа S. Эллипсис при подчинении сегментов, как и эллипсис при сочинении сегментов, определяется не типом сегментов и не видом или направлением связи сегментов при подчинении, а только самим фактом наличия их связи и практически всегда – фактом их непосредственного соседства. При этом эллипсис возможен и в первом из двух связанных (м.б., соседних) сегментов, и во втором.

4. Обнаружение эллипсисов типа Ia

Ак в S может: 1) быть субстантивировано (*данные этих опытов его огорчили*); 2) выступать в роли согласованного определения (*он стоял около высокого дома*); 3) входить в сложное сказуемое: *он хотел быть первым; он стал сильный; он высокий* (с нулевой связкой) и т.п.

Если в результате работы первых 3 модулей Ак не имеет N-хозяина, не является вершиной обособленного согласованного определения, не входит в состав сказуемого и не субстантивировано, то это Ак представляет ИГ с эллипсисом типа Ia, так как:

- в блоке разрешения омонимии частей речи [Кобзарева, Афанасьев, 2002] определены случаи субстантивации А;
- в блоке предсегментации определены А-слуги существительных, числительных, имен собственных в кавычках и т.д. в любых возможных в русском языке линейных конфигурациях атрибутивных ИГ, определены А – вершины обособленных определительных оборотов [Кобзарева, 2007], будут найдены А в составе сказуемого (в частности, в настоящем времени без глагола-связки).

5. Синтаксический базис восстановления эллипсисов типа Ia

Рассмотрим контексты, в которых возможен эллипсис типа Ia.

1) Сочинение ИГ или предложных групп (ПГ), где в одной из ИГ опущен N-хозяин: *Нужно взять красную чашку и большую синюю Ø. Нужно поставить чашку на первый стол и на второй Ø. Нужно поставить чашку не рядом с первой тарелкой, а слева от второй Ø. Возьми не красную чашку, а синюю Ø. Возьми не только красную Ø, но и синюю чашку* и т.д.

N – вершина полной ИГ – является антецедентом эллипсиса при условии согласования: Ак и полная ИГ при сочинении ИГ (но не при сочинении ПГ) должны согласоваться по падежу. Для всех эллипсисов I типа: если Ак в ед.ч., то род Ак должен совпадать с грамматическим родом N-антецедента.

2) Соподчинение ИГ\ПГ: есть Ак без хозяина и ИГп, соподчиненная ИГ с эллипсисом: N – вершина полной ИГп = антецедент эллипсиса при выполнении условий согласования: 2а. Соподчинение компаративу: *Красный кубик большие синего Ø \, чем синий Ø*; 2б. соподчинение глаголу: *заменил \ заменяя \ заменивший красный кубик синим Ø / на синий Ø* и др.

3) Сочинение предикатов или сегментов – простых S: *Он взял первую книжку, но забыл вторую Ø* (книжку). *Я взял красную чашку, а он взял синюю Ø* (чашку). При этом предикаты могут быть разными: *Я мыл красную чашку, а он любовался своей синей Ø*. При совпадении сказуемых второе сказуемое, скорее всего, будет опущено.

4) Вложение придаточного или деепричастного оборота в некоторый сегмент. *Я знаю, что он уже разбил свою красную чашку, когда я мыл мою синюю Ø* (чашку). *Когда я мыл красную чашку, он разбил синюю Ø* (чашку). *Когда я мыл красную Ø* (чашку), *он разбил свою синюю чашку*. *Забыв дома красный карандаш, он стал исправлять ошибки синим Ø* (карандашом). *Он искал синий Ø* (карандаш), забыв *красный карандаш* дома.

Ак и антецедент его отсутствующего хозяина могут быть в разных числах и, как, например, в случае, если они оказываются слугами разных предлогов, в разных падежах: *Когда я вымыл синие чашки, он еще любовался красной Ø. Когда я отказался пить из синей Ø, он потянулся за красной чашкой.*

6. Синтаксический базис восстановления эллипсиса типа Iб

Для этого типа эллипсисов мы ищем синтаксические структуры, подобные рассмотренным в разделе 5, например, сочинение двух ИГ, в этом случае – с числительными, во второй из которых эlimинирован повтор полнозначного существительного: *Возьмите две чашки муки и одну Ø (чашку) – сахарного песка. Они принесли 5 коробок с давно прочитанными книгами, а две Ø (коробки) – со старыми альбомами.*

Так как при поиске этих эллипсисов в предсегментации уже построены ИГ с числительными, сигналом, что в структуре есть эллипсис, служит появление количественного числительного \ количественного местоимения (*много, несколько, большинство* и др.) без слуги – полнозначного N. Для всех ситуаций эллипсиса типа Iб вводится особое условие согласования числительного в неполной ИГ с антецедентом по роду, а при сочинении ИГ и в некоторых других случаях – и по падежу.

7. Восстановление эллипсисов типа II, III и IV

Для эллипсисов этих типов поиск эллипсиса опирается только на структуру контекстов, которые, как и для поиска антецедентов других типов эллипсисов, задаются линейными конфигурациями.

Указанием на факт этих типов эллипсисов может быть появление (факультативного) тире на месте опущенного элемента S.

На предложение накладывается маска: задаются условия, необходимые для появления эллипсиса, – совпадения двух цепочек слов, где порядок следования некоторых частей речи и соответствующих форм слов сегмента с эллипсисом совпадает с частями речи и формами слов начала и конца сегмента без эллипсиса.

8. Объектное моделирование алгоритмов восстановления эллипсисов

С целью коррекции и дальнейшего совершенствования алгоритмов синтаксического анализа выполняется их объектное моделирование в специально разработанной инструментальной программной среде ЭСЛА (т.е. среде для Экспериментов С Лингвистическими Алгоритмами), обеспечивающей возможность внесения по ходу

отладки разрабатываемой системы ПСА изменений в алгоритмы и программы анализа без полного перепрограммирования всей системы (о необходимости такого подхода см. «Введение» к [Епифанов и др., 2010]). Входной язык ЭСЛА, применяемый для определения в ней объектной модели лингвистического алгоритма, структурно «приближен» к способу записи алгоритмов, используемому лингвистом, что облегчает процесс программирования и отладки.

Процесс отладки в ЭСЛА выполняется аналогично тому, как это делает программист в современных средах для разработки программ. Здесь тоже имеется возможность пошагового выполнения алгоритма с установкой прерывания на актуальных его узлах, просмотра контекста выполнения и пробного вычисления выражений элементов правил на промежуточных шагах. В процессе применения алгоритма к тестовому примеру (чаще всего это – отдельное русское предложение) строится трасса его вычисления (получение с той или иной степенью детализации протокола его применения к тестовому предложению). Имеются развитые средства просмотра результата применения алгоритма к тестовому примеру и протокола вычисления.

В настоящее время выполняются объектное моделирование и первичная отладка двух алгоритмов восстановления эллипсисов:

- существительного (N) с сохранением согласованного с ним определения (A);
- существительного в ИГ с Nnum и факультативным тире – знаком эллипсиса.

Оба эти алгоритма применяются в конце блока предсегментации, т.е. до них к анализируемому предложению уже были применены алгоритмы блоков постморфологии и разрешения частеречной омонимии. В самом блоке предсегментации уже были применены (среди прочих) алгоритмы выделения в тексте определительных (атрибутивных) отношений ИГ и предложных групп ПГ, анализа сложного сказуемого.

Заключение

В основе механизмов эллипсиса лежит, как и при анафорическом замещении, внутренняя установка языка на избегание повтора, однообразия лексического и структурного. В работе рассмотрено, как грамматические условия появления эллипсиса – аномалии,

которые возникают в синтаксической структуре S при элиминировании потенциального повтора, – можно использовать при поиске его антецедента.

СПИСОК ЛИТЕРАТУРЫ

- [Епифанов и др., 2010] Епифанов М.Е., Антонова А.Ю., Баталина А.М., Кобзарева Т.Ю., Лахути Д.Г. Итеративное применение алгоритмов снятия частеречной омонимии в русском тексте // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (2010). (Бекасово, 26–30 мая 2010 г.). Вып. 9 (16). – М.: Изд-во РГГУ, 2010. С. 119–123.
- [Кобзарева, Афанасьев, 2002] Кобзарева Т.Ю., Афанасьев Р.Н. Универсальный модуль предсинтаксического анализа омонимии частей речи в русском языке на основе словаря диагностических ситуаций // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2002. Протвино, 2002. Т.2. С. 258–268.
- [Кобзарева, 2007] Кобзарева Т.Ю. Иерархия задач поверхностно-синтаксического анализа русского предложения // НТИ, Сер.2, №1, 2007. С. 23–35.
- [Мельчук, 1964] Мельчук И.А. Автоматический синтаксический анализ. 1964. Том 1. Общие принципы. Внутрисегментный синтаксический анализ. Новосибирск. 1964.
- [Падучева, 1974] Падучева Е.В. О семантике синтаксиса. Материалы к трансформационной грамматике русского языка. – М.: Наука, 1974.
- [Теньер, 1988] Теньер Люсьен. Основы структурного синтаксиса. — М.: Прогресс, 1988.

УДК 004.7

СЕТЬ ЕСТЕСТВЕННЫХ ИЕРАРХИЙ ТЕРМИНОВ НА ПРИМЕРЕ АНАЛИЗА НАУЧНЫХ ТЕКСТОВ

Д.В. Ландэ (*dwlande@gmail.com*)

Институт проблем регистрации информации НАН Украины, Киев

А.А. Снарский (*asnarskii@gmail.com*)

НТУУ «Киевский политехнический институт», Украина, Киев

Е.В. Ягунова (*iagounova.elena@gmail.com*)

Санкт-Петербургский гос. университет, Санкт-Петербург, Россия

Описывается методика построения сетей естественных иерархий терминов на основе анализа текстовых корпусов. Методика базируется на применении компактифицированных графов горизонтальной видимости для терминов, а также установлении связей включения между ними. Построена и исследована сеть языка, сформированная на основе обработки полных текстов конференции OSTIS-2014.

Введение

Построение большой отраслевой онтологии – сложная проблема, которая требует больших ресурсных затрат. При этом определенным этапом построения общих онтологий является построение словарных номенклатур, тезаурусов, терминологических онтологий. Эффективный автоматический отбор отдельных терминов для таких конструкций – не решенная окончательно задача, есть отдельные попытки ее решения методами «взвешивания» слов и устойчивых словосочетаний, выполняемых с помощью различных алгоритмов [Лукашевич и др., 2007]. Проблема установления связей при построении сетей из таких терминов также остается открытой.

Ниже описана методика построения сети естественной иерархии терминов (СЕИТ), которую можно рассматривать как терминологическую основу для формирования соответствующей терминологической онтологии. Сеть естественной иерархии терминов базируется на информационно-значимых элементах текста, опорных терминах [Yagunova и др., 2012], методология выявления которых приведена в [Lande и др., 2013-1, 2, 3]. Использование таких эле-

ментов позволяет формировать терминологические сети, охватывать целые области знаний в качестве основ для дальнейшего построения общих онтологий. Опорные термины при этом обычно выбираются с учетом такого их свойства, как «дискриминантная сила» [Salton, 1983]. Вместе с тем одного этого свойства часто оказывается недостаточно для построения терминологических сетей. Иногда слова с низкой дискриминантной силой, в частности, наиболее частотные из выбранной предметной области (например, слова «Информация», «Семантика», «Интеллект» в корпусе текстов по семантическим технологиям) являются важными для рассматриваемой задачи.

1. Алгоритм формирования СЕИТ

Формирование сети естественных иерархий терминов (СЕИТ) базируется на контенте текстовых корпусов выбранной для анализа направленности. «Естественность» иерархий терминов в этом случае понимается как отказ при формировании сети от специальных методов семантического анализа; все связи в такой сети определяются естественным взаимным положением слов и словосочетаний, которые экстрагируются из текстов статистически значимых объемов (как показывает практика, минимальный объем для такого анализа – 20 КБ). СЕИТ, создаваемая полностью автоматически, может рассматриваться как основа для формирования онтологии с участием экспертов. Алгоритм формирования СЕИТ предусматривает реализацию последовательности шагов [Lande, 2014]:

1. На первом этапе выбирается исходный текстовый корпус, в качестве которого рассматриваются тексты научных докладов, вошедшие в сборник трудов научно-технической конференции OSTIS-2014 (Open Semantic Technologies for Intelligent Systems – OSTIS), прошедшей в г. Минске в феврале 2014 года [OSTIS, 2014]. В сборник включено около 100 докладов общим объемом свыше 2 млн. символов. Предварительная обработка такого текстового корпуса предусматривает выделение фрагментов текстов (докладов, абзацев, предложений, слов), исключение нетекстовых символов, отсечение флексивных окончаний слов.

2. На втором этапе каждому отдельному термину из текста – слову, двух- или трехсловному сочетанию (биграмме или триграмме) ставится в соответствие оценка его «дискриминантная сила», а

именно TFIDF, которая в каноническом виде определяется как произведение частоты соответствующего термина (Term Frequency) из фрагмента текста на двоичный логарифм от величины, обратной к количеству фрагментов текста, в которых этот терм встретился (Inverse Document Frequency).

3. Для последовательностей терминов и их весовых значений по TFIDF строятся компактифицированные графы горизонтальной видимости (CHVG) и выполняется повторное определение весовых значений слов уже по этому алгоритму. Данная процедура позволяет учитывать в дальнейшем кроме терминов с большой дискриминантной силой также высокочастотные термины, которые имеют большое значение для общей тематики текстового корпуса [Lande и др., 2013-1]. В качестве весовых оценок терминов используются степени соответствующих им узлов в CHVG. После этого все термины текста сортируются по убыванию рассчитанных весовых значений соответствующих узлов CHVG. Дальнейшему анализу не подлежат термины из так называемого стоп-словаря, являющиеся важными для связности текста, но не несущие информационной нагрузки. Используемый в рамках данной работы стоп-словарь был построен на основе различных стоп-словарей в доступном виде на веб-ресурсах (<http://code.google.com/p/stop-words/source/browse/trunk/stop-words/> stop-words-russian.txt; <http://www.ranks.nl/stopwords/>; <https://trac.mysvn.ru/punbb/punbb/browser/trunk/Russian/stopwords.txt>).

4. Экспертным методом определяется необходимый размер СЕИТ (число N), после чего выбирается соответствующее количество единичных слов, биграмм и триграмм (всего $N+N+N$ элементов) с наибольшими весовыми значениями по CHVG.

5. Из отобранных терминов строятся СЕИТ, в которых как узлы рассматриваются сами термины, а связи соответствуют вхождениям одних терминов в другие. На рис. 1 проиллюстрирован принцип построения связей СЕИТ. Различные геометрические фигуры на рис. 1 соответствуют различным словам. Первой строке соответствует выбранное множество единичных слов, второму – множество биграмм, а третьему – множество триграмм. Если единичное слово входит в биграмму или триграмму, или биграмма входит в триграмму, образуется связь. Множество узлов и связей образует трехуровневую сеть естественной иерархии терминов.

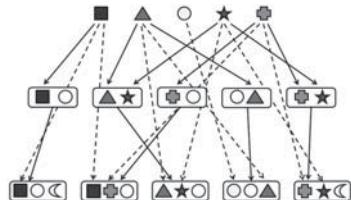


Рис. 1. Трехуровневая сеть естественной иерархии терминов

После формирования СЕИТ осуществляется ее отображение программными средствами анализа и визуализации графов. Для загрузки СЕИТ в базы данных формируется матрица инцидентности в формате csv. В табл. 1 приведены списки 20 наиболее весомых терминов (слов, биграмм и триграмм) из сборника трудов конференции.

Таблица 1

№	Слова	Биграммы	Триграммы
1	СИСТЕМА	ПРЕДМЕТНАЯ ОБЛАСТЬ	ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ
2	ЗНАНИЯ	БАЗА ЗНАНИЙ	ОНТОЛОГИЯ ПРЕДМЕТНОЙ ОБЛАСТИ
3	ОНТОЛОГИЯ	СЕМАНТИЧЕСКАЯ СЕТЬ	ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
4	МОДЕЛЬ	ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА	УНИВЕРСАЛЬНЫЙ СЕМАНТИЧЕСКИЙ КОД
5	ДАННЫЕ	ЕСТЕСТВЕННЫЙ ЯЗЫК	ХРАНИЛИЩЕ СВЯЗАННЫХ ДАННЫХ
6	СЕМАНТИЧЕСКИЙ	РЕШЕНИЕ ЗАДАЧ	ПОНЯТИЯ ПРЕДМЕТНОЙ ОБЛАСТИ
7	ЯЗЫК	ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ	ЯЗЫК ПРЕДСТАВЛЕНИЯ ЗНАНИЙ
8	ЗАДАЧА	ПРИНЯТИЕ РЕШЕНИЙ	ЗНАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ
9	ОБЛАСТЬ	ПРЕДСТАВЛЕНИЕ ЗНАНИЙ	ОСНОВАННАЯ НА ЗНАНИЯХ СИСТЕМА
10	ОБЪЕКТ	БАЗА ДАННЫХ	МАШИННАЯ ОБРАБОТКА ЗНАНИЙ
11	РЕШЕНИЕ	НЕЙРОННАЯ СЕТЬ	БАЗА КОНТЕКСТНЫХ ПРАВИЛ
12	СЕТЬ	СИСТЕМА УПРАВЛЕНИЯ	КЛАСС СЕМАНТИЧЕСКИХ СЕТЕЙ
13	ТЕКСТ	ИНФОРМАЦИОННАЯ СИСТЕМА	РЕШЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

№	Слова	Биграммы	Триграммы
14	ПОНЯТИЕ	СИТУАЦИОННОЕ УПРАВЛЕНИЕ	ИНФОРМАЦИОННЫЕ ПОТРЕБНОСТИ ПОЛЬЗОВАТЕЛЕЙ
15	МНОЖЕСТВО	ЖИЗНЕННЫЙ ЦИКЛ	УПРАВЛЕНИЕ ПОЛЕТАМИ МКС
16	ПРЕДСТАВЛЕНИЕ	СЕМАНТИЧЕСКИЙ АНАЛИЗ	СИСТЕМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА
17	ОТНОШЕНИЕ	УПРАВЛЕНИЕ ПРОЕКТАМИ	ГРАФ ГОРИЗОНТАЛЬНОЙ ВИДИМОСТИ
18	ОСНОВА	ИНФОРМАЦИОННЫЙ РЕСУРС	ЭЛЕКТРОННОЕ УЧЕБНОЕ ИЗДАНИЕ
19	ИНФОРМАЦИОННОЕ	РУССКИЙ ЯЗЫК	СЕМИОТИЧЕСКАЯ КОГНИТИВНАЯ КАРТА
20	УПРАВЛЕНИЕ	ОБЛАСТЬ ЗНАНИЙ	ОТКРЫТЫЕ СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ
21	ПОЛЬЗОВАТЕЛЬ	ОБУЧАЮЩАЯ ВЫБОРКА	МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА
22	ОПИСАНИЕ	КОГНИТИВНАЯ КАРТА	ДАННЫЕ ПРЕДМЕТНОЙ ОБЛАСТИ
23	БАЗА	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС	ПОРТАЛ НАУЧНЫХ ЗНАНИЙ

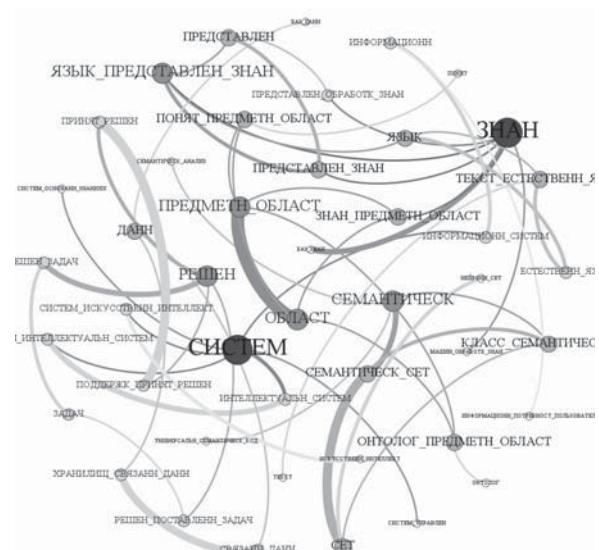


Рис. 2. Вид СЕИТ
размером 20+20+20

На рис. 2 представлена небольшая сеть естественной иерархии терминов размером 20+20+20 (20 слов, 20 биграмм, 20 триграмм), которая визуализирована средствами системы Gephi (<https://gephi.org/>).

2. Исследование СЕИТ

Для построенных сетей естественных иерархий терминов различных размеров по выбранному тексту было определено распределение исходящих степеней узлов, которое оказалось близким к степенному ($p(k) = Ck^a$), т.е. эти сети являются безмасштабными. Оказалось, что коэффициент a для сетей различных размеров (от 20+20+20 до 200+200+200) составляет от 2,1 до 2,3, что вполне соответствует сетям языка (Language Networks) [Большакова и др., 2011].

Очевидно, что в соответствии с предложенным алгоритмом максимальное количество входных связей для узлов данной сети составляет 5 (для узлов из одного слова – 0 входящих связей, для узлов из 2 слов – максимально 2 связи, для узлов из 3 слов – максимально 5 связей – три связи от отдельных слов и две от пар слов).

Наиболее интересными с семантической точки зрения в рассматриваемой СЕИТ оказались узлы с максимальным количеством входных связей, среди которых можно выделить такие словосочетания: «язык представления знаний»; «система искусственного интеллекта»; «поддержка принятия решений»; «онтология предметной области»; «универсальный семантический код».

Из материалов конференции было выбрано 5 докладов из сборника трудов конференции, в которые входили данные словосочетания, а именно доклады:

- Ефименко И.В., Хорошевский В.Ф. УСК Мартынова – тридцать лет спустя;
- Тарасов В.Б. От семантического кода к когнитивной лингвистике и информатике: наследие В.В. Мартынова;
- Массель Л.В., Массель А.Г. Ситуационное управление и семантическое моделирование в энергетике;
- Стефанюк В.Л. Условно-рефлекторная основа запоминания знаний;
- Кулнич А.А. Когнитивное моделирование в условиях неопределенности (семиотический подход).

По отдельным докладам также были рассчитаны значения CHVG для слов, биграмм и триграмм, построены сети естественных иерархий терминов. Взаимосвязь терминов из выбранных докладов приведена на рис. 3, на котором можно видеть, что каждому докладу (узлы, идентифицированные фамилиями авторов) соответствуют термины. При этом в центральной части сети располагаются термины, общие для нескольких докладов (О-зона), а «гребешки» на периферии соответствуют специальным терминам, отражающим специфику конкретных докладов (С-зоны).

О-зона вовсе не обязательно включает термины из абсолютно всех докладов, достаточно, чтобы термины соответствовали лишь их определенной части (порогу), например половине. Чем больше в докладе терминов, попадающих в центральную часть, тем он лучше вписывается в тематику конференции, тем он точнее попадает в ее тренд. В данном случае (рис. 3) именно доклады Хорошевского и Тарасова в большей мере соответствуют тематическому направлению конференции. В соответствии с этим наблюдением можно предложить такой лингвистический критерий «релевантности» доклада тематике конференции: чем большая часть топ-лексики из него попадает в О-зону, тем он более релевантен.

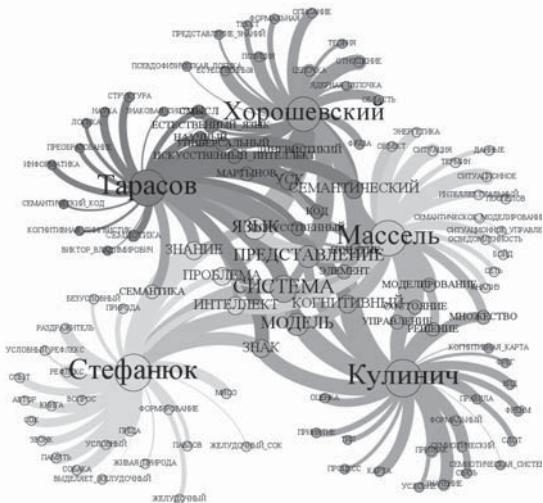


Рис. 3. Сеть связи терминов выбранных докладов

Заключение

Представления об информационной значимости наборов терминов для построения СЕИТ, степени их важности для отражения смысла научного текста были подтверждены в ходе экспериментов с информантами. Так, для всех текстов были проведены эксперименты со стандартной инструкцией «Прочитайте текст. Подумайте над его содержанием. Выпишите 10–15 слов, наиболее важных для его содержания» (более 20 информантов для текста каждого из исследуемых докладов, а также всего сборника) [Ягунова, 2010]. Полнота отражения информационно-значимых слов, биграмм и триграмм для сборника в этом случае превысила 50%.

Таким образом, в результате проведенных исследований:

- Описан алгоритм построения сетей естественных иерархий терминов на основе анализа текстов.
- На основании этого алгоритма по текстам докладов научно-технической конференции построена сеть естественной иерархии терминов.
- Сеть естественных иерархий терминов оказалась скайл-фри по исходящим связям.
- Выбраны программные средства визуализации сети естественных иерархий терминов.
- Предложен критерий релевантности доклада по тематике конференции.

Сеть языка, построенную с помощью предложенной методики, можно использовать в качестве базы для построения общей онтологии (в рассмотренном примере – по тематике семантических технологий), использовать на практике в качестве готового к применению средства навигации в базах данных, а также для организации контекстных подсказок пользователям информационно-поисковых систем.

СПИСОК ЛИТЕРАТУРЫ

[Лукашевич и др., 2007] Лукашевич Н.В., Добров Б.В., Чуйко Д.С. Отбор словосочетаний для словаря системы автоматической обработки текстов // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции «Диалог-2008». М., 2008. С. 339–344.

[Yagunova и др., 2012] Yagunova E., D. Lande D. Dynamic Frequency Features as the Basis for the Structural Description of Diverse Linguistic Objects

// CEUR Workshop Proceedings. Proceedings of the 14th All-Russian Scientific Conference «Digital libraries: Advanced Methods and Technologies, Digital Collections» Pereslavl-Zalesky, Russia, October 15-18, 2012. – P. 150–159.

[**Lande и др., 2013-1**] Lande D.V., Snarskii A.A. Compactified HVG for the Language Network // International Conference on Intelligent Information Systems: The Conference is dedicated to the 50th anniversary of the Institute of Mathematics and Computer Science, 20–23 aug. 2013, Chisinau, Moldova: Proceedings IIS / Institute of Mathematics and Computer Science, 2013. – P. 108–113.

[**Lande и др., 2013-2**] Lande D., Snarskii A., Yagunova E. The Use Of Horizontal Visibility Graphs To Identify The Words That Define The Information Structure Of The Text // CEUR Workshop Proceedings. Vol-1108 urn:nbn:de:0074-1108-1. ISSN 1613-0073. Selected Papers of the 15th All-Russian Scientific Conference «Digital libraries: Advanced Methods and Technologies, Digital Collections» Yaroslavl, Russia, October 14–17, 2013. – P. 158–164.

[**Lande и др., 2013-3**] Lande D.V., Snarskii A.A., Yagunova E.V., Pronoza E.V. The Use of Horizontal Visibility Graphs to Identify the Words that Define the Informational Structure of a Text // 12th Mexican International Conference on Artificial Intelligence, 2013. – P. 209–215.

[**OSTIS, 2014**] Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2014): материалы IV междунар. науч.-техн. конф. (Минск 20–22 февраля 2014 года) / – Минск: БГУИР, 2014. – 576 с.

[**Lande, 2014**] Lande D.V. Building of Networks of Natural Hierarchies of Terms Based on Analysis of Texts Corpora // E-preprint viXra 1404.0069.

[**Salton, 1983**] Salton G., McGill M.J. Introduction to Modern Information Retrieval. – New York : McGraw-Hill, 1983. – 448 p.

[**Luque и др., 2009**] Luque B., Lacasa L., Ballesteros F., Luque J. Horizontal visibility graphs: Exact results for random time series // Phys. Review E, 2009. – P. 046103-1 – 046103-11.

[**Большакова и др., 2011**] Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. – М.: МИЭМ, 2011. – 272 с.

[**Ягунова, 2010**] Ягунова Е.В. Эксперимент и вычисления в анализе ключевых слов художественного текста // Сборник научных трудов кафедры иностранных языков и философии ПНЦ УрО РАН. Философия языка. Лингвистика. Лингводидактика. – Пермь, 2010. – Вып. 1. – С. 85–91.

УДК 004.822

ТЕЗАУРУС РУТЕЗ: ПУБЛИКАЦИЯ ПЕРВОЙ ВЕРСИИ

Н.В. Лукашевич (*louk_nat@mail.ru*)
НИВЦ МГУ им. М.В. Ломоносова
Б.В. Добров (*dobrov_bv@mail.ru*)
НИВЦ МГУ им. М.В. Ломоносова

В работе представлена публично доступная версия лингвистической онтологии – тезауруса русского языка РуТез, созданного для автоматической обработки текстов. Доступная версия РуТез-lite была порождена из полного тезауруса на основе наиболее частотных слов в современном новостном корпусе. Данная статья описывает структуру тезауруса РуТез-lite, а также возможности его просмотра и получения для некоммерческого использования.

Введение

Одним из популярных лексических ресурсов, используемых при автоматической обработке текстов, является тезаурус WordNet [Fellbaum, 1998]. Тезаурус WordNet представляет собой совокупность лексических понятий – синсетов (наборов синонимов), между которыми установлены такие отношения, как гипоним-гипероним (родовидовые отношения), мероним-холоним (отношения часть – целое) и некоторые другие. Сеть понятий подразделена на четыре подсети, соответствующие четырем основным частям речи (существительное, прилагательное, глагол, наречие). Описание синсетов в каждой подсети содержит несколько отличающихся типов отношений и принципы их описания [Fellbaum, 1998]. Последняя версия тезауруса WordNet включает описание более 150 тысяч слов и выражений английского языка.

Подобные ресурсы стали создаваться и для многих других языков. В частности, известно, по крайней мере, четыре попытки создать wordnet для русского языка [Азарова, 2010; Гельфенбейн и др., 2003; Balkova et al., 2008; Braslavskiy и др., 2013].

Несмотря на свою популярность в компьютерных приложениях, WordNet изначально создавался для обоснования психолингви-

стической теории [Miller, 1998], выбор принципов его структуры, описания отношений между элементами базировалась на психолингвистических экспериментах. Такая психолингвистическая ориентация привела к тому, что WordNet исходно имел некоторые конструктивные особенности, которые могут усложнять его применение в приложениях автоматической обработки текстов, такие как отсутствие связей между сетями, построенными для разных частей речи, отсутствие связей между синсетами, относящимися к одной и той же тематической области (так называемая «теннисная проблема») и др. [Fellbaum, 1998], которые постепенно корректируются.

Понятно, что исследования по улучшению структурных свойств сверхбольших лингвистических ресурсов для автоматической обработки текстов очень сложны и трудоемки, поскольку необходимо не только предложить новый подход, но и в соответствии с предложенными принципами создать такой ресурс, а также продемонстрировать его работу в компьютерных приложениях.

В течение более 15 лет мы создавали тезаурус русского языка РуТез, который делался специально как инструмент для автоматической обработки текстов [Лукашевич, Добров, 2002]. Был предложен и протестирован ряд технологий с использованием этого тезауруса, включая автоматическую рубрикацию и кластеризацию текстов, автоматическое разрешение многозначности, автоматическое аннотирование одного и многих документов и др. [Лукашевич, 2011]; тезаурус использовался в проектах с государственными и коммерческими организациями. В настоящее время тезаурус представляет собой сеть понятий, к которым приписаны более 155 тысяч слов и выражений русского языка.

В 2014 году была подготовлена общественно доступная версия тезауруса РуТез – РуТез-lite, которая опубликована на сайте <http://www.labinform.ru/ruthes/index.htm>. Данная версия тезауруса включает около 96 тысяч уникальных слов и выражений русского языка. Версия РуТез-lite предназначена для некоммерческого применения. Доступны также файлы тезауруса РуТез-lite в формате xml.

В данной статье мы обсудим структуру и состав тезауруса РуТез-lite, особенности его формирования, распространения и применения.

1. Структура тезауруса РуТез

Тезаурус РуТез представляет собой лингвистическую онтологию, т.е. онтологию, понятия которой опираются на значения существующих в языке слов и выражений. Каждое понятие онтологии имеет уникальное и однозначное имя. Каждое понятие онтологии связано с набором слов и выражений, посредством которых данное понятие может выражаться в тексте – т.н. текстовых входов понятия.

Набор текстовых входов понятия может включать собственно синонимы, слова разных частей речи (так называемые дериваты), устойчивые словосочетания и другие типы выражений. Например, понятие ПРИВАТИЗАЦИЯ имеет такие текстовые входы, как *приватизация, приватизировать, приватизационный, передать в частные руки*. Особенностью ведения тезауруса является включение в состав его текстовых входов большого количества словосочетаний, представляющих собой более однозначные способы упоминания понятия, чем отдельные слова.

Значения многозначных слов представлены посредством отнесения слова к двум и более понятиям. Например, многозначное слово *бolid* отнесено к понятиям ГОНОЧНЫЙ БОЛИД и КОСМИЧЕСКИЙ БОЛИД.

Выбор набора отношений для описаний отношений между словами в лингвистическом ресурсе широкой тематики представляет собой сложную проблему. Для стабильной работы на разных типах текстов в тезаурусе РуТез используется небольшой набор надежных отношений, т.е. отношений, которые не исчезают, не меняются в течение всего срока существования любого или подавляющего большинства экземпляров понятия. В результате исследований и экспериментов мы пришли к набору отношений ресурса, предназначенного для эффективной автоматической работы в информационно-поисковых приложениях и приложениях автоматической обработки текстов.

В тезаурусе РуТез имеется четыре основных типа отношений.

Первый тип отношений – **родовидовое отношение ниже – выше**, представляет собой отношение класс – подкласс, обладает свойствами транзитивности и наследования.

Второй тип отношений – **отношение часть – целое**. Используется не только для описания физических частей, но и для дру-

гих внутренних сущностей понятия, таких как свойства или роли для ситуаций. Важным условием при установлении этого отношения является то, что понятия-части должны быть жестко связаны со своим целым, то есть каждый пример понятия-части должен в течение всего времени своего существования являться частью для понятия-целого, и не относиться к чему-либо другому. В этих условиях удается выполнить свойство транзитивности введенного таким образом отношения *часть-целое*, что очень важно для автоматического вывода в процессе автоматической обработки текстов. Описание отношений часть-целое в тезаурусе наиболее близко к так называемым внутренним отношениям (internal relations), обсуждаемым Н. Гуарино в работе [Guarino, 2009].

Еще один тип отношения, называемого **несимметричной ассоциацией** $asc_2 - asc_1$, связывает два понятия, которые не могут быть связаны выше рассмотренными отношениями, но когда одно из которых не существовало бы без существования другого. Например, понятие **КИПЕНИЕ** (в физическом смысле) требует существования понятия **ЖИДКОСТЬ**. В онтологических исследованиях такое отношение называется отношением онтологической зависимости [Guarino, 2009; Guzzardi, 2011].

Последний тип отношений – **симметричная ассоциация** asc связывает, например, понятия, очень близкие по смыслу, но которые разработчики не решились соединить в одно понятие (предсинонимия).

2. Формирование текстовой коллекции для порождения РуТез-lite

Для формирования словарного состава доступной версии Рутез-lite было принято следующее решение. Необходимо было отразить наиболее частотную лексику современного русского языка, ограничив выдачу терминологии, относящейся к конкретным предметным областям и проектам. Базой для выявления частотных характеристик слов был выбран поток современных новостных сообщений, поскольку новости – это наиболее востребованный для обработки жанр текстов.

Для этого была сформирована коллекция новостных сообщений на русском языке размером 2 млн. документов. Коллекция не-

обходима для проверки частот употребления слов в современной русской деловой прозе. Собранный коллекция была обработана графематическим и морфологическим анализаторами, произведено сопоставление с текущей версией тезауруса РуТез. Такое сопоставление позволило определить частотность текстовых входов тезауруса РуТез. Был построен список текстовых входов тезауруса, упорядоченный по мере снижения частотности их встречаемости в новостной коллекции.

Для верхней части полученного списка (около 30 тысяч самых частотных текстовых входов) была произведена вычитка, которая состояла в том, что из списка были удалены финансовые, экономические и спортивные термины, которые в большом количестве содержатся в современном новостном потоке. А также были удалены случайно совпавшие со словами русского языка опечатки, фамилии, фрагменты слов, возникшие из-за переносов, таблиц и т.п.

Начало списка (наиболее частотные слова) выглядит вполне традиционно: *быть, год, сообщать, мочь, сообщить, время* и т.п. Примеры слов из конца списка для выгрузки включают: *мандариновый, маятиться, краснуха, контрмера* и др.

Для подготовки версии РуТез-lite была разработана программа, которая по заданному списку текстовых входов тезауруса порождает связный подтезаурус. Для этого производятся следующие операции:

- текстовые входы из списка сопоставляются с понятиями тезауруса – создается исходное множество понятий;
- к исходному множеству понятий присоединяются все понятия, расположенные выше них по иерархии отношений (используются отношения *выше, целое, acq₁*);
- наконец, все понятия, как сопоставленные, так и вышестоящие по иерархии вместе с их отношениями выгружаются в новый тезаурус.

3. Опубликованная версия РуТез-Lite

В результате подготовлена версия тезауруса РуТез-lite, которая включает 26365 понятий, к которым приписано около 115 тысяч слов и выражений (с учетом их многозначности), между понятиями установлено около 108 тысяч отношений. Версия тезауруса выпущена на сайт <http://www.labinform.ru/ruthes/index.htm>.

Произведенная выкладка позволяет просматривать тезаурус по алфавиту (рис. 1).

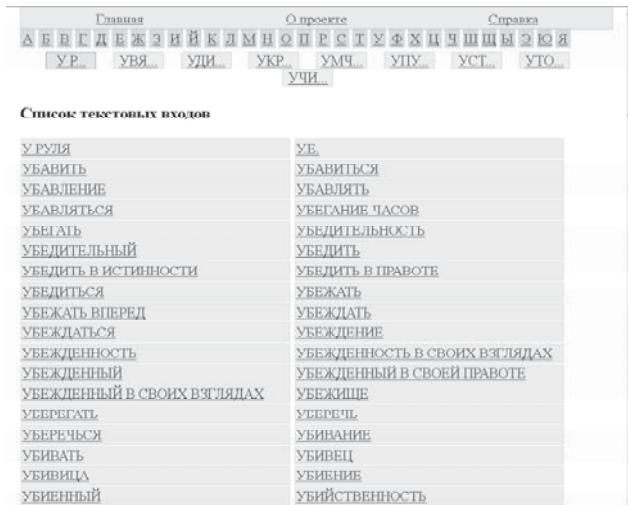


Рис. 1. Алфавитный перечень текстовых входов тезауруса PyТез

Просмотр конкретного текстового входа, например, *убирать* (рис. 2) позволяет увидеть совокупность понятий, которым приписано данное слово, а именно к понятиям *УБОРОЧНЫЕ РАБОТЫ; УБРАТЬ В ДРУГОЕ МЕСТО; УБРАТЬ, СДЕЛАТЬ УБОРКУ*. Для каждого понятия указаны полные списки текстовых входов, включающие слова разных частей речи, а также словосочетания. Кроме того, указаны отношения с другими понятиями.

Указанные в статьях понятий текстовые входы и другие понятия оформлены как ссылки, поэтому на них можно переходить. Так, на рис. 3 изображена статья понятия *СЕЛЬСКОХОЗЯЙСТВЕННЫЕ РАБОТЫ*, на которую можно перейти со статьи понятия *УБОРОЧНЫЕ РАБОТЫ*.

Точно так же можно перейти на другой текстовый вход, например *прибирать*, и увидеть, что он также многозначный и соответствует трем понятиям: УБРАТЬ В ДРУГОЕ МЕСТО; УБРАТЬ, СДЕЛАТЬ УБОРКУ; ПРИСВОИТЬ, САМОВОЛЬНО ВЗЯТЬ (рис. 4).

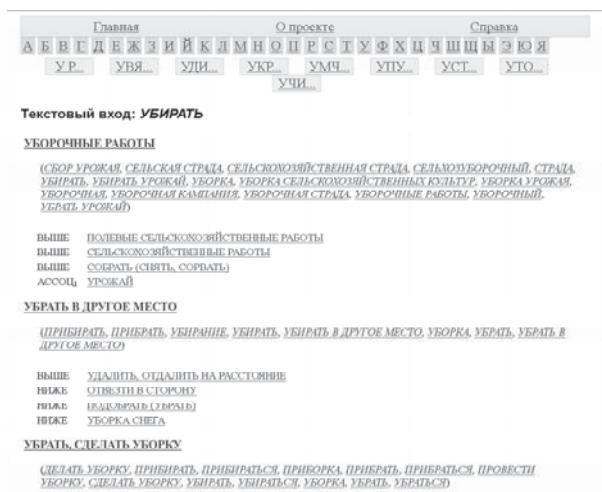


Рис.2. Описание понятий, соответствующих слову *убирать*

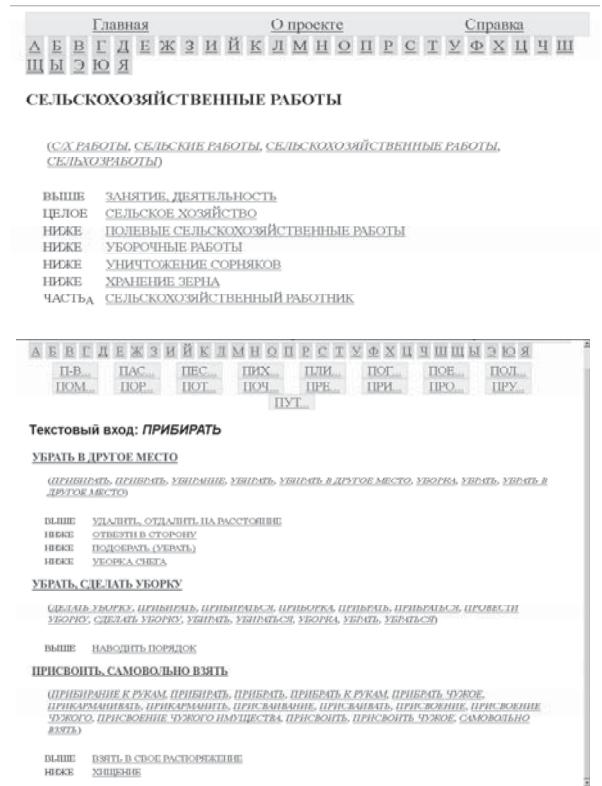


Рис. 3. Статья понятия
СЕЛЬСКОХОЗЯЙСТВЕННЫЕ РАБОТЫ

4. Версия PyTез-Lite в формате XML

Тезаурус PyTез-Lite может быть получен в виде архива файлов в формате XML. Набор файлов включает в себя:

- список понятий тезауруса, для которых указана предметная область (общий лексикон, общественно-политическая область, география); также для большей части понятий имеются толкования, автоматически извлеченные из Викисловаря [Лукашевич и др. 2014];
- список отношений между понятиями тезауруса;
- список текстовых входов тезауруса. Описание текстового входа содержит лемматическое представление текстового входа, синтаксический тип (именная группа, глагольная группа и др.), главное слово именной группы [Лукашевич и др., 2014];
- список соответствий текстовых входов понятиями тезауруса.

Заключение

В данной работе мы представили опубликованную версию тезауруса PyTез – PyTез-Lite. Тезаурус включает более 115 тысяч лексических единиц русского языка. PyTез-Lite доступен для просмотра на сайте и может быть получен в виде файлов в формате XML. Мы надеемся, что публикация такого тезауруса будет способствовать развитию различных методов автоматической обработки текстов на русском языке.

Благодарности. Работа поддержана фондом «Династия» Дмитрия Зимина при финансовой поддержке основателей компании Яндекс.

СПИСОК ЛИТЕРАТУРЫ

[Balkova et al., 2008] Balkova V., Suhonogov A., Yablonsky S. Some Issues in the Construction of a Russian WordNet Grid // Proceedings of the Forth International WordNet Conference, Szeged, Hungary, 2008. pp. 44–55.

[Fellbaum, 1998] Fellbaum Ch. WordNet: An Electronic Lexical Database, Cambridge, MA, MIT Press, 1998.

[Guarino, 2009] Guarino N. The ontological Level: Revisiting 30 years of Knowledge Representation. Conceptual Modeling: Foundations and Applications. Springer-Verlag Berlin, Heidelberg, 2009. pp. 52-67.

[Guizzardi, 2011] Guizzardi G. Ontological foundations for conceptual part-wholes relation: the case of collectives and their parts. Advanced Information Systems Engineering, Springer CAiSE, LNCS 6741, 2011. pp. 138–153.

- [Loukachevitch, Dobrov, 2014] Loukachevitch N., Dobrov B. *RuThes Linguistic Ontology vs. Russian Wordnets. Proceedings of Global WordNet Conference GWC-2014, Tartu, 2014.*
- [Miller, 1998] Miller G. A. Nouns in wordnet //WordNet: An electronic lexical database. 1998. pp. 24–45.
- [Азарова, 2010] Азарова И. В. RussNet как компьютерный тезаурус нового типа //Структурная и прикладная лингвистика. 2010. №. 7.
- [Браславский и др., 2013] Braslavskiy P., Muixin M., Lyahevskaya O. N., Bonch-Osmolovskaya A. A., Krzhizhanovskiy A., Egorov P. YARN: начало. Труды Международной конференции по компьютерной лингвистике и интеллектуальным технологиям Диалог-2013, 2013.
- [Гельфейнбайн и др., 2003] Гельфейнбайн И. Г., Гончарук А. В., Лехельт В. П., Липатов А. А., Шило В. В. Автоматический перевод семантической сети WORDNET на русский язык. Труды конференции «Диалог-2003», 2003.
- [Лукашевич, Добров, 2002] Лукашевич Н. В., Добров Б. В. Тезаурус русского языка для автоматической обработки больших текстовых коллекций //Компьютерная лингвистика и интеллектуальные технологии: Труды Международного семинара Диалог. 2002. С. 338–346.
- [Лукашевич, 2011] Лукашевич Н.В. Тезаурусы в задачах информационного поиска. – М.: Изд-во Московского университета, 2011.
- [Лукашевич и др., 2014] Лукашевич Н.В., Добров Б.В., Четверкин И.И. РуТез-Lite, опубликованная версия тезауруса русского языка РуТез // Международная конференция по компьютерной лингвистике Диалог-2014, 2014.

УДК 004.8

**КОНЦЕПТУАЛЬНАЯ СХЕМА СИСТЕМЫ
АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВЫХ
ДОКУМЕНТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ
С ИСПОЛЬЗОВАНИЕМ ДСМ-МЕТОДА**

Н.Д. Лыфенко (*LyfenkoNick@yandex.ru*)

Российский государственный гуманитарный университет, Москва

В статье рассматривается задача автоматической классификации текстовых документов на естественном языке. Данная задача решается с помощью системы классификации текстов, использующей разнообразные методы интеллектуального анализа данных, в том числе с помощью ДСМ метода автоматического порождения гипотез.

Введение

Задача автоматической классификации текстовых документов на естественном языке – это задача определения принадлежности нового документа заранее заданному классу, которая решается с помощью системы классификации текстов ADC (Automatic Document Classifier), использующей разнообразные методы интеллектуального анализа данных, в том числе с помощью ДСМ метода автоматического порождения гипотез [Финн, 1976-83]. Для системы представлена структурная схема, описаны основные концепции, рассмотрены особенности реализации ДСМ-метода для задач интеллектуальной обработки текста.

Над автоматической классификацией текстов стали работать уже в конце 1950-х и начале 1960-х гг. Одними из первых исследований, посвященных рассматриваемой теме, были работы Ирвинга Джона Гуда и Роберта Артура Фэрторна (Fairthorne). Ученые предложили использовать классификацию документов в области информационного поиска [Good, 1958], [Fairthorne, 1961]. Одной из первых реализаций была программа Sussenguth'a [Sussenguth, 1964], где уже были явным образом перечислены необходимые этапы предварительной обработки текста. Вплоть до конца 1980-х годов наиболее эффективным подходом к классификации был «ручной», с помощью методов из области инженерии знаний (knowledge engineering)

[Сэлтон, 1973], но в 1990-х годах в связи с развитием технологий машинного обучения данный подход потерял популярность.

За последние годы был проявлен высокий интерес к рассматриваемой задаче. Ведутся коммерческие разработки, например, Irosoft¹, Artyl's², TextAnalyst³, проводятся исследования [Bloom, 2012], [Fukumoto, Suzuki, 2011], [Sebastiani, 2002], разрабатываются новые теории представления документа и взвешивания термов⁴, создаются интегрированные среды разработки, такие как RapidMiner, Gate и др.

Перечислим некоторые особенности предлагаемой в докладе системы:

- предлагается не просто программа для решения прикладных задач по классификации текста, а платформа для проведения научных экспериментов по сравнению эффективности различных методов классификации текстовых документов, снабженная удобным пользовательским интерфейсом и развитыми средствами формализации результатов экспериментов для последующей их оценки;
- среди реализуемых с помощью предлагаемой платформы методов классификации наряду с широко применяемыми методами (наподобие метода опорных векторов и k-ближайших соседей) присутствуют и некоторые варианты ДСМ-метода автоматического порождения гипотез, который для анализа текстов ранее почти не использовался, хотя данные отдельных экспериментов свидетельствуют о хороших возможностях этого метода для анализа текстовых данных [Котельников, 2012];
- настройка параметров системы (выбор модели представления термов и модели их взвешивания, выбор алгоритма классификации и др.) позволяет использовать оптимальные значения для решения конкретной задачи классификации.

Поскольку основной особенностью данной системы является попытка использования ДСМ-метода для классификации, то, к сожалению, нет возможности пользоваться хорошо известными и рекомендовавшими себя системами по автоматической обработке

¹ <http://www.irosoft.com/en/communiques-presse/irosoft-adds-automatic-document-classification-module-docutheque-entreprise>

² http://www.artsyltech.com/da_classification.html

³ <http://www.megaputer.com/site/textanalyst.php>

⁴ Термами в данной работе мы называем объекты анализа – словосочетания длиной не более *n*.

текста, такими как RapidMiner, Gate и пр. Также это обусловлено отсутствием качественного аппарата работы с текстами на русском языке (например, для решения задач нормализации слов) и нереализованной идеей фильтрации текстовых документов по словарям предметной области.

Концептуальная схема системы ADC

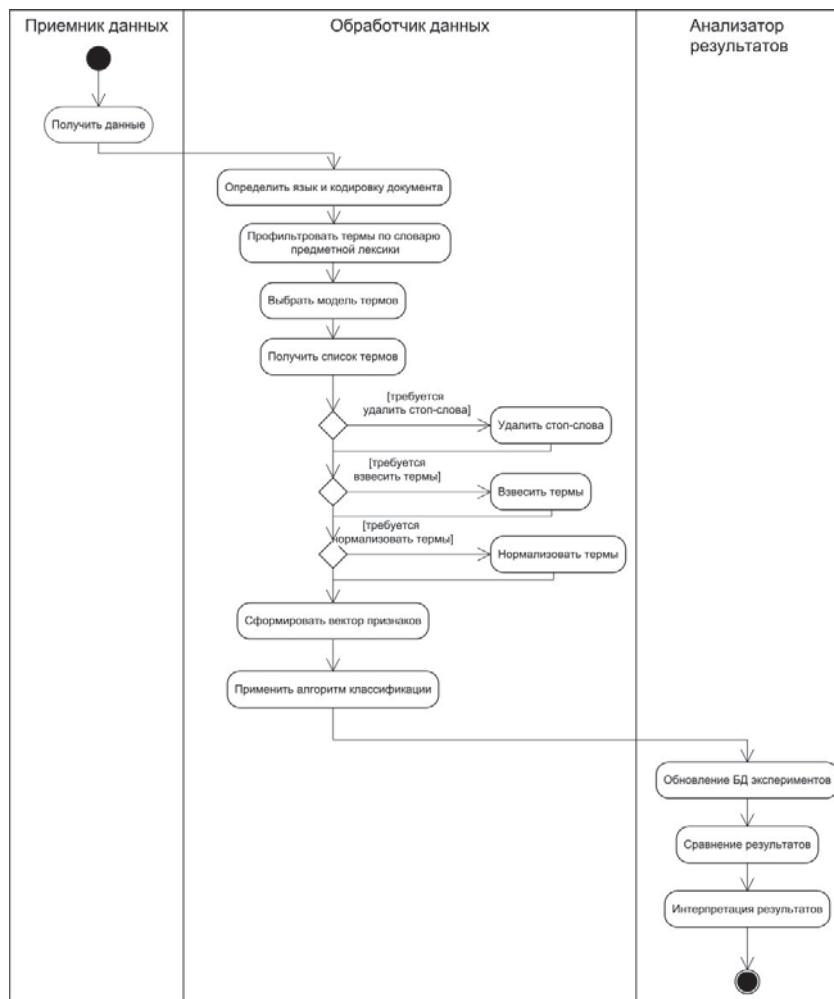


Рис.1. Основной алгоритм работы ADC

Система ADC состоит из трех модулей: приемник данных, обработчик данных и анализатор результатов. Задача приемника данных заключается в создании корпуса документов, при необходимости в преобразовании входного документа (например, HTML файл) в текстовый документ. В модуле обработки данных предварительно обрабатываются текстовые документы (выделяются, нормализуются и взвешиваются термы) и применяется алгоритм классификации. С помощью анализатора результатов происходит интерпретация экспериментов и вычисление различных метрик эффективности (точность, аккуратность, полнота).

На рис. 1 представлен основной алгоритм работы системы ADC в виде диаграммы деятельности UML.

Получение данных

Первый этап состоит в конвертации документов в текстовый файл, содержащий только значимую текстовую информацию. В рамках данного исследования предполагается использовать в качестве данных массивы документов из сети Интернет и представленные html-страницами. Очевидно, что теги в html-документе не несут семантической нагрузки и их можно не учитывать. Но обычно удаление тегов не приводит к безошибочному получению чистого текста (plain text) из-за наличия большого количества рекламы, новостного мусора, комментариев, которые вносят большой шум в обучающую выборку. Поэтому был предложен и реализован механизм более точного извлечения чистого текста с учетом тегов, который повысил качество получения текстовых данных из html-документов. Идея механизма состоит в том, что на основе html-страницы строится гистограмма, учитывающая длину строки текста (количество слов/букв) и тег, в который заключен текст. Анализируются только те части документа, которые получат максимальное значение в результате такой композиции.

Обработка данных и классификация

На втором этапе проводится предварительная обработка документа и классификация. Используя статистический анализ, определяется кодировка документа из заданных списком значений кодовых страниц (UTF-8, UTF-16, CP1251).

В данной работе предполагается, что не все выделенные термы несут семантическую нагрузку. Если построить распределение

термов по частоте их употребления, то большее значение получат стоп-слова, меньшее – узко специализированные термины, а оставшаяся часть характеризует документ в целом. Поэтому предлагаются использовать автоматически сгенерированные словари предметной лексики, для проведения фильтрации извлеченных термов из текста по ним.

Для представления текста в виде n -мерного вектора используются концепция мешка слов (bag of words), различные модели взвешивания термов: частотный, TF-IDF, LOWBOW, MFS [Lebanon et al., 2007], [Cabera et al., 2013].

Проводится нормализация термов по сформированному обратному словарю на основе словаря А.А. Зализняка и предложена методика для нормализации неизвестных слов на основе поиска максимально длинной подстроки в инвертированном слове.

Предполагается изменять вес у некоторых термов: так, у именованных сущностей имеет смысл увеличивать вес и уменьшать его у незначимых термов.

После полученного вектора признаков нужно применить алгоритм классификации. Для решения данной задачи широко используются следующие методы: метод k -ближайших соседей, нейронные сети, байесовский классификатор, машина опорных векторов (SVM) и др. В данной работе подробнее рассмотрим ДСМ-метод, т.к. данный метод показывает неплохую эффективность в задаче определения тональности текста [Котельников, 2012]. Метод существенно превосходит базовый классификатор, он на 4% лучше SVM без подбора оптимального числа характеристик и менее чем на 1% отличается от SVM с подбором характеристик. Поэтому есть все основания использовать данный метод и предположить, что и для другого круга задач автоматической классификации текстов (задачи определения темы текста и авторства) данный метод будет эффективен.

В основе ДСМ-метода лежат формализованные правила правдоподобных рассуждений. ДСМ-метод реализует синтез трех познавательных процедур – индукции, аналогии и абдукции. Свое сокращенное название метод получил в честь английского философа и логика Д.С. Милля. В ДСМ-методе используются две классические для машинного обучения фазы: выявление закономерностей в данных (обучение) и предсказание значения целевых свойств.

Анализ результатов

В силу специфики применения различных методов интеллектуального анализа данных следует выяснить, какие из имеющихся алгоритмов и моделей представления текста дают наиболее релевантную классификацию документов для конкретной предметной задачи, т.е. требуется инструментарий для хранения результатов экспериментов и их сравнения.

Для формирования базы данных экспериментов с различными опциями обработки текста и классификации необходимо иметь инструментарий, позволяющий быстро и удобно производить настройку экспериментов. Для решения данной задачи была разработана объектная модель проекта системы. Проект состоит из имени (путь к файлу проекта), даты проведения, настройки проекта и серии экспериментов. Последнее поле включает имя эксперимента (уникальный идентификатор), дату проведения, список файлов для обучения и тестирования классификатора, результаты и настройки. Поля «результаты» и «настройка эксперимента» – это указатели на соответствующие поля в базе данных по экспериментам.

Особенности реализации ДСМ-метода для задач автоматической обработки текста

Установим соответствие между терминами автоматической обработки текста и терминами, используемыми в ДСМ-методе.

Объектами мы будем называть тексты. Термы или результаты применения хэш-функции к ним назовем атомами. Целевое свойство объекта – это класс, который нужно предсказать для каждого примера. Задачу n -арной классификации будем рассматривать как последовательность решения задач бинарной классификации, т.е. многократный запуск ДСМ-процедур.

Обычно для представления объектов в задачах с использованием ДСМ-метода используют множества, которые на более низком уровне представляются в виде битовых строк. Такое представление считается самой эффективной структурой, на которой можно определить операцию сходства. На вычисление сходства (пересечение множеств) тратится большая часть времени на этапе формирования гипотез о наличии или отсутствии свойств у объектов (индукция). Но веса термов представлены вещественным числом в

промежутке $[0,1]$, поэтому необходимо преобразовать число в битовое представление. Можно разделить интервал $[0,1]$ на 8 или 4 отрезка таким образом, что каждый последующий интервал включает предыдущий. Например, таким образом:

$$\begin{cases} w_i = 0, \text{нулевое значение, } 0000 \\ w_i \in (0;0,4], \text{низкое значение, } 0001 \\ w_i \in (0;0,6], \text{низкое значение, } 0011 \\ w_i \in (0;0,8], \text{значение не выше повышенного, } 0111 \\ w_i \in (0;1], \text{все возможные значения, } 1111 \end{cases}$$

где w_i – вес i -ого терма, словесное описание значения веса терма, битовое представление терма.

После таких преобразований, конечно, увеличится размерность вектора признаков, но в силу применения идеи фильтрации мы будем работать только с подмножеством множества термов документа. Для этой цели мы будем использовать небольшое количество словарей сравнительно узких предметных областей.

Решая задачу тематической автоматической классификации документов, будем выбирать для рассмотрения некоторую предметную область, формировать узкоспециализированный словарь этой предметной области и отдельно создавать список меток рубрик. Мы предполагаем, что связь между рубриками и совокупностями термов заранее неизвестна. Она будет обнаружена на этапе индукции. Рубрики будут соответствовать подобластям рассматриваемой предметной области.

Мы будем обрабатывать не множество всех значащих термов документа, а множество только тех термов документа, которые входят в специализированный словарь предметной области. Именно это множество будет рассматриваться как объект ДСМ-системы. Количество терминов словаря предметной области предполагается небольшим (несколько сот терминов). Это позволит существенно уменьшить размерность битового вектора, представляющего документ (объект ДСМ-системы), и позволит нам использовать известные алгоритмы, предлагаемые для работы с теоретико-множественным представлением объектов ДСМ-метода.

Мы будем допускать такую ситуацию, когда выбранный заранее набор рубрик не обязательно целиком включается в одну предметную область, для которой имеется узкоспециализированный словарь. Поэтому процедура индукции будет включать перебор всех имеющихся в нашем распоряжении узкоспециализированных словарей, фильтрацию по каждому словарю и порождению гипотез по выделению набора термов для автоматической рубрикации (по отдельности для каждого словаря).

Не исключена и такая ситуация, когда одна рубрика будет относиться к нескольким предметным областям. Тогда разным предметным областям, вообще говоря, могут соответствовать разные наборы термов, относящиеся к одной рубрике.

Заключение

В данной работе рассмотрена задача автоматической классификации текстовых документов на естественном языке с использованием ДСМ-метода. Предложена концептуальная схема системы, рассмотрены некоторые особенности реализации ДСМ-метода для задач автоматической обработки текста. На текущий момент задача получения вектора признаков и его формирования решена для частотной и TF-IDF моделей представления термов. Реализованы удобный пользовательский интерфейс для проведения экспериментов и их анализа, объектная модель представления проекта в системе ADC с помощью xml-файла проекта.

СПИСОК ЛИТЕРАТУРЫ

[Котельников, 2012] Котельников Е.В. Опыт применения ДСМ-метода для определения тональности текста // Труды Тринадцатой Национальной конференции по искусственному интеллекту с международным участием КИИ-2012 (16–20 октября 2012 г., Белгород). – Т. 1. – Белгород: Изд-во БГТУ, 2012. – С. 135–142.

[Сэлтон, 1973] Сэлтон Г. Автоматическая обработка, хранение и поиск информации. Нью-Йорк, 1968. Пер. с англ., под ред. А. И. Китова. М.: Сов. радио, 1973, 580 с.

[Финн, 1976] Финн В.К. О возможностях формализации правдоподобных рассуждений средствами многозначных логик // VII Всесоюзный симпозиум по логике и методологии науки. – Киев: Наукова думка, 1976. – С. 82–83.

[Финн, 1981] Финн В.К. Базы данных с неполной информацией и новый метод автоматического порождения гипотез // Диалоговые и фактографические системы информационного обеспечения. – М., 1981. – С. 153–156.

[Финн, 1983] Финн В.К. О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона – Д.С. Милля // Семиотика и информатика. – 1983. – Вып. 20. – С. 35–101.

[Cabera et al., 2013] Cabera J.M., Escalante H. J., Montes-y-Gómez M.: Distributional Term Representations for Short-Text Categorization. 14th International Conference on Text Processing and Computational Linguistics (CICLING 2013). Samos, Greece, 2013.

[Bloom, 2012] Bloom, N., Using natural language processing to improve document categorization with associative networks Proceedings of the 17th international conference on Applications of Natural Language Processing and Information Systems, Groningen, The Netherlands, pp. 117-182, 2012.

[Fairthorne, 1961] Fairthorne, R.A., ‘The mathematics of classification’. Towards Information Retrieval, Butterworths, London, 1961.

[Fukumoto, Suzuki, 2011] Fukumoto, F. and Suzuki, Y., Cluster Labeling based on Concepts in a Machine-Readable Dictionary Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 1371–1375, November 2011.

[Good, 1958] Good, I.J., Speculations Concerning Information Retrieval, Research Report PC-78, IBM Research Centre, Yorktown Heights, New York 1958.

[Lebanon et al., 2007] Lebanon G., Mao Y., Dillon M. The Locally Weighted Bag of Words Framework for Document Representation // Journal of Machine Learning Research. – 2007. – Vol. 8. – P. 2405–2441.

[Sebastiani, 2002] Sebastiani, F:Machine learning in automated text categorization.J.ACM Comput. Surv. 34(1): 1–47, 2002.

[Sussenguth, 1964] Sussenguth, E. H., Jr. The Sentence Matching Program-Graph, 1964.

УДК 004.853

**АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ ТЕРМИНОВ
ДЛЯ ЗАДАЧИ ТОНОВОЙ КЛАССИФИКАЦИИ
В ПОСТОЯННО ОБНОВЛЯЮЩИХСЯ ТЕКСТОВЫХ
КОЛЛЕКЦИЯХ**

Ю.В. Рубцова (*yu.rubtsova@gmail.com*)

Институт систем информатики им. А.П. Ершова СО РАН,
Новосибирск

В работе описывается подход к извлечению терминов из текстов и вычислению весов для построения словаря, который постоянно обновляется. Полученный словарь используется для тоновой классификации текстов на три класса: положительные, нейтральные, отрицательные. Приведены сравнения методов, основанных на различных статистических мерах, показана вычислительная сложность пересчета весов терминов в зависимости от используемых методов.

Введение

Мы живем в постоянно меняющемся мире. Вместе с глобальными изменениями происходят изменения естественного языка. Язык постоянно трансформируется и развивается: одни слова входят в активный словарь, другие перестают употребляться. Каждый день рождаются новые слова, около половины из них – сленг. Сленг активно используют в разговорной речи и для письменного общения в социальных сетях, а также для выражения эмоционального отношения к тому или иному вопросу. В связи с этим важно учитывать изменения в естественном языке и сленг при разработке тоновых классификаторов. Более того, так как активный словарный запас регулярно пополняется новыми терминами, словари эмоциональной лексики также должны регулярно обновляться, а веса терминов в этих словарях пересчитываться.

В данной работе представлен подход к созданию тонового классификатора, извлечению терминов и назначению им весов для построения словаря эмоциональной лексики, который постоянно обновляется. Приведены сравнения результатов работы тонового классификатора в зависимости от подходов назначения весов, основанных на различных статистических мерах, показана вычисли-

тельная сложность пересчета весов терминов словаря. Все эксперименты проводились на двух коллекциях:

1. Коллекция коротких сообщений микроблогов;
2. Новостная коллекция.

1. Описание текстовых коллекций

1.1. Коллекция коротких сообщений микроблогов. В предыдущей работе автора [Рубцова, 2013] подробно описан подход к построению корпуса коротких текстов на русском языке на основе сообщений социальной сети Twitter. Twitter – это социальная сеть и сервис микробlogginga, который позволяет пользователям писать сообщения в реальном времени. Иногда сообщение пишется с мобильного устройства с места события, что добавляет сообщению эмоциональности. Из-за ограничения платформы длина twitter-сообщения не превышает 140 символов. В связи с этими особенностями сервиса люди используют аббревиатуры, сокращают слова, используют смайлики, пишут с орфографическими ошибками и опечатками. Twitter имеет особенности социальной сети, пользователи Twitter имеют возможность выражать свое мнение относительно различных вопросов: от качества телефонов до экономических и политических событий в мире. Часто новые сленговые слова появляются из сокращений слов или объединения двух и более слов в одно – это делается для того, чтобы уложиться в отведенные 140 символов. Из-за своих особенностей площадка twitter интересна для исследователей и проведения научных экспериментов с ней.

В открытом доступе нет размеченных общетематических корпусов коротких сообщений, поэтому с помощью API twitter была собрана коллекция, состоящая из около 15 миллионов коротких сообщений. На основе собранных сообщений, с помощью метода [Read, 2005] и предложенной автором [Рубцова, 2013] фильтрации, был сформирован сбалансированный корпус коротких сообщений, состоящий из следующих коллекций:

- коллекция положительных сообщений 114 991 записи;
- коллекция негативных сообщений 111 923 записей;
- коллекция нейтральных сообщений 107 990 записей.

Количество словоформ и уникальных словоформ в коллекциях представлено в таблице 1. Корпус доступен для публичного скачивания и ознакомления по ссылке <http://study.mokoron.com>.

Таблица 1

Тип коллекции	Количество словоформ в коллекции	Количество уникальных словоформ в коллекции
Положительные сообщения	1 559 176	150 720
Негативные сообщения	1 445 517	191 677
Нейтральные сообщения	1 852 995	105 239

1.2. Коллекция новостных сайтов. Новостные коллекции были собраны на новостных web-сайтах. Разметка корпуса на положительные, нейтральные и негативные коллекции производилась экспертами вручную. Отличие новостной коллекции от коллекции коротких текстов в том, что новости менее эмоциональны, лексика новостей более нейтральна и не изобилует жаргонизмами, сокращениями и ненормативной лексикой. Как правило, новостные тексты не содержат орфографических ошибок и не содержат символов, обозначающих эмоции на письме (смайликов). Новостные тексты существенно длиннее 140 символов.

Корпус новостных текстов состоит из следующих коллекций:

- коллекция положительных документов состоит из 46 339 новостей;
- коллекция негативных документов состоит из 46 337 новостей;
- коллекция нейтральных документов состоит из 46 340 новостей.

Количество словоформ и уникальных словоформ в коллекциях представлено в таблице 2.

Таблица 2

Тип коллекции	Количество словоформ в коллекции	Количество уникальных словоформ в коллекции
Положительные сообщения	4 553 010	104 001
Негативные сообщения	10 400 699	202 354
Нейтральные сообщения	7 667 441	155 538

2. Описание весовых схем

Большинство из существующих методов автоматического и полуавтоматического извлечения слов из текстов строится на условии, что все данные заранее известны, доступны и статичны. Например, для использования метода, основанного на мере TF-IDF [Salton et al., 1988], необходимо знать частоту встречаемости термина в документе, следовательно, набор данных не должен меняться во время расчета. Так, например, при добавлении нового текста в коллекцию требуется пересчитать веса для всех терминов коллекции. Вычислительная сложность перерасчета весов всех терминов в коллекции равна $O(N^2)$. Это существенно усложняет вычисления, если требуется провести обсчет данных в реальном времени.

Для того, чтобы решить проблему поиска и расчета весов терминов в режиме реального времени, была предложена [Reed et al., 2006] мера Term Frequency – Inverse Corpus Frequency (TF-ICF). Для расчета TF-ICF не требуется информации о частоте использования термина в других документах коллекции, таким образом, вычислительная сложность перерасчета весов терминов в коллекции линейна. Чтобы оценить эффективность метода, основанного на мере TF-ICF, для задачи извлечения оценочных терминов для словаря эмоциональной лексики было проведено сравнение результатов работы методов, основанных на мерах TF-ICF и TF-IDF.

Формула для вычисления меры TF-IDF:

$$tfidf = tf \times \frac{\log T}{T(t_i)}, \quad (2.1)$$

где tf – это частота встречаемости термина в коллекции твитов; T – общее число сообщений в коллекциях положительных и отрицательных; $T(t_i)$ – число сообщений в положительной и отрицательной коллекциях, содержащих термин.

Для вычисления меры TF-ICF используется следующая формула:

$$tficf = tf \times \log\left(1 + \frac{|C|}{cf(t_i)}\right), \quad (2.2)$$

где C – это число категорий; cf – число категорий, в которых встречается взвешиваемый термин.

В мере TF-IDF оба фактора взвешивания оценивают термин на уровне документа. Предложенная мера TF-ICF является смешанной: TF оценивает термин на уровне документа, а ICF на уровне категории. Необходимость подробно описывать TF-ICF обусловлена возможными разнотечениями. Например [Lertnatteed et al., 2004], предложил использовать TF-ICF, в которой фактор TF оценивает частоту встречаемости термина на уровне категории, фактор ICF аналогичен тому, что используется в этой работе.

3. Подготовка к эксперименту

3.1. Подготовка словаря и текстов для классификатора.

Прежде чем применять текстовый классификатор, необходимо представить тексты в векторном виде. Для этого коллекция коротких сообщений, описанная в пункте 1.1, подверглась фильтрации. Для составления словаря из коллекции были удалены:

- Знаки препинания (запятые, двоеточия, кавычки), восклицательные и вопросительные знаки, а также троеточия были оставлены;
- Удалены упоминания значимых личностей и событий – со временем отношение к ним может изменяться, но обученный на «старых текстах» классификатор не сможет быстро перестроиться;
- Удалены имена собственные;
- Удалены цифры (упоминания годов и времени суток были оставлены);
- Удалены слова, которые встречаются в коллекциях менее 5 раз;
- Все ссылки были объединены словом «Link» и считались в совокупности.

В результате первая версия словаря содержит 21 481 слово.

С помощью формул (2.1) и (2.2) для каждого слова в словаре были рассчитаны и сохранены веса этого слова. Далее, на основе полученного словаря, тексты коллекции коротких сообщений и тексты новостной коллекции были представлены в векторном виде.

3.2. Классификатор. Для классификации текстов на три класса был использован хорошо зарекомендовавший себя [Lan et al., 2009; Sebastiani, 2002] метод опорных векторов (Support Vector Machine).

В качестве реализации метода опорных векторов использовалась библиотека LibSVM [Chang et al., 2001; LIBSVM, 2014] с линейным ядром и регулирующим параметром C=1.

3.3. Оценка качества классификации. В качестве метрик для оценки качества классификации текстов на три класса были выбраны: accuracy, precision, recall и F-measure.

Значения точности и полноты были рассчитаны с помощью матрицы неточностей (confusion matrix). Размерность матрицы соответствует количеству классов, на которые классифицируются тексты, и равна трем. Столбцы матрицы неточностей зарезервированы за экспертными решениями, а строки – за решениями классификатора. Когда происходит классификация документа из тестовой выборки, инкрементируется число, стоящее на пересечении строки класса, который вернул классификатор, и столбца класса, к которому действительно относится документ.

Точность (Precision) равняется отношению соответствующего диагонального элемента матрицы и суммы всей строки класса. Формула для расчета точности:

$$Precision_x = \frac{A_{x,x}}{\sum_{i=1}^n A_{x,i}}$$

Полнота (Recall) – это отношение диагонального элемента матрицы и суммы всего столбца класса:

$$Recall_x = \frac{A_{x,x}}{\sum_{i=1}^n A_{i,x}}.$$

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремятся к нулю. F-мера вычисляется по формуле:

$$F = 2 \frac{Precision \times Recall}{Precision + Recall}.$$

4. Результаты эксперимента

С целью сравнения точности классификатора в зависимости от выбранного метода определения веса термина в коллекции было проведено несколько экспериментов на двух разных по структуре наборах данных. Число текстов в коллекции и, соответственно, тер-

минов в словаре было зафиксировано для эксперимента. Текстовые коллекции постоянно пополняются, поэтому словарь необходимо обновлять, а веса терминов в словаре пересчитывать.

4.1. Корпус коротких текстов. На первом этапе эксперимент был проведен для корпуса коротких текстов. Для этого корпус случайным образом был разделен на тренировочный и тестовый. Отношение количества положительных, нейтральных и отрицательных текстов сохранено в тренировочной и тестовых коллекциях. Тренировочная коллекция состоит из 267924 документов, тестовая – из 66980. Результаты работы классификатора представлены в таблице 3.

Таблица 3

	TF-IDF	TF-ICF
Accuracy	95,5981	95,0664
Precision	0,955204837	0,94984672
Recall	0,958092631	0,953112184
F-measure	0,956646554	0,95147665

Из таблицы видно, что применение метода для расчета весов, основанного на мере TF-ICF, незначительно ухудшает точность классификации коротких текстов. Следовательно, для расчета весов в динамически обновляемых коллекциях коротких сообщений могут быть применимы методы, основанные на мере TF-ICF. Таким образом, можно сократить вычислительную сложность и ускорить время работы алгоритмов.

4.2. Корпус новостей. Аналогичный эксперимент был проведен для более длинных текстов, не ограниченных верхним лимитом в 140 символов – коллекции новостей. Новостная коллекция также была разделена на тренировочную и тестовую коллекции с сохранением отношения количества текстов в каждой из категорий. В тренировочную коллекцию вошло 111 214 документов, в тестовую – 27 802. Эксперимент показал, что на длинных текстах методы, основанные на мере TF-ICF, показывают значительно худшие результаты, чем основанные на методе TF-IDF. Результаты работы классификатора представлены в таблице 4.

Таблица 4

	TF-IDF	TF-ICF
Accuracy	69,8619	58,1397
Precision	0,698624505	0,581402868
Recall	0,709246342	0,61278022
F-measure	0,703895355	0,596679322

Заключение

В работе показан подход к автоматическому построению словарей для задачи тоновой классификации текстов на три класса. Словарь строится на основе коллекций, автоматически размеченных на три класса, и является общетематическим, то есть не принадлежит к какой-либо заранее определенной предметной области. Веса в словарях вычисляются с помощью методов, основанных на двух статистических мерах. Для методов определена их вычислительная сложность при обновлении коллекции – добавлении новых сообщений. В отличие от методов, основанных на перерасчете всех весов терминов коллекции, вычислительная сложность метода, основанного на мере TF-ICF, линейна.

Экспериментально показано: несмотря на существенное снижение точности классификации длинных текстов на три класса, при использовании методов, основанных на мере TF-ICF, точность классификации коротких текстов снижается незначительно.

Полученный в результате работы программный модуль позволяет пересчитывать веса терминов словаря в режиме реального времени, отслеживать и учитывать во времени лексические изменения речи, такие как ввод в активный словарный запас новых терминов или снижение частоты использования некоторых слов, и пересчитывать веса этих терминов в зависимости от принадлежности к коллекции.

К перспективам данной работы относится применение N-грамм и морфологической разметки текстов к коллекции новостей, для сокращения разницы между методами, основанными на мерах TF-IDF и TF-ICF, а также для увеличения точности классификации текстов на три класса: положительный, нейтральный, отрицательный.

СПИСОК ЛИТЕРАТУРЫ

- [Рубцова, 2013] Рубцова Ю.В. Метод построения и анализа корпуса коротких текстов для задачи классификации отзывов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XV Всероссийской научной конференции RCDL'2013, Ярославль, Россия, 2013.
- [Chang et al., 2001] Chang C. C., Lin C. J. LIBSVM: A library for support vector machines // ACM Transactions on Intelligent Systems and Technology (TIST). 2011. №2(3).
- [Lan et al., 2009] Lan M., Tan C. L., Su J., Lu Y. Supervised and traditional term weighting methods for automatic text categorization // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009. Vol. 31, No. 4.
- [Lertnattee et al., 2004] Lertnattee V., Theeramunkong T. Analysis of inverse class frequency in centroid-based text classification // Proceedings of the 4th International Symposium on Communication and Information Technology, Japan, 2004.
- [Lertnattee et al., 2008] Lertnattee V., Theeramunkong T. Improving Thai academic web page classification using inverse class frequency and web link information // Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Work-shops, Japan, 2008.
- [LIBSVM, 2014] LIBSVM – A Library for Support Vector Machines., – <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [Read, 2005] Read J. Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification // Proceedings of the Student Research Workshop at the Annual Meeting of the Association for Computational Linguistics, USA, 2005.
- [Reed et al., 2006] Reed J., Jiao Y., Potok T. TF-ICF: A new term weighting scheme for clustering dynamic data streams // Proceedings of the 5th International Conference on Machine Learning and Applications, USA, 2006.
- [Salton et al., 1988] Salton G., Buckley C. Term-weighting approaches in automatic text retrieval // Journal of Information Processing and management. 1988. Vol. 24, Is. 5.
- [Sebastiani, 2002] Sebastiani F., Machine learning in automated text categorization // ACM Computing Surveys. 2002. Vol. 34, Is. 1.

УДК 004.52

**СИСТЕМА РАСПОЗНАВАНИЯ ТАТАРСКОЙ РЕЧИ
С УЧЕТОМ МОРФОЛОГИЧЕСКИХ ОСОБЕННОСТЕЙ
ТАТАРСКОГО ЯЗЫКА**

Д.Ш. Сулейманов (*dvdt.slt@gmail.com*)

А.Ф. Хусаинов (*khusainov.aidar@gmail.com*)

«Научно-исследовательский институт «Прикладная семиотика»

Академии наук Республики Татарстан», Казань

Казанский (Приволжский) федеральный университет, Казань

В работе описывается процесс создания системы распознавания слитной татарской речи, включающий в себя разработку средств статистического анализа текстов на татарском языке, процедуру создания акустических правил татарской речи, а также реализацию метода распознавания речи с использованием моделирования особенностей языка на морфемном уровне. Предлагаемая система показала 75%-ное качество распознавания слов тестового корпуса.

Введение

Создание систем автоматического анализа речи для множества малоресурсных языков является актуальной задачей, требующей комплексного изучения особенностей целевого языка, а также разработки программных средств для моделирования выявленных закономерностей [Berment, 2004].

В данной работе рассматривается подход к созданию системы распознавания слитной речи на татарском языке, включающий в себя следующие основные этапы исследования:

1. Проектирование и создание многодикторного корпуса звучащей татарской речи.
2. Выявление акустических особенностей татарского языка: разработка и реализация правил транскрибирования татарских текстов.
3. Анализ созданного корпуса звучащей татарской речи и создание на его основе акустических моделей фонем татарского языка.
4. Формализация и использование для улучшения качества распознавания закономерностей, существующих в татарском языке на морфемном уровне.

5. Программная реализация системы автоматического распознавания фонем.

1. Система распознавания татарской речи

1.1. Корпус звучащей татарской речи

Процесс создания корпуса звучащей речи начинается с этапа подготовки текстового корпуса для озвучки. Подобранные тексты должны обладать определенными характеристиками, продиктованными конечной стоящей задачей распознавания. В случае создания дикторонезависимых программных средств распознавания слитной речи необходимо учесть как фонетические, так и лингвистические особенности татарского языка.

Так, с лингвистической точки зрения, тексты должны обладать теми же особенностями, что и произносимые в процессе работы системы распознавания предложения. А учитывая, что на этапе проектирования системы никаких ограничений на произносимые фразы не накладывается, тексты должны включать в себя различные по происхождению материалы: литературные, новостные материалы, а также спонтанные фразы, используемые в каждодневном общении. Кроме того, был учтен факт использования технологии phoneme-alignment для фонетической разметки записанного речевого корпуса на основе транскрипций. Данная технология подразумевает итерационный процесс по уточнению границ произнесения каждой фонемы в каждой записи, и для лучшего качества его работы в текстовый корпус были добавлены отдельные татарские слова. Это позволит на начальных итерациях на небольшом наборе фонем в отдельных словах более точно установить их границы и улучшить процесс работы алгоритма.

В результате был создан текстовый корпус, который можно представить состоящим из 4 основных частей:

1. Литературных фрагментов;
2. Отдельных слов;
3. Новостных материалов;
4. Спонтанных фраз.

Общий размер полученного корпуса составил 776 предложений, включающих 6913 слов. Средняя длина предложений составила 8,9 слова.

Для того чтобы программные средства распознавания могли работать в дикторонезависимом режиме, то есть распознавать речевые фрагменты, произнесенные дикторами, неизвестными системе на этапе обучения системы, необходимо на этапе создания акустических моделей предоставить информацию о большом количестве вариантов произнесения одних и тех же фонем языка. Для этих целей была произведена запись 251 диктора. В состав дикторов вошли как мужчины, так и женщины, люди различных возрастов и особенностей произношения.

Полученный в результате проектирования и записи речевой корпус имеет параметры, представленные в таблице 1.

Таблица 1

Параметр	Весь корпус	Обучающая часть	Тестовая часть
Количество файлов	8638	8125	513
Продолжительность записей	8:14:24	7:48:12	26:12
Количество дикторов	251	236	15

1.2. Акустические особенности татарского языка

Для работы программных средств распознавания речи необходима информация о составе графического и фонемного алфавитов анализируемого языка, а также о правилах перехода между грамматической записью и фонетической. Алфавит татарского языка состоит из 39 символов: к буквам русского алфавита в нем добавлены татарские буквы Ә-ә, Ө-ө, Ү-ү, Ж-җ, Н-ң, Й-һ. Вопрос о составе фонемного алфавита татарского языка не имеет однозначного ответа, поэтому при его формировании учитывались как имеющиеся на данный момент результаты фонетических исследований татарского языка, так и особенности работы программных средств распознавания речи, заключающиеся в необходимости выделения основных звуков языка, которые влияют на смысл произнесенного, группируя при этом схожие по звучанию звуки. В результате проведенного исследования был сформирован алфавит из 57 фонем татарского языка. На основе определенного алфавита фонем были сформулированы акустические закономерности в татарском языке. В конечном итоге для случая татарского языка было выделено 37 правил фонетической транскрипции [Хусаинов, 2013].

1.3. Акустические модели фонем татарского языка

Основной частью программных средств распознавания фонем является набор акустических моделей языка. Каждая фонема была смоделирована скрытой Марковской моделью, состоящей из трех состояний, с ограничениями на переход на более ранние состояния. Каждое из трех состояний моделировалось, в свою очередь, смесью Гауссовских распределений [Gales et al., 2007]. Процесс обучения акустических моделей производился итерационно: итерация увеличения количества Гауссовских распределений в смеси сопровождалась двумя итерациями переобучения на данных обучающего корпуса.

Было произведено описание параметров, с помощью которых будет описываться речевой сигнал, а также проведены эксперименты по определению количества итераций и числу Гауссовских распределений для каждого состояния моделей, при которых качество распознавания было бы наилучшим. В качестве параметров, рассчитываемых для речевых фрагментов, был выбран вектор кепстральных коэффициентов MFCC [Deller et al., 1999]. Необходимое число итераций обучения было определено экспериментальным путем, для чего количество итераций постепенно увеличивалось с 10 до 170. В ходе эксперимента количество Гауссовских распределений в смесях постепенно наращивалось, после каждого увеличения происходило два цикла переобучения всех моделей.

На основе данных, полученных экспериментальным путем, при построении фонетического распознавателя было решено выбрать модели фонем, полученные на 91 итерации. В этих моделях число распределений в Гауссовских смесях равняется 31.

1.4. Морфемная модель татарского языка

Языковая модель призвана описывать существующие в языке закономерности и на их основе уметь оценивать вероятности произнесения конкретных последовательностей слов.

В качестве языковой модели, чаще всего, используют основанную на статистике модель n-грамм [Manning et al., 1999]. Модель n-грамм исходит из предположения, что вероятность произнесения слова можно рассчитать на основе последовательности предшествующих слов, а умев вычислять вероятность появления каждого слова w_i во фразе $\{w_1, \dots, w_n\}$, можно рассчитать и вероятность произнесения фразы целиком:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, w_2, \dots, w_{i-1}) \quad (1.1)$$

Вероятности произнесения каждого слова в различных контекстах определяются на этапе подготовки языковой модели на основе больших текстовых корпусов. В качестве оценки для условных вероятностей берется отношение количества наблюдений последовательностей слов:

$$P(w_i | w_1, \dots, w_{i-1}) = \frac{N(w_1, \dots, w_i)}{N(w_1, \dots, w_{i-1})}, \quad (1.2)$$

где $N(w_1, \dots, w_j)$ – количество наблюдений последовательности слов $\{w_1, \dots, w_j\}$ в корпусе.

В данной работе был использован вариант n-граммных моделей, которые основываются на элементах, по размеру меньших слова (particle-based models). В этом случае слова представляются в виде морфем, и происходит анализ статистических закономерностей между ними, а не целыми словами. Данная особенность ценна для случаев распознавания языков с богатой морфологией, например, для агглютинативных языков, каким является и татарский язык.

Все слова текстового корпуса были разбиты на составляющие их морфемы, к последней морфеме слова добавлялся специальный знак #. Далее, был сформирован словарь для распознавания, который в контексте данной работы состоял не из слов, а из отдельных морфем, что позволило уменьшить общий размер словаря при увеличении полноты покрытия слов языка.

1.5. Результаты экспериментов

Этап распознавания является заключительным и основывается на всей информации, полученной на предыдущих этапах. Таким образом, используются построенные и обученные акустические модели, список выделенных морфем для распознавания, а также их транскрипции. Программные средства распознавания также умеют рассчитывать вероятности той или иной фразы на основе языковой модели.

Для проведения экспериментов по распознаванию речи была использована тестовая часть созданного речевого корпуса. Для всех морфем, вошедших в словарь, была автоматически создана фонетическая транскрипция, которая послужила основой для

объединения статистических моделей отдельных фонем в модели морфем. На этапе распознавания с помощью алгоритма Витерби рассчитывались наиболее вероятная последовательность произнесенных фонем языка, а затем вероятности соответствия найденной последовательности моделям морфем и слов языка. Слово, соответствующее модели с максимальной вычисленной вероятностью, принималось в качестве результата распознавания.

По результатам работы системы на тестовом корпусе точность распознавания составила 75%. Стоит отметить, что корректность распознавания фонем при этом составила 63%, что позволяет говорить о качестве работы системы, сопоставимом с существующими для других языков образцами систем.

На базе созданных программных средств распознавания речи было разработано приложение для распознавания слитной татарской речи, рис. 1. Пользовательский интерфейс системы распознавания фонем состоит из кнопок, отвечающих за начало, окончание записи файла, кнопки, запускающей процесс распознавания, а также двух текстовых полей, отображающих текущее расположение аудио-файла и результат проведенного распознавания. Приложение также предоставляет возможность записи речевого фрагмента с помощью микрофона и загрузки звуковых файлов. Форма приложения отображает информацию о текущем выбранном устройстве записи звука, текущем аудио-файле (или об их количестве). Текстовый файл со списком произнесенных слов может быть

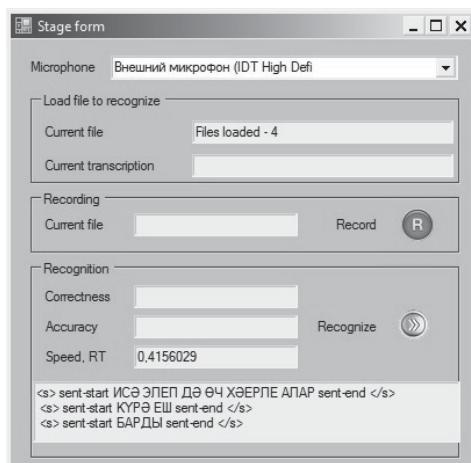


Рис. 1. Модуль распознавания речи

загружен в систему для обеспечения возможности расчета качества проведенного распознавания. Рассчитанные характеристики качества распознавания в таком случае отображаются в соответствующих текстовых полях данных. Кроме того, на экран выводятся результаты распознавания слов в текущих файлах.

Заключение

Построение и реализация алгоритма автоматического транскрибирования татарских текстов, а также создание аннотированного корпуса звучащей татарской речи позволило реализовать программный модуль автоматического распознавания речи на татарском языке. Особенностью созданной системы являлось использование морфемного уровня анализа при моделировании закономерностей татарского языка. В рамках проведенных экспериментов по распознаванию слитной речи татарского языка построенная система показала 75%-ное качество распознавания, что позволило использовать ее при создании пользовательского интерфейса для распознавания речи с возможностями записи речевого фрагмента с микрофона.

СПИСОК ЛИТЕРАТУРЫ

- [Berment, 2004] Berment, V. Méthodes pour informatiser des langues et des groupes de langues peu dote's.: Ph.D. Thesis, J. Fourier University. – Grenoble I, 2004.
- [Хусаинов, 2013] Хусаинов А.Ф. Система автоматического распознавания фонем татарского языка // Компьютерная обработка тюркских языков: труды первой международной конференции. (Астана, 3–4 октября, 2013). Астана: ЕНУ им. Л. Н. Гумилева, 2013. С. 211–217.
- [Gales et al., 2007] Gales, M, Young, S. The Application of Hidden Markov Models in Speech Recognition // Foundations and Trends in Signal Processing. 2007. Vol. 1, № 3. P. 195–304.
- [Deller et al., 1999] Deller, R. Jr., Hansen, H. L., Proakis, G. Discrete -Time Processing of Speech Signals. – Wiley-IEEE Press, 1999. – 936 p.
- [Manning et al., 1999] Manning, C.D, Schutze H. Foundations of Statistical Natural Language Processing. – Cambridge, Massachusetts: MIT - Press, 1999. – 704 p.

УДК 004.9

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: КАРТА НАУЧНОГО НАПРАВЛЕНИЯ В ТРУДАХ КОНФЕРЕНЦИЙ РАИИ

В.Ф. Хорошевский (*khor@ccas.ru*)

Вычислительный центр РАН, Москва

И.В. Ефименко (*iefimenko@hse.ru*)

ЦИАС ИСИЭЗ НИУ ВШЭ, Москва

В работе обсуждаются вопросы построения карты научного направления «Искусственный интеллект», сформированной на основе наукометрического анализа публикаций, представленных в трудах конференций РАИИ. Обработка корпуса выполнена с использованием методов извлечения информации из текстов статей под управлением системы онтологических моделей и кластеризации полученных результатов на базе различных методов. Полученные карты научного направления исследованы с использованием методов анализа временных рядов, облаков тэгов и методов выявления скрытых коллективов.

Введение

В настоящее время наукометрия и библиометрия становятся «горячими точками», где пересекаются интересы специалистов, работающих в различных областях науки и техники, менеджеров, занятых управлением наукой, и государства, заинтересованного в эффективном использовании средств, отпускаемых на развитие новых научных направлений. При этом для оценки ситуации в той или иной области, как правило, используются статистические методы библиометрического и (реже) наукометрического анализа публикаций, которые далеко не всегда адекватны задачам построения действительно информативных карт науки.

Вместе с тем в рамках научного направления «Искусственный интеллект» уже давно разрабатываются и используются достаточно мощные методы анализа данных, с одной стороны, и методы извлечения информации из текстов на естественных языках – с другой. Как представляется, совместное использование этих методов может дать значимые и семантически интерпретируемые результаты в области построения карт научно-технических направлений.

Учитывая вышеизложенное, целью настоящей работы является обсуждение вопросов построения карты научного направления «Искусственный интеллект» (ИИ), сформированной на основе анализа публикаций, представленных в трудах конференций РАИИ.

1. Технология построения карты научного направления ИИ

1.1. Предварительные замечания

Как отмечалось выше, для построения карты науки представляется целесообразным использовать гибридный подход, где методы кластеризации и классификации работают на данных, сформированных с помощью методов извлечения информации из текстов под управлением онтологий [Kim, et al., 2009; Wang, et al., 2010; Li, et al., 2011; Erdi, et al., 2013]. В основе подхода к выделению из публикаций семантически значимой системы терминов предметной области, которые специфицируют характеристические векторы публикаций, лежит принцип «черного ящика» [Efimenko, et al., 2014], а интеграция результатов обработки коллекций отдельных жанров осуществляется на основе результатов статистической обработки коллекций публикаций. Спецификой предлагаемого подхода является и то, что после лингвистической и статистической обработки текстов происходит автоматическая генерация семантических представлений результатов, которые затем отражаются в экземплярную часть OWL-онтологии ПО «Искусственный интеллект», в XML-файлы для системы кластеризации Cartot, а также в спецификации графического представления карты научного направления и временные ряды терминов.

1.2. Онтологическое моделирование предметной области ИИ

Для автоматизации процессов построения карты научного направления ИИ была разработана система онтологических моделей, в которых специфицированы ключевые типы объектов данной ПО с их атрибутами, отношения между объектами, а также система лингвистических маркеров-индикаторов для выявления семантически значимых терминов. При этом в качестве исходной модели научного направления ИИ используется тезаурус, построенный на базе анализа тематических рубрик, представленных в трудах КИИ

[Долматова, 1992; Гаврилова и др., 2000; Кузнецов, 2009]. Для примера, на рис. 1 представлен фрагмент онтологической модели предметной области ИИ.

Заметим, что разработанная система моделей используется как на этапе извлечения информации из текстов, так и в процессе генерации результатов обработки.

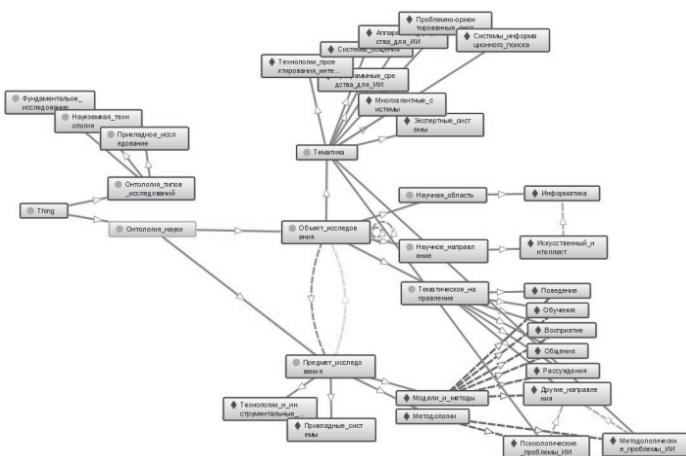


Рис. 1. Фрагмент онтологической модели предметной области ИИ

1.3. Автоматизация процессов построения карт научных направлений

Как показывает анализ литературы по средствам автоматизированного построения карт науки [Shibata, et al., 2008; Li, et al., 2011], в общей схеме обработки информации целесообразно выделять этапы выявления центров компетенции (включая организации и авторские коллективы) в предметной области; формирования презентативной коллекции документов с учетом выявленных на предыдущем этапе центров превосходства и компетенций и собственно обработки сформированных коллекций документов.

Обсуждение методов и средств автоматизированного формирования экспертных групп и выявления центров экспертной компетенции в определенных предметных областях представлено в [Хорошевский, 2010].

Общая архитектура системы построения карт науки представлена на рис. 2.

В качестве инструментария для извлечения информации из текстов в настоящем исследовании использована платформа GATE, расширенная плагином Russian Morph Tagger¹, где используется открытая версия модуля русской морфологии компании Яндекс², а также специально разработанными модулями обработки именных групп RusChunkerNP и генерации характеристических векторов документов (Bag of Terms) с расчетом TF*IDF входящих в коллекции терминов. Таким образом, в процессе обработки документов коллекций формируется информация, достаточная для последующего анализа экспертами на предмет наличия в системе коллекций статистически значимой информации, необходимой для построения карты научного направления.

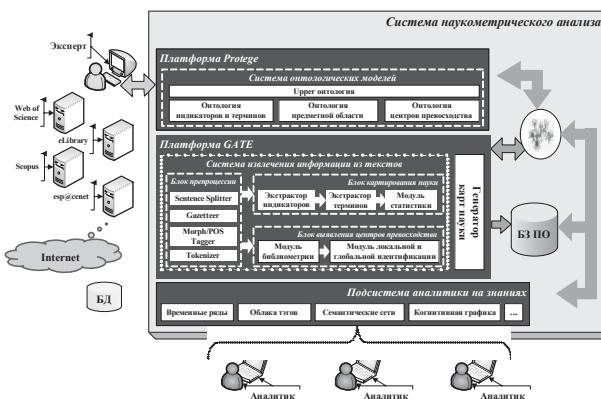


Рис. 2. Общая архитектура системы построения карт науки

Для удобства анализа результатов обработки коллекций документов был разработан и реализован специальный модуль (OWL Generator), который базируется на идеях, представленных в работе [Witte, et al., 2010].

На вход этого модуля поступает характеристический вектор коллекции (системы коллекций), а на выходе формируется ее OWL-представление, соответствующее онтологии ПО ИИ. Полученные OWL-представления загружаются в систему онтологического

¹ <http://gate.ac.uk/>

² <http://company.yandex.ru/technology/mystem/>

инжиниринга Protégé¹ и используются экспертами для дальнейшего анализа. В системе построения карт науки экспертами также используются мощный инструментарий кластеризации Carrot² и система визуализации облаков тэгов TagCloud³.

2. Карта научного направления ИИ в трудах КИИ

2.1. Коллекция документов

Для построения карты научного направления ИИ авторами был сформирован корпус статей из трудов конференций КИИ. Распределение статей по отдельным конференциям дано в табл. 1. К сожалению, из 2000 статей, которые есть в трудах конференций КИИ, в электронном виде удалось собрать для обработки только 581 статью в коллекцию для дальнейшей обработки.

Таблица 1

Распределение статей по отдельным конференциям

КИИ	1988	1990	1992	1994	1996	1998	2000	2002	2004	2006	2008	2010	2012
Всего статей	308	174	168	–	126	73	–	–	116	68	98	138	151
Обр. статей					126					68	98	138	151

2.2. Схема обработки коллекции

Обработка коллекции производилась в соответствии со схемой, представленной на рис. 3.

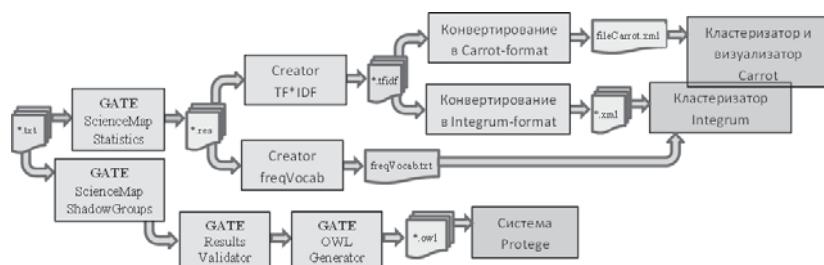


Рис. 3. Схема обработки коллекции статей КИИ

¹ <http://www.protege.com>

² <http://www.Carrot.com>

³ <http://www.TagCloud.com>

2.3. Методика сбора информации и анализ результатов

Методика сбора результатов обработки коллекции состояла в следующем:

- Отдельные статьи обрабатывались гибридным модулем извлечения информации из текстов, на выходе которого формировались семантическая сеть отдельной статьи и характеристический вектор многословных и однословных терминов, представленных в этой статье.
- Характеристические векторы статей, составляющих отдельные секции одной конференции, интегрировались в характеристические векторы секций, а затем – в характеристические векторы отдельной конференции.
- Семантические сети статей, составляющих отдельные секции одной конференции, интегрировались в семантическую сеть секций, а затем – в общую семантическую сеть отдельной конференции.
- Кластеризация характеристических векторов отдельных конференций осуществлялась с помощью разных методов и разными системами кластеризации.
- На результатах кластеризации отдельных конференций строились временные ряды терминов.
- На результатах объединения семантических сетей конференции проводился анализ выявленных скрытых коллективов авторов.

К сожалению, объем настоящей статьи не позволяет представить все, даже наиболее интересные, результаты. Поэтому ниже приведены лишь некоторые фрагменты общей карты науки направления «Искусственный интеллект», как она сформирована на основе автоматического анализа имеющейся в распоряжении авторов электронной коллекции документов.

Так, на рис. 4 представлена экранная форма результатов обработки одной из статей гибридным модулем извлечения информации из текстов, на рис. 5 – семантическая сеть той же статьи в системе Protégé, а на рис. 6 – результаты кластеризации коллекции статей КИИ-2006 в системе Carot.



Рис. 4. Экранная форма результатов извлечения информации из одной статьи

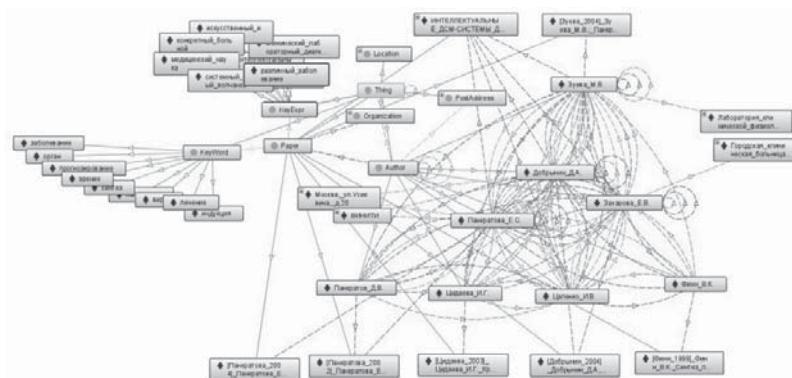


Рис. 5. Семантическая сеть статьи в системе Protégé

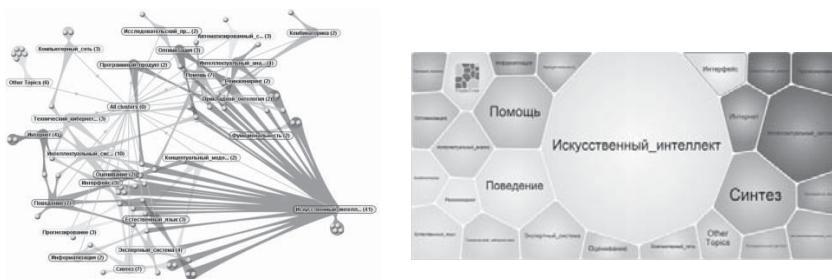


Рис. 6. Результаты кластеризации коллекции статей КИИ-2006 в системе Carot

Следует отметить, что полученные результаты показывают значительное число автоцитирований авторов статей в конференциях серии КИИ, а также существенную инкапсуляцию цитирований в пределах одной организации и/или проекта. При этом количество цитирований, которые учитываются при подсчете индексов цитируемости и индексов Хирша авторов, существенно меньше, чем это требуется для выявления скрытых коллективов даже в отдельных направлениях исследований и разработок в области ИИ.

Заключение

В работе представлена методика построения карт науки на основе автоматической обработки корпуса научных статей на примере карты научного направления «Искусственный интеллект», сформированной на основе наукометрического анализа трудов конференций РАИИ. Обработка корпуса выполнена с использованием методов извлечения информации из текстов статей под управлением системы онтологических моделей и кластеризации полученных результатов на базе различных методов. Полученные карты научного направления исследованы с использованием методов анализа временных рядов, облаков тэгов и методов выявления скрытых коллективов.

СПИСОК ЛИТЕРАТУРЫ

- [Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник, С.-Петербург: Питер пресс, 2000.
- [Долматова, 1992] Долматова Л.М. От первой до третьей конференции АИИ: как оценить ускользающую реальность? //Новости искусственного интеллекта, № 3, 1992, с. 61–124.
- [Кузнецов, 2009] Кузнецов О.П. Искусственный интеллект и прикладные проблемы информатики, Учебное пособие. М.: КОС-ИНФ, 2009.
- [Хорошевский, 2010] Хорошевский В.Ф. Извлечение информации из текстов на конференциях серии ДИАЛОГ: взгляд соседа по лестничной клетке // Труды международной конференции «Диалог 2010». – М.: Наука, 2010.
- [Efimenko, et al., 2014] Efimenko I., Khoroshevsky V. New Technology Trends Watch: an Approach and Case Study. In: Proc. of AIMSA-2014.

[Erdi, et al., 2013] Erdi P., Makovi K., and et al. Prediction of emerging technologies based on analysis of the US patent citation network. *Scientometrics*, 2013, 95 (1), pp. 225–242 (2013).

[Kim, et al., 2009] Kim Y., Tian Y., Jeong Y., Jihee R Myaeng., S.-H. Automatic Discovery of Technology Trends from Patent Text. In: Proc, SAC'09, March 8-12, 2009, Honolulu, Hawaii, U.S.A., (2009).

[Li, et al., 2011] Li H., Xu F., Uszkoreit H.: TechWatchTool: Innovation and Trend Monitoring. In: Proc. of the International Conference on Recent Advances in Natural Language Processing 2011 RANLP 2011, Tissar, Bulgaria, pp. 660–665 (2011).

[Shibata, et al., 2008] Shibata N., Kajikawa Yu., Takeda Y., Matsushima K.: Detecting emerging research fronts based on topological measures in citation networks of scientific publications, *Technovation*, Vol. 28, Issue 11, November 2008, pp. 758–775 (2008).

[Wang, et al., 2010] Wang et al. Identifying technology trends for RD planning using TRIZ and text mining, *RD Management*, vol. 40, № 5, (2010).

[Witte, et al., 2010] Witte R., Khamis N., Rilling J., Flexible ontology population from text: The owlexporter. In International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, 05/2010 2010.

СЕКЦИЯ 3 | КОГНИТИВНОЕ МОДЕЛИРОВАНИЕ
И КОГНИТИВНАЯ ГРАФИКА

УДК 007:519.816

ИНТЕЛЛЕКТУАЛЬНАЯ КОГНИТИВНАЯ СИСТЕМА
РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ КОНТРОЛЯ БЕЗОПАСНОСТИ
РАБОТЫ АТОМНЫХ ЭЛЕКТРОСТАНЦИЙ КОНЦЕРНА
«РОСЭНЕРГОАТОМ»

А.А. Башлыков (*aleks-ltd@mail.ru*)
Национальный исследовательский университет «МЭИ»,
М.А. Лыгин (*Lygin@transenergostroy.ru*)
ООО «ТРАНСЭНЕРГОСТРОЙ»

В статье рассматриваются вопросы применения средств когнитивной графики для оперативной поддержки принятия решений оперативным персоналом Кризисного центра концерна «Росэнергоатом» при решении задач оперативного мониторинга и контроля выполнения графика несения нагрузки энергоблоками атомных станций по обобщенным параметрам.

Введение

Мыслительный процесс человека-оператора в процессе управления сложным технологическим объектом является процедурой, направленной на своевременное решение проблем, возникающих на объекте контроля и управления [Пушкин, 1965].

Мышление человека-оператора определяется умением увидеть проблему там, где она существует, и потребностью найти ее решение [Гришин, 1982].

Наглядно-образное мышление – один из видов мышления человека-оператора. Оно связано с образным представлением ситуации и отслеживанием изменений в ней. С помощью наглядно-

образного мышления наиболее полно воссоздается все многообразие различных фактических характеристик объекта контроля и управления.

Средства образного представления информации для принятия решений, основанные на методах когнитивной графики [Зенкин, 1991], [Солсо, 1996], ориентированы на использование механизмов наглядно-образного мышления. Они в последнее время находят все большее применение в информационных системах интеллектуальной поддержки принятия решений для оперативного человеко-машинного управления сложными и экологически опасными объектами и технологиями [Башлыков, 1996], [Трахтенгерц, 1992]. Когнитивная графика – это совокупность приемов и методов образного представления условий задачи, которое позволяет либо сразу увидеть *решение*, либо получить *подсказку* для его нахождения [Поспелов, 1992].

Однако до настоящего времени не существует обобщенного взгляда на методы и принципы построения средств образного представления для интеллектуальных систем поддержки принятия управляющих решений (СППР-РВ), функционирующих в режиме реального времени [Вагин и др., 2001], [Башлыков А.А. и др., 2001]. Один из реализованных подходов к построению такого класса СППР-РВ «ОКО» [Башлыков А.А. и др., 2003] предлагается рассмотреть в настоящей статье.

1. Функции СППР-РВ «ОКО»

СППР-РВ «ОКО» реализует функции информационной поддержки лиц диспетчерского и административного персонала Кризисного центра концерна «Росэнергоатом» в части:

1. Контроля состояния энергоблоков и идентификации факта превышения пределов безопасной эксплуатации энергоблоков, в том числе:

- энерговыделения в реакторе;
- обеспечения условий теплоотвода из активной зоны;
- обеспечения условий локализации и надежного удержания радиоактивных продуктов;

2. Контроля графика несения нагрузки идентификацией фактов изменения нагрузки энергоблоков АЭС (изменение нагрузки на величину более ΔN за время менее ΔT);

3. Идентификации факта срабатывания технологических и аварийных защит, устройств и систем безопасности, в том числе:

- автоматической разгрузки реактора действием технологической защиты;
- останова реактора действием аварийной защиты;
- срабатывания устройств и систем безопасности.

Функции интеллектуальной информационной поддержки в системе «ОКО» реализуются, в свою очередь, следующими средствами:

1. Оперативного дистанционного сбора информации из баз данных ЦДУ, СПП и АСКРО концерна «Росэнергоатом», содержащих текущие состояния технологических, радиационных параметров и параметров безопасности по следующим атомным электрическим станциям (АЭС): Балаковской АЭС; Белоярской АЭС; Калининской АЭС; Кольской АЭС; Курской АЭС; Нововоронежской АЭС; Ростовской (Волгодонской) АЭС; Смоленской АЭС.

2. Оперативного ситуационного анализа собранной информации на предмет:

- контроля текущего состояния технологических, радиационных и параметров безопасности АЭС;
- контроля выполнения графика несения нагрузки АЭС;
- контроля динамики изменения основных технологических, радиационных и параметров безопасности АЭС.

3. Оперативного оповещения диспетчерского персонала о:

- текущем состоянии технологических, радиационных и параметров безопасности АЭС;
- выполнении графика несения нагрузки АЭС;
- фактах превышения параметрами предупредительных или аварийных технологических уставок;
- фактах отклонения от графика несения нагрузки АЭС.

4. Экспертной системы диагностирования нештатных ситуаций, осуществляющей поиск диагнозов, прогнозов их развития и рекомендуемых управляющих действий в этих ситуациях и аномальных состояниях;

5. Гипертекстовой информационной поддержки пользователя при работе с регламентной и эксплуатационной информацией.

6. Когнитивного пользовательского интерфейса, ориентированного на диспетчерский персонал.

2. Принципы организации когнитивного пользовательского интерфейса СППР «ОКО»

Когнитивный пользовательский интерфейс системы «ОКО» построен в виде четырехоконной системы представления информации (рис. 1, 2).

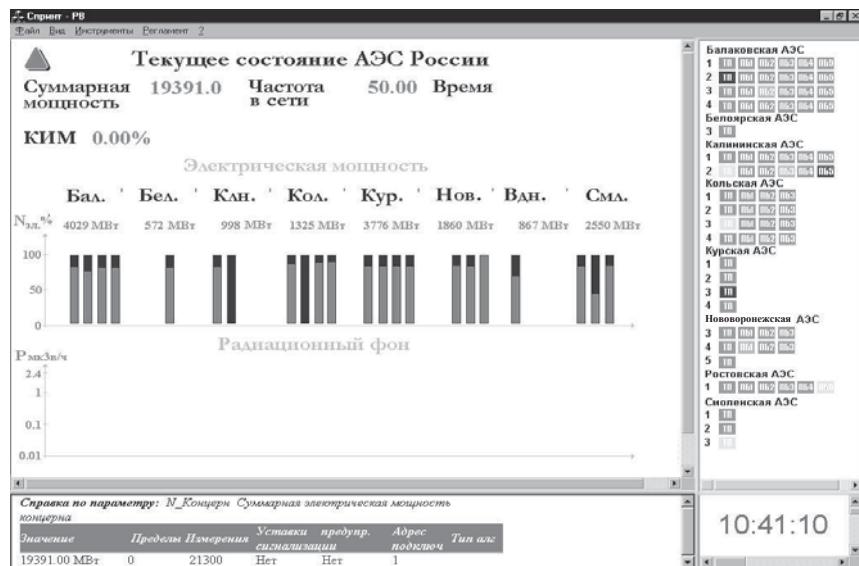


Рис. 1. Вид четырехоконного пользовательского интерфейса административного персонала Кризисного центра

В каждом из окон отображается определенный вид информации. Левое верхнее окно (*окно состояния*) используется для представления текущего состояния технологических параметров (ТП), радиационных параметров (РП) и параметров безопасности (ПБ) для каждой из площадок АЭС.

Правое верхнее окно (*окно диагностики*) используется только для отображения состояния контролируемых параметров как элементов групповой сигнализации. Левое нижнее окно (*окно гипертекстовой информации*) предназначено для отображения технологических регламентов (это может быть и иная регламентирующая информация) и справок по любой переменной состояния.

Окно состояния объекта контроля предназначено для отображения информационной модели объекта контроля в виде иерархии динамических мнемосхем. На мнемосхемах представляется оперативная обобщенная и детальная информация о контролируемом объекте. Динамика элементов мнемосхем определяется управлением значениями переменных, ситуационным изменением их конфигурации и изменением цвета.

На обобщенных мнемосхемах, на мнемосхемах второго уровня (см. ниже) контролируемые состояния, в которых отсутствуют факты нарушения регламентных границ, окрашены в зеленый цвет. Контролируемые состояния, в которых имеются факты нарушения состояний, окрашиваются в красный цвет.

На детальных мнемосхемах, характеризующих состояние технологических, радиационных параметров и параметров безопасности нормальное состояние аналогового параметра отображается зеленым цветом, состояние параметра, вышедшее за уставки, может отображаться (в зависимости от типа уставки) желтым, оранжевым, красным цветом.

При состоянии включено/открыто технологического оборудования, приборов, устройств, характеризуемых в системе дискретными параметрами, информация о них отображается зеленым ровным цветом.

Состояние технологического оборудования, приборов, устройств отключено/закрыто отображается красным цветом.

Нормальное состояние защитных систем, устройств, когда они не сработали, отображается зеленым цветом. Срабатывание защитных систем, устройств отображается красным цветом. Для дополнительного привлечения внимания можно использовать мигание динамического образа. Недостоверная информация выделяется белым цветом. Промежуточное состояние объекта отображается желтым цветом. Фон статического изображения черного либо темно-серого цвета.

На рис. 1 в окне состояния показан образ текущего состояния объекта для **административного** персонала концерна.

При инициализации системы «ОКО» на рабочем месте **диспетчерского** персонала в окне состояния объекта контроля (рис. 2) отображается когнитивный образ, представляющий обобщенную информацию о текущем состоянии всех площадок АЭС (*мнемосхема первого уровня информационной модели*).

Образ «ромашка» в центре характеризует состояние всех видов контролируемой информации по площадкам АЭС. Полукружья на концах «ромашки» показывают отношение плана по выработке электроэнергии к фактической выработке на каждой из площадок АЭС. Два полукруга в центре «ромашки» показывают соотношение плановой выработки электроэнергии к фактической выработке для концерна «Росэнергоатом» в целом.

В правом окне представляется состояние всех энергоблоков, расположенных на конкретной площадке АЭС.

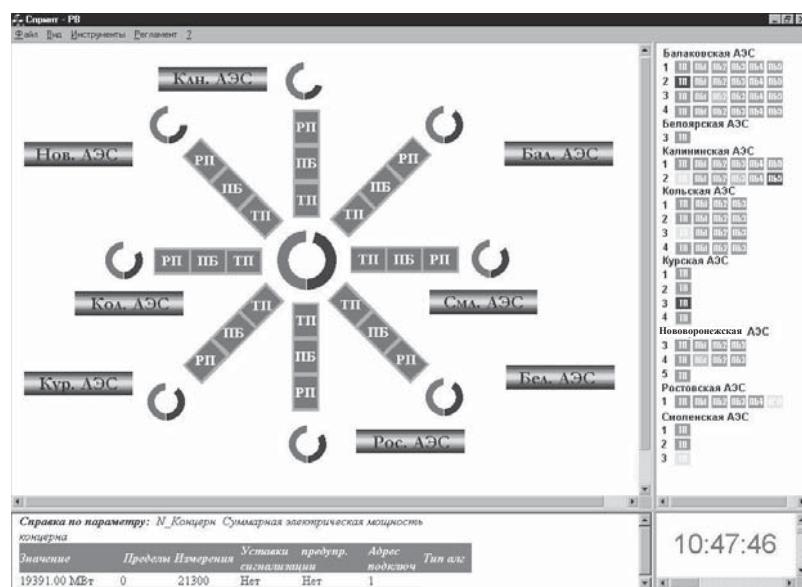


Рис. 2. Вид четырехоконного пользовательского интерфейса оперативного диспетчерского персонала Кризисного центра

Образы состояния технологических, радиационных и параметров безопасности используются для *процесса навигации*, реализуемого пользователем по информационной модели объекта контроля, т.е. перехода с мнемосхемы на мнемосхему методами «вглубь» и «вверх».

Для осуществления перехода на мнемосхему более низкого уровня применяется метод навигации «вглубь» по информационной модели. При реализации этого метода необходимо подвести

курсор мыши к соответствующему образу, вокруг которого появится зеленый прямоугольник, и нажать левую клавишу мыши. После этого в основном окне программы будет отображена мнемосхема, содержащая обобщенную информацию о текущем состоянии выбранной АС и всех ее блоков (*мнемосхема второго уровня информационной модели*).

На мнемосхемах *второго уровня информационной модели* каждый энергоблок выбранной АЭС представлен отдельной «строкой» таблицы, состоящей (рис. 3) из:

- образа, характеризующего текущее значение мощности АЭС;
- образа, характеризующего состояние технологических параметров (ТП);
- образов, характеризующих состояние параметров безопасности (ПБ).

Балаковская АЭС

Блок	N%	Тех. пар.	Парам. безопасности				
1	83	ТП	ПБ1	ПБ2	ПБ3	ПБ4	ПБ5
2	77	ТП	ПБ1	ПБ2	ПБ3	ПБ4	ПБ5
3	82	ТП	ПБ1	ПБ2	ПБ3	ПБ4	ПБ5
4	62	ТП	ПБ1	ПБ2	ПБ3	ПБ4	ПБ5

Рис. 3. Пример характеристик контролируемых параметров на площадках АЭС

Образы, характеризующие состояние технологических (ТП) и параметров безопасности (ПБ), полностью аналогичны соответствующим образам на главной мнемосхеме. В нормальном состоянии (когда значения всех параметров мнемосхемы находятся в пределах уставок) цвет фона образа – зеленый. В случае выхода какого-либо из параметров соответствующего типа за уставки, цвет фона образа – красный.

Переход на мнемосхемы *третьего уровня информационной модели* осуществляется аналогично переходу с мнемосхемы первого уровня на второй.

На мнемосхемах *третьего уровня информационной модели* представлена наиболее детальная информация обо всех переменных состояния (рис. 4).

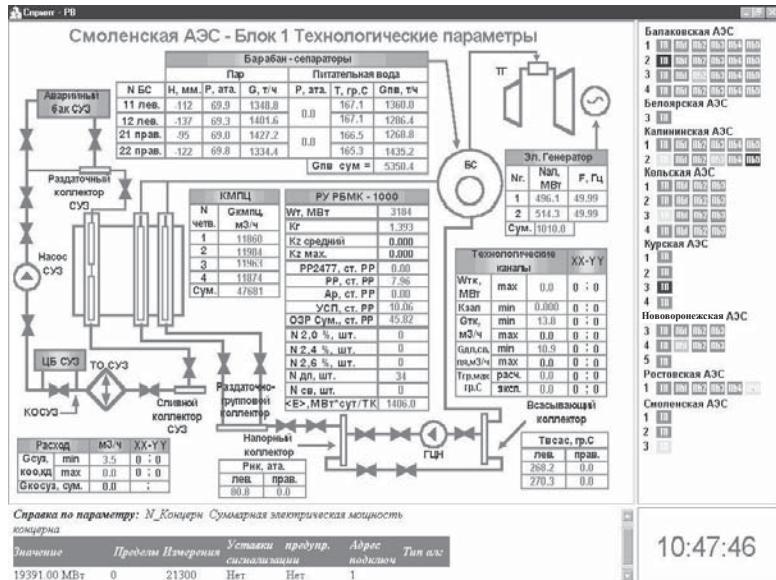


Рис. 4. Вид пользовательского интерфейса диспетчерского персонала Кризисного центра для представления технологических параметров

Окно групповой сигнализации отклонений контролируемых переменных состояния предназначено для отображения образа групповой сигнализации в виде фактов наличия отклонений в контролируемых элементах технологического оборудования и процесса.

На этом образе факт отклонения переменных состояния, представленных на каждой из мнемосхем, входящих в информационную модель объекта контроля (мнемосхемы третьего уровня информационной модели), представлен отдельной кнопкой.

В нормальном состоянии (когда значения всех параметров мнемосхемы находятся в пределах уставок) цвет фона кнопки – серый. В случае выхода какого-либо из параметров мнемосхемы за уставки, цвет фона кнопки – красный мигающий. Мнемосхема группово-

Балаковская АЭС
1 III
2 III
3 III
4 III
Белоярская АЭС
3 III
Калининская АЭС
1 III
2 III
3 III
4 III
Нововоронежская АЭС
3 III
4 III
5 III
Ростовская АЭС
1 III
2 III
3 III
Смоленская АЭС
1 III
2 III
3 III

вой сигнализации может также использоваться для быстрого перехода на мнемосхемы третьего уровня (минуя второй).

Окно гипертекстовой информации – технологических регламентов эксплуатации предназначено для отображения технологических регламентов (иной регламентирующей информации) и справок по параметрам состояния в формате HTML.

Навигация по гипертексту в системе осуществляется по методу «вглубь» и «вверх».

Технологические регламенты отображаются в окне гипертекстовой информации автоматически при загрузке соответствующей мнемосхемы (рис. 5).

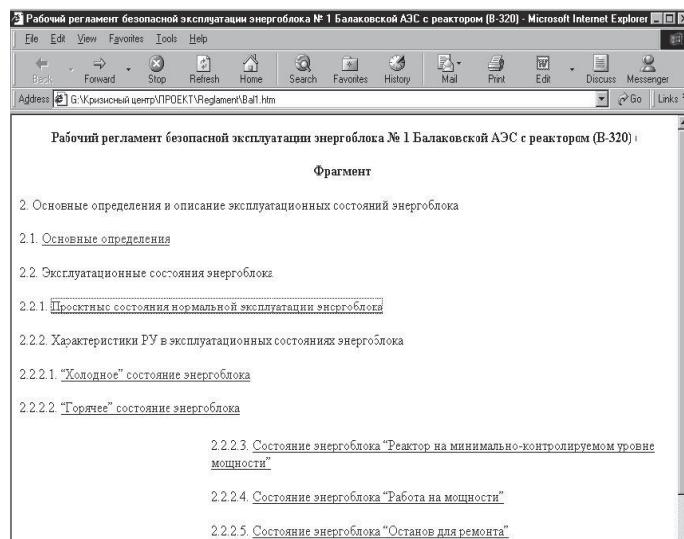


Рис. 5. Представление технологического регламента
в виде гипертекста

В системе предусмотрена возможность отображения графических зависимостей выводимых в специальные временные окна. В качестве таких графических зависимостей реализованы:

- графики зависимости параметра (или параметров) от времени (до 8 параметров на одном графике);
- графики зависимости параметра от параметра;
- гистограммы (до 8 параметров на одной гистограмме);

- графики зависимости скорости изменения параметра от времени (тренды) (до 8 параметров на одном тренде).

Пример графика приведен на рис. 6.

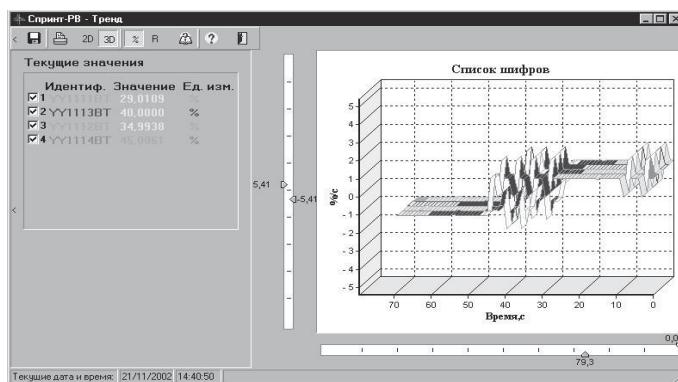


Рис. 6. Вид представления переменных состояния в виде графиков

В интерфейсе пользователя СППР-РВ информация диагностической экспертной системы (ДИЭКС-РВ) доступна через окно групповой сигнализации. Структурно эта информация представляется набором данных, приведенных на рис. 7.

Динамический стек технологических систем (ТС), в нештатном состоянии			
Имя ТС, Время	Имя ТС, Время	Имя ТС, Время	Имя ТС, Время
Имя вызванной ТС, Время возникновения нештатной ситуации			
Описание нештатной ситуации в вызванной технологической системе (ТС) и время возникновения нештатной ситуации (Время)	Диагноз - найденная причина возникновения нештатной ситуации возникшая в вызванной ТС		
	Прогноз последствий от возникшей нештатной ситуации в ТС		
	Рекомендуемые управляющие решения – команды управления		

Рис. 7. Структура информации, генерируемой ДИЭКС-РВ

Заключение

Рассмотрена реализация операторского интерфейса, основанная на методах когнитивной графики как средстве интеллектуальной поддержки принятия решений для оперативного и административного персонала Кризисного центра концерна «Росэнергоатом». Данные работы проводятся в ООО «ТАСМО-БИТ».

СПИСОК ЛИТЕРАТУРЫ

- [**Башлыков, 1994**] Башлыков А.А. Методы когнитивной графики для образного представления состояний энергоблока АЭС. Труды международного форума информатизации. М.: Машиностроение, 1994, с. 92–98.
- [**Башлыков А.А. и др., 2001**] Башлыков А.А. Интеллектуальные системы поддержки принятия решений в нештатных ситуациях с использованием информации о состоянии природной среды. – М.: Издательство «Эдиториал УРСС», 2001, 303 с.
- [**Башлыков А.А. и др., 2003**] Башлыков А.А., Ильинский А.Н., Абраменко В.О., Колесник А.И. «ОКО» – информационная корпоративная система контроля безопасности работы энергоблоков и выполнения графика несения нагрузки по обобщенным параметрам. Приборы. № 9. 2003, с. 22–30.
- [**Вагин и др., 2001**] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Известия РАН. Теория и системы управления, 2001. – № 6. – С. 114–123.
- [**Гришин, 1982**] Гришин В. Г. Образный анализ экспериментальных данных. М.: Наука, 1982.
- [**Зенкин, 1991**] Зенкин А. А. Когнитивная компьютерная графика. М.: Наука, 1991.
- [**Поспелов, 1992**] Поспелов Д.А. Когнитивная графика – окно в новый мир // Программные продукты и системы. 1992. С. 4–6.
- [**Пушкин, 1965**] Пушкин В.Н. Оперативное мышление в больших системах. М.: Энергия, 1965.
- [**Солсо, 1996**] Солсо Р.Л. Когнитивная психология. – М.: Тривола, 1996. – С. 598.
- [**Трахтенгерц, 1992**] Трахтенгерц Э.А. Компьютерная поддержка принятия решений. – М.: СИНТЕГ, 1998, 360 с.

УДК 007:681.512.2

КОГНИТИВНАЯ СЕМИОТИКА: СЕМИОЗИСЫ И ГЕШТАЛЬТЫ

Ю.Р. Валькман (*yur@valkman.kiev.ua*)

Международный научно-учебный центр информационных
технологий и систем НАН и МОН Украины, Киев

Показано, что основными ресурсами в когнитивных семиотических операциях являются гештальты, а базовыми процессами – семиозисы. Исследованы свойства гештальтов. Рассмотрена классификация схем семиозиса. С помощью этих схем производится интерпретация знаков. Исследованы принципы построения схем семиозиса с целью построения формального аппарата для реализации, далее, компьютерных технологий образного мышления.

Введение

Данная работа является продолжением исследований семиотических структур и когнитивных процессов (см., в частности, [Валькман, 2011; 2012]). *Объектом* этих исследований является моделирование когнитивных процессов в вычислительной среде. Здесь *предмет* исследования – гештальты и схемы семиозиса в когнитивных процессах. Общее *направление исследований* – разработка формальных аппаратов для представления гештальтов и схем семиозиса в вычислительной среде. Здесь цель исследования – общий анализ взаимодействия гештальтов и семиозиса.

1. Когнитивные процессы, знаки и семиотика

Понятие «*когнитивные процессы*» чаще применяют к таким процессам, как *память, внимание, восприятие, действие, принятие решений и воображение*. Нам здесь интересны, в большей мере, те процессы, которые можно рассматривать как операции по обработке информации или знаний. Такие процессы должны на входе и выходе иметь некоторые информационные структуры. Эти информационные структуры, очевидно, правомерно назвать когнитивными. *Когнитивные структуры* – инварианты человеческого восприятия, которые отражают содержание и структуру знаний человека о действительном и возможном мирах и определяют их кате-

горизацию и систему опыта человека [Бодалев, 2011]. И очень важно, что когнитивная система человека как иерархическая совокупность *когнитивных структур* позволяет составить представление о мире, опираясь исключительно на внутренние репрезентации.

Когнитивная деятельность поэтому – манипулирование репрезентациями, главной характеристикой которых является их интерпретируемость (символичность).

В любом случае все эти репрезентации имеют знаковую структуру. Известно [Пирс, 1999] высказывание «мы думаем только в знаках». Таким образом, мы приходим к семиотике. Широко известен семиотический треугольник Г. Фреге, К. Огдена, И. Ричардса, С. Ульриха, А. Черча и др., который связывает *S* (знак), *C* (концепт, понятие) и *D* (денотат). Вершине *C* соответствуют трактовка знака *S* и денотата *D* (точнее отношения *S-D* для обеспечения перехода от знака к денотату или наоборот), которая производится в голове интерпретатора.

Но для исследования когнитивных процессов целесообразно (вслед за [Щедровицкий, 1974], [Мельников, 1978] и [Рыков, 2004]) рассматривать четырехугольник. Схема его представлена на рис. 1.

В этом четырехугольнике вершина «концепт» заменяется отношением *S"-D"*, в котором *S"-* образ знака в голове интерпретатора (ментальная модель знака), а *D"-* – ментальный образ денотата. Будем его называть *десигнатом*. Термин заимствован у Ч. У. Морриса, но под ним понимается не «вид поведения» [Моррис, 1983], а *субъективный* образ денотата, субъективные образы любых понятий, представлений и т. д., в т. ч. образы, не имеющие объективных денотатов. Обозначение *"* заимствовано из символики [Мельников, 1978]. Им помечаются все объекты, которые представлены в голове интерпретатора (ментальные модели, по сути).

Очевидно, важнейшим свойством знака является его функция отображения некоторого объекта, т.е. любой знак всегда знак некоторого объекта. И это могут быть объекты *материального мира*, объекты нашего сознания или *идеального мира*, *знаковые структуры*.

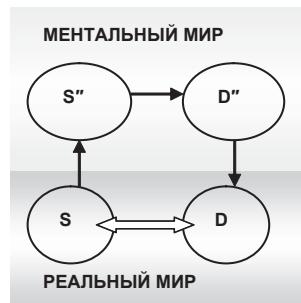


Рис. 1. Отношения «знаки – денотаты» в семиотике

Заметим, один и тот же объект может быть денотатом в одной структуре и знаком в другой.

Известна коммутативная диаграмма Г. П. Мельникова [Мельников, 1978] (см. рис. 1): $S \rightarrow S'' \rightarrow D'' \rightarrow D \Leftrightarrow S$. По сути – это определение знака. Хорошая интерпретация этого определения приведена в [Рыков, 2004].

«Если мы видим (слышим, обоняем, осязаем – далее мы будем пользоваться без ущерба для общности зрительными аналогиями) объект из внешнего мира S , то в нашем сознании возникает его зрительный образ S'' . Если мы видим другой объект – D , то происходит то же самое – возникает образ этого объекта D'' . Если в нашем сознании есть устойчивая связь от S'' к D'' , то может происходить следующее. Мы видим только объект S , активизируется его образ S'' , затем образ D'' . То есть мы не видим объекта D , но его образ D'' активизируется или возникает в нашем сознании! В этом случае мы говорим, что S – это знак, а D – это его денотат. Или, другими словами, знак (S) – это то, что замещает в нашем сознании другой объект (D)» [Рыков, 2004].

Заметим, что основные семиотические процессы $S'' \rightarrow D''$ или $D'' \rightarrow S''$, восприятия знаков $S \rightarrow S''$ или их порождения $S'' \rightarrow S$, восприятия денотатов $D \rightarrow D''$ проходят в ментальном мире (см. рис. 1). Тем самым показано, что *семиотика во многом имеет когнитивный характер*.

2. Разновидности и схемы семиозиса

Пирс мыслил семиозис как постоянное движение знака. Семиозис у него включает в себя две одинаково важные части: производство знаков и их интерпретацию.

Для нашей когнитивной семиотики (см. в [Валькман, 2011; 2012]) такое понимание семиозиса соответствует анализу и синтезу структур образной информации и образных знаний.

Для Пирса именно понятие *семиозиса* было центральным понятием его семиотической теории. По мысли Пирса, знак не функционирует как знак до тех пор, пока он не осмысливается как таковой. Иначе говоря, *знаки должны быть интерпретированы, чтобы быть знаками*. Моррис ввел [Моррис, 1983] следующее определение: Семиозис – «Процесс, в котором нечто функционирует как знак».

Ранее мы уже использовали понятие *интерпретатор*. Теперь определим его более точно. Интерпретатор – то, что (или кто) воспринимает и/или воспроизводит знакососители.

Происходящие в интерпретаторе в случае простейшей знаковой коммуникации между интерпретаторами **A** и **B** [Мельников, 1978] минимально необходимые звенья процесса коммуникации будут (см. рис. 1) [Юцкевич, 2006]:

В интерпретаторе A: Возбуждение десигната D_a'' ; Возбуждение ассоциаций между десигнатором и образами соответствующих знаков ($D_a'' \rightarrow S_a''$); Возбуждение образов знаков S_a'' ; Обратимое отражение (по терминологии Мельникова [Мельников, 1978]), в результате чего во внешней среде появляется физический носитель знаков S_a ;

В интерпретаторе B: Восприятие органами рецепции интерпретатора **B** физических носителей знаков S_a ; Возбуждение образов S_e'' носителей знаков S_a ; Возбуждение ассоциаций между образами знаков S_e'' и соответствующим десигнатором D_e'' ; Возбуждение десигната D_e'' .

Сема (семиозиса) общения (см. рис. 1): $D_a'' \rightarrow S_a'' \rightarrow S_a \rightarrow S_e'' \rightarrow D_e''$.

Обратим внимание на начало и конец этой цепи, соответственно,

D_a'' – «мысли интерпретатора-передатчика сообщения» и

D_e'' – «мысли интерпретатора-приемника сообщения».

Изэстетические (греческий термин «из (о)» – одинаковый, «эстетикос»— чувствующий) *интерпретаторы* – такие, у которых возбужденные от одного и того же знака (комбинации знаков) десигнаторы подобны, изоморфны или хотя бы гомоморфны [Мельников, 1978].

Обратим внимание, что каждый из интерпретаторов, участвующих в любой знаковой коммуникации, чаще всего участвует в каждый момент времени только одной своей половиной: либо воспроизводит знаки, либо воспринимает их. Соответственно этому целесообразно выделить два частных вида семиозиса, в каждом из которых участвуют разные половинки интерпретаторов: *моторный семиозис* – процесс от возбуждения десигната до генерации знака (знаков) и *сенсорный семиозис* – процесс от восприятия знака (знаков) до возбуждения десигната.

Указанные виды семиозиса сформулированы относительно «внешней» (по отношению к мозгу субъекта) функциональности:

производства и восприятия знаков. Характеризуя указанные виды семиозиса относительно «внутренней» (по отношению к мозгу) функциональности, необходимо признать, что эти виды семиозиса всегда (в той или иной степени) сопровождаются психической деятельностью. Поэтому имеет смысл выделить два частных вида психосемиозиса, соответственно связанных с мотосемиозисом и сенсосемиозисом: *психомотосемиозис* и *психосенсесемиозис* [Юцкевич, 2006].

Интересная классификация семиотик по критерию вида синтеза знаковых структур предлагается в [Тарасов, 2002].

На рис. 2 представлена (см. в [Бел и др., 1996]) обобщенная схема примера различных интерпретаций одного и того же образа различными интерпретантами. Натюрморт, изображающий вазу с фруктами, помимо прочего есть знак, или репрезентант. У рассматривающих его зрителей этот натюрморт с чем-то ассоциируется. Это что-то – ментальный образ – и есть *интерпретант*. Интерпретант с чем-то соотносится, указывает на какой-то иной предмет. При этом у каждого воспринимающего – свой предмет, с которым связан ментальный образ натюрморта: у одного – это *реальные фрукты*; у другого – *живопись*; у третьего – *масса денег, которых он стоит*; у четвертого – «*Дания... XVII век*» и т.д. Схемы семиозиса: $S \rightarrow S'' \rightarrow D_1'', S \rightarrow S'' \rightarrow D_2'', S \rightarrow S'' \rightarrow D_3'', S \rightarrow S'' \rightarrow D_4''$. Это пример неизэстетичности.

Итак, объект, или референт, принципиально субъективен и зависит от воспринимающего. Широко известно определение знака, данное Пирсом (см. в [Бел и др., 1996]): «*Знак, или репрезентант, есть нечто, что соотнесено с иным в каком-либо отношении или по какому-либо признаку*». Знак адресован, то есть создает в уме человека эквивалентный или, возможно, более сложный знак. Последний есть *интерпретант первого*. Знак замещает нечто, а именно свой объект. Он замещает его не во всех отношениях, но в соотнесении с той идеей, которую я называю *базисом, или основанием знака*».

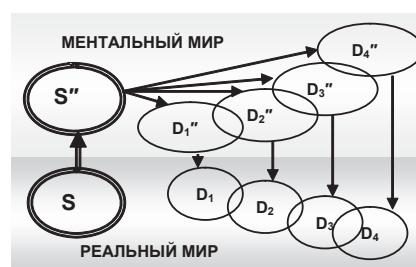


Рис. 2. Многозначность смыслов – субъективность десигнатов

На рис. 3 представлен пример многозначности субъективных десигнатов одного денотата. Рассмотрим в роли денотата *самолет*, а в качестве *десигнаторов* – его отражения (представления) у стюардессы, пилота, пассажира и диспетчера. Естественно, каждому десигнату будет соответствовать свой образ знака, и каждому образу знака свой знак. Объединение знаков в некоторой степени будет способствовать более полному представлению денотата. Схемы соответствующего семиозиса: $D \rightarrow D_1'' \rightarrow S_1'' \rightarrow S_1$, $D \rightarrow D_2'' \rightarrow S_2'' \rightarrow S_2$, $D \rightarrow D_3'' \rightarrow S_3'' \rightarrow S_3$, $D \rightarrow D_4'' \rightarrow S_4'' \rightarrow S_4$. Однако схемы семиозиса могут быть намного сложнее. Так как каждый знак производит интерпретанту, которая, в свою очередь, является репрезентантом следующего знака, процесс семиозиса оказывается «*последовательным рядом интерпретант» ad infinitum* [Пирс, 1999]. В бесконечном процессе семиозиса ($\dots \rightarrow S_i'' \rightarrow D_i'' \rightarrow S_{i+1}'' \rightarrow \dots \rightarrow S_i'' \rightarrow D_i'' \rightarrow \dots$) нет *первого* и нет *последнего* знака.

Однако мысль о бесконечном семиозисе не означает отсутствия семиотических ориентиров, но отсылает к актуальному взгляду, согласно которому «*мышление всегда осуществляется в форме диалога – диалога между различными фазами „я“*», так что мышление, поскольку оно диалогично, в сущности, состоит из знаков» [Пирс, 1999]. Так как каждая мысль должна быть обращена к другой мысли, непрерывный процесс семиозиса (или мышления) может быть только прерван, но не может быть окончен. Заметим, что когнитивная семиотика в такой интерпретации исследует только процессы возбуждения образов, их отражения, трансформации и интерпретации. Но существует еще много когнитивных операций (см., например, в [Рыков, 2004]).

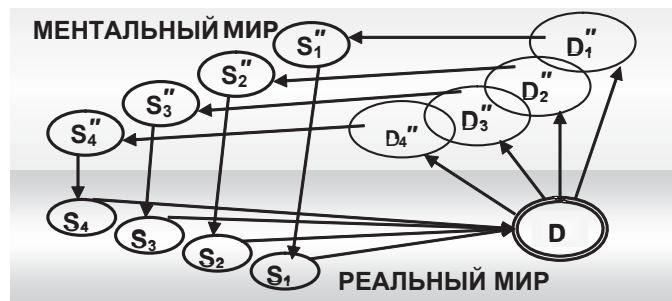


Рис. 3. Многозначность субъективных десигнатов одного денотата

С нашей точки зрения, схемы различных видов семиозиса являются основой разработки моделей синтеза и анализа образной информации.

3. Гештальты в семиозисе

Использование понятия *гештальта* [Вертгеймер, 1975] чрезвычайно важно в семиозисе и вообще в когнитивных процессах. Тот внутренний врожденный (или встроенный в автомат) *интенциальный образ*, который позволяет определять определенные внешние объекты (знаки *S* и денотаты *D*) без предварительного обучения, а только благодаря правильной работе рецепторов, назовем *априорным гештальтом* [Мельников, 1978]. Их наличие позволяет не только обеспечивать отнесение внешних отражаемых объектов к определенному множеству, но и ускорять процесс выработки *новых интенциальных образов* для различения подмножеств этого множества и даже индивидуальных его представителей. Такие образы можно было бы назвать *апостериорными гештальтами* [Мельников, 1978].

Без априорных гештальтов невозможно замыкание цепочек семиозиса. Естественно предположить, что всегда **любой акт опознания (в семиозисе) должен завершаться возбуждением априорного гештальта** под влиянием возникновения образов опознаваемого объекта.

Очевидна соотнесенность априорного гештальта с конкретными рецепторами, например, обоняние, слух, звук и т.д. Естественно, что априорный гештальт, представляя отражательный опыт вида субъекта, не имеет прообраза в виде конкретного объекта, т. е. уникального объекта в уникальных обстоятельствах (такие объекты называются *окказиональными* – «случайными»). Являясь обобщенным и не возникающим, а лишь воспроизводимым для определенного функционального использования, априорный гештальт должен быть отнесен к числу *узуальных* («общепринятых») *образов*. Однако среди *окказиональных* образов, а также обобщенных знаков, ставших *узуальными* образами, выработанными в практике самого интерпретатора и поэтому уже не априорными, а *апостериорными*, априорные гештальты остаются полноправными членами во всех отношениях. Как известно, гештальт не делится на части. Но можно ли составлять из отдельных гештальтов новый гештальт?

Содержание восприятия *гештальтно*; оно таково изначально, а не в результате акта соединения отдельных ощущений. Однако в

восприятие любого предмета входит многое такое, что само может быть предметом самостоятельного восприятия. Например, слово, которое мы сейчас читаем. В качестве слова оно является *отдельным гештальтом*. Однако очевидно и то, что в то же время мы можем воспринять и отдельные буквы; *в этом случае каждая из них также будет представлять собой предмет отдельного восприятия и иметь собственный гештальт*. Другой пример, очевидно, *дерево* является гештальтом; но *яблоко*, которое висит на ветке, как таковое, дано в содержании восприятия в виде гештальта. *Одним словом, предметы как целостные гештальты состоят из частей, в свою очередь также представляющих собой отдельные гештальты* [Вертгеймер, 1975].

В [Кузнецов, 2012] показано, что гештальт занимает *центральное место* в когнитивных процессах мозга.

В [Хокинс и др., 2007] предложена модель «память – предсказание» в качестве модели естественного интеллекта. Хокинс приводит особенности памяти неокортекса. Неокортекс: *запоминает* последовательности элементов, а не отдельные элементы окружающего мира; *вспоминает* последовательности автоассоциативно; *запоминает* последовательности в инвариантной форме; *сохраняет* последовательности иерархически. С нашей точки зрения – это свойства гештальтов или технологий их синтеза/анализа.

Заключение

Семиотика и семиозис во многом имеют *когнитивный* характер. Основными ресурсами в когнитивных семиотических операциях являются *гештальты*, а базовыми процессами – *семиозисы*. Знаки должны быть интерпретированы, чтобы быть знаками. А значит, в семиозисе должен участвовать *интерпретатор* (человек или компьютер в будущем). Следовательно, все операции распознавания знаков (отображение их в значение) являются когнитивными. Рассмотрена классификация знаков с ориентацией на построение схем семиозиса. Именно с помощью этих схем производится трактовка знаков. Отношения между компонентами этих схем строятся в основном с помощью анализа ассоциаций свойств компонентов. Исследованы принципы построения схем семиозиса с целью построения формального аппарата для реализации, далее, компьютерных технологий образного мышления. Начат процесс классификации этих схем. Рассмотрены понятия априорных и апо-

стериорных гештальтов. Показано, что они являются необходимым атрибутом семиозиса. Начато исследование свойств гештальтов и их классификация. Полагаем, что это направление исследований должно обеспечить возможность разработки формального аппарата для моделирования образного мышления в компьютерных технологиях.

СПИСОК ЛИТЕРАТУРЫ

- [Бел и др., 1996] Бел М., Брайсен Н. Семиотика и искусствознание http://www.philosophy.ru/library/aesthetics/bal_bryson_semiotics_art.pdf.
- [Бодалев, 2011] Психология общения. Энциклопедический словарь. Под общ. ред. А.А. Бодалева. – М.: Изд-во «Когито-Центр», 2011.
- [Валькман, 2011] Валькман Ю. Р. Когнитивная семиотика в моделировании образного мышления // Моделювання та інформаційні технології, Збірник наукових праць ІІІМЕ, Київ, 2011, Випуск 62 с. 71–84
- [Валькман, 2012] Валькман Ю. Р. Когнитивная семиотика // Сб. научн. тр. 1 Междунар. симпозиума «Гибридные и синергетические интеллектуальные системы: теория и практика». Калининград: Физматлит, май, 2012, с. 98–116.
- [Вертгеймер, 1975] Вертгеймер М. Продуктивное мышление. М., 1987.
- [Кузнецов, 2012] Кузнецов О.П. Когнитивная семантика и искусственный интеллект// Искусственный интеллект и принятие решений, 2012, №4, с. 32–42.
- [Мельников, 1978] Мельников Г. П. Системология и языковые аспекты кибернетики. М.: Советское радио, 1978.
- [Моррис, 1983] Моррис Ч. У. Основания теории знаков // Семиотика, М.: Радуга, 1983.
- [Пирс, 1999] Пирс Ч.С. Логика как семиотика: теория знаков // Метафизические исследования. Вып. 11. Язык. СПб, 1999. С. 199–217.
- [Рыков, 2004] Рыков В. В. Извлечение знаний – реализация когнитивной семиотики. <http://tykov-cs.narod.ru/dlg9.html>.
- [Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – Эдиториал УРСС, 2002. – 352 с.
- [Хокинс и др., 2007] Хокинс Д., Блейксли С. Об интеллекте. М.: Изд. Дом. Вильямс, 2007.
- [Щедровицкий, 1974] Щедровицкий Г.П. Смысл и значение // Проблемы семиотики. М., 1974. С. 76–111.
- [Юцкевич, 2006] Юцкевич О. А. Основные виды семиозиса. <http://www.cit.bsu.by/site1/ovs.htm>.

УДК 007.5:510.66:159.955.5

ВИЗУАЛЬНЫЕ ОБРАЗЫ-СИМВОЛЫ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Б.А. Кобринский (*bakob@pedklin.ru*)
ГБОУ ВПО «Российский национальный исследовательский
медицинский университет
им. Н.И. Пирогова» Минздрава России, Москва

В работе описываются принципы перехода к использованию визуальных образов в интеллектуальных системах. Рассматривается применение образных рядов различного вида (нечетких, временных и других) в базах знаний на примере медицинских диагностических систем.

Введение

Вне-словесны (по Ф. Розинеру [1993]) музыка и живопись, изложение которых берут на себя музыко- и искусствоведы. То же касается и фотографий, особенно художественных. Хотя иногда и поясняющие слова не требуются. Так, при словах «Картина Репина «Отказ от исповеди» сразу возникает образ сидящего узника и стоящего около него священнослужителя, если же картина не известна, то более общий визуальный образ включает камеру с фигурами приговоренного и служителя церкви с крестом. Но обратимся к еще одной, на первый взгляд не интуитивно-образной области уже естественных наук – к медицине. Интуитивные и образные представления позволяют высококвалифицированному врачу, имеющему большой опыт наблюдения больных и диагностики их заболеваний, строить первичную гипотезу на основе анализа внешнего вида и поведения пациента [Кобринский, 1996]. Зачастую и окончательный диагноз включает образные характеристики патологических проявлений. Этому соответствует у А.Ф. Лосева [1993] метафорическое и интуитивное толкование «эйдоса» (облик, образ) и «идеи» как лика или понятия. При обращении же к средневековой и современной философии мы обнаруживаем, что это категориальная структура, интерпретирующая исходную семантику какого-либо понятия. Это позволяет говорить в отношении и верbalных, и образных характеристик как о символах, используемых, в частности, в интеллектуальных системах. При этом нужно

иметь в виду, что одинаковые названия могут иметь разное смысловое содержание, включая и визуальные образы. Примером могут служить творения художников. Например, «Красный уголок» конструктивиста Родченко был частью реального пространства и, одновременно, организационной единицей. Нельзя сбрасывать со счетов и оптические иллюзии, т.е. неоднозначную классификацию зрительных впечатлений (невольничий рынок с невидимым/видимым бюстом Вольтера на картине Дали). Оптические иллюзии могут быть связаны с влиянием цветового контраста, освещенности окружающих предметов, движения объекта и др.

Отдельная область «мифотворчество» у символистов и метафорический образ у имажинистов. Как пишет Д.А. Поспелов [2007], «художник Петр Фатеев был интуитивистом. Его созерцание окружающего мира было лишь толчком к той внутренней работе, которая потом материализовалась в его произведениях». Его картины таковы, что в них реальность, опирающаяся на наши конкретные впечатления, раскрывает недоступное глазу и воздействует прямо на сознание. Как представляется, под недоступным глазу здесь следует понимать нечто, что смотрящий как бы не видит, но ощущает его воздействие. Сродни этому и диагностическая работа врача-интуитивиста, взгляд которого обнаруживает нечто, порождающее гипотезу о возможном диагнозе.

Дефрагментация образа с последующей реинтеграцией неприемлема, так как при этом искажается «взаимная функциональная зависимость элементов» [Лифшиц, 2002]. Поэтому врачами используются различные интегральные понятия (например, грубые или гаргоилоподобные черты лица). Когда мы говорим о гаргоилоподобных чертах лица, пишет Камю [1997], «ясность оборачивается метафорой», что характерно и для разных других проявлений при заболевании. Опытный врач может быстро определить направление диагностического поиска, взглянув на больного. Здесь уместно высказывание Гарсия Лорки, что метафора связует противолежащие миры одним скачком воображения [Гарсия Лорка, 1986]. Если обратиться к сообществу агентов, то можно сказать, что «рассудочный» (deliberative) агент позволяет принимать решения на основе символьных рассуждений, например, метода сравнения по образцу [Wooldridge, Jennings, 1995], в качестве которого можно рассма-

тровать образ или тождественную метафору. Область визуальных образных представлений до сих пор не нашла своего отражения в базах знаний интеллектуальных систем.

1. Образы в медицинской диагностике

Эффект озарения – объяснение формирования образа болезни «по наитию», без подкрепления дополнительными фактами – это удел немногих очень опытных врачей. Но формирование гипотез, пусть даже вызывающих сомнения, может способствовать движению к истинному диагнозу, построению дифференциально-диагностического ряда, уточнению прогноза течения болезни. Восникающие у врача воспоминания носят часто размытый характер, но здесь важна последовательность сходных или дополняющих друг друга образов – визуальный образный ряд [Кобринский, 2008]. Врач часто не помнит фамилии и имени больного, но у него сохраняются в памяти клинические проявления патологического процесса, диагноз и «ответы» организма на применявшиеся медикаменты. При анализе течения болезни врачи встречаются также с динамикой проявлений, которые могут иметь последовательность визуальных образов, т.е. представлять собой временной образный ряд – кортеж образов, зависящих от изменений клинической картины болезни, в особенности при прогредиентно текущих заболеваниях (нарастающих ранее и возникающих новых симптомах).

Образно-эмоциональная система субъективна, в отличие от логико-лингвистической, являющейся объективной. Но именно эмоциональная система преобладает у больных, особенно мнительных, излагающих врачу свои жалобы. В ответ у врача возникает осознанная рецепция как элемент психики вследствие традиционной мыслительной деятельности и неосознанная рецепция образно-интуитивных представлений, которые, как отмечено выше, могут быть представлены рядом взаимодополняющихся образов разного порядка. Мысленные образы, иногда довольно размытые, при встрече с новым случаем заболевания запускают определенные ассоциации и позволяют сформировать дифференциальный ряд, включающий те или иные образы. Наиболее выражено это у врачей с правополушарным мышлением, хотя и не исключает активности логических аспектов левого полушария. Не следует забывать здесь и о рефлексивности, присутствующей как у пациентов, так и у врачей.

Значительный интерес представляет рассмотрение специфической ситуации, в которой визуальные характеристики и мимика больного имеют принципиальное значение для решения диагностической задачи. В этом смысле интересным примером является известная автору статьи работа русскоговорящего врача-психиатра, на первом этапе его работы в Израиле принимающего пациентов, говорящих на иврите, при том, что он лишь частично владел другими языками, а пациенты не владели русским языком. Примером этого может служить известный автору опыт русскоговорящего врача-психиатра. С одной стороны, то, что врач-психиатр видит и что позволяет ему принять диагностическое решение, далеко не всегда удается отобразить в записи так, чтобы можно было говорить о полном соответствии. С другой стороны, большое значение имеют поведение и внешний вид пациента, излагающего свои жалобы. Их понимание не зависит в полной мере от знания языка, на котором говорит больной, т.е. от полноты излагаемых пациентом данных. При этом записи «беседы» делались на русском языке. Этот феномен позволяет уверенно подтвердить значение образно-интуитивных представлений в процессе общения «врач – пациент». Хотя это и нельзя распространять на все классы болезней, а также на врачей, не имеющих достаточного опыта в данной предметной области. Однако большая предшествующая практика позволяет ассоциировать вербализуемые и образно-эмоциональные признаки.

В этом смысле можно привести слова врача и писателя В.В. Вересаева [1961. Записки врача // Собрание сочинений. М.: Б-ка «Огонек»; Правда, т.1, с.337)], который считал, будучи молодым врачом, что настоящий врач мог бы «поставить правильный диагноз: его совершенно особенная творческая наблюдательность уцепилась бы за массу неуловимых признаков, которые ускользнули от меня, бессознательным вдохновением он возмestил бы отсутствие ясных симптомов и почувял бы то, чего не в силах познать».

2. Образные ряды и их формализация

Образные ряды фенотипических проявлений болезни могут служить основой для визуальной фенотипической диагностики (если можно так выразиться, преддиагностики заболеваний) заболеваний. Включение в образный ряд может осуществляться на основе различных принципов:

- по сходству – близкие лица, объекты, ситуации в том числе по отдельным фрагментам;
- по метафорическому отражению – гаргоилы и синдром Хурлер, колокольчик и «похоронный звон колокольчиков», Мюнхгаузен и определенный тип бреда;
- по ассоциации.

Образы одного визуального ряда, имеющие те или иные отличия друг от друга, представляют собой как бы условно нечеткий образ типичного представителя ряда. Рассматривая понятие образного ряда, нужно понимать его как образный квазиконтинуум близких или относительно близких изображений, что предполагает нечеткость переходов между отдельными представителями. Наиболее типичный представитель ряда, представляющий собой образный архетип, – это или наиболее ранний из образов, или наиболее известный. Таким путем достигается упорядоченность рядов (в частности временных). Используя понятие образов на оппозиционных шкалах, можно говорить о дискриминируемых рядах образов.

В медицине члены образного ряда демонстрируют различные варианты фенотипического сходства и различия в клинической картине заболевания, в том числе наблюдавшиеся специалистами в разных странах, т.е. у людей различающихся этносов и в различные возрастные периоды, соответствующие разным фазам болезни.

Использование образных рядов может позволить ставить и решать задачи поиска характерных объединяющих особенностей среди образов одного типа (например, лица больных с одним заболеванием или лицо человека в процессе развития заболевания, т.е. временной образный ряд).

Аналогично тому, как по фрагменту виденной ранее картины (кинофильма, проявления болезни) может восстанавливаться целостная картина (иногда характерная ее часть) или возникать ряд взаимодополняющих, по ассоциации, изображений, также и в медицине определенное изображение как ключевое слово может иногда позволить врачу сформулировать гипотезу диагноза.

В рамках модели денотативного пространства [Нариньяни, 2003] отдельные визуальные изображения представляют собой денотаты «подлинника» в образном ряде, которые могут соответствовать отдельным денотатам отражения в когнитивном процессе. Денотаты в образных рядах могут быть поставлены в соответствие

метаденотатам, с каждым из которых соотносится класс, т.е. открытое множество наблюдавшихся конкретных или виртуальных (мифических) образных денотатов, выделенных в данный класс по принципу определенного сходства. Метаденотатом в медицине можно считать маски заболеваний, маркирующих определенные ряды болезней, например, острые хирургические и инфекционные болезни брюшной полости.

3. Образно-лингвистическая интеллектуальная система

Включение в базы знаний, наряду с логико-лингвистическими понятиями, визуальных образов (рисунков, фотографий), рассматриваемых как символы, может позволить перейти к отражению невербализуемых характеристик и использовать их в системе рассуждений, логико-аргументационных построений и принятии решений.

Условие выдвижения гипотезы об «образе» в базе знаний интеллектуальной системы должно осуществляться в интерактивном режиме, на каждом шаге вновь выдвигаемой на основе лингвистических характеристик гипотезы и сопровождаться указанием эксперта о степени его уверенности в отношении наблюданного образа или образного ряда. Результатом может быть подтверждение выдвинутой гипотезы или ее отклонение. Т.е. визуальный ряд включается в качестве второй составляющей, наряду с лингвистической компонентой, в традиционные семантические базы знаний, которые в этом случае будут базироваться на двух символьных представлениях – лингвистических и иконических, для формализации которых могут быть использованы фреймовые структуры [Кобринский, 2008].

Визуальные образные ряды можно было бы строить по типу фреймоподобной структуры, где фрейму соответствует традиционное представление определенного образа (или типичный представитель образного ряда, своего рода архетип или родительский фрейм), а слоты представлены образами данного типа («индивидуумами» или элементами образного кортежа), отличающимися по отдельным невербализуемым или трудно вербализуемым характеристикам (к примеру, особенностям отдельных проявлений болезни, в том числе в разные периоды заболевания). Контекст во

фреймах может пояснить (сопровождать) визуальные образы, но и изображения ряда могут быть контекстом по отношению к семантическим представлениям. В этом случае, при вводе определенных признаков, «демон» должен будет приводить соответствующий фрейм с изображениями (образами) в активное состояние с последующей демонстрацией его врачу-пользователю. То есть в процессе формирования промежуточных результатов поиска, на каждой стадии логического вывода, после ввода определенной порции признаков, запуска демона и активизации традиционного фрейма должен активизироваться связанный с ним образный фрейм, который и представляется врачу-пользователю, который подтверждает выдвинутую гипотезу или отвергает ее. Таким образом, в интеллектуальной системе образных рядов должны будут включаться, с точки зрения оценки их полезности экспертом / пользователем, в процесс разрешения конфликтов при активизации более одного правила в сети фреймов. Следовательно, у пользователя будет использоваться традиционный подход к воспроизведению из памяти зрительных образов при предъявлении видеорядов интеллектуальной системой.

Реализация интеллектуальной системы на фреймах может включать решатель продукционного типа.

Заключение

Реализация лингвообразного подхода предполагает переход к созданию баз знаний на основе символов двух типов – лингвистических и иконических, последние из которых должны опираться на концепцию образных рядов [Кобринский, 2008]. В естественных науках особый интерес данный подход представляет для медицинской диагностики, где характерные внешние проявления могут служить маркерами конкретных заболеваний.

В развитии данного представления лингвистическая и образная компоненты – разнотипные формализованные символные характеристики, должны сделаться взаимодополняющими составляющими в базах знаний. Их представление во фреймовой форме можно рассматривать как один из потенциально возможных вариантов решения для логико-лингво-иконических интеллектуальных систем.

Методология сочетанных лингво-фреймовых структур в системах искусственного интеллекта может найти применение в естествознании и гуманитарных науках.

СПИСОК ЛИТЕРАТУРЫ

- [Розинер, 1993] Розинер Ф. Искусство Чюрлёниса. – М.: ТЕПРА – TERRA, 1993.
- [Кобринский, 1996] Кобринский Б.А. К вопросу учета образного мышления и интуиции в экспертных медицинских системах // V Национальная конференция с международным участием «Искусственный интеллект-96»: Сб. науч. тр. Т.2. – М.: Физматлит, 1996.
- [Лосев, 1993] Лосев А.Ф. Очерки античного символизма и мифологии. – М.: Мысль, 1993.
- [Поспелов, 2007] Поспелов Д.А. Амаравелла: мистическая живопись Петра Фатеева. – М.: Фантом Пресс, 2007.
- [Лифшиц, 2002] Лифшиц Б. Полугораглазый стрелец. – М.: Захаров, 2002.
- [Камю, 1997] Камю А. Миф о Сизифе // Сочинения. – Харьков: Фолио, 1997. – Т.2.
- [Гарсиа Лорка, 1986] Гарсиа Лорка Ф. Поэтический образ дона Луиса де Гонгоры // Избранные произведения. Т.1. – М.: Худ. лит., 1986.
- [Wooldridge et al., 1995] Wooldridge M., Jennings N. Intelligent Agents – Theory and Practice // Knowledge Engineering Review. – 1995. – Vol.10, No.2.
- [Кобринский, 2008] Кобринский Б.А. Образные ряды и их отображение в базе знаний // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008: Тр. конф. Т.1. – М.: Ленанд, 2008.
- [Вересаев, 1961] Вересаев В.В. Записки врача // Собрание сочинений. – Т.1. – М.: Б-ка «Огонек»; Правда. – 1961.
- [Нариньяни, 2003] Нариньяни А.С. Не-факторы: Неоднозначность (доформальное исследование) (1-я часть) // Новости искусственного интеллекта. – 2003. – №5 (59).

УДК 159.9.07

**РАЗВИТИЕ МЕТОДОЛОГИЧЕСКИХ ОСНОВ
КОГНИТИВНЫХ ИССЛЕДОВАНИЙ ВИРТУАЛЬНОЙ
КОММУНИКАЦИИ¹**

Ю.М. Кузнецова (*kuzjum@yandex.ru*)

Н.В. Чудова (*nchudova@gmail.com*)

Институт системного анализа РАН, Москва

В работе проанализирована возможность реконструкции картины мира субъекта интернет-коммуникации с помощью методов психолингвистики и методов искусственного интеллекта.

Искусственный интеллект развивается в рамках междисциплинарного научного направления, получившего название «Когнитивная наука». Впервые центр когнитивных исследований был создан в Гарварде в 1960 г. Наряду с искусственным интеллектом в семейство когнитивных наук входят психология, лингвистика, нейрофизиология, философия познания и ряд других дисциплин. Как признается многими исследователями, «локомотивом когнитивного содружества, конечно, является теория искусственного интеллекта – ведь за ней стоят стремительно развивающиеся отрасли промышленности, связанные с производством компьютеров и электроники, развитием сети телекоммуникаций. Поэтому в когнитологии доминирует технологический подход к изучению знаний, а критерием качества когнитивных теорий является практическая реализация» [Плотинский, 2001]. За полувековую историю совместного развития когнитивных наук их взаимодействие выстраивалось в самые разные конфигурации. «Первоначально объединение дисциплин в когнитивную науку происходило под флагом компьютерной метафоры и вычислительного подхода, в рамках которого всякая теория должна была в явном виде строиться как математическая модель или компьютерная программа» [Величковский, 2006]. В настоящее время новой точкой роста становится коммуникативная метафора и развитие когнитивных исследований будет

¹ Работа выполнена при финансовой поддержке РФФИ (проект №12-07-00611a).

связано с работами, в которых в центре внимания окажется субъект когнитивных, информационных и социальных процессов. Как подчеркивает Б.М. Величковский, руководитель НБИКС-центра Курчатовского института, «новым концептуальным моментом стало недавнее добавление к этой аббревиатуре буквы «С» – социо-. Возникновение «социального человека» связано с появлением рефлексивных форм сознания, т.е. таких актов сознания, в которых человек обнаруживает способность к самопознанию. С другой стороны, связь социального и когнитивного дополняется столь же очевидной его связью с информационными процессами, т.е. с процессами коммуникации, лежащими в основе работы любого социального организма. [Величковский Б.М., 2013].

На наш взгляд, одним из перспективных направлений работ, в котором когнитивные исследования могут продвинуться уже в ближайшее время, является моделирование средствами ИИ картины мира, объективированной в продуктах речемыслительной деятельности участников виртуальной коммуникации. Предметом таких работ могут стать социолингвистические аспекты взаимодействия субъектов конструирования социальной реальности, опосредованного интернет-технологиями, и методы их исследования.

Рассмотрим имеющиеся на данный момент теоретические и эмпирические предпосылки таких исследований.

В рамках когнитивного подхода психические процессы рассматриваются как процессы моделирования реальности. Исследуя такие «построенные психикой» модели, когнитивная психология одновременно получает и характеристики различных моделей и открывает для себя возможность описания механизмов их построения. В гуманитарных науках, и в том числе в psychology, для описания результата процессов моделирования реальности существует понятие «картина мира». В отечественной psychology А.Н. Леонтьевым [Леонтьев, 1975] введено представление о существовании образа Мира как интегрального образования всей познавательной сферы, которое, имея амодальный характер и будучи аффективно насыщенным, задает для частных предметных гипотез, порождаемых картиной мира, мотивационно-смысловый контекст. Существование двух теоретических конструктов, один из которых позволяет изучать проявления процессов моделирования в конкретном материале («перцептивный мир» по Е.Ю.Артемьевой

[Артемьева, 1999]), а другой позволяет связать предметные представления с аффективно-смысловой сферой личности, дает возможность ставить задачу исследования моделирующей функции психики в рамках подхода, получившего в мировой психологии название «аффективно-когнитивный подход».

Появление сети Интернет и развитие ее социально-ориентированных сегментов (социальных сетей, электронных СМИ, инструментов сбора мнений, инструментов выражения коллективной позиции и проч.) создали беспрецедентный по объему материала и числу участников архив «протоколов» для психологического исследования. Впервые появилась возможность анализировать одновременно и индивидуальную картину мира, проявленную в текстах, порожденных автором на протяжении значительного временного промежутка (достаточного для того, чтобы можно было экстрагировать из текстов мотивационную и ценностную ориентации автора), и групповую картину мира, возникающую в виртуальной коммуникации и проявляющуюся в текстах «единомышленников», группирующихся в ходе обсуждения определенной темы. Отметим, что субъектом виртуальной коммуникации выступают общности, обладающие новыми с точки зрения методологии исследования свойствами. Отраженная, точнее, смоделированная реальность становится сферой существования и деятельности людей, практически независимой от реальности «офф-лайн». При этом т.н. «легкая социальность», формируемая коммуникацией в Сети, обеспечивает принадлежность одного автора множеству групп и саму подвижность этих групп – как по числу и составу участников, так и по количеству тем, в обсуждение которых включены отдельные члены групп. Кроме того, «он-лайн» обсуждения позволяют наблюдать процесс формирования мнений и закрепления их на уровне коллективной картины мира – в отличие от процедуры социологического анкетирования здесь мы имеем дело не с уже готовой позицией анонимного респондента, а с ее становлением и оформлением «коллективного мнения» тех, чьи представления и чьи переживания нам уже могут быть известны из ретроспективы авторских текстов и выраженных сетевыми средствами предпочтений.

Можно констатировать, что назрела необходимость формирования научных представлений о сущности, проявлениях и разви-

тии новых форм взаимодействия человека с реальностью – форм, возникающих в процессе усложнения механизма опосредствования такого взаимодействия, встраивания в него новых звеньев в виде постоянно обновляемых высокотехнологичных устройств, возникновения существенно отличающейся от традиционной среды жизнедеятельности человека. Отметим, что при этом трансформируется не только реальность, но изменяются также и условия и средства ее психического отражения, что составляет объективное условие для возникновения ранее не наблюдавшихся видов картин мира. Очевидно, что для изучения процессов семиозиса, опосредованного интернет-технологиями, необходимы исследовательские средства, позволяющие учитывать его специфику.

Итак, в совместных междисциплинарных работах психологов, лингвистов и специалистов по ИИ может быть поставлена задача разработки методологии и создания конкретных методов изучения картины мира, формирующейся в виртуальной коммуникации.

Одним из важных современных трендов развития искусственного интеллекта является усовершенствование методов анализа текста, способов извлечения из текста имплицитной информации, в том числе выявление различного рода пресуппозиций, а также описание особенностей мотивации, когнитивной и эмоциональной сфер автора высказывания. Специалисты в сфере интернет-технологий активно занимаются разработкой средств, которые позволяют проводить оперативный мониторинг динамичной картины отношения участников виртуальной коммуникации к различным предметам и событиям, отражаемого в публикуемых текстах. С другой стороны, в таких направлениях научных исследований, как когнитивная лингвистика, психолингвистика, лингвосоциология, лингвокультурология, на материале «традиционных» форм коммуникации выработаны подходы, позволяющие описывать и исследовать языковую, концептуальную, эмоциональную картины мира, носителями которых являются как отдельные индивиды, так и общности различного масштаба.

Анализ существующих на данный момент методов, используемых для автоматической оценки в режиме реального времени материалов электронных СМИ, дискуссионных форумов, сообщений блогеров и т.д., представлен нами в [Девяткин и др., 2014]. В качестве общей характеристики развития данной сферы знания вы-

ступает направленность на создание средств, позволяющих переходить от фиксации эмоционального отношения авторов сообщений, которое определяется в рамках оппозиции «положительное – отрицательное», к описанию различных оттенков аффективного содержания текста («сентимента») с использованием все более дифференцированной системы категорий. Перед компьютерной психолингвистикой открываются перспективы изучения не только речевых проявлений эмоциональной оценки, но и причин ее возникновения и динамики. Благодаря этому оказывается принципиально возможной постановка задачи такого рода анализа больших массивов текстов, в результате которого выявляются психологические по своему содержанию закономерности.

Однако, как показывает практика, на настоящий момент времени усилия специалистов направлены на решение частных вопросов – либо маркетинговых (изучение отношения к продукту/производителю), либо социально-прикладных (изучение отношения к персоне/событию). Методов, позволяющих решать задачу изучения картины мира, объективированной в текстах виртуальной коммуникации, на данный момент в мировой науке не существует. Задачи же развития психологических знаний и теоретических знаний в области ИИ требуют создания именно таких методов. В основу развития когнитивных методов изучения картины мира должно быть положено представление об отражении в формализуемых показателях текста картины мира индивидуального и группового субъекта.

Текст, являясь продуктом речевой деятельности человека, объективирует всю совокупность индивидуально-психологических особенностей ее субъекта, в силу чего последние становятся доступными для наблюдения и изучения. С позиций традиционной лингвистики отражающими характеристиками субъекта выступают такие параметры, как тематическое содержание речи, дискурсивные стратегии, средства и способы выражения. Подход с позиций компьютерной лингвистики к диагностике субъективных особенностей, проявляющихся в продукте речемыслительной деятельности, позволяет вводить новые интегральные категории анализа и описания исследуемого предмета. Введение в научный оборот текстов, порожденных в виртуальной коммуникации, в ка-

честве материалов психолингвистического исследования, требует разработки программного инструментария, основанного на методах искусственного интеллекта.

В наших работах, выполняемых в сотрудничестве с коллективом лаборатории интеллектуальных динамических систем ИСА РАН, мы исходим из представления о деятельностной природе психики, что позволяет нам отнести к явлению виртуальной коммуникации как к особому виду деятельности, в ходе которой возникает новый тип сознания – общественное сознание коммуникантов, формирующих свои представления о реальности и оценке отдельных событий в ходе сетевого взаимодействия. Концепция образа мира, представленная в работах А.Н.Леонтьева [Леонтьев, 1983], Е.Ю.Артемьевой [Артемьева, 1999], В.В.Петухова [Петухов, 1984], В.Ф.Петренко [Петренко, 2001], С.Д.Смирнова [Смирнов, 1985], позволяет нам рассматривать продукты виртуальной коммуникации – тексты социальных медиа – как объективированные фрагменты картины мира, конструируемой участниками коммуникации. Социальный характер возникающей в ходе сетевых дискуссий картины мира позволяет ставить задачу реконструкции по текстам таких социальных эмоций, как гнев и презрение, таких социальных форм поведения, как агрессия и группирование, таких социальных явлений, как социальная напряженность и аномия.

В центре наших исследовательских интересов находятся явление фruстрации и проблеморазрешающее поведение как концепты, позволяющие описать активность субъекта при столкновении с препятствием. Здесь мы опираемся на положение Дм.Узгадзе [Узгадзе, 2001] об объективации как процедуре введения в сознание того, что препятствует разворачиванию деятельности под влиянием установки. Именно это – препятствия, возникающие при реализации привычного поведения, – является основными темами обсуждения в виртуальной коммуникации: обсуждения в Сети всегда вращаются вокруг возникших для их участников затруднений как в сфере частной жизни, так и в сфере политики (от рецептов блинов до рецептов разрешения международных конфликтов). В совокупности своих отношений и меняющегося отношения к ним субъекта эти препятствия выступают объектами психического отражения и могут рассматриваться как единицы анализа при

изучении картины мира. В теории искусственного интеллекта, для моделирования таких явлений разработан аппарат динамических неоднородных семантических сетей, что позволяет ставить задачу информационного моделирования различных типов картины мира, допускающих разные типы реакции на фрустрацию и служащие основой для различных вариантов группового проблеморазрешающего поведения.

В качестве исследовательского средства могут быть использованы методы семантического анализа текстов на естественном языке, основанные на понятии значения в языке [Смирнов и др., 2014]. В их основу положена теория коммуникативной грамматики языка [Золотова и др., 2004], согласно которой главное место в конструировании осмысленных высказываний занимают синтаксемы – минимальные семантико-синтаксические единицы языка, выступающие одновременно как носители элементарного смысла, и как конструктивные компоненты более сложных синтаксических построений. В конкретном предложении слово выступает в качестве единицы смысла именно как синтаксема. Таким образом, при работе с текстом необходимо оперировать не лексическими единицами (словами), а семантико-синтаксическими (синтаксемами). Выделив в тексте синтаксемы и установив смысловые значения, которые они могут принимать согласно правилам грамматики, можно выявить смысл всего текста.

Для представления семантики текста предлагается использовать неоднородные семантические сети (НСС) – семейство графов, имеющих общее множество вершин [Осипов, 1997]. Вершинам НСС соответствуют синтаксемы с их значениями, ребрам – элементы бинарных отношений на множестве значений синтаксем. Таким образом, НСС текста представляют ситуацию, описываемую в тексте. Неоднородные семантические сети являются эффективным способом представления семантики предложений текста. Они используются для вычисления степени смысловой близости текстов при решении задач информационного поиска, а также для эффективного решения многих других задач обработки текстов. Для представления семантики текста в виде неоднородной семантической сети может быть использован реляционно-ситуационный анализ текстов [Осипов и др., 2008].

СПИСОК ЛИТЕРАТУРЫ

- [Артемьева, 1999] Артемьева Е.Ю. Основы психологии субъективной семантики. М.: Издательство «Смысл». 1999.
- [Величковский, 2013] Величковский Б.М. Конвергенция сознания и технологический прогресс // В мире науки. 2013. Спец. выпуск.
- [Величковский и др., 2006] Величковский Б.М., Соловьев В.Д., Ушаков Д.В. Предисловие // Когнитивные исследования. Сборник научных трудов. Вып.1. М.: Институт психологии РАН, 2006.
- [Девяткин и др., 2014] Девяткин Д.А., Кузнецова Ю.М., Чудова Н.В., Швец А.В. Интеллектуальный анализ проявлений вербальной агрессивности в текстах сетевых сообществ // Искусственный интеллект и принятие решений, 2014, №2.
- [Золотова и др., 2004] Золотова Г.А., Онищенко Н.К., Сидорова М.Ю. Коммуникативная грамматика русского языка. М.: ИРЯ РАН, 2004.
- [Леонтьев, 1983] Леонтьев А.Н. Образ мира. Избр. психолог. произведения, М.: Педагогика, 1983.
- [Осипов и др., 2008] Осипов Г.С., Смирнов И.В., Тихомиров И.А. Реляционно-ситуационный метод поиска и анализа текстов и его приложения // Искусственный интеллект и принятие решений. 2008. № 2.
- [Осипов, 1997] Осипов Г.С. Приобретение знаний интеллектуальными системами. М.: Наука. Физматлит, 1997.
- [Петухов, 1984] Петухов В. В. Образ мира и психологическое изучение мышления // Вестник Московского университета. Серия 14. Психология, 1984, № 4.
- [Петренко, 2001] Петренко В.Ф. Многомерное сознание: психосемантическая парадигма. М.: Изд-во. «Новый хронограф». 2010.
- [Плотинский, 2001] Плотинский Ю.М. Модели социальных процессов: Учебное пособие для высших учебных заведений. – Изд. 2-е, перераб. и доп. – М.: Логос, 2001.
- [Смирнов, 1985] Смирнов С.Д. Психология образа: проблема активности психического отражения. – М.: Издательство Московского университета, 1985.
- [Смирнов и др., 2014] Смирнов И.В., Шелманов А.О., Кузнецова Е.С., Храмоин И.В. Семантико-синтаксический анализ естественных языков. Часть II. Метод семантико-синтаксического анализа текстов // Искусственный интеллект и принятие решений. М.: ИСА РАН. 2014. № 1.
- [Узгадзе, 2001] Узгадзе Д. Н. Психология установки. – СПб.: Питер, 2001. («Психология-классика»).

УДК 007

ЗАКОНОМЕРНОСТИ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОГО КОГНИТИВНОГО СТИЛЯ¹

А.В. Павлов (*pavlov@phoi.ifmo.ru*)

К.А. Исаков

Национальный исследовательский университет информационных
технологий, механики и оптики, Санкт-Петербург

Формирование индивидуального когнитивного стиля рассмотрено в рамках логики с исключением на алгебре фурье-дуальных определяющих модель операций. Выявлена связь когнитивного стиля с описываемыми диссипативным фактором индивидуальными физическими свойствами и опытом познающего агента. Проведено моделирование ряда сценариев редуцирования когнитивного диссонанса, показана закономерность возникновения истерик.

Введение

Результаты когнитивных исследований подтверждают общий тезис о физической обусловленности математических моделей, применительно к теме – зависимости модели когнитивного агента от его индивидуального физического и социального опыта [Кузнецов, 2012]. Социальный опыт в результате обучения – установления и запоминания весов синаптических связей – также переводится в категорию физических свойств субъекта.

В рамках когнитивного подхода к обработке информации на первый план выходят вопросы включения новой информации в индивидуальный тезаурус. Формирование новой внутренней картины мира, субъективно непротиворечивой и адекватной реальности, вызывает наибольшие трудности, если новая информация противоречит старому знанию. Конфликт двух когниций проявляется как когнитивный диссонанс (КД) [Корсини и др., 2006]. Невозможность жизни в состоянии диссонанса порождает защитный механизм его редуцирования [Хекхаузен, 2003], сценарий которого ярко проявляет когнитивный стиль субъекта.

Все многообразие когнитивных стилей в первом приближении можно разделить на два: научный и обыденный [Кузнецов, 1998].

¹ Работа выполнена при финансовой поддержке РФФИ (грант № 12-01-00418).

Для первого, имеющего приоритетом логичность и адекватность знаний, КД может рассматриваться как ошибка обучения и, соответственно, редуцироваться также методами обучения – дополнением конфликтующих когниций для их согласования на более высоком уровне. Для второго, ориентированного на устойчивость внутренней картины мира, КД относится уже не столько к формально-логической, сколько к психологической противоречивости и обучение здесь неэффективно в силу иных приоритетов субъекта.

Одна из моделей, описывающих включение новой информации в индивидуальное знание, – логика с исключением [Вагин и др., 2008]. В статье [Павлов, 2012] в рамках бионического подхода дана модель логики с исключением в классе нечетко-значимых на алгебре фурье-дуальных операций, в которой системе предъявляется только посылка основного правила «Обобщенный Modus Ponens», заключение по которому ассоциативно формирует значение исключения, которое, в свою очередь, модифицирует первоначальное заключение. В работе [Павлов, 2013] показан феномен возникновения и редуцирования в данной модели КД. В статье [Павлов, 2014] показано, что редуцирование КД обусловлено диссипативным характером модели. В настоящем докладе, в развитие подхода, дан анализ диссипативного фактора и показана связь сценария редуцирования КД с оценкой диссипативного фактора.

1. Модель

Логика с исключением на алгебре фурье-дуальных операций порождается трехслойной нейронной сетью (НС) рис.1.а [Павлов, 2012].

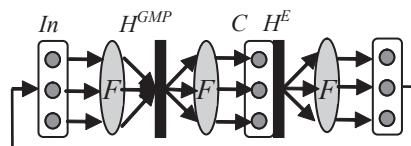


Рис.1.а. Модель НС. In , C , E – слои входной, заключения и исключения, H^{GMP} и H^E – матрицы весов «обобщенный Modus Ponens» и исключения, соответственно, F – преобразование Фурье

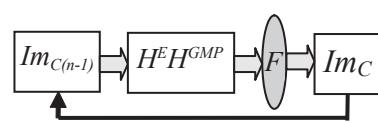


Рис.1.б. Блок-схема трехслойной НС. H^{GMP} и H^E – матрицы весов связей правила «обобщенный Modus Ponens» (ОМП) и исключения, формирующие составную матрицу

В рамках биологически мотивированного подхода смыслы значений входных лингвистических переменных представляются НС рис.1.а паттернами внутренней репрезентации (ПВР) в слое *In*, а в слое *C* формируются ПВР смыслов логических выводов. Матрица связей правила «Обобщенный Modus Ponens» (ОМП) реализуется каскадом двойного преобразования Фурье, синаптические веса формируются в пространстве Фурье и запоминаются в виде голограммы Фурье H^{GMP}

$$H(v_x, v_y) = \eta(F(\delta(x, y))F^*(Im_A(x, y))), \quad (1)$$

где (v_x, v_y) – пространственные частоты, η – оператор, учитывающий нелинейность регистрирующих сред для записи синаптических весов (РС), δ – дельта-функция, описывающая нейрон, $Im_A(x, y)$ – эталонный ПВР при обучении правилу ОМП, F – символ преобразования Фурье, астериск – комплексное сопряжение. При предъявлении в слое *In* входного ПВР $Im_{In}(x, y)$, матрица H^{GMP} формируется в слое *C* ПВР заключения

$$Im_C(\zeta, \eta) = [Im_m(x, y) \otimes Im_A(x, y)^* \delta(x, y)]_\eta, \quad (2)$$

где (ζ, η) – координаты в слое *C* (курсивом), \otimes и $*$ – символы корреляции и свертки, соответственно, нижний индекс η обозначает фильтрацию на матрице весов (1), обусловленную нелинейностью РС. Условимся считать формирование ПВР вывода (2) нулевой итерацией.

Матрица синаптических весов исключения описывается

$$H^E(\zeta, \eta) = \eta^E(F(Im_{ER}(\chi_x, \chi_y))Im_{CR}^*(\zeta, \eta)), \quad (3)$$

где η^E – оператор, учитывающий нелинейность РС для записи H^E , $Im_{ER}(\chi_x, \chi_y)$ и $Im_{CR}^*(\zeta, \eta)$ – ПВР в слое *E* и *C* (отклик (2) по правилу ОМП), используемые для обучения НС исключению.

Тогда входной ПВР $Im_{In}(x, y)$ преобразуется матрицей (1) в ПВР вывода (2) в слое *C*, который, проходя матрицу (3), формирует в слое *E* ПВР:

$$Im_E(\chi_x, \chi_y) = F\{Im_C(\zeta, \eta)\eta^E[F(Im_{ER}(\chi_x, \chi_y))Im_{CR}^*(\zeta, \eta)]\}.$$

Этот ПВР снова поступает в слой *In* и, в результате итерации *In*→*C*, в слое *C* формируется ПВР модифицированного заключения

$$Im_{C2}(\zeta, \eta) = Im_E(\chi_x, \chi_y) \otimes Im_{A\eta}(x, y).$$

Если выполнено требование идентичности спектров фаз входных ПВР

$$\{Im_{In}(x, y)\}, \forall i, j; \Phi(Im_{In}) = \Phi(Im_{Inj}), \quad (4)$$

то (χ_x, χ_y) в слое E можно заменить на (x, y) и тогда заключение

$$Im_{Cn}(\zeta, \eta) = F(Im_{C(n-1)}(\zeta, \eta)) * \left\{ [Im_{ER\eta^E}(x, y) \otimes F(Im_{CR}(\zeta, \eta))] \otimes Im_{A\eta}(x, y) \right\}.$$

Член в фигурных скобках – диссипативный фактор ($\Delta\Phi$). С учетом (1) и (3) это выражение можно представить в двух эквивалентных видах:

$$Im_{Cn}(\zeta, \eta) = F(Im_{C(n-1)}(\zeta, \eta)) * F(H^E H^{GMP}), \quad (5)$$

$$F(Im_{Cn}(\zeta, \eta)) = (Im_{C(n-1)}(\zeta, \eta))(H^E H^{GMP}). \quad (6)$$

Выражение (6) удобно для анализа динамики НС в пространстве Фурье в увязке с определяющими свойства матриц (1) и (3) условиями обучения, представив НС, начиная с 1-й итерации, в виде, данном на рис.1.б.

2. Анализ диссипативного фактора

При минимальном значении диссипативного члена, т.е.

$$\left\{ [Im_{ER\eta^C}(x, y) \otimes F(Im_{CR}(\zeta, \eta))] \otimes Im_{A\eta}(x, y) \right\} = \delta,$$

система была бы консервативна и, в силу фурье-дуальности значений заключений на соседних итерациях, ее динамика – циклической. Рассмотрим роль отдельных членов, входящих в диссипативный фактор.

Первый член $\Delta\Phi$ описывает условия записи матрицы (3) как в выборе ПВР эталона исключения $Im_{ER}(x, y)$, так и свойств РС (нелинейности и ограниченности динамического диапазона), ведущих к дополнительной фильтрации. Очевидна изотонность $Im_{ER\eta^C}(x, y)$ и $\Delta\Phi$ в целом.

Влияние выбора эталона исключения $Im_{ER}(x, y)$ на $\Delta\Phi$ определяется тем, что если $Im_{ER}(x, y) = \min \{Im_{In}(x, y)\}$, т.е. эталон $Im_{ER}(x, y)$ – самый высокочастотный, то и $\Delta\Phi$ в целом минимизируется, и наоборот.

Условия обучения в части записи матрицы (3) определяют дополнительную фильтрацию, обусловленную нелинейностью РС. Высокочастотная фильтрация, т.е. повышенное внимание деталям, имеет следствием $Im_{ER\eta}(x,y) \prec Im_{ER}(x,y)$, т.е. уменьшает ДФ, а низкочастотная, т.е. общность взгляда в ущерб его детальности, напротив, $Im_{ER\eta}(x,y) \succ Im_{ER}(x,y)$, т.е. ведет к росту ДФ и скорости конвергенции.

Второй член описывает отклик голограммы основного правила ОМП (1) сразу за ее плоскостью на этапе записи матрицы весов исключения (3):

$$F(Im_{CR}(\zeta, \eta)) = F(Im_{mR}(x, y))\eta(F(\delta(x, y))F^*(Im_A(x, y))).$$

Этот операнд учитывает:

- условия записи (1) в части выбора эталонов $Im_A(x, y)$ и $\delta(x, y)$;
- условия записи (1) в части свойств РС и условий записи, определяющих фильтрацию, описываемую оператором η ;
- выбор ПВР $Im_{mR}(x, y)$ для формирования заключения $Im_{CR}^*(\zeta, \eta)$ при обучении исключению, т.е. также и условия записи матрицы (3).

Поскольку ПВР $Im_A(x, y)$, $\delta(x, y)$ и $Im_{mR}(x, y)$ входят во второй член своими фурье-образами, то они антитонны ему. Второй операнд, в свою очередь, как вычитаемое антитонен ДФ в целом. Соответственно, высокочастотная фильтрация на (1) ведет к росту вклада второго операнда в ДФ и росту скорости конвергенции, а низкочастотная – к уменьшению.

Антитонность второго операнда ДФ может нарушаться при росте высокочастотности фильтрации вследствие нарушения следующей из $Im_{ER}(\chi_x, \chi_y) \in \{Im_m(x, y)\}$ унимодальности как ДФ, так и ПВР заключения вследствие роста боковых максимумов функции автокорреляции (АКФ).

Выражение в квадратных скобках описывает импульсный отклик матрицы исключения (3) и зависит как от условий ее записи, в том числе выбора эталона $Im_{ER}(x, y)$, так и от условий записи (1) (второй член).

Третий член зависит только от условий записи матрицы весов правила ОМП (1) и изотонен как первому операнду, так и ДФ в целом.

Вернемся к выражению (6) и обозначим составную голограмму

$$H^E(\zeta, \eta) H^{GMP}(v_x, v_y) = HH(\zeta, \eta). \quad (7)$$

Устойчивое состояние сети определяется как решение уравнения

$$F(Im_{Cn}(\zeta, \eta)) = (Im_{Cn}(\zeta, \eta))HH(\zeta, \eta). \quad (8)$$

Условие (4) определяет, что, начиная с нулевой итерации (2), ПВР $Im_{Cn}(\zeta, \eta)$ действителен. Также из (4), с учетом (1) и (3), следует, что составная голограмма (7) также действительна. Действительность левой части (8), начиная с нулевой итерации (2), следует из (1). Из действительности правой части (8), вкупе с действительностью левой, следует четность последней и всех операндов (8). Отсюда следует:

1. Если составная голограмма $HH(\zeta, \eta)$ унимодальна, то и решение уравнения (8) – устойчивое состояние $Im_{C\infty}(\zeta, \eta)$ также унимодально. В работах [Павлов, 2013, 2014] показано, что это устойчивое состояние соответствует формированию толерантности – независимости итогового заключения от значений входных переменных. Из (5) и (6) видно, что формирование толерантности обусловлено пассивностью восприятия исключения в смысле отсутствия самостоятельного дообучения как активного поиска информации для дополнения конфликтующих когниций. Новый тезаурус описывается как составная голограмма (7).

Случай унимодальной матрицы (7) исследован ранее [Павлов, 2013, 2014], где показано, что скорость конвергенции связана с оценкой диссипативности системы, в качестве которой взят $r_{0.606}$ – радиус АКФ ДФ по уровню 0.606. Поэтому рассматривать этот вариант сейчас не будем.

2. Если составная голограмма $HH(\zeta, \eta)$ не унимодальна, то и решения уравнения (8) могут быть не унимодальны. Нарушение унимодальности (7) возможно вследствие высокочастотной фильтрации, включая режектирование низких частот. Высокочастотная фильтрация обусловлена условиями обучения вкупе с фундаментальным свойством ограниченности динамического диапазона РС и соответствует повышению внимания к деталям в ущерб общности взгляда. Согласно (5), ДФ $F(H^E H^{GMP})$ представляет собой импульсный отклик системы, а (7) – передаточную функцию. Соответственно, свертка (5) с мультиодальным импульсным откликом может, в зависимости от соотношения входящих в (5) опе-

рандов, вести к мультимодальности результата, даже при унимодальном $F(Im_{C(n-1)}(\zeta, \eta))$.

Рассмотрим динамику сети и тип устойчивого решения в зависимости от вида нового тезауруса, представленного в пространстве Фурье, т.е. (7).

3. Моделирование

Моделировалась динамика системы рис.1 в зависимости от параметров высокочастотного фильтра (ВЧФ) на голограмме (7). Как следует из (7), ВЧФ может вводиться на любой из голограмм, составляющих (7).

Моделировались ПВР со спектрами амплитуд, описываемых функциями Гаусса с параметром $v_{Im0.606}$, и случайными равномерными распределениями фаз на интервале $[0, 2\pi]$. ВЧФ задавался функцией Гаусса с центральной частотой v_0 и параметром (полушириной) $v_{0.606}$.

На рис.2.а даны зависимости показателя затухания γ , определяемого из $|r_{0.606}(n) - r_{0.606}(n-1)| = y_0 + \exp(\gamma n)$, от радиуса АКФ ДФ $r_{0.606}$ для ряда условий записи матриц весов (1) и (3). Здесь нас интересует диапазон небольших радиусов АКФ ДФ, в котором квазилинейный характер зависимостей нарушен в силу роста боковых максимумов (БМ) ДФ и (7).

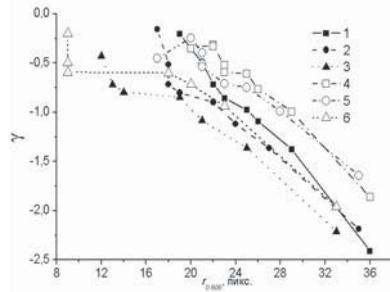


Рис.2.а. Зависимости показателя затухания γ от $r_{0.606}$ для радиусов АКФ ПВР Im_A : (1-3) – $r=14$; (4-6) – $r=9$; условий записи H^E : 1,4 – линейная; 2,5 – $|F(Im_{ER})| = \text{Const}(v)$; 3,6 – $|F(Im_{ER})| = |F(Im_{CR})| = \text{Const}(v)$

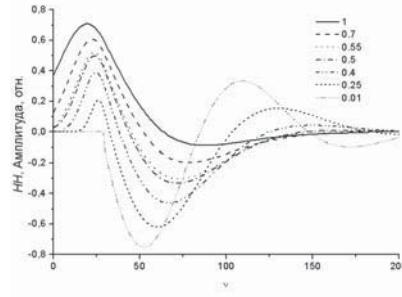


Рис.2.б. Сечения передаточной функции $H^E H^{GMP}$ для ряда значений v_R

На рис.2.б для $v_0=30$ даны сечения составной голограммы (7), для ряда относительных ширин полос ВЧФ $v_R = \frac{v_0}{v_{0.606}}$ при $\frac{v_{0.606}}{v_0} = 5$.

При незначительном ослаблении низких частот, в диапазоне $v_{0.606}$ [30, 17.373165] (v_R [1,0.579]) динамика конвергентна, в устойчивом режиме формируется унимодальный отклик (рис.3.а). При дальнейшем уменьшении v_R динамика вначале выглядит как конвергентная (рис.3.б), но на очень большом числе итераций (после 18 000) появляются истерики – отклик теряет свойство унимодальности, а потом система вновь возвращается к унимодальному отклику ... до следующей истерики. По мере уменьшения v_R частота появления истерик нарастает (рис.4.в), они сливаются и при $v_{0.606}=16$ ($v_R=0.533$) вновь наблюдается конвергентная динамика (рис.3.г), но устойчивый отклик уже не унимодален.

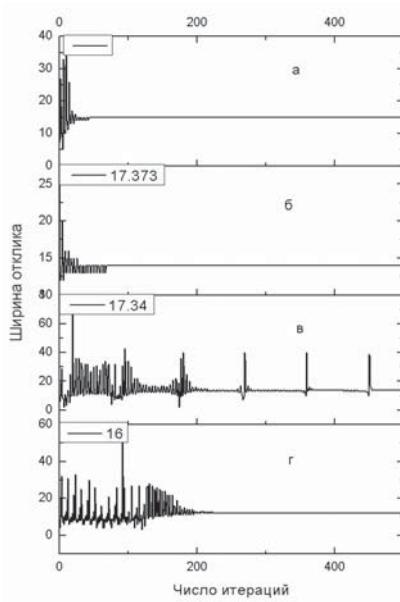


Рис.3. а-г. Динамика в зависимости от ширины фильтра

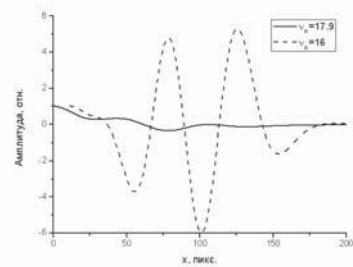


Рис.3.д. Отклики в устойчивом режиме для двух фильтров

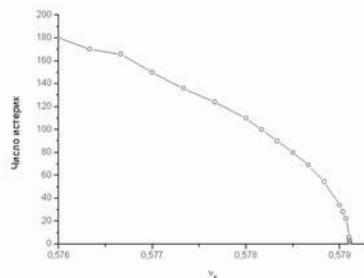


Рис.3.е. Зависимость числа истерик на 10000 итераций от параметра фильтра

Появление периодических истерик может рассматриваться по аналогии с эффектом перемежаемости. Квази-устойчивый унимодальный отклик (рис.3.д, сплошная кривая) на протяжении многих итераций имеет стабильную форму глобального максимума, но при этом на каждой итерации происходит медленный рост амплитуд боковых максимумов, резкое ускорение которого предшествует появлению истерики – переходу к мультимодальному отклику и последующим возвращением к унимодальному. Таким образом, в этом диапазоне параметров фильтра решение (8) неустойчиво. Отношение амплитуд бокового максимума к глобальному может рассматриваться как параметр порядка системы.

Применительно к переходу от «истерического» к устойчивому режиму мультимодального отклика как устойчивому решению (8) существенно, что зависимость амплитуды глобального максимума АКФ ДФ от относительной ширины фильтра имеет экстремум (минимум), соответствующий данному изменению типа динамики. В силу ограниченности объема статьи иллюстрации здесь не приводим.

Таким образом, когнитивный стиль при редуцировании когнитивного диссонанса в рамках логики с исключением определяется способностью к дополнительному самообучению – при ее наличии возможна реализация научного стиля, при отсутствии – обыденного. В последнем случае диссипативный характер модели определяет редуцирование как схождение к состоянию толерантности. Высокочастотная фильтрация – излишнее внимание к деталям – ведет вначале к появлению периодических истерик, а затем, при увеличении внимания к деталям, к формированию неадекватного заключения, не отвечающего критерию унимодальности, обязательному для подачи на исполнительные органы.

Благодарности. Авторы считают приятным долгом выразить свою благодарность проф., д.т.н. О.П. Кузнецову, проф., д.т.н. И.Б. Фоминых и с.н.с., к.т.н. А.А. Кулиничу за ценные дискуссии и замечания.

СПИСОК ЛИТЕРАТУРЫ

[**Вагин и др., 2008]** Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах. 2-е изд. М.: ФИЗМАТЛИТ, 2008. 712 с.

- [Корсини и др., 2006]** Психологическая энциклопедия / Под ред. Р. Корсини, А. Ауэрбаха. 2-е изд. СПб: Питер, 2006. 1096 с.
- [Кузнецов, 1998]** Кузнецов О.П. Быстрые процессы мозга и обработка образов // Новости искусственного интеллекта. 1998. № 2.
- [Кузнецов, 2012]** Кузнецов О.П. Когнитивная семантика и искусственный интеллект // Искусственный интеллект и принятие решений. 2012. № 4.
- [Павлов, 2012]** Павлов А.В. Алгебра фурье-дуальных операций: логика с исключением // Искусственный интеллект и принятие решений. 2012. № 3.
- [Павлов, 2013]** Павлов А.В. Логика с исключением на алгебре фурье-дуальных операций: феномен сомнений и колебаний // VII Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте». Коломна, 20–22 мая 2013. Сборник научных трудов. М.: ФИЗМАТЛИТ, 2013. Т. 3.
- [Павлов, 2014]** Павлов А.В. Логика с исключением на алгебре фурье-дуальных операций: нейросетевой механизм редукции когнитивного диссонанса // Научно-технический вестник информационных технологий, механики и оптики. 2014. Т. 89. №1.
- [Хекхаузен, 2003]** Хекхаузен Х. Мотивация и деятельность. 2-е изд. СПб.: Питер; М.: Смысл, 2003. – 860 с.

УДК 007:681.518.2

СПОСОБЫ РАЗРЕШЕНИЯ КОНФЛИКТОВ В ГИБРИДНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ¹

В.В. Смирнов (*vitaly_smirnov@mail.ru*)

Делис Архив, Москва

И.Б. Фоминых (*igborfomin@mail.ru*)

МЭИ, Москва

В работе рассмотрены способы разрешения конфликтов в гибридных интеллектуальных системах, показана применимость этих способов в зависимости от уровней восприятия внешней и внутренней среды, объединяющих в себе несколько распространенных точек зрения на психические процессы восприятия.

Введение

Междисциплинарное направление «гибридные интеллектуальные системы» объединяет специалистов, занимающихся разработкой и совместным использованием различных методов имитации интеллектуальной деятельности человека. Когда речь идет о гибридных интеллектуальных информационных системах, часто имеют в виду системы, объединяющие работу со знаниями и традиционную обработку информации. Одной из проблем такой обработки является проблема разрешения конфликтов, которая может решаться по-разному, в зависимости от используемых методов извлечения, представления и обработки данных и знаний.

Например, в реляционных базах данных (БД) под конфликтами обычно понимают противоречивость данных, которая может возникать вследствие дублирования одной и той же информации в нескольких записях одной и той же или различных таблицах, что в случае внесения изменений может приводить к разноточням. Для устранения возможности возникновения таких противоречий, обычно являющихся следствием избыточности данных, принято выполнять нормализацию БД, а также использовать такие механизмы, как первичные и внешние ключи, трансакции, триггеры, значения по умолчанию, предопределенные типы данных и др.

¹ Работа выполнена при финансовой поддержке РФФИ (проекты 14-07-00373, 12-07-00083, 13-07-00396).

В экспертных системах (ЭС) решение проблемы разрешения конфликтов может начинаться еще на этапах их разработки, когда противоречивость знаний выявляется в процессах их извлечения, структурирования и формализации [Рыбина и др., 2007].

На этапах реализации, тестирования и внедрения ЭС разрешение конфликтов обеспечивается стратегиями, которые являются частью общей стратегии логического вывода. Обычно стратегии разрешения конфликтов ориентированы на выбор наиболее предпочтительных правил, цепочек правил, путей в семантической сети или других элементов БЗ для того, чтобы использовать их для выполнения изменений рабочей памяти на текущем шаге логического вывода. Гораздо реже стратегии разрешения конфликтов ориентированы на достижение наиболее полного соответствия исходных данных и информации, содержащейся в рабочей памяти. Примеры таких стратегий рассмотрены в работах [Рыбина и др., 2012; Моросин и др., 2012].

Противоречия могут быть связаны как с избыточностью, так и с неполнотой данных и знаний. Например, выше было отмечено, что избыточность данных может приводить к их противоречивости. Что касается связи проблемы противоречивости знаний с их неполнотой, то эта проблема рассматривалась, в частности, в работе Д.А. Поспелова и его коллег, например, [Кондрашина и др., 1989]. При разработке и использовании гибридных интеллектуальных информационных систем может применяться комбинация методов извлечения, представления или обработки данных и знаний. При этом конфликты могут возникать при выборе наиболее эффективного сочетания таких методов для решения конкретной задачи. Следует учесть, что в сложных проблемных областях может быть выделено несколько слабо связанных групп исходных данных и знаний, при этом для одних групп может иметь место неполнота, а для других – избыточность. Как показано на рис. 1, названные проблемы, потенциально способные приводить к конфликтам в процессе решения задач, можно представить, используя шкалы «неполнота исходных данных», «избыточность исходных данных», «неполнота знаний» и «избыточность знаний». В центральной части рисунка расположена область, соответствующая ситуациям, когда условий для возникновения конфликтов меньше, чем для области, примыкающей к краям рисунка.

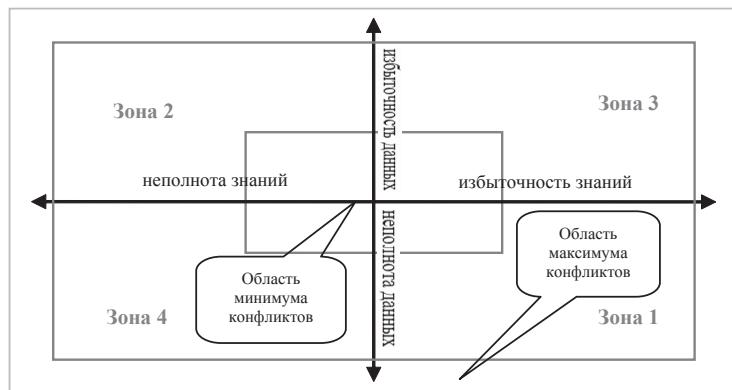


Рис.1. Влияние неполноты и избыточности данных и знаний на конфликты, возникающие в процессе решения задач

При моделировании конфликтов, связанных с этими двумя группами проблем, в рамках данного исследования планируется провести аналогии между методами разрешения конфликтов в гибридных интеллектуальных информационных системах и способами разрешения конфликтов, характерных для различных психических уровней восприятия людьми внутренней среды своего организма, окружающей среды и собственного жизненного опыта. При этом предполагается, что стратегия разрешения конфликтов, возникающих при решении интеллектуальных задач, должна формироваться в зависимости от того, какая из двух групп проблем доминирует при решении конкретной задачи.

1. Моделирование конфликтов в зависимости от уровня восприятия внешней и внутренней среды

В психологии существует множество различных точек зрения на понятие *восприятие*. Например, в работе [Маклаков, 2004] под восприятием понимается психический процесс, приводящий к рождению чувственного образа, структурированного по определенным принципам и содержащего в качестве одного из исследуемых элементов самого наблюдателя. В ряде работ делается попытка определить элементы, одновременное функционирование которых составляет суть процесса восприятия. В качестве таких элементов

выбирают образы, значения, личностные смыслы [Леонтьев, 1983]; используют личностные конструкты, устанавливающие отношение типа сходства – различия между воспринимаемыми явлениями [Келли, 2000]; и др.

В других работах акцент делается на иерархической структуре восприятия. По З. Фрейду, личность составляют три структурных элемента: «Оно», «Я» и «Сверх-Я», каждый из них участвует как в сознательных, так и в бессознательных процессах [Хайгл-Эверс и др., 2002]. В работе [Грановская, 1988] рассматриваются различия иерархической структуры «левополушарного» и «правополушарного» восприятия.

Полезно выделить работы, посвященные *интуитивному восприятию*. В работе [Спрингер и др., 1983] интуитивными считаются процессы, происходящие в правом полушарии; иногда интуитивными считаются «надсознательные» процессы [Барбитова, 2010] или подсознательные процессы [Гаврилов, 2004].

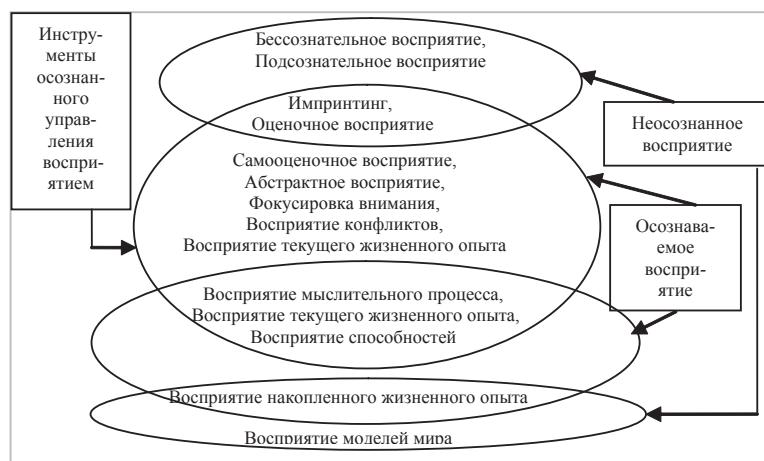


Рис. 2. Уровни осознанности процессов восприятия

Авторами исследования, представляющего в данной работе, выделены следующие основные уровни восприятия (рис. 2): 1) Бессознательное инстинктивное восприятие; 2) Подсознательное восприятие, импринтинг; 3) Оценочное и самооценочное восприятие; 4) Абстрактное восприятие, фокусировка внимания, восприятие накопленного жизненного опыта.

конфликтов; 5) Восприятие мыслительного процесса и текущего жизненного опыта, в том числе эмоционального; 6) Восприятие способностей, в том числе психических и интеллектуальных, восприятие накопленного жизненного опыта; 7) Восприятие моделей мира. Такая многоуровневая структура, с одной стороны, может быть соотнесена с распространенными точками зрения на психические процессы восприятия, а с другой – классифицировать методы моделирования конфликтов.

2. Бессознательный, подсознательный и оценочный уровни восприятия

На 1-м и 2-м уровнях восприятие конфликтов при решении задач является неосознанным. Представляется наиболее перспективным моделирование таких конфликтов с помощью интеллектуальных агентов. Выделены следующие четыре типа конфликтов: конфликты восприятия внешней среды агентом, конфликты целей агента, конфликты рассуждений агента, конфликты коллективного поведения агентов.

Конфликты восприятия внешней среды могут быть связаны, с одной стороны, с неполнотой данных о внешней среде, что приводит к неоднозначности идентификации ее состояния, а с другой стороны – с избыточностью информации о внешней среде, когда на выполнение действий агентом влияет лишь часть воспринимаемых данных [Бугайченко и др., 2005].

При моделировании конфликтов целей агентов полезно различать имманентные и редуцированные цели, определять, является ли многоагентная система централизованной, децентрализованной или распределенной, учитывать возможность межуровневых конфликтов [Алгазин, 2012; Тарасов, 2002]. Рассматривая возможные конфликты рассуждений агентов, необходимо учитывать как различные модели представления знаний, так и различные типы рассуждений [Тарасов, 2002].

В работе [Городецкий и др., 1998] представлена классификация конфликтов в многоагентных системах и рассмотрен один из подходов к разрешению конфликтов в поведении агентов, основанный на снятии логического противоречия вида $p \wedge q \Rightarrow \text{false}$.

Рассмотрим подробнее один из возможных подходов к моделированию конфликтов на 3-м уровне. Предполагая, что на оце-

ночном и самооценочном уровнях, как и на предыдущих уровнях, само восприятие конфликтов при решении задач является неосознанным и имеют место избыточность исходных данных и неполнота знаний, представляется весьма перспективным для моделирования конфликтов и их разрешения использовать алгебру эмоций [Фоминых, 1999].

Как известно [Анохин, 1964], эмоции в живых организмах выполняют функцию локального критерия управления, подсказывая организму, что хорошо и что плохо в данных конкретных условиях, т.е. эмоция, по словам П.К.Анохина, играет «пеленга», который подсказывает организму, движется ли он к цели или от цели. В первом случае эмоция положительна («удовольствие»), во втором – отрицательна («страдание»). При формализации эмоций будем опираться на введенный Г.А.Голицыным принцип максимума взаимной информации между условиями среды и реакциями системы [Голицын и др., 1991]. Согласно этому принципу эмоции рассматриваются как средства квазиоптимального управления поведением системы (субъекта), направляющие ее к достижению максимума ее целевой функции (максимума взаимной информации между условиями среды и реакциями системы).

Увеличение целевой функции L сопровождается положительными эмоциями, уменьшение – отрицательными эмоциями. Поскольку L зависит от некоторых переменных x_i , то эмоции e вызываются изменениями этих переменных или в формальном виде: $e = dL/dt = dL/dx_i dx_i/dt$, где dL означает изменение L , обусловленное изменением переменной x_i .

Более того, эмоции, по существу, можно считать результатом интегральной оценки ситуации не по всем описывающим ее параметрам, которых может оказаться слишком много, а лишь по нескольким наиболее важным характеристикам. Соответственно, эмоции могут «запускать» поведение, не обязательно оптимальное в данной плохо изученной ситуации, но такое, которое в наибольшей степени позволит избежать (пусть с большими потерями), катастрофических последствий превышения некоторого важного (например, временного) ресурса.

Существует гипотеза [Golitsyn and other, 1995], что алгебра эмоций реализуется в мозгу человека с помощью нейроподобных/распределенных вычислительных структур, задача которых опери-

ровать оценками (причем как в области материальных благ, так и благ нематериальных: поступков, мыслей, чувств и т.д.).

3. Абстрактное восприятие, восприятие мыслительного процесса, способностей и моделей мира

Интеллектуальная система, способная к восприятию 4-го уровня, должна содержать элементы, которые, с одной стороны, не зависят от конфликтов других элементов и не являются результатом их функционирования, а, с другой стороны, способны управлять процессом разрешения конфликтов. В зависимости от типа интеллектуальной системы конфликтующие и управляющие элементы могут различаться. Например, в экспертных системах, когда ситуация относится к зоне 1, представленной на рис.1, конфликты могут возникать между правилами или цепочками правил, а в качестве элементов, обеспечивающих разрешение конфликтов, могут выступать стратегии. Для той же зоны 1 в многоагентных системах конфликты между агентами могут разрешаться при использовании арбитражных протоколов взаимодействия [Тарасов, 2002].

Собственный жизненный опыт, помогающий в разрешении конфликтов, может различным образом восприниматься интеллектуальной системой, в зависимости от того, в какой степени она обладает способностью к рефлексии. Есть мнение, что эту способность формируют оценивание и целеполагание, мотивация, самоорганизация, активирование средств решения задачи, адаптация, самообучение, саморасширение [Чечкин, 2002].

На 7-м уровне восприятия интеллектуальная система, обладающая способностью к рефлексии, должна различать конфликты между различными моделями мира. Под моделью мира будем понимать сложившуюся систему представлений о реальности. Когда речь идет о моделях, используемых в интеллектуальных системах, термин «модель мира» может определяться через понятия «предметная область» [Шалютин, 2006], «проблемная область» или «картина мира» [Святогор и др., 2009].

В работе [Кузнецова и др., 2012] рассматривается семиотическое представление «картины мира», в котором применяется семантическая сеть с узлами, соответствующими таким элементам сознания, как образы, значения и личностные смыслы, с заданными оценками и предпочтениями.

Заключение

Использованная авторами многоуровневая структура процессов восприятия позволила систематизировать существующие подходы к моделированию процессов восприятия и способы разрешения конфликтов в гибридных интеллектуальных системах.

Анализ существующих способов разрешения конфликтов показал, что независимо от уровня восприятия наименее исследованными являются конфликты, происходящие в условиях, когда одновременно имеет место неполнота данных и знаний, и когда одновременно имеет место избыточность данных и знаний.

СПИСОК ЛИТЕРАТУРЫ

[Рыбина и др., 2007] Рыбина Г.В., Смирнов В.В. Методы и алгоритмы верификации баз знаний в интегрированных экспертных системах // Известия РАН. Теория и системы управления. 2007. № 4. С. 91–102.

[Рыбина и др., 2012] Рыбина Г.В., Мозгачев А.В. Приобретение, представление и обработка темпоральных знаний в динамических интегрированных экспертных системах // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012: Труды конференции. – Белгород: Изд-во БГТУ, 2012. Т.2. С. 170–176.

[Моросин и др. 2012] Вагин В.Н., Моросин О.Л. Система аргументации для логики предикатов первого порядка // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012: Труды конференции. – Белгород: Изд-во БГТУ, 2012. Т.1. С. 34–42.

[Маклаков, 2004] Маклаков А.Г. Общая психология. – СПб., 2004.

[Леонтьев, 1983] Леонтьев А. Н. Избранные психологические произведения: В 2-х т. Т. I – М.: Педагогика, 1983. – 392 с.

[Келли, 2000] Келли Д. Теория личности: психология личных конструктов. – СПб.: Речь, 2000. – 249 с.

[Спрингер и др., 1983] Спрингер С., Дейч Г. Левый мозг, правый мозг. – М.: Мир, 1983. – 265 с.

[Барбитова, 2010] Барбитова А.Д. Интуитивный подход в педагогической деятельности // Вектор науки Тольяттинского государственного университета. №. 1 (11). – Тольятти: Тольяттинский государственный университет, 2010. С. 165–169.

[Грановская, 1988] Грановская Р.М. Элементы практической психологии. – 2-е изд. – Л.: Издательство Ленинградского университета, 1988.– 560 с.

[Хайгл-Эверс и др., 2002] Хайгл-Эверс, Аннелизе; Хайгл, Франц; Отт, Юрген; Рюгер, Ульрих. Бессознательное // Базисное руководство по

психотерапии. – Речь, Восточно-Европейский институт психоанализа, 2002. – 790 с.

[Кондрашина и др., 1989] Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. Представление знаний о времени и пространстве интеллектуальных систем. М., 1989.

[Алгазин, 2012] Алгазин Г. И. Централизация и децентрализация в базовых игровых моделях организационных систем // УБС, 36 (2012), 144–172.

[Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. М.: Эдиториал УРСС, 2002.

[Бугайченко и др., 2005] Бугайченко Д. Ю., Соловьев И. П. Абстрактная архитектура интеллектуального агента и методы ее реализации // Системное программирование. – 2005. – №1. С 36–67.

[Городецкий и др., 1998] Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор) // Новости искусственного интеллекта, №2, 1998, с. 64–116.

[Анохин, 1964] Анохин П.К. Эмоции // Большая медицинская энциклопедия, т.35. – М.: Медгиз, 1964.

[Голицын и др., 1991] Голицын Г.А., Петров В.М. Информация – поведение – творчество. – М.: Наука, 1991.

[Golitsyn and other, 1995] Golitsyn G. A., Petrov V. M. Information and Creation. – Basel: Birkhauser Verlag, 1995.

[Фоминых, 1999] Фоминых И.Б. Адаптивные системы и информационная модель эмоций. – В сб. трудов Международной конф. «Интеллектуальное управление: новые интеллектуальные технологии в задачах управления» (ICIT'99) М.: Изд. Физ.-мат. литературы, 1999.

[Ксенчук, 2011] Ксенчук Е.В. Системное мышление. Границы ментальных моделей и системное видение мира. – М.: Издательский дом «Дело» РАНХиГС, 2011. – 368 с.

[Чечкин, 2002] Чечкин А. В. Рефлексия – центральная особенность интеллектуальных систем / В научно-методических материалах «Математические методы решения инженерных задач» под ред. профессоров В. В. Блаженкова и А. В. Чечкина. – М.: МО РФ, 2002.

[Шалютин, 2006] Шалютин С.М. Искусственный интеллект – гносеологический аспект. – М., 2006. – 172 с.

[Кузнецова и др., 2012] Кузнецова Ю.В., Осипов Г.С., Панов А.В., Чудова Н.В. Механизмы сознания и самосознания в управлении целенаправленным поведением // Институт системного анализа РАН, 2012. – <http://www.raai.org/news/pii/ppt/Osipov2012.pdf>.

[Святогор и др., 2009] Святогор Л., Гладун В. Семантический анализ текстов естественного языка: цели и средства // XV th International Conference «Knowledge-Dialogue-Solution» KDS-2 2009, Киев, Украина, Октябрь, 2009.

УДК 316.472.45

**МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ УЧАСТНИКОВ
ВИРТУАЛЬНЫХ СООБЩЕСТВ НА ПОРТАЛЕ
ASPIRANTURA.SPБ**

В.И. Тищенко (*tishchenkovictor@gmail.com*)

А.Л. Прочко (*alexei@prochko.ru*)

Институт системного анализа РАН, Москва

В работе обоснована возможность использования принципов теории сложных систем (complex networks) для моделирования виртуальных коммуникаций. В результате исследования научного сообщества на портале aspirantura.spb.ru получено графическое изображение (граф) социальной сети участников форума. Показано, что использование лувинского алгоритма позволяет выявить из сети виртуальные сообщества, участники которых характеризуются различной активностью и степенью «связности».

Введение

Информационные и коммуникационные цифровые технологии (ИКТ) перестают быть инновацией в современном обществе, превращаясь в рутинное средство социальной коммуникации. Новые способы изначально межличностного, а затем группового и массового взаимодействия на основе ИКТ, в свою очередь, приводят к изменению коммуникативной практики и культуры организации социального пространства [DiMaggio et al., 2001] и, как следствие, возникновению виртуального пространства социального взаимодействия. Именно это способствовало возникновению так называемых виртуальных (сетевых) сообществ, в которых, как утверждал Х. Рейнгольд, автор этого термина, «...люди делают все то же самое, что делают люди в реальной жизни. Но при этом их тела не взаимодействуют. В виртуальной комнате некого поцеловать и некому ударить вас кулаком в нос» [Rheingold, 1993].

Виртуальные сообщества (для обозначения сообществ в Интернете используют также термины «Интернет-сообщества», «онлайн-сообщества», «сетевые сообщества»), если рассматривать их не в операциональном или технологическом контексте, а в категориальном смысле, – достаточно «широкий» термин, включающий множество вариантов и конкретных примеров коммуни-

кации в сети. В этой связи оказывается описать поведение участников виртуального сообщества не просто, потому что и сообщества встречаются разные, и взаимодействие может быть разного уровня. Определенная трудность также возникает в связи с тем, что понятие «виртуальное сообщество» подобно многим иным понятиям, напрямую связанным с коммуникативными практиками в виртуальном пространстве, нередко используется в метафорическом или публицистическом значении.

В соответствии с концепцией, сформулированной и изложенной в работах, посвященных анализу становления сетевых взаимодействий в пространстве Интернета [Тищенко и др., 2014], под виртуальным сообществом мы будем понимать устойчивую содержательную коммуникацию пользователей информационной среды, которая опосредована компьютерными сетями. Предлагаемое определение не исключает возможности понимания виртуальных сообществ в качестве социально организованных групп людей со сходными интересами и общающихся через Интернет, используемое в социологии, политологии или социальной психологии при анализе структуры, социальных норм, статуса или личных отношений участников социальных образований в Интернете [Бондаренко, 2004; Патаракин, 2006]. В то же время определение виртуального сообщества в качестве специфической формы сетевой коммуникации позволяет использовать для анализа поведения сетевого сообщества теорию сложных сетей (complex networks) [Newman, 2010], формализовав тем самым это коммуникативное понятие.

В контексте сетевой парадигмы различные исследователи по-разному выделяют сообщества в сложных сетях [Gibson et al., 1998; Kumar et al., 1999]. В то же время, несмотря на некоторые различия, все эти определения рассматривают сообщества как такие части сети G , в которых связи между узлами C_{ij} сильнее, многочисленнее и насыщеннее, чем между узлами иных частей сети [Flake et al., 2000; Hedetniemi et al., 2004]. В соответствии с определением сообщества как части сложной сети проблема анализа поведения участников виртуальных сообществ распадается на две задачи – обнаружение в сети сообществ (1) и последующая их идентификация (2).

Исходя из теории графов, первая задача решается как процедура распределения узлов графа G на подмножества $C_i \subseteq V$, $0 \leq i < k$, так

что узлы, принадлежащие подмножеству C_i , имеют связи теснее и насыщеннее. В то же время задача идентификации сообщества решается как выделение сообщества C_i , к которому принадлежит множество узлов $S \subseteq V$ [Balakrishnan et al., 2004].

Для решения этой совокупности задач в последние годы предложен ряд алгоритмов [Newman et al., 2004; Fortunato, 2010; Vasudevan, 2012]. В своей основе эти методы используют принцип поэтапного удаления связей с максимальной важностью (betweenness centrality). Каждое такое удаление изменяет структуру кратчайших путей в сети, а следовательно, и важность каждой связи, и поэтому эти параметры пересчитываются после каждого удаления. На некотором шаге сеть оказывается разделенной на два кластера – два самых больших сообщества, и далее процедура продолжается. В результате получается дерево, в котором сообщества малых размеров включены в большие сообщества. Распределение по размерам сообществ, выявляемых в результате этой процедуры, в большинстве реальных сетей подчинено степенному закону.

1. Сеть сообщества участников форума на портале «aspirantura.spb.ru»

С целью анализа особенностей коммуникаций и построения модели коммуникаций виртуального сообщества нами было проведено экспериментальное исследование научного сообщества, сформировавшегося на портале аспирантов (<http://www.aspirantura.spb.ru>).

В качестве объекта анализа на этом портале был выбран форум, который состоит из 19 разделов, соответствующих различным наукам, 525 тем, 11 000 сообщений. Ежедневная аудитория портала доходит до 5000 пользователей, месячная аудитория сайта достигает 105 000 человек (статистика Рейтинг@Mail.ru).

Структура каждой темы представляет собой совокупность коммуникационных связей, состоящую из стартового сообщения, в котором предлагается обсудить некую проблему, и последовательных ответов-комментариев участников форума. В каждой тематической «нитке» форума существуют определенные стандартные функции, кроме стартового сообщения, – «цитата сообщения», «обращение к автору», которые в будущем помогут при подсчете и обработке связей. Мы будем называть связью между

двумя пользователями именно использование этих двух функций. В своей совокупности участники форума и связи между ними образуют сложную социальную сеть, в которой участников сети можно рассматривать в качестве узлов, соединенных множеством связей (ребер).

В качестве эмпирической предпосылки анализа сообщества и последующего графического отображения социальной сети, сложившейся на форуме портала, была сформирована база данных форума в форме MySql. Программа для получения данных, программа для визуализации данных были написаны на языке PHP.

В итоге, осуществляя анализ данных таблиц, мы получили таблицу связности между пользователями, которые принимали участие в обсуждениях на форуме портала. Каждая связь имеет направление (тот, на кого ссылаются) и вес (количество таких ссылок). В результате мы получили таблицу связей между 230 пользователями форума. Анализируя связи, мы получаем таблицу связности или граф из этих связей, где количество обращений или цитирований задает силу связи.

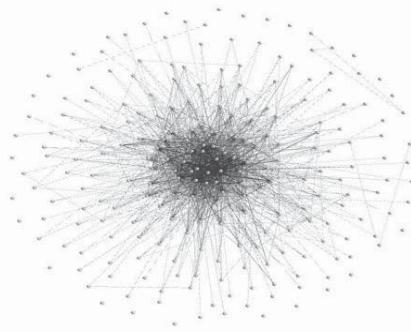


Рис. 1. Граф цитирований
и обращений участников
форума

Мы обнаружили около 5000 обращений и цитирований у этих 230 участников. На рисунке 1 схематично изображен график связей между этими пользователями.

Результаты построения графической модели рассматриваемого форума дают наглядную иллюстрацию о персональном вкладе каждого участника в обсуждение, модальность его суждений (активность). Кроме того, информация, отображенная в графическом расположении, дает исчерпывающее представление о том, кто из участников может претендовать на статус лидера мнений.

В представленной социальной сети велика роль активных участников форума, обладающих значительным влиянием в сети пользователей, существующей на форуме. Именно они фактически формируют структуру сети. Неудивительно, что пользователи с большим количеством сообщений оказались участниками с максимальной мощностью и показателем важности (betweenness centrality). В своей работе мы считаем лидерами именно таких пользователей. Они и являются связующими звеньями в сети в целом.

2. Выделение и идентификация сообществ. Меры важности узлов и связей

После построения социальной сети и ее графического отображения мы приступили к решению задач – выделение и идентификация сообществ. Для выделения сообществ был использован Лувенский метод [Blondel, 2008], который позволяет выделять сообщества на больших объемах данных и обладает высокой скоростью.

На первом шаге алгоритма каждому пользователю присваивается свое сообщество. Затем рассматриваются «соседи», и анализируются показатели, которые получаются, если поместить выбранную вершину в соседнее сообщество. Если показатель улучшится, вершина перемещается в соседний модуль. На следующем шаге рассматриваются уже не вершины, а полученные объединения – модули. Если в модуле несколько вершин, то вместо веса связи рассматривается сумма связей. Так продолжается до тех пор, пока улучшение становится невозможным.

Результаты оцениваются следующим образом, изменение от включения i в модуль C :

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

где \sum_{in} – сумма весов ребер внутри C , \sum_{tot} – сумма весов ребер, входящих в C , k_i – входящих в i , $k_{i,in}$ – сумма весов из i в C , а m – сумма весов во всей сети.

Мы провели разделение нашего графа на модули с различным показателем resolution. Всего было проведено 11 разделений с показателями от 0.01 (минимальные по размеру модули) до 2 (несколько крупных). Получили таблицу, в которой 230 пользователей

имеют 11 показателей – номер сообщества в каждом из разбиений. После этого были подсчитаны парные включения – если два пользователя в каком-то из разбиений попадали в один модуль, их совместный показатель увеличивался.

После этого были подсчитаны парные включения – если два пользователя в каком-то из разбиений попадали в один модуль, их совместный показатель увеличивался. Так была получена таблица с количеством «общих» модулей для всех пар.

Для каждого пользователя был подсчитан средний показатель по строке, т.е. среднее количество модулей, в которые он входит с каждым пользователем в отдельности. Если два пользователя при различных разбиениях оказываются в одном сообществе, то можно предположить, что у них есть «крепкая» связь.

Затем пользователи были отсортированы по количеству сообщений. После этого высчитывался показатель «Среднее количество модулей, в которые входят пользователи». Для этого бралась группа пользователей и считалось среднее число количества парных включений в этой группе. Добавлялись по одному пользователю в порядке убывания сообщений, и каждый раз пересчитывался показатель.

Оказалось, что средний показатель по мере включения новых пользователей падал, но незначительно. Для верхушки пользователей он составлял порядка 3.5, после присоединения 100 пользователей – 3.2, после включения 200 он стал равным 3. Получилось, что при 11 разбиениях, в среднем, каждые два пользователя из группы оказываются в среднем в трех разбиениях в одной группе. Так как показатель практически не меняется, можно утверждать, что с точки зрения силы связей между пользователями данное сообщество равномерно.

Далее, из таблицы с данными о парных включениях в различные модули был сформирован новый граф. Мы его назвали «Граф силы связи между двумя пользователями». Это полный неориентированный граф, где любые две вершины соединены ребром, а вес ребра – сила связи (количество общих модулей) для каждого двух пользователей.

Для него, по Лувинскому алгоритму, было осуществлено разбиение на модули при трех разрешениях. Оказалось, что все пользователи при этих трех разрешениях, несмотря на то, что при не-

которых показателях «resolution»» получается более двух модулей, в целом практически разбиваются на два больших модуля (которые включают меньшие модули в себя).

Возникает вопрос – совпадает ли алгоритмическое деление сообщества, основанное на коммуникациях с делением на сообщества полного графа силы связей? Для этого мы вернулись к исходным данным и постарались подобрать такой показатель разрешения, чтобы разбить все сообщество на 2–3 модуля через коммуникации.

Оказалось, что при таких разбиениях и граф связей через коммуникации, и полный график связей через введенную нами «силу связи» распадаются на два одинаковых подграфа с одними и теми же пользователями. Совпадение в группах составляет около 85–90%.

Для наглядности приведем «укладку» подграфа для каждого из графов с помощью алгоритма Force Atlas. Разным цветом обозначены разные сообщества.

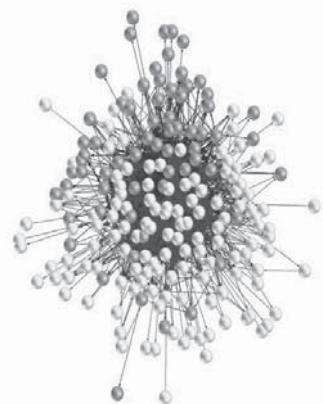


Рис. 2. Применение алгоритма Force Atlas для «укладки» графа цитирований и обращений

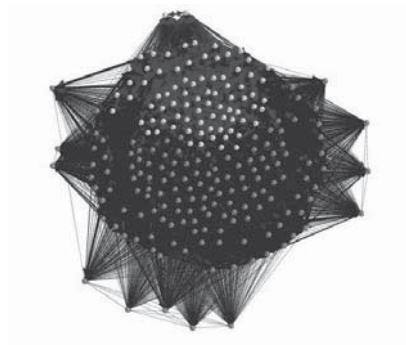


Рис. 3. Применение алгоритма Force Atlas для «укладки» графа «силы связей»

За счет полноты графа, после «укладки» график силы связи демонстрирует более показательную картину при делении на сообщества.

Заключение

В результате исследования получается, что «активные» пользователи, как бы «отбирая» близких им по содержанию общения участников форума, самоорганизуются и формируют соответствующие сообщества. Другими словами, используемая методология позволяет выявить («обнаружить») из множества существующих в виртуальной среде пользователей форума такие сообщества, участники которых характеризуются формализованным показателем – степенью «связности».

Выделение виртуальных сообществ по степени связности позволяет объективно выделить (идентифицировать) из множества пользователей форума таких участников, которые объединены в виртуальные сообщества общими целями и/или интересами, не прибегая к текстовому анализу сообщений участников форума.

И наконец, применение принципов анализа сложных сетей позволяет смоделировать структуру виртуального сообщества, выделив «влиятельных» участников сообществ («лидеров»), и тем самым заложить основы построения модели управления поведением виртуального сообщества.

СПИСОК ЛИТЕРАТУРЫ

- [Бондаренко, 2004] Бондаренко С.В. Социальная структура виртуальных сетевых сообществ. – Ростов-на-Дону: Издательство Рост. гос. ун-та, 2004.
- [Патаракин, 2006] Патаракин Е.Д. Сетевые сообщества и обучение. М.:ПЕР СЭ, 2006.
- [Тищенко и др., 2014] Тищенко В.И., Жукова Т.И., Попков Ю.С. Сетевые взаимодействия. Предмет исследования и объект моделирования. М., URSS. 2014.
- [Balakrishnan et al., 2006] Hemant Balakrishnan, Hemant Balakrishnan Discovering Communities in Complex Networks // ACMSE'06, March 10–12, 2006, Melbourne, Florida, USA. Copyright 2006 ACM/
- [Blondel et al., 2008] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), Jul 2008.
- [Flake et al., 2000] G. W. Flake, S. Lawrence, and C. L. Giles, «Efficient Identification of Web Communities,» presented at Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.

- [Fortunato, 2010]** Santo Fortunato Community detection in graphs // Physical Reports. V.486, issues 3–5, February 2010, pp. 75–174. arXiv:0906.0612v2.
- [Gibson et al., 1998]** D. Gibson, J. M. Kleinberg, and P. Raghavan, «Inferring Web Communities from Link Topology» presented at 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA, 1998.
- [Hedetniemi et al., 2004]** S. M. Hedetniemi, S. T. Hedetniemi, and P. Kristiansen, «Alliances in Graphs,» Journal of Combinatorial Mathematics and Combinatorial Computing, vol. 48, pp. 157–177, 2004.
- [Kumar et al., 1999]** S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, «Extracting Large-Scale Knowledge Bases from the Web» presented at Proceedings of 25th International Conference on Very Large Data Bases, 1999.
- [Newman, 2010]** Newman M. Networks. An Introduction. Oxford University Press, 2010.
- [Newman et al., 2004]** Newman, M.E.J. and Girvan, M. 2004, Finding and evaluating community structure in networks, Phys. Rev. E 69, 026113; cond-mat/0308217.
- [Rheingold, 1993]** Rheingold H. «The Virtual Community: Homesteading on the Electronic Frontier» – Perennial, 1993.
- [Vasudevan, 2012]** Vasudevan M. Algorithms for community identification in complex networks Spring Term 2012.

УДК 321.3

**АЛГОРИТМЫ, ИНСПИРИРОВАННЫЕ ПРИРОДНЫМИ
СИСТЕМАМИ ПРИ РЕШЕНИИ ЗАДАЧ
КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ
И ОПТИМИЗАЦИИ¹**

В.В. Бова (*vvbova@yandex.ru*)

В.В. Курейчик (*vkur@tgn.sfedu.ru*)

Вл.Вл. Курейчик (*vkur@tgn.sfedu.ru*)

Южный федеральный университет, Таганрог

В статье описываются алгоритмы, инспирированные природными системами при решении задач конструкторского проектирования и оптимизации. Эти алгоритмы позволяют распараллеливать процесс поиска и частично устранять проблему предварительной сходимости, а также получать наборы оптимальных и квазиоптимальных решений за полиномиальное время. Временная сложность алгоритмов в лучшем случае $\approx O(n \log n)$, в худшем случае – $O(n^2)$.

Введение

В связи с развитием нанометровых технологий в области производства возникает проблема перехода на новые методы проектирования. Это требует развития новых методов и алгоритмов. Одним из таких подходов являются бионические методы и биоинспирированные алгоритмы (БИА), использующие стратегии эволюционного моделирования и принципы природных механизмов принятия решений [Курейчик и др., 2009], [Гладков и др., 2010], [Курейчик

¹ Работа выполнена при поддержке гранта РФФИ № 13-01-00371.

и др., 2012а]. Бионические методы включают в себя генетические алгоритмы (ГА), эволюционные алгоритмы (ЭА), алгоритмы, моделирующие механизмы принятия решений природными системами. Это методы роевого интеллекта (муравьиные (МА), пчелиные (ПА) и др.), основанные на принципах коллективного поведения децентрализованной самоорганизующейся системы. Данный класс алгоритмов анализирует различные области пространства решений одновременно, и они более приспособлены к нахождению новых областей с оптимальными значениями целевой функции (ЦФ) при решении задач проектирования.

1. Генетический алгоритм

Генетические алгоритмы осуществляют поиск баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений», в неопределенных и нечетких условиях. При решении задач конструкторского проектирования и оптимизации с использованием ГА обычно выполняют четыре предварительных этапа: выбор способа представления решения; разработка операторов случайных изменений; определение способов «выживания» решений; создание начальной популяции альтернативных решений.

Приведем модифицированный генетический алгоритм, адаптированный для решения задач конструкторского проектирования и оптимизации (рис.1) [Гладков и др., 2010].

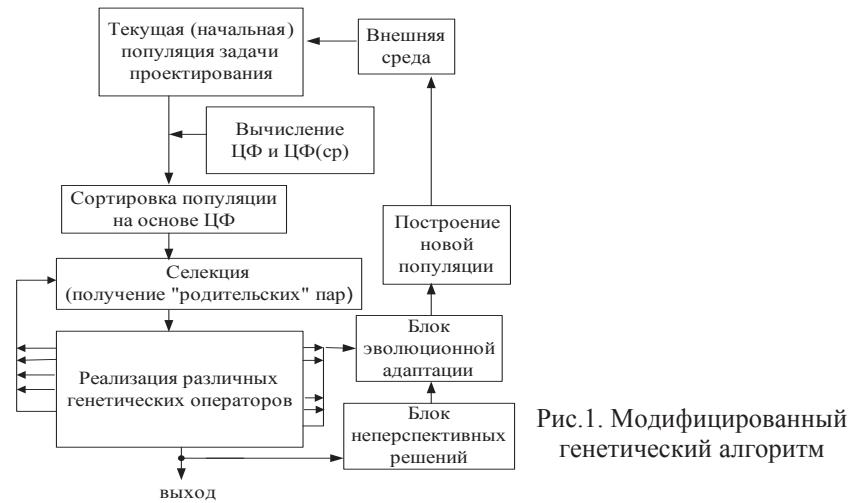


Рис.1. Модифицированный генетический алгоритм

Отметим, что периодически в каждой итерации ГА можно проводить различные изменения в перспективных, неперспективных и других решениях, что позволяет концентрировать поиск на получение более перспективных решений. Предложенная схема генетического поиска позволяет варьировать размер популяции от генерации к генерации, что позволяет частично предотвращать преждевременную сходимость алгоритмов в задачах конструкторского проектирования [Курейчик и др., 2012б]. Можно предложить большое число аналогичных схем поиска. Их эффективность проверяется экспериментальным путем.

2. Муравьиный алгоритм

Одним из биоинспирированных подходов является метод роевого интеллекта, включающий в себя муравьиные алгоритмы, пчелиные алгоритмы, метод роя частиц и др. [Курейчик и др., 2009] [Dorigo, 1996].

Приведем модифицированный муравьиный алгоритм (рис.2), ориентированный на решение задач конструкторского проектирования.

Муравьиные алгоритмы представляют собой вероятностную жадную эвристику, где вероятности устанавливаются, исходя из информации о качестве решения, полученного из предыдущих решений [Dorigo, 1996]. Муравьи относятся к социальным насекомым, живущим внутри некоторого колLECTива – колонии. Основу «социального» поведения муравьев составляет самоорганизация – множество динамических механизмов, обеспечивающих достижение системой глобальной цели в результате низкоуровневого взаимодействия ее элементов. Принципиальной особенностью такого взаимодействия является использование элементами системы только локальной информации. При этом исключается любое централизованное управление.

В сравнении с ГА муравьиные алгоритмы имеют некоторые преимущества: опираются на память всей колонии вместо памяти только о предыдущем поколении и меньше подвержены неоптимальным начальным решениям (из-за случайного выбора пути и памяти колонии).

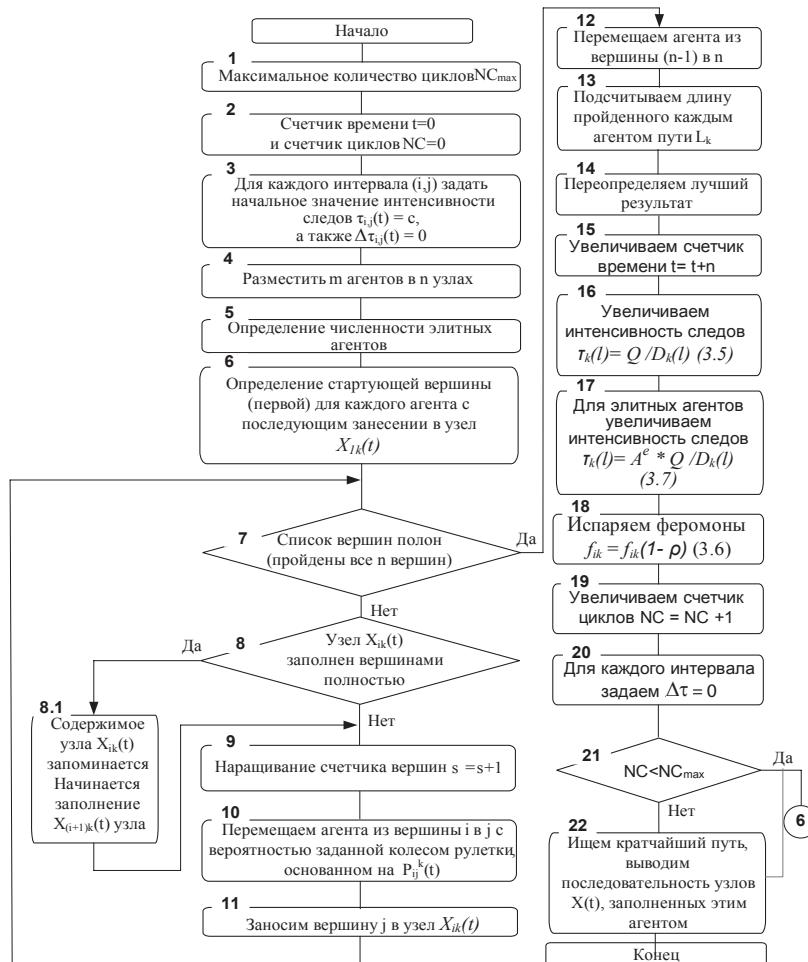


Рис.2. Модифицированный муравьиный алгоритм

Ряд экспериментов показывает, что эффективность муравьиных алгоритмов растет с ростом размерности решаемых задач оптимизации и для нестационарных систем с изменяемыми во времени параметрами.

Важным свойством муравьиных алгоритмов является неконвергентность: даже после большого числа итераций одновременно

исследуется множество вариантов решения, что частично решает проблему предварительной сходимости. Перспективными путями улучшения муравьиных алгоритмов являются их гибридизация с другими методами природных вычислений, например генетическими алгоритмами [Курейчик и др., 2012b]. Гибридизация может осуществляться по островной схеме, когда различные алгоритмы решают задачу параллельно и автономно, или по принципу «мастер-подмастерье», когда основной алгоритм – «мастер» передает решение типовых подзадач «подмастерью».

3. Пчелиный алгоритм

Пчелиный алгоритм – это оптимизационный алгоритм, в основе которого лежит поведение пчел в живой природе [Karaboga, 2005] [Запорожец и др., 2010]. Применительно к задачам конструкторского проектирования и оптимизации в пчелином алгоритме каждое решение представляется в виде пчелы, которая знает (хранит) расположение (координаты или параметры многомерной функции) какого-то участка. Выделим два варианта поведения пчел.

В первом варианте две пчелы нашли два разных пересекающихся участка, и оба этих участка следует отметить как лучшие или выбранные. Во втором варианте будем считать, что один участок, центр которого находится в точке, соответствующей большому значению целевой функции. Второй вариант поведения менее подвержен попаданию в локальные оптимумы за счет просмотра перспективных мест и их окрестностей. Причем на каждой итерации алгоритма область просмотра уменьшается.

Основная идея пчелиного алгоритма заключается в том, что все пчелы на каждом шаге будут выбирать как элитные участки для исследования, так и участки в окрестности элитных, что позволит, во-первых, разнообразить популяцию решений на последующих итерациях, во-вторых, увеличить вероятность обнаружения близких к оптимальным решений. После чего в окрестности остальных участков, в зависимости от значения их ЦФ, отправляются рабочие пчелы. Таким образом, работа алгоритма зависит от следующих основных параметров: общего числа пчел-разведчиков; общего числа участков; числа элитных участков; числа пчел-разведчиков на элитных участках; числа пчел на остальных участках; началь-

ного размера участков, вместе с их окрестностями; максимального числа итераций.

Приведем словесное описание алгоритма пчел [Запорожец и др., 2010].

1° В соответствии с постановкой задачи проектирования и исходными данными формируется популяция пчел (хромосом);

2° Отправка пчел-исследователей. Определение месторасположения источников нектара. Для каждой пчелы случайным образом задается начальная позиция;

3° Оценка ЦФ пчел в популяции. Выбор источника нектара пчелой-исследователем с определенной вероятностью, в зависимости от его качества. Для каждой пчелы определяется лучший (элитный) участок, и значение целевой функции на этом участке. Участки, на которых значения ЦФ больше, отбираются для поиска решений в их окрестностях;

4° Выбор пчел с лучшими значениями ЦФ с каждого источника;

5° Если решение на исследуемом участке не улучшается с течением нескольких итераций, переход к п.6, иначе п.3;

6° Отправка пчел-разведчиков, осуществляющих случайный поиск, и оценка их ЦФ;

7° Формирование новой популяции пчел, в состав которой будут входить как пчелы с лучшими значениями ЦФ с элитных участков, так и пчелы со случайными значениями ЦФ;

8° Проверка условий окончания алгоритма. Если они выполняются, переход к п. 9, иначе п. 2;

9° Конец работы алгоритма.

Таким образом, ключевой операцией алгоритма пчел является совместное исследование перспективных областей и их окрестностей. Отличительной особенностью алгоритма является способность динамически разбивать поисковое пространство на области, что уменьшает время работы алгоритма. Данный алгоритм иллюстрирует стратегию поиска «Разделяй и властвуй». Главным преимуществом является тот факт, что резко снижается вероятность попадания в локальный оптимум, а за счет распараллеливания уменьшается время.

Пчелиный алгоритм, в отличие от ГА, имеет лишь один оператор и легко распределяется на несколько параллельных процессов, за счет чего значительно увеличивается его скорость.

Отметим, что пчелиный алгоритм основывается на социальном поведении роя, а генетический алгоритм имитирует процесс эволюции и отбора. За счет этого есть возможность комбинирования этих методов.

Заключение

Отличительной особенностью этих алгоритмов является то, что на их основе можно создавать многоуровневые, комбинированные и гибридные архитектуры поиска. Это позволяет распараллеливать процесс поиска, получать оптимальные и квазиоптимальные результаты за полиномиальное время и частично устраниТЬ преждевременную сходимость алгоритмов. Разработана программная среда. Проведен вычислительный эксперимент. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов конструкторского проектирования и их поведение для схем различной структуры. В лучшем случае времененная сложность алгоритмов $\approx O(n \log n)$, в худшем случае – $O(n^2)$.

СПИСОК ЛИТЕРАТУРЫ

- [Курейчик и др., 2009] Курейчик В.В., Курейчик В.М., Гладков Л.А., Сороколетов П.В. Бионспириванные методы в оптимизации. – М.: Физматлит, 2009.
- [Гладков и др., 2010] Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010.
- [Курейчик и др., 2012а] Курейчик В.В., Курейчик В.М., Родзин С.И. Теория эволюционных вычислений. – М.: Физматлит, 2012.
- [Курейчик и др., 2012б] Курейчик В.В., Курейчик Вл.Вл. Архитектура гибридного поиска при проектировании//Известия Южного федерального университета. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2012, № 7 (132). С. 22–27.
- [Dorigo, 1996] Dorigo M., Maniezzo V., Colorni A. The Ant System: Optimization by a colony of cooperating objects // IEEE Trans. on Systems, Man, and Cybernetics. –1996. – Part B. – N 26(1) – pp. 29–41.
- [Karaboga, 2005] Karaboga, D. An idea based on honey bee swarm for numerical optimization // Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [Запорожец и др., 2010] Запорожец Д.Ю., Курейчик В.В. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2010, № 7 (108), с. 28–32.

УДК 519.712.2

РЕШЕНИЕ ЗАДАЧИ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ СХЕМ ЭВА НА ОСНОВЕ НЕЧЕТКОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Л.А. Гладков (leo_gladkov@mail.ru)
Н.В. Гладкова (nadyusha.gladkova77@mail.ru)
С.Н. Лейба (leo@tgn.sfedu.ru)
Южный федеральный университет

В работе рассматривается задача размещения элементов схем электронно-вычислительной аппаратуры (ЭВА) на коммутационном поле. Данная задача относится к задачам конструкторского этапа проектирования ЭВА и является NP-полной задачей. В статье предлагается подход к решению поставленной задачи на основе интеграции генетических алгоритмов поиска и нечетких моделей управления параметрами алгоритма. Дано описание структуры используемого нечеткого логического контроллера. Приведено краткое описание программы и результатов ее тестирования, подтверждающих эффективность предложенного метода.

Введение

Проектирование СБИС представляет собой многоуровневый процесс, где на каждом уровне используется свое математическое и программное обеспечение. Одним из важных этапов процесса проектирования СБИС является этап конструкторского проектирования. Задачи конструкторского проектирования, как правило, характеризуются большой вычислительной сложностью, обусловленной необходимостью перебора огромного числа различных вариантов решений. Причем для получения точного решения требуется выполнить полный перебор, что не представляется возможным. К числу задач конструкторского этапа относятся задачи разбиения, размещения, трассировки и др. [Норенков, 2006].

Одним из направлений повышения эффективности решения задач автоматизированного проектирования сложных технических систем, в том числе СБИС, содержащих миллионы компонентов, являются эволюционные и генетические алгоритмы, другие инспирированные природой методы и подходы [Cohoon et al., 2003].

Новым этапом развития теории генетических алгоритмов стали гибридные системы. Они основаны на совмещении различных научных направлений, например, таких как генетические алгоритмы, нечеткие системы и нейронные сети. Существуют различные способы гибридизации этих трех подходов. Один из них – нечеткие генетические алгоритмы [Heggara et al., 2003], в которых методы нечеткой математики используются для настройки параметров ГА. Также используют нечеткие операторы и нечеткие правила для создания генетических операторов с различными свойствами; системы нечеткого логического контроля параметров ГА в соответствии с принятыми критериями; нечеткие критерии остановки процесса генетического поиска. Математический аппарат теории нечетких систем используется в данном случае для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов.

1. Постановка задачи

Исходной информацией при решении задач размещения являются: данные о конфигурации и размерах коммутационного пространства, определяемые требованиями установки и крепления данной сборочной единицы в аппаратуре; количество и геометрические размеры конструктивных элементов, подлежащих размещению; схема соединений, а также ограничения на взаимное расположение отдельных элементов, учитывающие особенности разрабатываемой конструкции [Shervani, 1995]. Задача сводится к отысканию для каждого размещаемого элемента таких позиций, при которых оптимизируется выбранный показатель качества и обеспечиваются наиболее благоприятные условия для последующего электрического монтажа.

К числу критериев, которые могут использоваться для оценки качества размещения, можно отнести следующие критерии:

- 1) суммарная взвешенная длина соединений;
- 2) длина самой длинной связи;
- 3) число соединений, длина которых больше заданной;
- 4) число изгибов соединений;
- 5) число соединений между элементами, находящимися в соседних позициях либо в позициях, указанных разработчиком;

6) площадь кристалла.

Наибольшее распространение в алгоритмах размещения получил первый критерий, что объясняется следующими причинами: уменьшение длины связей позволяет улучшить электрические характеристики схемы, упрощает процесс последующей трассировки; кроме того, он сравнительно прост в реализации [Гладков, 2011].

Модель оптимизационной задачи может быть описана кортежем длины 3 следующего вида: $\langle \mathbf{X}, \mathbf{D}, \mathbf{Q} \rangle$, где \mathbf{X} – множество (пространство) решений (хромосом) для данной задачи, \mathbf{D} – ограничения, накладываемые на множество \mathbf{X} , для выделения допустимых решений и \mathbf{Q} – целевая функция (ЦФ), с помощью которой можно определить наилучшее (оптимальное) решение [Гладков и др., 2010а].

Пусть дано множество элементов $E = \{e_i \mid i=1, \dots, n\}$ и множество $P = \{p_i \mid i=1, \dots, c\}$ – множество позиций на коммутационном поле. Для размещения всех элементов необходимо выполнение условия $c \geq n$. Произвольное размещение элементов в позициях представляет собой перестановку $P = P_{(1)}, \dots, P_{(i)}, \dots, P_{(n)}$, где $P_{(i)}$ – номер позиции, в которую назначен элемент e_i . В зависимости от выбранного критерия для оценки результатов размещения вводится целевая функция $F(P)$. Таким образом, задача размещения состоит в отыскании оптимального значения функции F на множестве перестановок P .

2. Описание генетического алгоритма

Для решения задачи размещения будет использован простой генетический алгоритм. Начальная популяция будет задаваться методом дробовика, будет использована селекция методом рулетки, и будут применяться одноточечный оператор кроссинговера и двухточечный оператор мутации [Гладков и др., 2010а], [Курейчик, 2009].

Множество позиций представим в виде регулярной структуры (*решетки*). Каждая позиция p_i имеет координаты x_i, y_i . Первая позиция имеет координаты (1; 1). Установим взаимно-однозначное соответствие между множеством порядковых номеров позиций и множеством координат. С этой целью упорядочим и пронумеруем позиции по возрастанию параметра $W_i = x_i + y_i$. Так как позиции, расположенные на одной диагонали в решетке, имеют одно и то

же значение W_i , то в пределах диагонали они упорядочиваются и нумеруются по убыванию значения параметра $Q_i = x_i - y_i$.

Каждое решение R_i представляется в виде гомологичной хромосомы H_i . Порядковый номер гена в хромосоме соответствует порядковому номеру позиции на коммутационном поле. Значение гена соответствует номеру элемента, который размещен в позиции, соответствующей порядковому номеру гена. Число генов в хромосоме равно n . Если фактическое число размещаемых элементов меньше числа позиций, то вводятся «пустые элементы» (не имеющие связей) так, чтобы общее число элементов было равно числу позиций.

При реализации приложения будут использоваться разногабаритные элементы, имеющие прямоугольную форму, поскольку элемент любой геометрической формы может быть описан прямоугольником.

Каждый элемент имеет базовую точку O_i^δ и базовые оси координат $O_i^\delta X_i^\delta$, $O_i^\delta Y_i^\delta$, относительно которых задается контурное описание элемента a_i . Будем считать, что элемент a_i назначен в позицию p_j , если его базовая точка O_i^δ совмещена с точкой коммутационного поля, имеющей координаты (x_j, y_j) .

Например, хромосома

$<4\ 5\ 6\ 9\ 2\ 3\ 8\ 1\ 7>$

задает следующее размещение (рис. 1). Точки пересечения пунктирных линий соответствуют посадочным позициям на коммутационном поле.

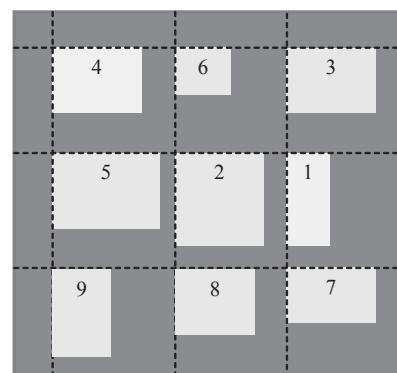


Рис. 1. Пример размещения элементов на плоскости

Для создания начальной популяции используется метод «дробовика». Он подразумевает случайный выбор альтернатив из всей области решений данной задачи [Гладков и др., 2010а].

При расчете значений ЦФ рассчитывается суммарная длина связей между точками, являющимися центрами тяжести описываемых прямоугольников (пересечение диагоналей).

$$F = \sum_{i=0}^n \sum_{j=0}^{n-i} k_{ij} \sqrt{(x_i + (l_i/2) - (x_j + (l_j/2)))^2 + (y_i + (h_i/2) - (y_j + (h_j/2)))^2},$$

где x_i, x_j, y_i, y_j – координаты позиций на коммутационном поле, в которых размещены элементы i и j ;

l_i, l_j, h_i, h_j – габариты элементов i и j относительно базовой точки;

k_{ij} – количество связей между парой элементов i и j ;

n – общее число элементов.

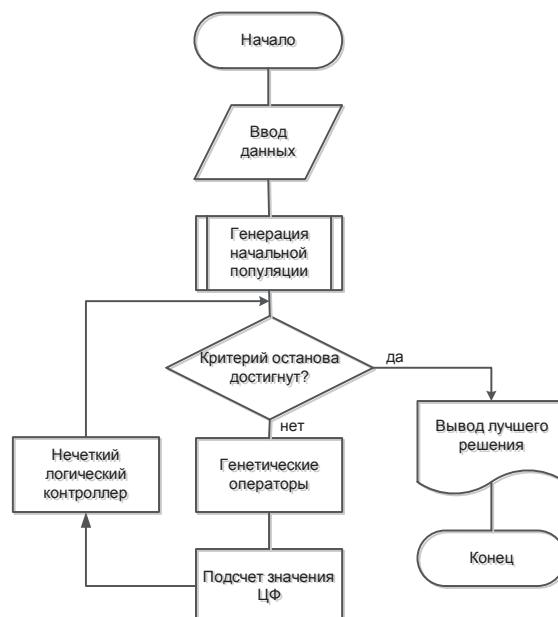


Рис. 2. Структурная схема генетического алгоритма размещения

После расчета ЦФ к популяции решений применяются генетические операторы. Перечислим эти операторы. Селекция на основе

«рулетки» – это простой и широко используемый метод. При его реализации каждому элементу в популяции соответствует зона на колесе рулетки, пропорциональная величине ЦФ. Тогда при повороте колеса рулетки элемент с большим значением ЦФ имеет большую вероятность для выбора. Также используются упорядоченный оператор кроссинговера, оператор многоточечной мутации, при котором число точек разреза пропорционально длине хромосомы [Гладков и др., 2010а].

Таким образом, с учетом использования блока нечеткого логического контроллера (НЛК) структурную схему генетического алгоритма размещения можно представить следующим образом (рис. 2).

3. Блок управления параметрами алгоритма

Рассмотрим вкратце работу блока управления параметрами генетического алгоритма (нечеткого логического контроллера). Он состоит из следующих блоков.

База правил. Знания, используемые для корректного функционирования модуля нечеткого управления, записываются в виде нечетких правил типа [Ярушкина, 2004]:

$$R^k: \text{IF}(x_1 \text{ это } A_1^k \text{ AND } \dots \text{ AND } x_n \text{ это } A_n^k) \text{ THEN}(y \text{ это } B^k).$$

Можно также представить эти знания в виде нечетких множеств с функцией принадлежности, заданной выражением:

$$\mu_{R(k)}(x, y) = \mu_{A^k \rightarrow B^k}(x, y).$$

Блок вывода. Конкретная форма функции принадлежности нечеткого множества зависит от применяемой Т-нормы, определения нечеткой импликации и от способа задания декартова произведения нечетких множеств.

Блок фазификации. Значению входного параметра НЛК $\bar{x} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n)^T \in X$ в результате операции фазификации становится в соответствие нечеткое множество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$.

Блок дефазификации. Решается задача отображения нечетких множеств B^k (либо нечеткого множества B') в единственное значение $y \in Y$, которое представляет собой управляющее воздействие, действующее на объекты управления НЛК. В алгоритме

применяется метод дефазификации по среднему центру (*center average defuzzification*) [Ярушкина и др., 2007].

Завершающий этап в процессе проектирования модуля нечеткого управления – это определение формы представления нечетких множеств A_i^k , $i = 1, \dots, n$; $k = 1, \dots, N$. Например, функция принадлежности

$$\mu_{A_i^k}(x) = \exp\left(-\left(\frac{x_i - \bar{x}_i^k}{\sigma_i^k}\right)^2\right),$$

где параметр \bar{x}_i^k – это центр, а σ_i^k – ширина функции принадлежности (функции Гаусса). Эти параметры могут изменяться в процессе обучения, что позволяет изменять положение и структуру нечетких множеств.

Для повышения качества результатов генетического поиска решается задача включения в контур эволюции экспертной информации путем построения НЛК, регулирующего значения параметров процесса эволюции на основе экспертных знаний.

В качестве входных параметров используется среднее и лучшее значение ЦФ.

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)}, \quad e_2(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{best}(t)},$$

где t – номер текущей итерации; $f_{max}(t)$ – лучшее значение ЦФ на итерации t ; $f_{ave}(t)$ и $f_{ave}(t-1)$ – среднее значение ЦФ на итерации t и $(t-1)$ соответственно [Гладков и др., 2010б].

На основании оценки динамики значений e_1 и e_2 изменяются значения вероятностей выполнения операторов кроссинговера и мутации

$$\begin{aligned} P_c(t) &= P_c(t-1) + \Delta P_c(t), \\ P_m(t) &= P_m(t-1) + \Delta P_m(t). \end{aligned}$$

4. Программная реализация алгоритма

Для разработки приложения была выбрана среда Microsoft Visual Studio 2010. Был разработан простой интерфейс программы, отображающий основные данные, которые необходимо выводить в процессе выполнения программы. Это график, на котором отображается среднее и минимальное значение ЦФ, а также текстовое поле, в которое выводились различные данные в процессе отладки приложения.

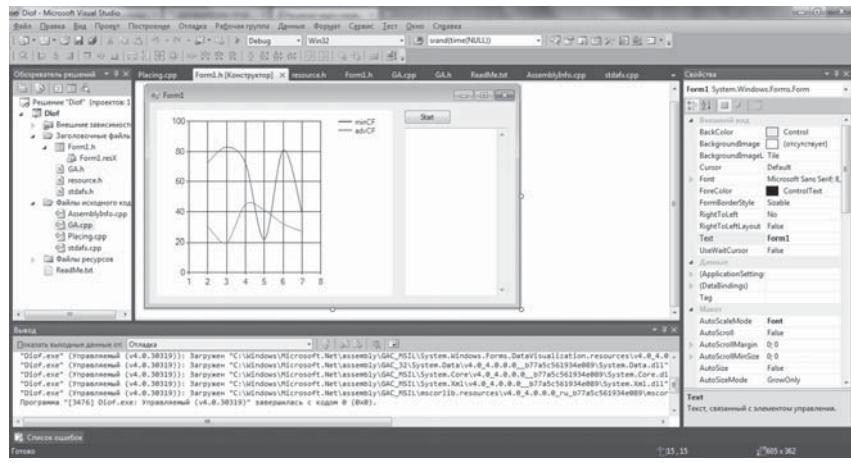


Рис. 3. Окно программы

Для сравнения эффективности исследовались тестовые задачи, решаемые с использованием НЛК и без него. Из рис. 4 (а и б) видно, что эффективность алгоритма с контроллером значительно выше, чем эффективность алгоритма без него. Эффективность использования контроллера повышается после введения блока обучения на основе модели искусственной нейронной сети. Значения параметров коэффициентов (x_i^k , σ_i^k и y^k), используемых при работе ИНС, задаются случайно или на основе направленного поиска (для этого можно использовать генетический алгоритм).

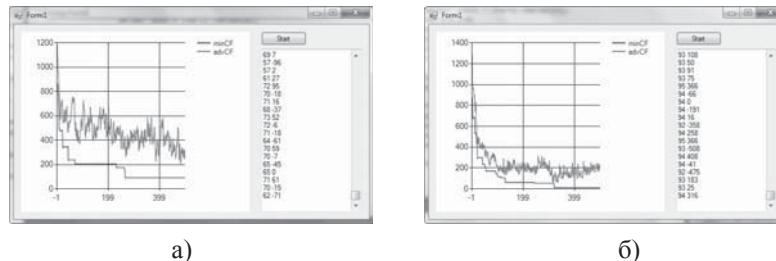


Рис. 4. Эффективность работы алгоритма без использования (а) и с использованием НЛК

Благодарности. Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-00829).

СПИСОК ЛИТЕРАТУРЫ

- [Норенков, 2006] Норенков И.П. Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Баумана, 2006.
- [Cohoon et al., 2003] Cohoon J.P., Karro J., Lienig J. Evolutionary Algorithms for the Physical Design of VLSI Circuits. Advances in Evolutionary Computing: Theory and Applications, Ghosh, A., Tsutsui, S. (eds.) Springer Verlag, London, 2003. – pp. 683–712.
- [Herrera et al., 2003] Herrera F., Lozano M. Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions. // Soft Computing 7(2003), Springer-Verlag, 2003. – p. 545–562.
- [Shervani, 1995] Shervani N. Algorithms for VLSI physical design automation. – USA, Kluwer Academy Publisher, 1995. – 538 р.
- [Гладков, 2011] Гладков Л.А. Интегрированный алгоритм решения задач размещения и трассировки на основе нечетких генетических методов // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР», № 7 (120), 2011. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – С. 22–30.
- [Гладков и др., 2010а] Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010.
- [Курейчик, 2009] Курейчик В.М. Модифицированные генетические операторы // Известия ЮФУ. Технические науки, № 12.– Таганрог: Изд-во ТТИ ЮФУ, 2009. – С. 7–15.
- [Ярушкина, 2004] Ярушкина Н.Г. Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004.
- [Ярушкина и др., 2007] Батыршин И.З., Недосекин А.О. и др. Нечеткие гибридные системы. Теория и практика / Под ред. Н.Г. Ярушкиной. – М.: Физматлит, 2007.
- [Гладков и др., 2010б] Гладков Л.А., Гладкова Н.В. Новые подходы к построению систем анализа и извлечения знаний на основе гибридных методов. // Известия ЮФУ. Технические науки, № 7 (108), Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТТИ ЮФУ 2010. – С. 146–154.

УДК 004.89+004.021

ИСПОЛЬЗОВАНИЕ САМОАДАПТАЦИИ РАЗМЕРА ПОПУЛЯЦИИ В ГЕНЕТИЧЕСКИХ АЛГОРИТМАХ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ

П.В. Казаков (*pvk_mail@list.ru*)

Брянский государственный технический университет, Брянск

В работе описывается оригинальный способ повышения эффективности генетических алгоритмов многокритериальной оптимизации посредством динамического управления размером популяции. Предлагаемый способ представляет собой модификацию для любых эволюционных методов многокритериальной оптимизации. Его эффективность исследована при решении набора тестовых задач многокритериальной оптимизации разной сложности.

Введение

Для любого эволюционного алгоритма размер популяции является одним из главных параметров. От его значения зависит как скорость, так и точность поиска: недостаточный размер популяции ограничивает объем исследуемого пространства поиска, приводит к быстрому схождению алгоритма только к локально оптимальным решениям; слишком большой – замедляет поиск, снижает разнообразие популяции и эффект от оператора отбора. Выбор размера популяции является особенно важным в работе генетических алгоритмов многокритериальной оптимизации (МГА) [Deb et al., 2009], так как определяет результирующее множество и число найденных оптимальных решений. Эффективное управление популяцией подразумевает сохранение баланса между уже найденными хромосомами, кодирующими оптимальные решения, и хромосомами, предполагающими их дальнейшее изменение для продолжения исследования поискового пространства. При этом в составе популяции должно поддерживаться разнообразие в течение всего процесса оптимизации. Очевидно, что разные участки пространства поиска имеют отличающиеся по сложности топологии, требующие для их исследования различное число хромосом. Однако традиционное использование в МГА фиксированного размера популяции не позволяет изменять плотность распределения хромосом по поис-

ковому пространству без нарушения их баланса в популяции и сохранения ее разнообразия. Данное ограничение может быть снято при использовании в МГА изменяемого (самоадаптирующегося) размера популяции в зависимости от характера поиска и свойств уже найденных решений. Такая модификация позволит расширить функциональные возможности генетического алгоритма и снизить вероятность его схождения к локально-оптимальным решениям.

1. Модель динамического изменения размера популяции

Модель динамического изменения размера популяции (Self-Adaptive Population Size – SAPS) использует концепцию хромосомы, имеющей возраст и фиксированное время жизни [Michalewicz, 1996], [Гладков и др., 2002]. Возраст хромосомы c_i $age(c_i, t)$ $i = 1, \dots, n_p(t)$, где $n_p(t)$ – размер популяции в поколении t , совпадает с числом поколений, которые остаются в популяции и зависит только от значения пригодности хромосомы. В SAPS в явном виде не используется оператор отбора, что позволяет соответствующим хромосомам большее число поколений оставаться в популяции. С одной стороны, это приводит к изменению численности популяции, в частности, ее росту, с другой – к сохранению в ней множества Парето-оптимальных решений [Ногин, 2004], [Соболь и др., 2006]. Для размещения получаемых в результате новых хромосом на каждом поколении t производится увеличение размера популяции на величину $n_p^+(t) = n_p(t) \cdot \rho$, где $\rho \in (0, 1]$ – коэффициент репродукции. Кроме текущего возраста с каждой хромосомой связывается параметр времени жизни (LifeTime) $LT(c_i, t), i = 1, \dots, n_p(t)$. Его значение ограничивает число поколений существования хромосомы. В модели SAPS значение $LT(c_i, t), i = 1, \dots, n_p(t)$ вычисляется каждый раз после оценки пригодности хромосомы, для чего получены следующие зависимости $\forall c_i, i = 1, \dots, n_p(t)$:

$$\left\{ \begin{array}{l} LT(c_i, t) = age(c_i, t) + \text{round}\left(\frac{f(c_i, t) - f(c_i, t)^{\min}}{f(c_i, t)^{\max} - f(c_i, t)^{\min}}\right), \text{ если } rank(c_i, t) = 1; \\ LT(c_i, t) = \frac{1 + \max\{rank(c_i, t)\}}{2} + (rank(c_i, t-1) - rank(c_i, t)) \cdot \\ \times \frac{f(c_i, t) - f(c_i, t)^{\min}}{f(c_i, t)^{\max} - f(c_i, t)^{\min}}, \text{ иначе.} \end{array} \right. \quad (1)$$

Здесь $rank(c_i, t), i = 1, \dots, n_p(t)$ – ранг Парето хромосомы в поколении t . У недоминируемых хромосом ранг равен 1. Значение $f(c_i, t), i = 1, \dots, n_p(t)$ определяет пригодность каждой хромосомы, предполагается, что большее ее значение соответствует лучшему показателю. Параметры $f(c_i, t)^{\min}, f(c_i, t)^{\max}, i = 1, \dots, n_p(t)$ представляют собой соответственно минимальную и максимальную пригодность среди хромосом поколения t . Для начальной популяции ($t = 0$) устанавливается, что $LT(c_i, t) = rank(c_i, t), i = 1, \dots, n_p(t)$.

Таким образом, время жизни хромосомы зависит только от ее ранга Парето и значения пригодности. Решения, представляемые хромосомами с рангом Парето, равным единице, относительно данной популяции являются оптимальными. Поэтому для таких хромосом время жизни вычисляется по отдельной зависимости (первое выражение системы (1)). Согласно ей, каждая недоминируемая хромосома с пригодностью выше средней гарантированно переходит в следующее поколение. Для остальных хромосом наряду с пригодностью при определении времени жизни учитывается изменение ранга Парето в сравнении с предыдущим поколением. Если ранг улучшился (уменьшился), то соответствующие хромосомы могут увеличить время своего пребывания в популяции и тем самым вероятность участия в операторе кроссинговера.

Известно, что ранг Парето во всех МГА вычисляется одинаково [Deb et al., 2009], в то время как пригодность хромосомы зависит от используемого в алгоритме выражения fitness-функции. В ней могут учитываться различные показатели качества решения, например, особенности его расположения на границе Парето, плотность его соседства на ней с другими точками и т.п. В итоге хромосомы с одинаковым рангом, но разной пригодностью могут иметь различное время жизни. Таким образом, в следующее поколение преимущественно переходят те хромосомы, у которых ранг Парето равен единице, либо он улучшился, а значение их пригодности при этом выше среднего. Также предлагаемая модель вычисления времени жизни хромосом отличается достаточно низкой вычислительной сложностью, не использует не относящихся к базовому МГА специальных переменных и функций. Для МГА это важный фактор, так как параметр $LT(c_i, t), i = 1, \dots, n_p(t)$ должен вычисляться для каждой хромосомы каждого поколения. После вычисления значения $LT(c_i, t), i = 1, \dots, n_p(t)$ популяция со-

кращается посредством удаления из нее всех хромосом, у которых $age(c_i, t) > LT(c_i, t), i = 1, \dots, n_p(t)$.

Предлагаемый способ динамического управления размером популяцией может быть использован в любом МГА. В этом случае важное влияние на точность определения множества Парето будет оказывать основной управляемый параметр SAPS – коэффициент репродукции. Его значение может быть определено экспериментально в зависимости от особенностей решаемой задачи МО.

2. Выбор значения коэффициента репродукции

Для анализа влияния различных значений коэффициента репродукции на изменение размера и состава популяции был проведен ряд экспериментов. В них применялся один из наиболее известных генетических алгоритмов МО NSGA-II [Deb et al., 2002] с модификацией SAPS (NSGA-II+SAPS) для решения тестовой задачи МО DTLZ1 [Казаков, 2012б]. Число критериев m задавалось равным двум, анализировалось множество различных значений параметра $\rho = \{0,1; 0,3; 0,5; 0,8\}$. Для каждого из них задавались одинаковые размер начальной популяции $n_p(t=0) = 100$, число поколений, а также значения вероятностей операторов кроссинговера и мутации. В процессе поиска множества Парето оценивались динамика изменения размера популяции (рис. 1), число и доля в ней недоминируемых решений. Анализ полученных результатов позволил сделать следующие выводы.

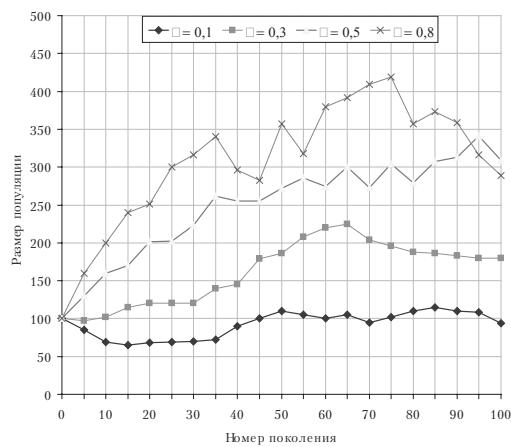


Рис.1. Изменение размера популяции для разных значений коэффициента репродукции

1. Для малых значений коэффициента репродукции ($\rho < 0,3$) не наблюдается заметного роста численности популяции. Она сначала снижается, далее растет, но достигает и превышает свой начальный размер, только начиная со второй половины поколений. При этом доля недоминируемых решений в популяции относительно высока (более 30%) и почти не изменяется в течение всего поиска. Это позволяет сделать вывод о его быстрой стагнации из-за недостаточного разнообразия популяции - преобладания в ней одинаковых, постоянно повторяющихся хромосом.

2. С ростом значения коэффициента репродукции ($\rho \geq 0,3$) у популяции заметен рост численности уже в первой трети поколений. При достижении $\rho = 0,5$ популяция, а также число в ней недоминируемых решений быстро удваиваются. В последней третьи поколений число и доля недоминируемых решений для $\rho = 0,3$ практически не изменяется, что свидетельствует о схождении алгоритма в целом. В сравнении с этим, изменение числа и доли недоминируемых решений при $\rho = 0,5$ в этот интервал времени все еще подвержено заметным колебаниям, не позволяющим сделать окончательный вывод о схождении алгоритма к заданному числу поколений.

3. Дальнейшее увеличение значения коэффициента репродукции ($\rho > 0,5$) привело к быстрому росту численности популяции в самом начале работы МГА. Впоследствии динамика изменения размера популяции и число недоминируемых решений в ней принимают хаотичный характер. Такое свойственно исключительному применению оператора мутации для получения новых членов популяции. Соотношение размера популяции и числа недоминируемых решений в ней носит преимущественно случайный характер. Это позволяет сделать вывод о существенных проблемах схождения МГА с такими значениями коэффициента репродукции, причем не только в условиях решаемой задачи МО.

В итоге было выявлено, что для решаемой тестовой задачи наилучший баланс между скоростью роста популяции и числом недоминируемых решений в ней достигается при $\rho \in [0,2, \dots, 0,4]$. В то же время решение рассматриваемой задачи МО при тех же условиях экспериментов, но с большим числом критериев ($m = 8$) показало оправданность использования коэффициента репродукции $\rho > 0,5$ ($\rho = 0,7$). Обосновать это можно нелинейным увеличением скоро-

сти заполнения популяции недоминируемыми решениями при росте числа оптимизируемых критериев. Тем самым только возможность быстрого роста численности популяции в моменты времени процесса оптимизации, когда требуется увеличение разнообразия популяции, позволяет сохранить баланс между исследованными и неисследованными участками пространства поиска.

3. Исследование эффективности модификации SAPS

Для оценки эффективности использования предложенной модели самоадаптации размера популяции был проведен ряд экспериментов. Они заключались в решении набора тестовых задач многокритериальной оптимизации (DTLZ1, DTLZ2, DTLZ3, DTLZ6) [Казаков, 2012б] с различным числом критериев. Это позволило оценить влияние модификации SAPS на сохранение эффективности базовых МГА при росте сложности решаемых задач. В качестве таких МГА использовались наиболее известные SPEA2(Strength Pareto Evolutionary Algorithm)[Zitzler et al., 2002] и NSGA-II(Nondominated Sorting Genetic Algorithm) [Deb et al., 2002].

Результаты многокритериальной оптимизации оценивались по нескольким показателям качества. Для их вычисления использовался набор индикаторов [Казаков, 2012а], позволяющий оценить различные свойства найденного МГА множества Парето в пространстве решений или критериев:

I_{ONVG} (Overall Nondominated Vector Generation) – определяет мощность найденного множества Парето;

I_s (Spacing) – используется для оценки равномерности распределения решений вдоль границы Парето;

I_{DE} (Dimensions Extent) – позволяет оценить максимальную протяженность границы Парето по каждой из размерностей;

I_{GD} (Generational Distance) – позволяет оценить степень близости между полученной и заданной эталонной границами Парето;

I_{OT} (Overall Time Computing) – ориентирован на оценку временной эффективности МГА при определении множества Парето.

Для каждого МГА экспериментально были подобраны значения управляющих параметров: размеры популяции и Парето-архива, число поколений, а также вероятности кроссинговера и

мутации. Значения для всех алгоритмов совпадали и зависели от решаемой задачи [Казаков, 2012b]. Результаты решения всех задач приведены в табл. 1. Значения всех индикаторов усреднены по выполненному числу запусков МГА. Значения показателей, которые удалось улучшить, используя модификацию SAPS, выделены полужирным.

Таблица 1

Задача	Индикатор	SPEA2	NSGA-II	SPEA2 +SAPS	NSGA-II + SAPS
DTLZ1 ($m = 2$)	I_{ONVG}	28	35	28	66
	I_{GD}	0,071	0,067	0,068	0,062
	I_S	0,186	0,223	0,188	0,214
	I_{DE}	0,971	0,964	0,968	0,973
	I_{OT} (сек.)	6,3	4,8	9,3	7,4
DTLZ2 ($m = 4$)	I_{ONVG}	113	138	113	179
	I_{GD}	5,732	6,124	5,441	6,132
	I_S	0,133	0,124	0,121	0,125
	I_{DE}	1,714	1,847	1,793	1,892
	I_{OT} (сек.)	191,4	31,6	267,8	48,5
DTLZ3 ($m = 6$)	I_{ONVG}	188	207	208	236
	I_{GD}	226,953	310,603	231,762	314,547
	I_S	0,328	0,287	0,297	0,276
	I_{DE}	2,137	2,312	2,167	2,376
	I_{OT} (сек.)	6217,3	698,7	10342,7	1095,4
DTLZ6 ($m = 8$)	I_{ONVG}	319	371	360	401
	I_{GD}	16,623	11,237	14,547	11,142
	I_S	0,251	0,201	0,247	0,196
	I_{DE}	1,862	1,913	2,034	2,173
	I_{OT} (сек.)	76784,3	2843,8	128132,5	4546,4

Как видно из таблицы, основной эффект от использования модификации SAPS связан с увеличением во всех случаях числа найденных Парето-оптимальных решений. Это, в свою очередь,

повлияло на улучшение значений ряда других индикаторов. В частности, I_{DE} , предназначенного для оценки протяженности границы Парето. В отдельных случаях улучшились результаты по индикаторам I_{GD} и I_S , что, вероятно, вызвано увеличением плотности заполнения границы Парето большим числом решений, чем у оригинальных SPEA2 и NSGA-II. В то же время ни одна из модифицированных версий этих МГА не оказалась лучшей во всех задачах. Однако заметим, что в двух из них (DTLZ1, DTLZ6) алгоритм NSGA-II+SAPS полностью превзошел SPEA2+SAPS. Также обе версии NSGA-II во всех экспериментах оказались быстрее своего конкурента. Причем с ростом числа критериев разница во времени между SPEA2 и NSGA-II становится существенной. Вместе все это позволяет сделать вывод о предпочтительности использования предложенной модели динамического изменения размера популяции именно в NSGA-II.

Заключение

Предложенный способ динамического управления размером популяции МГА в процессе многокритериальной оптимизации позволяет обеспечить не только поддержание разнообразия популяции за счет постоянного обновления времени жизни хромосом, но и увеличить число найденных Парето-оптимальных решений. Это особенно важно, когда их число ограничивается первоначально заданным фиксированным размером популяции. Так, использование созданной модификации в МГА SPEA2 и NSGA-II позволило достичь увеличения числа найденных Парето-оптимальных решений в сочетании с улучшением качества связанный с ними границы Парето. Рост временной сложности при интеграции в МГА механизмов динамического управления численностью популяции зависит от сложности топологии пространства критериев, а также выбора значения для единственного дополнительного управляющего параметра (коэффициент репродукции).

СПИСОК ЛИТЕРАТУРЫ

[Гладков и др., 2002] Гладков Л.А., Зинченко Л.А., Курейчик В.В., Курейчик В.М., Лебедев Б.К., Нужнов Е.В., Сорокин С.Н. Методы генетического поиска. – Таганрог: Изд-во ТРТУ, 2002.

[Казаков, 2012а] Казаков П.В. Оценка эффективности генетических алгоритмов многокритериальной оптимизации. Часть 1 // Информационные технологии. – 2012. – №8.

[Казаков, 2012б] Казаков П.В. Оценка эффективности генетических алгоритмов многокритериальной оптимизации. Часть 2 // Информационные технологии. – 2012. – №9.

[Ногин, 2004] Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход. – 2-е изд., испр. и доп. – М.: ФИЗМАТЛИТ, 2004.

[Соболь и др., 2006] Соболь И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями. – 2-е изд., перераб. и доп. – М.: Дрофа, 2006.

[Deb et al., 2002] Deb K., Pratap A., Agarwal S., Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation, vol. 6, №2, 2002.

[Deb et al., 2009] K. Deb Multi-Objective Optimization Using Evolutionary Algorithms. – Hoboken: Wiley, 2009.

[Michalewicz, 1996] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. – Berlin: Springer-Verlag, 1996.

[Zitzler et al., 2002] Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm // Proceedings of the EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 2002.

УДК 32.81

МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ МЕЖДУ ОБУЧЕНИЕМ И ЭВОЛЮЦИЕЙ: МЕХАНИЗМЫ ВЗАИМОДЕЙСТВИЯ¹

В.Г. Редько (*vgredko@gmail.com*)

Научно-исследовательский институт системных исследований РАН,
Москва

Построена и исследована модель взаимодействия между обучением и эволюцией. Проанализированы следующие механизмы этого взаимодействия. 1) Механизм генетической ассилияции, при которой индивидуально приобретаемые путем обучения навыки «переизобращаются» эволюцией и становятся наследуемыми. 2) Механизм эффекта экранирования, при котором сильное обучение тормозит эволюционный поиск. 3) Механизм влияния нагрузки на обучение, которая приводит к значительному ускорению эволюции.

Введение

Попытки моделирования и анализа взаимодействия между обучением и эволюцией проводились рядом специалистов в области бионических моделей вычислительного интеллекта. В известной работе Хинтона и Новлана [Hinton et al., 1987] продемонстрировано, что обучение может направлять эволюционный процесс к нахождению оптимума. В работе Мейли [Mayley, 1997] показано, что сильное обучение может и препятствовать эволюционной оптимизации, если отбор производится по конечному фенотипу, а обучение увеличивает шансы нахождения хорошего фенотипа независимо от генотипа особи. Этот эффект был назван эффектом экранирования. Также в этой работе было учтено, что процесс обучения имеет дополнительную нагрузку на особь (обучение требует времени и ресурсов) и приспособленность особи может уменьшаться под влиянием этой нагрузки.

Однако в указанных работах для моделирования эволюции использовались довольно сложные схемы генетического алгоритма со скрещиванием, поэтому были продемонстрированы сами эффек-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 13-01-00399).

ты взаимодействия между обучением и эволюцией, а механизмы этого взаимодействия представлены недостаточно четко. В настоящей работе для анализа механизмов взаимодействия используем одну из наиболее четких моделей эволюции – модель квазивидов [Эйген, 1973; Эйген и др., 1982] и наши количественные оценки эффективности эволюционных алгоритмов [Редько и др., 2005, 2011]. Модель квазивидов рассматривает отбор и мутации особей (обычно без скрещивания) и характеризует основные свойства эволюционных процессов.

1. Основная модель

Рассматривается эволюционирующая популяция особей. Каждая особь имеет генотип и фенотип. Генотип и фенотип определяются одинаковыми цепочками символов.

Генотип k -й особи S_{Gk} представляет собой цепочку символов S_{Gki} , $k = 1, \dots, n$, $i = 1, \dots, N$. Длина цепочки N и численность популяции n велики: $N, n \gg 1, 2^N \gg n$. Символы S_{Gki} равны 0 либо 1. Генотипы исходной популяции случайны. Генотип особи не меняется в течение ее жизни и передается (с малыми мутациями) потомкам особи.

В момент рождения k -й особи ее фенотип S_{Pk} равен генотипу: $S_{Pk}(t=1) = S_{Gk}$. Время дискретно: $t = 1, \dots, T$. T – время жизни одного поколения. Фенотип S_{Pk} меняется в течение жизни особи путем обучения.

Имеется оптимальная цепочка S_M длины N (символы которой также равны 0 или 1), которая ищется в процессе эволюции и обучения особей. При обучении меняются текущие символы фенотипа S_{Pk} : каждый тик времени t , каждый символ S_{Pki} меняется на случайный (равный 0 либо 1), и если новый символ S_{Pki} совпадает с соответствующим символом S_{Mi} оптимальной цепочки S_M , то этот символ сохраняется, в противном случае происходит возврат к старому символу фенотипа S_{Pk} .

В конце поколения происходит отбор особей в соответствии с их приспособленностями, определяемыми конечными фенотипами при $t = T$. Обозначим $S_{Fk} = S_{Pk}(t=T)$. Приспособленность k -й особи определяется расстоянием по Хеммингу $\rho(S_{Fk}, S_M)$ между S_{Fk} и S_M :

$$f_k = \exp[-\beta\rho(S_{Fk}, S_M)] + \varepsilon, \quad (1)$$

где $\beta (> 0)$ – интенсивность отбора, ε характеризует влияние случайных факторов на приспособленность особей, $0 < \varepsilon \ll 1$.

Дополнительно учтем роль нагрузки на обучения. С этой целью будем рассматривать модифицированные приспособленности особей:

$$f_{mk} = \exp(-ad) \{ \exp[-\beta\rho(\mathbf{S}_{Fk}, \mathbf{S}_M)] + \varepsilon \}, \quad (2)$$

где $a > 0$, $d = \rho(\mathbf{S}_{Gk}, \mathbf{S}_{Fk})$ – расстояние по Хеммингу между начальным и конечным фенотипом рассматриваемой особи.

Поскольку начальные генотипы особей в исходной популяции \mathbf{S}_{Gk} случайны, то среднее расстояние по Хеммингу между этими цепочками и оптимальной \mathbf{S}_M равно $N/2$. Такое расстояние нужно преодолеть путем обучения и эволюционного поиска.

Отбор особей в новое поколение производится хорошо известным рулеточным методом, для которого вероятность селекции определенной особи пропорциональна ее приспособленности. Селекция особей в каждое поколение производится одинаковое число раз, за счет этого численность популяции n в процессе эволюции постоянна.

Таким образом, обучаемые особи отбираются по конечным фенотипам особей $\mathbf{S}_{Fk} = \mathbf{S}_{Pk}(t=T)$, т.е. по конечному результату обучения, а по наследству потомкам передаются генотипы \mathbf{S}_{Gk} .

2. Результаты моделирования по основной модели

Параметры моделирования. Параметры модели выбираем таким образом, чтобы эволюционный поиск проходил достаточно эффективно, с учетом опыта работ [Редько и др., 2005, 2011]. Полагаем, что длина цепочек достаточно велика: $N=100$. Параметр отбора считаем равным $\beta = 1$, это соответствует достаточно высокой интенсивности отбора, при которой время (число поколений) отбора сравнительно мало, а время эволюционного поиска определяется интенсивностью мутаций. Интенсивность мутаций должна быть не слишком высока, чтобы не было мутационных потерь уже найденных в процессе эволюции удачных особей, но и не слишком мала, для того, чтобы мутационный поиск происходил достаточно быстро. С учетом этого полагаем, что вероятность замены любого символа в одном поколении в генотипах \mathbf{S}_{Gk} при мутациях равна $p_m = N^{-1} = 0.01$. При такой интенсивности мутаций p_m за одно по-

коление происходит замена примерно одного символа в генотипе особи, т.е. в течение одного поколения за счет мутаций расстояние по Хеммингу ρ между S_{Gk} и S_M изменяется в среднем на 1. Отбор приводит к уменьшению этого расстояния. Поскольку расстояние по Хеммингу между генотипами особей нулевого поколения и S_M порядка N , то весь процесс эволюции займет порядка $G_T \sim N$ поколений. Такая оценка скорости эволюции справедлива, если численность популяции n достаточно велика и нейтральным отбором особей (независящим от приспособленности особей) можно пре-небречь. Для выполнения этого условия достаточно потребовать, чтобы характерное время (число поколений) нейтрального отбора (по порядку величины равное n [Кимура, 1985]) было не меньше G_T , поэтому полагаем $n = G_T = N$.

Итак, параметры расчета равны: $N = 100$, $\beta = 1$, $p_m = N^{-1}$, $n = N$.

Также считаем, что $T = 2$, $\varepsilon = 10^{-6}$.

Сопоставление режимов с обучением и без обучения. На рис. 1 представлена зависимость от номера поколения G среднего по популяции расстояния по Хеммингу $\rho = \rho(S_{Gk}, S_M)$ между генотипами S_{Gk} и оптимальной цепочкой S_M для режимов с обучением и без обучения. Кривая 1 – режим с обучением, кривая 2 – режим без обучения (режим «чистой эволюции»). Приспособленности особей определяются выражением (1). Видно, что без обучения чисто эволюционный процесс не обеспечивает оптимизации генотипов, а при наличии обучения явно происходит приближение к оптимальной цепочке S_M .

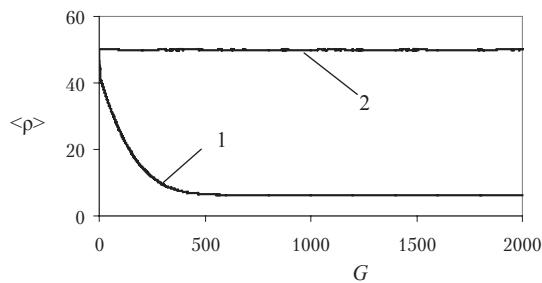


Рис. 1. Зависимость среднего по популяции расстояния до оптимума $\langle\rho\rangle = \langle\rho(S_{Gk}, S_M)\rangle$ от номера поколения G ; 1 – режим с обучением, 2 – режим чистой эволюции (усреднено по 1000 расчетам)

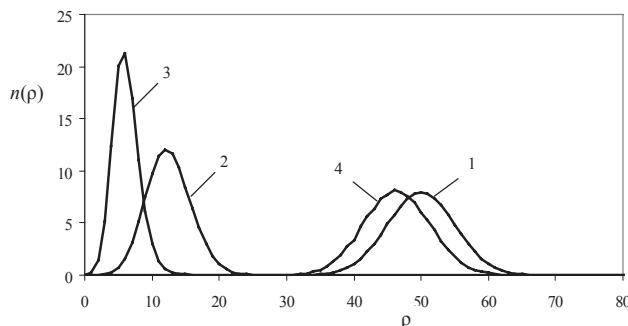


Рис. 2. Распределение особей $n(\rho)$ по величинам ρ в первом поколении эволюции: 1 – распределение по $\rho = \rho(S_{Gk}, S_M)$ для исходных генотипов до обучения, 2 – распределение по $\rho = \rho(S_{Fk}, S_M)$ для особей после обучения, но еще до отбора, 3 – распределение по $\rho = \rho(S_{Fk}, S_M)$ для отобранных особей, 4 – распределение по $\rho = \rho(S_{Gk}, S_M)$ для генотипов отобранных особей в конце поколения (усреднено по 10000 расчетам)

Уменьшения ρ в режиме чистой эволюции не происходит, так как в исходной популяции $\rho = \rho(S_{Gk}, S_M) \sim N/2 = 50$, следовательно, $\exp(-\rho) \sim 10^{-22} \ll \varepsilon$. Поэтому все особи популяции имеют приблизенно одинаковое значение приспособленности, равное ε , и, следовательно, в случае чистой эволюции вообще не происходит селекции особей. Приближение к S_M происходит только при наличии обучения.

Механизм ускорения эволюционного процесса за счет обучения иллюстрируется рис. 2, на котором представлено распределение особей $n(\rho)$ по ρ в разные моменты первого поколения. Кривая 1 показывает распределение по $\rho = \rho(S_{Gk}, S_M)$ для исходных генотипов особей в начале поколения. Кривая 2 показывает распределение по $\rho = \rho(S_{Fk}, S_M)$ для особей после обучения, но еще до отбора. Кривая 3 показывает распределение по $\rho = \rho(S_{Fk}, S_M)$ для отобранных особей. Кривая 4 показывает распределение по $\rho = \rho(S_{Gk}, S_M)$ для генотипов отобранных особей в конце поколения. Генотипы отобранных особей S_{Gk} достаточно близки к фенотипам обученных и отобранных особей S_{Fk} , поэтому распределение по $\rho = \rho(S_{Gk}, S_M)$ для генотипов (кривая 4) перемещается в сторону распределения для конечных фенотипов S_{Fk} (кривая 3). В следующих поколениях происходит аналогичное дальнейшее смещение распределения $n(\rho)$ в сторону меньших значений ρ . Такое смещение характеризует механизм генетической асимиляции: свойство приближения к S_M ,

полученное в результате индивидуального обучения, становится наследуемым.

Подчеркнем, что в результате обучения происходит смещение к достаточно малым ρ , при которых существенно различаются приспособленности (1) и происходит отбор особей с меньшими значениями $\rho(S_{Gk}, S_M)$. Для того, чтобы изложенный механизм работал, обучение должно быть достаточно сильным.

Эффект экранирования. Однако сильное обучение может не только способствовать, но и препятствовать эволюционному поиску оптимального генотипа $S_{G_{opt}} = S_M$. Как показывает кривая 1 на рис. 1, конечное значение $\langle \rho \rangle$ остается достаточно большим, приближенно равным 6.2. Это связано с тем, что для сильного обучения при $G \sim 1000$ нахождение оптимального фенотипа $S_{F_{opt}} = S_M$ происходит независимо от генотипа S_{Gk} . Поэтому при больших G не происходит приближения генотипов к S_M . Наблюдается эффект экранирования [Mayley, 1997].

Влияние нагрузки на обучение на моделируемые процессы. Был проведен анализ влияния нагрузки на обучение на моделируемые процессы эволюции с обучением. Для этого случая приспособленность определялась выражением (2). Расчет проводился для указанных выше параметров ($N = n = 100$, $\beta = 1$, $p_m = 0.01$, $T = 2$, $\varepsilon = 10^{-6}$), величина α полагалась равной 1. Результаты расчета представлены на рис. 3,4. На рис. 3 показана зависимость среднего по популяции расстояния по Хеммингу $\langle \rho \rangle = \langle \rho(S_{Gk}, S_M) \rangle$ между генотипами S_{Gk} и оптимальной цепочкой S_M от номера поколения G . На рис. 4 представлена динамика распределений особей $n(\rho)$ по величинам ρ в первом поколении эволюции.

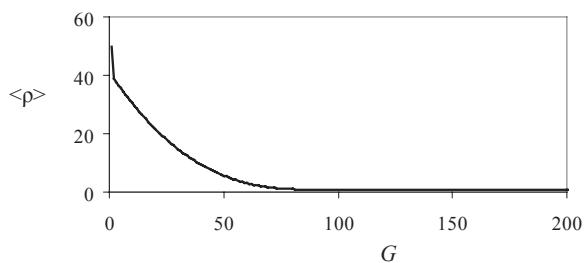


Рис. 3. Зависимость среднего по популяции расстояния до оптимума $\langle \rho \rangle$ для генотипов особей от номера поколения G с учетом нагрузки на обучение (усреднено по 1000 расчетам)

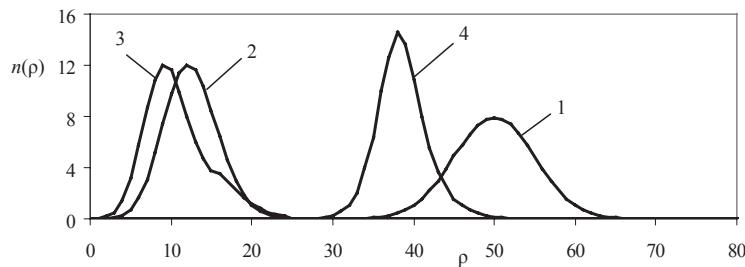


Рис. 4. Распределение особей $n(\rho)$ по величинам ρ в первом поколении эволюции с учетом нагрузки на обучение. 1 – распределение по $\rho = \rho(S_{Gk}, S_M)$ для исходных генотипов до обучения, 2 – распределение по $\rho = \rho(S_{Fk}, S_M)$ для особей после обучения, но еще до отбора, 3 – распределение по $\rho = \rho(S_{Fk}, S_M)$ для отобранных особей, 4 – распределение по $\rho = \rho(S_{Gk}, S_M)$ для генотипов отобранных особей в конце поколения. Смещение распределения 4 к меньшим значениям ρ существенно больше, чем на рис. 2 (усреднено по 10000 расчетам)

Сравнение рис. 1,2 и рис. 3,4 показывает, что учет нагрузки на обучение приводит к дополнительному значительному ускорению эволюции. При учете нагрузки на обучение в конце эволюционного процесса (при $G = 200$) оптимальные генотипы и фенотипы находятся. Эффект экранирования в этом случае отсутствует.

3. Сопоставление с подходом Хинтона и Новлана

Для сопоставления с хорошо известным подходом Хинтона и Новлана [Hinton et al., 1987] была построена дополнительная модель, также основанная на модели квазивидов. Дополнительная модель строится подобно основной. Каждая особь эволюционирующей популяции имеет генотип и фенотип, фенотипы модифицируются путем такого же механизма обучения. Отбор в следующее поколение происходит в соответствии с конечным фенотипом, полученным в результате обучения. Только приспособленность особей в дополнительной модели изменялась следующим образом.

При наличии обучения приспособленность k -й особи определяется конечным фенотипом S_{pk} при $t = T$:

$$f_k = \exp[-\beta\rho(S_{Fk}, S_M)], \quad (3)$$

где $S_{Fk} = S_{Pk}(t = T)$, $\rho = \rho(S_{Fk}, S_M)$ – расстояние по Хеммингу между S_{Fk} и оптимальной цепочкой S_M .

При отсутствии обучения приспособленность k -й особи равна 1, если генотип равен оптимальной цепочке и 0 во всех остальных случаях:

$$f_k = 1 \text{ при } S_{Gk} = S_M; \quad f_k = 0 \text{ при } S_{Gk} \neq S_M. \quad (4)$$

Также рассматривалось влияние нагрузки на обучение. В этом случае приспособленность модифицировалась:

$$f_{mk} = \exp(-ad) \exp[-\beta\rho(S_{Fk}, S_M)], \quad (5)$$

где $d = \rho(S_{Gk}, S_{Fk})$.

Анализ дополнительной модели проводился путем компьютерного моделирования для эволюции с обучением. Результаты почти совпадали с результатами исследования основной модели. Для дополнительной модели не было только явного сопоставления режимов эволюции с обучением и эволюции без обучения.

Совпадение существенных результатов для основной и дополнительной моделей показывает малую роль параметра ε в основной модели (см. выражения (1) и (2)). Этот параметр важен только для явного сопоставления режимов эволюции с обучением и чистой эволюции (см. рис.1).

Заключение

Таким образом, построена и исследована модель взаимодействия между обучением и эволюцией. Количественно проанализированы механизмы взаимодействия между обучением и эволюцией.

Проведенное исследование показывает, что а) генетическая асимиляция, б) эффект экранирования и в) значительное ускорение генетической асимиляции и эволюционного процесса под влиянием нагрузки на обучение наблюдаются при следующих предположениях:

- 1) Каждая особь эволюционирующей популяции имеет генотип и фенотип.
- 2) Генотип и фенотип представляют собой цепочки символов одинаковой формы.
- 3) Генотипы передаются от родителей к потомкам с небольшими мутациями. Генотип особи не меняется в течение ее жизни.

- 4) Начальный фенотип особи равен ее генотипу.
- 5) Имеется определенная оптимальная цепочка символов, которая ищется в процессе обучения и эволюции.
- 6) Фенотип существенно модифицируется путем обучения в течение жизни особи. В процессе обучения фенотип приближается к оптимальной цепочке.
- 7) Отбор особей в новое поколение определяется конечными фенотипами особей.

СПИСОК ЛИТЕРАТУРЫ

- [Кимура, 1985]** Кимура М. Молекулярная эволюция: теория нейтральности. – М.: Мир, 1985.
- [Редько и др., 2005]** Редько В.Г., Цой Ю.Р. Оценка эффективности эволюционных алгоритмов // Доклады АН, 2005. Т. 404. № 3.
- [Редько и др., 2011]** Редько В.Г., Цой Ю.Р. Оценка скорости и эффективности эволюционных алгоритмов // Бионические информационные системы и их практические применения. – М.: Физматлит, 2011.
- [Эйген, 1973]** Эйген М. Самоорганизация материи и эволюция биологических макромолекул. – М.: Мир, 1973.
- [Эйген и др., 1982]** Эйген М., Шустер П. Гиперцикл. Принципы самоорганизации макромолекул. – М.: Мир, 1982.
- [Hinton et al., 1987]** Hinton G.E, Nowlan S.J. How learning can guide evolution // Complex Systems. 1987. V. 1. No. 3.
- [Mayley, 1997]** Mayley G. Guiding or hiding: Explorations into the effects of learning on the rate of evolution // Proceedings of the Fourth European Conference on Artificial Life (ECAL 97). – Cambridge, Massachusetts: MIT Press, 1997.

УДК 519.7

КРИПТОГРАФИЯ С СИММЕТРИЧНЫМ КЛЮЧОМ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКОГО АЛГОРИТМА¹

В.Л. Степанюк (*stefanuk@iitp.ru*)

Институт проблем передачи информации, Москва

А.Х. Алхуссайн (*amanie-alhussain@hotmail.com*)

Российский университет дружбы народов, Москва

В отличие от использования генетических алгоритмов (ГА) для формирования ключа шифрования в настоящей работе предлагается включить ГА непосредственно в тракт передачи шифрованной информации, где этот алгоритм позволяет улучшить вероятностные характеристики генератора случайных чисел (ГСЧ). Полезность данного подхода иллюстрируется на нескольких примерах. Кроме того, представлена действующая система криптографии, опирающаяся на описанные в статье принципы.

Введение

Криптография играет важную роль в области сетевой безопасности. Цель криптографии – защита передаваемой информации от чтения посторонними лицами.

Генетический алгоритм представляет собой эвристику, которая имитирует процесс естественной эволюции. Компьютерный вариант генетического алгоритма был предложен Голландом [Holland, 1975], [Poornima, 2014]. Он был успешно применен в целом ряде приложений для решения задач нелинейной оптимизации, встречающихся в таких областях, как искусственный интеллект.

В литературе по криптографии был представлен целый ряд методов, использующих генетические алгоритмы.

Например, в работах [Menezes, 2001] и [Goldberg, 1989] была описана новая блочная система шифрования, а именно ICIGA («Улучшенная криптография, навеянная идеями генетических ал-

¹ Работа выполнена при финансовой поддержке по программе президиума РАН «Информационные, управляемые и интеллектуальные технологии и системы» и фонда РФФИ (проект № 12-07-00209-а). Исследование проводилось в ИППИ РАН и на кафедре информационных технологий Российского университета дружбы народов.

горитмов»), в которой в ходе случайного процесса генерируется ключ сеанса. Размер блока и длина ключа могут изменяться и могут быть установлены конечным пользователем в начале процесса шифрования. Система ICIGA является усовершенствованием системы ГПК [Tragha, 2005], [Tragha, 2006].

В настоящей работе используется детерминированный вариант ГА, который обеспечивает однозначное восстановление зашифрованного текста получателем.

Проведенные нами эксперименты показали, что уровень случайности для типичных генераторов случайных чисел (ГПСЧ) не очень высокий и может быть несколько улучшен с помощью генетического алгоритма. В настоящей статье детерминированный ГА используется для улучшения свойств ГПСЧ и повышения криптографической защиты.

В данной работе мы предлагаем особый подход к шифрованию. Сначала генерируется последовательность псевдослучайных чисел с помощью линейного конгруэнтного генератора (либо с помощью генератора на основе чисел Фибоначчи). Затем эта последовательность используется в детерминированном ГА с применением процедур мутации и кроссовера¹.

Настоящая работа организована следующим образом. В разделе 1 мы описываем используемые нами генераторы случайных чисел. В разделе 2 рассматривается роль детерминированного ГА. В разделе 3 объясняется предлагаемый способ криптографической защиты. В разделе 4 приводятся экспериментальные результаты, а выводы содержатся в заключительном разделе.

1. Используемые генераторы псевдослучайных чисел

– Алгоритм ГПСЧ (Pseudorandom Number Generator, PRNG) призван создавать последовательность чисел, элементы которой практически независимы друг от друга и подчиняются заданному распределению (обычно равномерному). Числа мы будем считать первым поколением в случае применения ГА.

В предлагаемом алгоритме шифрования может быть использован один из следующих двух алгоритмов ГПСЧ:

¹ Хорошо известно, что блочные шифры, когда биты обрабатываются группами, надежнее, но медленнее поточных. Поэтому они здесь не используются.

Линейный конгруэнтный метод был предложен Д.Г.Лемером (Derrick Henry Lehmer) в 1949 году. Суть метода заключается в вычислении последовательности случайных чисел X_n , полагая:

$$X_{n+1} = (aX_n + c) \bmod m,$$

где m – значение модуля ($m \geq 0$), a – множитель, c – приращение, X_0 – начальное значение. (Модуль, множитель, приращение и начальное значение будут частью секретного ключа)

Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator.) – это другой метод генерации псевдослучайных чисел:

$$X_{n+1} = (X_n + X_{n-1}) \bmod m,$$

где m – модуль, X_0 – начальное значение. (Начальное значение и модуль будут использоваться как часть секретного ключа)

При помощи описанных ГПСЧ будет создано первое поколение случайных чисел, которое в дальнейшем нами будет модифицировано с применением ГА, описанного в следующем разделе.

2. Детерминированный генетический алгоритм

Мы используем детерминированную версию генетического алгоритма, подобную тому, что встречаются в некоторых алгоритмах. В силу использованного нами алгоритма передачи, который описан в разделе 3, необходимо обеспечить повторяемость результатов работы такого алгоритма. Поэтому введем понятия детерминированной операции кроссовера и детерминированной мутации, которые, разумеется, отличаются от того, что предлагается Холландом [Holland, 1975].

В детерминированных операциях, пример которых мы даем ниже, считается, что полное множество хромосом представляет собой коллекцию двоичных строк фиксированной длины.

Пример работы оператора кроссовера. Здесь выбраны две «хромосомы», а конкретное место для операции кроссовера выбирается в зависимости от размера хромосом (рис. 1а, 1б). После этого производится обмен частей хромосом (рис. 1с) (Напомним, что операция носит детерминированный, а не вероятностный характер.)

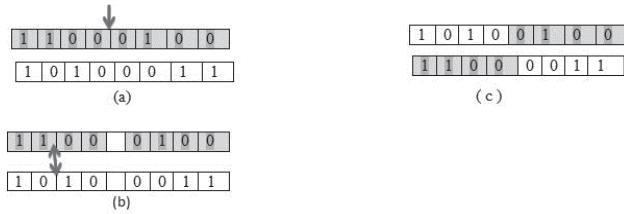


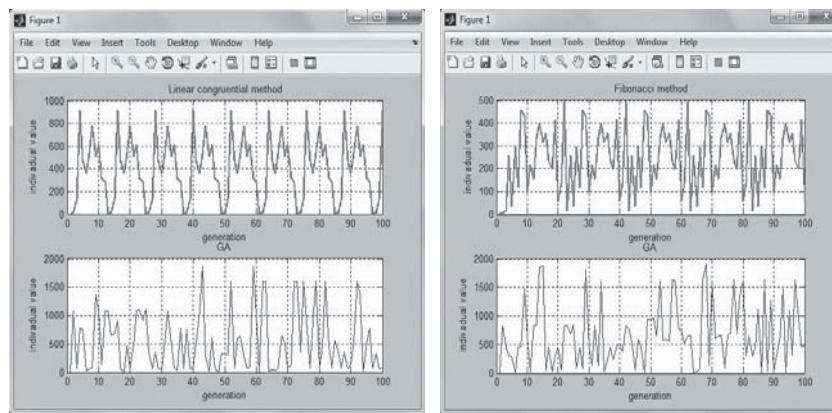
Рис. 1. Пример работы оператора детерминированного кроссовера

Пример работы *оператора мутации*, когда некоторые «гены», имеющие значение 0, изменяются на значение 1, показан на следующем рисунке.

0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 1

Рис. 2. Пример работы оператора мутации

Результаты, достигаемые при применении ГА к ГПСЧ, иллюстрируются на рис. 3.



3 а. Последовательность ПСЧ, созданная с использованием линейного конгруэнтного генератора (вверху) и модифицированная с помощью генетического алгоритма (снизу).

3 б. Последовательность ПСЧ, созданная с использованием генератора Фибоначчи (вверху) и модифицированная с помощью генетического алгоритма (снизу).

Рис. 3. Генетический алгоритм повышает уровень случайности

На рис. За сверху показан результат применения ГА в случае линейного конгруэнтного генератора, а ниже показано, что степень случайности существенно повышается после применения детерминированного ГА. Аналогично, на рис. 3б показана сходная ситуация для генератора Фибоначчи.

3. Симметричный ключ. Алгоритм шифрования

Симметричные алгоритмы требуют, чтобы ключ шифрования и ключ дешифрования совпадали. Отправитель и получатель обязаны, таким образом, согласовать ключ шифрования, прежде чем они смогут общаться безопасно.

Секретный ключ.

Секретные ключи в нашем подходе зависят от выбора исходного генератора случайных чисел:

Ключ для генератора Фибоначчи содержит несколько компонент – начальное значение, модуль, число поколений, размер хромосомы, конечное значение;

Ключ для конгруэнтного генератора также содержит несколько компонент.

Описание алгоритма

Этап шифрования.

Сначала выбирается ключ шифрования, сообщаемый как передатчику, так и приемнику информации. В зависимости от ключа генерируется последовательность псевдослучайных чисел с помощью линейного конгруэнтного генератора (либо с помощью генератора на основе чисел Фибоначчи), модифицированная с применением описанного выше детерминированного генетического алгоритма. Обозначим эту последовательность символом L .

Затем совершаются следующие шаги, связанные с передаваемым текстом.

1. Для очередного символа передаваемого текста, начиная с самого первого, находится значение ASCII кода;
2. Полученное значение делится по модулю на число, взятое из множества L . При этом значение модуля содержится в составе выбранного секретного ключа (последовательность L повторяет-

ся циклическим образом, если множество содержит недостаточное количество чисел.)

3. В качестве шифрованного текста вместо каждого символа будет передана пара – частное и остаток, полученные для этого символа текста;

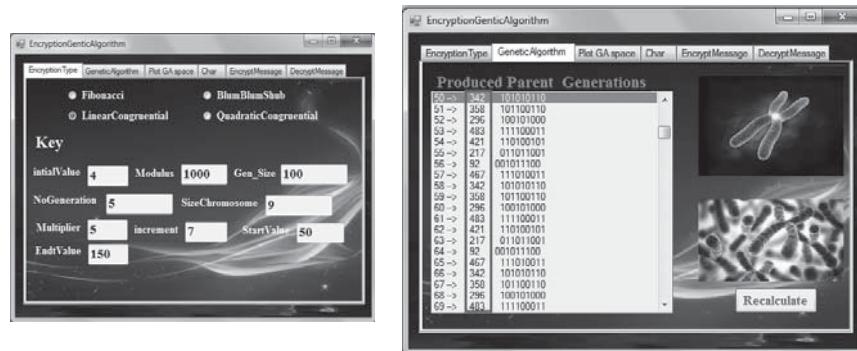
Этап дешифрования.

Этап дешифрования выполняется аналогичным образом, причем в силу детерминированного характера всех операций возникает возможность на приемном конце восстановить значения всех переданных символов.

4. Экспериментальные результаты

Для эксперимента был выбран простой текст (открытый текст): »genetic encryption«

При выборе линейного конгруэнтного генератора на основе секретного ключа (ключ показан на рисунке 4.а) построенное множество L содержит числа {342 358 296 483 421 217 92 467 342 358 296 483 421 217 92 467 342 358}, как показано на рисунке 4б.



(a) Секретный ключ

(b) Множество L

Рис. 4. Ключ шифрования и соответствующая последовательность ПСЧ

Ход шифрования показан в табл. 1 и на скриншорте рис. 5.

Таблица 1

Символ	ASCII Значение символ	Последовательность чисел L	Деление по модулю	Частное	Остаток
g	103	342	342/103	3	33
e	101	358	358/101	3	55
n	110	296	296/110	2	76
e	101	483	483/101	4	79
t	116	421	421/116	3	73
ı	105	217	217/105	2	7
c	99	92	92/99	0	99
	32	467	467/32	14	19
e	101	342	342/101	3	39
n	110	358	358/110	3	28
c	99	296	296/99	2	98
r	114	483	483/114	4	27
y	121	421	421/121	3	58
p	112	217	217/112	1	105
t	116	92	92/116	0	116
ı	105	467	467/105	4	47
o	11	342	342/11	3	9
n	110	358	358/110	3	28

Зашифрованный текст – это { 3 28 3 9 2 47 4 116 3 105 2 58 0 27 14 98 3 28 3 39 2 19 4 99 3 7 1 73 0 79 4 76 3 55 3 33 }, как показано на рис. 5.

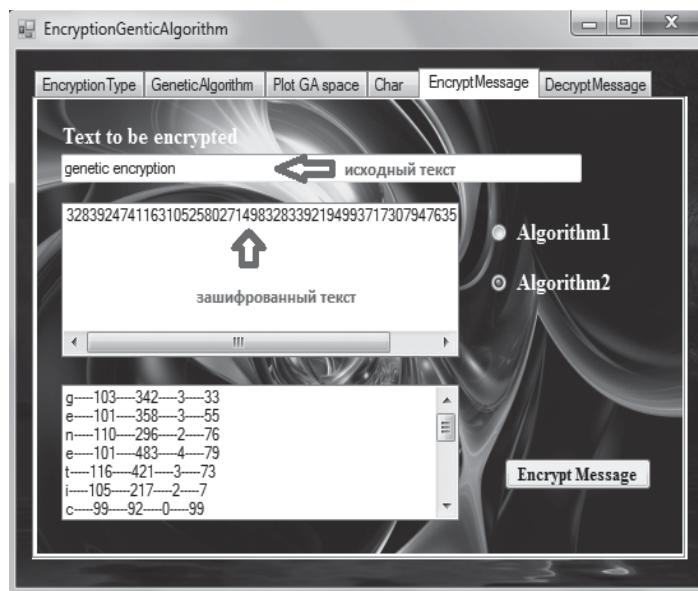


Рис.5. Процесс шифрования

Ход дешифрования в действующей системе показан в табл. 2 и на рис. 6.

Таблица 2

частное	остаток	Последовательность чисел L	Для цикла $I = 1$ до 256 сравнивать частное* I + остаток с последовательностью чисел и выбрать I	символ
3	33	342	103	g
3	55	358	101	e
2	76	296	110	n
4	79	483	101	E
3	73	421	116	t
2	7	217	105	ı
0	99	92	99	c
14	19	467	32	
3	39	342	101	e
3	28	358	110	n
2	98	296	99	c
4	27	483	114	r
3	58	421	121	y
1	105	217	112	p
0	116	92	116	t
4	47	467	105	ı
3	9	342	11	o
3	28	358	110	n



Рис. 6. Процесс дешифрования

Заключение

В настоящем исследовании был разработан потоковый криптографический алгоритм с использованием концепции детерминированного генетического алгоритма для улучшения свойств линейного конгруэнтного генератора или генератора Фибоначчи. Работа была реализована с использованием C# и MATLAB 7.8.0 в качестве платформы моделирования. Разработанный алгоритм тестиировался также на текстах из русских и арабских символов и показал успешную работу, аналогичную приведенной выше для текста на английском языке.

СПИСОК ЛИТЕРАТУРЫ

- [Tragha, 2005] Tragha A., Omary F., and A. Kriouile, «Genetic Algorithms Inspired Cryptography» A.M.S.E Association for the Advancement of Modeling & Simulation. Techniques in Enterprises, Series D: Computer Science and Statistics, 2005.
- [Tragha, 2006] Tragha A., Omary F., Mouloudi A., «ICIGA: Improved Cryptography Inspired by Genetic Algorithms», Proceedings of the International Conference on Hybrid Information Technology (ICHIT'06), pp. 335–341, 2006.
- [Holland, 1975] Holland, J., «Adaptation in Natural and Artificial Systems», University of Michigan Press, Ann Arbor, 1975.
- [Goldberg, 1989] Goldberg D.E., «Genetic algorithms in search optimization & Machine learning», Addison Wesley, 1989.
- [Poornima, 2014] Poornima G. Naik, Girish R. Naik. Symmetric Key Encryption using Genetic Algorithm NCI2TM: 2014 ISBN: 978-81-927230-0-6.
- [Menezes, 2001] Menzes A. J., Paul, C., Van Dorschot, V., Vanstone, S. A., «Handbook Of Applied Cryptography», CRS Press 5th Printing; 2001.

УДК 519.87

**АВТОМАТИЧЕСКОЕ ГЕНЕРИРОВАНИЕ
НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ
В ЗАДАЧЕ ПРОГНОЗИРОВАНИЯ УРОВНЯ
ЗАБОЛЕВАЕМОСТИ НАСЕЛЕНИЯ¹**

Д.И. Хриトンенко (hdmitry.91@mail.ru)

Е.С. Семенкин (eugeneselemenkin@yandex.ru)

Е.В. Сугак (sugak@mail.ru)

Е.Н. Потылицина (leonova_en@mail.ru)

Сибирский государственный аэрокосмический университет имени
академика М.Ф. Решетнева, Красноярск

Рассматривается автоматическое генерирование искусственных нейронных сетей самоконфигурирующимися эволюционными алгоритмами. Полученным алгоритмом решается задача прогнозирования уровня заболеваемости населения в зависимости от уровня атмосферного загрязнения. Проводится ряд тестов и сравнений различных вариантов предложенного алгоритма.

Введение

Использование искусственных нейронных сетей (далее ИНС) в задачах прогнозирования на сегодняшний день является одним из распространенных методов [Галушкин, 2010]. Однако при использовании такого подхода исследователи каждый раз сталкиваются со сложной задачей выбора архитектуры ИНС. С такой задачей могут успешно справляться эволюционные алгоритмы. Автоматизация построения архитектуры ИНС позволяет расширить область их применения, т.к. от конечного пользователя больше не требуются знания в области нейросетевого моделирования. Требуются знания лишь в области применения ИНС.

Эффективность эволюционных алгоритмов зависит от выбранной для них конфигурации [Meyer-Nieberg, 2007]. Для каждой конкретной задачи наиболее эффективная конфигурация алгоритма

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 14-06-00256).

заранее неизвестна. Встречаются случаи, когда изменения конфигурации алгоритма в процессе работы приводят к увеличению его эффективности. Таким образом, автоматическая процедура выбора конфигурации алгоритма в процессе работы (самоконфигурирование) является полезной модификацией, позволяющей пользователю частично отказаться от настроек алгоритма и повысить его эффективность.

В работе описывается алгоритм самоконфигурирования эволюционных алгоритмов. Рассматривается подход автоматического генерирования структуры и обучения искусственных нейронных сетей. Алгоритм генетического программирования также используется в задаче восстановления пропусков таблицы данных. В заключительной части итоговый алгоритм применяется для решения задачи прогнозирования уровня заболеваемости населения.

1. Самоконфигурирующийся эволюционный алгоритм

В эволюционных алгоритмах существует несколько групп генетических операторов: селекции, скрещивания, мутации, формирования новой популяции. Идея самоконфигурирования основывается на следующем принципе: «Внутри каждой из групп наиболее успешный из операторов поощряется, остальные штрафуются». Это происходит за счет увеличения и уменьшения вероятностей выбора генетических операторов внутри группы. Опишем алгоритм [Семенкина, 2013; Semenkin et al., 2012].

Для каждого из операторов внутри группы в начале работы алгоритма выбирается вероятность $p=1/k$, где k – число операторов внутри группы. Далее на каждой итерации алгоритма происходит оценка работы каждого из операторов, основываясь на средней пригодности потомков, полученных с его помощью. Это происходит по следующей формуле: $fitness=f/n$. Здесь f – суммарная пригодность потомков, n – их количество;

Перераспределение вероятностей выбора операторов происходит следующим образом: наиболее эффективный из операторов увеличивает свою вероятность выбора p на $((k-1)\cdot C)/(k\cdot N)$, в то время как остальные из операторов группы уменьшают свою вероятность на $C/(k\cdot N)$. Здесь k – число операторов внутри группы,

C – некоторый коэффициент, N – число поколений алгоритма. При перераспределении сумма вероятностей всегда равна единице и вероятность выбора каждого из операторов не может упасть ниже заданного порога.

Далее для каждого будущего потомка операторы, при помощи которых он будет порожден, выбираются в соответствии с полученными распределениями вероятностей.

Описанный подход позволяет отказаться от выбора конфигурации алгоритма пользователем.

Ранее такой подход был протестирован на ряде задач безусловной оптимизации [Khritonenko et al., 2013]. Один из полученных результатов представлен ниже:

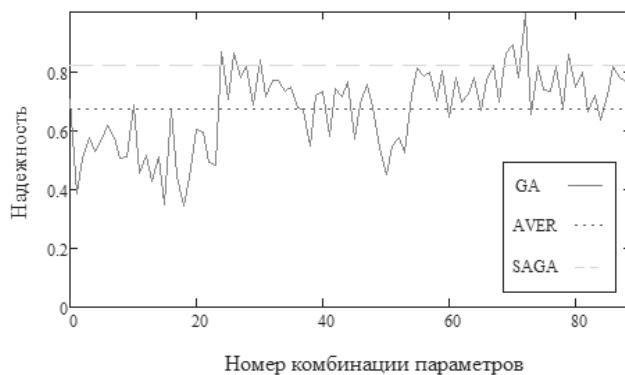


Рис. 1. Результаты сравнения эффективности стандартного и самоконфигурирующегося генетического алгоритма

На данном графике GA – базовая модель эволюционного алгоритма (генетический алгоритм), $AVER$ – среднее значение пригодности алгоритма, $SAGA$ – самоконфигурирующийся алгоритм. Как видно из представленного графика, модифицированный алгоритм выигрывает у базового по большему числу настроек, что, безусловно, является важным результатом, т.к. существенный проигрыш наблюдается только при одном варианте конфигурации ГА, который, однако, заранее неизвестен, а должен быть определен в результате сравнения 90 различных вариантов.

2. Применение алгоритма генетического программирования

Один из методов автоматического генерирования ИНС – эволюционные алгоритмы, а именно алгоритм генетического программирования (ГП) [Рутковская и др., 2006]. Для применения такого алгоритма в задаче формирования ИНС достаточно определить метод кодирования, заполнить терминальное и функциональное множество необходимыми элементами, соблюсти условия замкнутости и достаточности. В качестве элементов терминального множества используются все входы задачи, а также нейроны с различными активационными функциями. В качестве функционального множества – оператор объединения нейронов в один слой (+), а также операторы, отвечающие за межслойные связи (>). Пример простейшей нейронной сети можно увидеть на рисунке ниже.

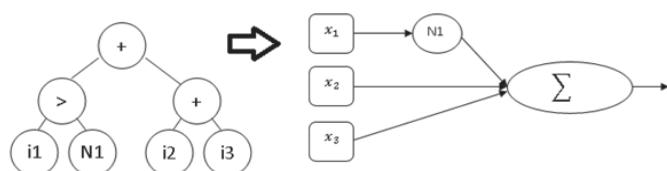


Рис. 2. Пример кодирования нейронной сети

На рисунке 2 $N1$ – нейрон с активационной функцией 1-го типа, $i1, i2, i3$ – соответствующие входы (атрибуты) задачи. Всего в алгоритме задействовано десять типов различных нейронов.

Зачастую при решении практических задач возникает проблема обработки и восстановления пропущенных значений [Снитюк, 2006]. Существует множество методов для решения данной проблемы, одним из которых также является метод генетического программирования. Для решения такой задачи необходимо наполнить функциональное множество алгоритма различными математическими операторами, а терминальное – числовыми параметрами и теми атрибутами, пропусков в которых не наблюдается. Таким образом, станет возможным восстановить нелинейную множественную регрессию и тем самым заполнить пропуски в массивах данных задачи.

В обоих из представленных случаев алгоритмам ГП необходима процедура обучения. В первом случае для настройки весовых коэффициентов нейронных сетей, во втором – для настройки различных параметров полученных в итоге символьных выражений. В данной работе в качестве такого метода используется самоконфигурирующийся генетический алгоритм.

Эффективность итогового алгоритма на ряде тестовых задач была продемонстрирована в предшествующих работах [Khritonenko et al., 2013]. Они показали полезность описанных модификаций. Это и обуславливает выбор описанного алгоритма для решения представленной далее задачи.

3. Описание решаемой задачи

Имеется выборка (x_i, y_i) , $i=1, 2, \dots, n$, где n – число измерений в выборке, x_i является вектором размерности m , где m – число атрибутов задачи, y_i – наблюдаемая величина. Также известно, что y_i может зависеть от предыдущих k своих значений, а также значений атрибутов x_i . Необходимо, основываясь на имеющейся выборке, установить зависимость между $x_i, x_{i-1}, \dots, x_{i-k}, y_i, y_{i-1}, \dots, y_{i-k}$ и y_{i+1} , т.е. спрогнозировать значения y_{i+1} или решить задачу прогнозирования временных рядов.

В данной работе выборка представляет собой массив с 39 столбцами, 28 из которых являются атрибутами задачи, а 10 – прогнозируемые измерения. Количество измерений равно 32. Один из столбцов – год, в котором были сделаны замеры. В таблице имеются пропуски. Выборка получена из отчетов организации «КРАССТАТ». Атрибуты показывают содержание различных вредных веществ в воздухе: медь, цинк, железо, хлор и т.д. Остальные измерения показывают различные возникшие заболевания и смертность от них. Таким образом, ставится задача прогнозирования заболеваемости населения в зависимости от экологического состояния воздуха в городе.

Для удобства дальнейшего восприятия полученной информации пронумеруем прогнозируемые величины от $Out1$ до $Out10$ и представим некоторую числовую информацию об этих величинах. В таблице 1 приводятся оригинальное название прогнозируемых величин, сопоставленная им нумерация, вычисляются минимальное и максимальное значения внутри столбца выборки, а также среднее значение и стандартное уклонение по нему.

Числовые характеристики

Таблица 1

Прогнозируемая величина	Рождаемость (на 1 тыс. чел.)	Смертность (на 1 тыс. чел.)	Смертность детей до 1 года (на 1 тыс. чел.)	Больных злокачественными новообразованиями (на 100 тыс. чел.)	Смертность от новообразований (на 100 тыс. чел.)
Нумерация	Out1	Out2	Out3	Out4	Out5
Мин.	8,3	8,9	4,8	1042	180,26
Макс.	14,3	15,5	21	2483,8	211,9
Среднее	10,98	12,41	14,26	1822,65	202,44
Стандарт. укл.	1,89	1,80	5,91	429,42	8,75
Прогнозируемая величина	Смертность Система кровообращения (на 100 тыс. чел.)	Смертность Органы дыхания (на 100 тыс. чел.)	Больных психических расстройствами (100 тыс. чел.)	Смертность от психических расстройств (100 тыс. чел.)	Смертность от болезней нервной системы (100 тыс. чел.)
Нумерация	Out6	Out7	Out8	Out9	Out10
Мин.	441,44	38,69	686,7	0,91	5,31
Макс.	726,05	72,16	1407,4	11,42	8,98
Среднее	594,03	55,93	931,07	5,83	6,79
Стандарт. укл.	76,16	8,70	198,96	3,91	1,13

4. Тестирование разработанного алгоритма

Все представленные алгоритмы были реализованы в виде программных систем на языке C++. Тестирование проводилось с использованием кросс-валидации [Efron, 1983] и разбиением выборки на десять частей случайным образом. Для каждой прогнозируемой переменной проводилась серия из ста экспериментов. Все выводы, полученные в дальнейшем, сделаны на основании непараметрического критерия Вилкоксона. Тестирование проводилось как для восстановленных, так и для «сырых» данных, с отбрасыванием некомплектных строк. Для корректного функционирования ИНС производилась нормировка данных на отрезок [-1, 1]. Также производилось тестирование алгоритма для различного числа входных переменных, т.е. бралось различное число предыдущих значений временного ряда (глубина). В таблицах ниже это число значений обозначено как k . При тестировании алгоритмов в качестве критерия выступает средняя относительная ошибка предсказания в процентах.

Тестирование для варианта без восстановления пропусков представлено ниже.

Таблица 2
Результаты тестирования

k	Out1	Out2	Out3	Out4	Out5	Out6	Out7	Out8	Out9	Out10
1	12,7	12,7	6,1	4,5	8,6	2,7	0,8	8,3	4,7	10,5
2	9,8	13,8	7	7,1	6,3	3,1	0,5	7,5	6	10,7
3	10,4	14,3	7,7	4,2	7,5	3,6	1,8	5,4	5,6	11,5
4	11,2	13,5	8,6	8,7	7,4	1,8	2	5,3	5,5	11,8
5	11	15,5	8,7	8	8	3,3	2,8	7	5,8	12
6	11,2	15,1	8	5,7	7,2	1,5	3,2	7,2	5,8	12,1
7	11,6	16,2	9,1	5,8	9,4	2,4	3,4	8,5	6,5	12,5

Стоит заметить, что для такого тестирования использовалось приблизительно 50% атрибутов, т.к. в другой половине встречаются пропуски данных, и они не могут быть использованы в алгоритме. Такая потеря информации недопустима, т.к. выборка данных является и без того небольшой.

Тестирование варианта с восстановленными пропусками в матрице данных методом ГП.

Таблица 3

Результаты тестирования

<i>k</i>	Out1	Out2	Out3	Out4	Out5	Out6	Out7	Out8	Out9	Out10
1	12,1	16	6,9	3	6,8	1	0,8	6,3	2,1	9,7
2	9,2	12,7	5,6	2,9	5,4	1	0,5	5,5	1	8,9
3	9,5	12,5	5,4	3,2	5,6	1,1	0,2	5,1	1	8,5
4	9,8	13,1	5,8	2,7	5,7	1	0,7	4,8	0,8	8,6
5	10,6	14,2	6,3	2,4	6,3	1,3	0,7	5,9	1,3	9,1
6	10,5	14,5	5,9	3,6	6,2	1,3	0,4	6,1	2,3	9,5
7	11,2	15,3	6,7	3,2	6,7	1,4	0,5	6,1	3,2	9,2

Из приведенных результатов видно, что использование процедуры восстановления пропусков в таблице данных позволяет повысить эффективность подхода. Это подтверждается при использовании критерия Вилкоксона.

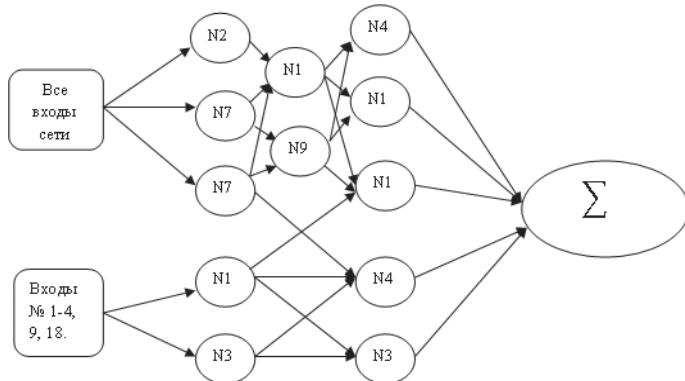


Рис. 3. Пример полученной нейронной сети

Рисунок 3 показывает одно из решений, сгенерированных алгоритмом для прогнозирования столбца *Out7*, – смертность от болезней, связанных с органами дыхания. Стоит заметить наличие атрибутов №1–4, 9, 18 в сети. Вместе с высокими весовыми коэф-

фициентами это означает значительный вклад данных атрибутов в прогнозируемую характеристику. Подобную картину можно наблюдать и в других ИНС.

Некоторые из прогнозируемых величин имеют довольно большую ошибку, т.к. в выборке данных не учтено множество факторов. Доказанным фактом является то, что на рождаемость, например, влияет не только физико-химический состав воздуха. Однако для различных заболеваний и их последствий точность прогноза оказалась высока. Кроме того, отслеживается явная взаимосвязь между такими параметрами, как, например, число больных психическими расстройствами и смертность от них. Взаимосвязь кажется нам очевидной, однако при работе с интеллектуальными информационными технологиями, в частности ИНС, важной является проверка на адекватность полученных результатов. Несмотря на то, что ИНС являются моделями черного ящика, косвенно влияние тех или иных факторов на выходное значение можно отследить посредством весовых коэффициентов входных атрибутов.

Использование самоконфигурирующихся эволюционных алгоритмов в задачах такого типа позволяет в автоматическом режиме строить адекватные модели для предсказания необходимых значений без обладания знаний как в области нейросетевого моделирования, так и в области эволюционных вычислений.

Заключение

В работе было рассмотрено применение эволюционных алгоритмов в задаче предсказания. Алгоритм автоматического генерирования искусственных нейронных сетей смог обеспечить необходимый уровень точности для ряда показателей. Реализованный алгоритм восстановления пропусков в таблице данных позволил повысить эффективность предложенного подхода. Однако следует отметить, что ряд величин не удалось прогнозировать с достаточной точностью. Это может быть связано с тем, что в статистике не учтены все влияющие факторы. Стоит подчеркнуть необходимость продолжения процесса накопления выборки данных, т.к. ее расширение позволит построить более точные модели.

Дальнейшими перспективными направлениями в данной области могут считаться применения:

- рекуррентных нейронных сетей;
- альтернативных методов обучения ИНС;
- привлечение других интеллектуальных информационных технологий;
- объединение ИИТ в коллектизы.

СПИСОК ЛИТЕРАТУРЫ

- [Галушкин, 2010] Галушкин А.И. Нейронные сети: основы теории. – М.: Горячая линия-Телеком, 2010. – 496 с.
- [Meyer-Nieberg, 2007] S. Meyer-Nieberg, H.-G. Beyer, Self-Adaptation in Evolutionary Algorithms, Studies in Computational Intelligence (SCI) 54, 47–75 (2007).
- [Семенкина, 2013] Семенкина М. Е. Самоадаптивные эволюционные алгоритмы проектирования информационных технологий интеллектуального анализа данных. – Искусственный интеллект и принятие решений. №1. 2013, С. 13–23.
- [Semenkin et al., 2012] E. Semenkin, M. Semenkina, Self-configuring genetic algorithm with modified uniform crossover operator, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7331 LNCS (PART 1), pp. 414–421.
- [Khritonenko et al., 2013] D. Khritonenko, E. Semenkin, Distributed self-configuring evolutionary algorithms for artificial neural networks design, Bulletin of the Siberian State Aerospace University named after academician M. Reshetnev. № 4 (50). Р. 112–116, 2013.
- [Рутковская и др., 2006] Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: пер. с польск. И. Д. Рудинского. М. : Горячая линия. Телеком, 2006. – 383 с.
- [Снитюк, 2006] Снитюк В.Е. Эволюционный метод восстановления пропусков в данных. – Сборник трудов VI Межд. конф. «Интеллектуальный анализ информации». Киев, 2006, с. 262–271.
- [Efron, 1983] Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation. J. Am. Stat. Assoc., 78:316-331, 1983.

УДК 681.3

СОДЕРЖАТЕЛЬНАЯ ИНТЕРПРЕТАЦИЯ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА¹

Л.Ш. Шигабутдинова (*liliya.scherbakova@gmail.com*)

Ульяновский государственный технический университет, Ульяновск

Данная статья посвящена проблеме содержательной интерпретации временных рядов. Вниманию читателя предлагается программный продукт, позволяющий распознавать тенденции временного ряда, а также получить лингвистические оценки их фрагментов. Работа системы включает в себя два этапа: преобразование исходного временно го ряда с помощью генетического алгоритма и онтологический анализ полученного ряда с целью получения экспертной лингвистической интерпретации его участков.

Введение

В условиях современной действительности очень важной является проблема извлечения знаний для принятия управленческих решений. Часто исходные данные представляют собой временные ряды. Интеллектуальный анализ временных рядов (Time-Series Data Mining) в последние десятилетия стал одним из перспективных и динамично развивающихся направлений Data Mining, которые включают в себя множество подходов и методов, среди которых как широко известные математико-статистические, так и методы нечеткого анализа, в том числе и гибридные подходы.

В определенных случаях решением обозначенной проблемы может являться распознавание тенденций временного ряда. Под ним понимаются определение участков ряда, обладающих определенным трендом, и его описание. Полученные тенденции могут в дальнейшем использоваться для прогнозирования временного ряда, его кластеризации, классификации. Для цели принятия решений очень важным является представление тенденций временного

¹ Исследование выполнено в рамках гранта РФФИ №14-01-31080 «Формирование навигационной структуры электронного архива технических документов на основе онтологических моделей».

ряда в лингвистических оценках тренда, так как содержательная интерпретация лучше всего воспринимается человеком. Таким образом, интерпретация в терминах экспертных оценок конкретной предметной области позволила бы быстрее и качественнее принять управленческое решение.

Одним из наиболее перспективных направлений является использование для задач содержательной интерпретации онтологий [Gruber, 1993], [Наместников и др., 2010], [Наместников, 2009]. Онтологическое моделирование в настоящее время является весьма популярным и быстроразвивающимся научно-техническим направлением. Предпосылками его были исследования и разработки в таких областях, как искусственный интеллект, информационные технологии и, в последнее время, Semantic Technologies и Semantic Web [Ефименко и др., 2011].

Извлеченные в результате распознавания тенденций временно-го ряда знания в виде лингвистических термов, понятных пользователю, могут быть интегрированы с базой знаний для использования в системах принятия решений и экспертных системах, как например, в работах [Ануреев и др., 2010] и [Гончар и др., 2006]. Среди преимуществ использования онтологий можно выделить возможность «переориентации» онтологической системы на решение задач других предметных областей путем замены онтологии, а также универсализацию процесса анализа временных рядов и наглядность формализованного представления проблемной области.

Задачей данного исследования является разработка модели и программной системы онтологического распознавания тенденций временных рядов.

Так как временные ряды могут описывать изменения значения показателя за довольно большой промежуток времени, актуальной задачей является приведение временного ряда к более компактному виду для максимального сокращения вычислительных мощностей на обработку информации. При этом важно удалить из исходного ряда только те точки, наличие или отсутствие которых минимально оказывается на общей динамике ряда, т.е. отбросить случайную компоненту и сосредоточиться на систематической. Для решения данной задачи предлагается использовать генетический алгоритм.

1. Преобразование исходного временного ряда

Работа системы включает в себя два этапа: преобразование исходного временного ряда с помощью генетического алгоритма и онтологический анализ полученного ряда.

Предварительное преобразование временного ряда реализуется для того, чтобы представить исходный ряд наиболее компактно.

Разрабатываемая система разбивает исходный ряд на интервалы, в которых каждое последующее значение показателя минимально отличается от предыдущего. Таким образом, крайние точки полученных интервалов приближенно определяют поведение всего ряда.

Для реализации данного этапа было выбрано использование генетических алгоритмов, которые часто используются для задач функциональной и структурной оптимизации.

Генетическими они называются потому, что строятся на принципах эволюции биологических организмов Чарльза Дарвина. Популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора.

Основные принципы генетических алгоритмов были сформулированы Дж. Холландом [Holland, 1975]. Генетический алгоритм приближенно моделирует процессы, происходящие в популяциях.

Наиболее приспособленные к окружающим условиям особи получают больше шансов воспроизвести потомство, следовательно, гены высокоадаптированных особей распространяются в популяции, количество их потомков возрастает в каждом последующем поколении, т.е. вид развивается за счет приспособления к среде обитания.

Для задачи данного исследования каждый возможный вариант решения (т.е. особь или хромосома) представляет собой набор нулей и единиц. Применительно к данной задаче единицей будет обозначаться сохранившаяся в ряде точка, нулем – удаленная точка. Исходный ряд со всеми точками всегда будет обозначаться набором единиц, причем их количество будет равным числу точек во временном ряде.

Например, имеется исходный временной ряд со значениями показателя для десяти точек во времени:

$$(0, 5) (1, 7) (2, 3) (3, 15) (4, 10) (5, 13) (6, 7) (7, 13) (8, 2) (9, 2).$$

Исходный ряд может быть закодирован как (1111111111).

Предположим, что в ходе работы генетического алгоритма (преобразования ряда) точки с номерами 2 и 5 (индексация с нуля) не вошли в результирующий ряд, тогда результатом работы алгоритма будет хромосома (1101101111).

Таким образом, зная исходные значения пар (время, показатель), можно соотнести эти значения с хромосомой и получить значения пар (время, показатель) для преобразованного ряда.

На начальном этапе работы генетического алгоритма случайным образом формируется исходная популяция, число особей (хромосом) в которой задается пользователем.

Оценивается приспособленность каждой особи с помощью целевой функции F . Приспособленность в этом случае определяется количеством точек, вошедших в результирующий временной ряд, а также суммой отклонений значений уровней ряда в последующий момент времени от предыдущего. Чем меньше значение F , тем оптимальнее решение.

Следующим этапом является скрещивание особей популяции, которое происходит с заданной вероятностью. Случайным образом определяется точка внутри хромосомы выбранных особей-родителей (двух особей с высокой приспособленностью), в которой обе хромосомы делятся на две части и обмениваются ими. Таким образом, особь-потомок получает генетический материал от обоих родителей.

В программной системе определен также оператор мутации. С небольшой вероятностью случайный ген потомка инвертируется на противоположный.

Приспособленность потомков также оценивается с помощью функции F , затем они помещаются в новое поколение.

Цикл воспроизводства повторяется до тех пор, пока не будет найдено удовлетворительное решение, либо не будет достигнуто заданное количество итераций.

Решение считается приемлемым, когда находится хромосома, для которой

$F(\text{хромосома}) < \text{Порог};$

Порог = СумНачОткл/(ДР* q),

где СумНачОткл – сумма отклонений каждого последующего значения показателя от значения показателя в предыдущий момент

времени, ДР – длина ряда или количество точек, а q – некий коэффициент, вычисленный экспериментальным путем.

Пример работы системы представлен на рисунке 1. Точки исходного ряда обозначены ромбами, результирующий ряд обозначен пунктирной линией.

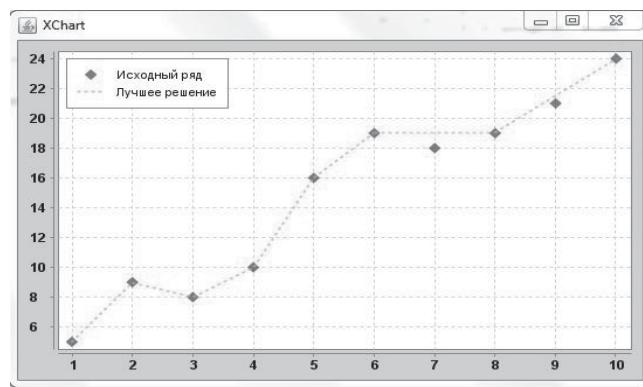


Рис. 1. Результат преобразования

Как видно из рисунка 1, наилучшим решением оказался временной ряд, из которого исключены точки 7 и 9, т.е. хромосома, закодированная как (1111110101).

На рисунке 2 изображен график изменения функции приспособленности и порога для каждого поколения.

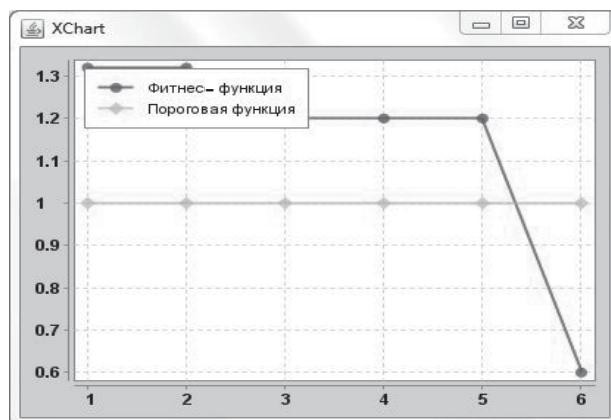


Рис. 2. Функция принадлежности и порог

2. Онтологический анализ временного ряда

Второй этап – онтологический анализ полученного временного ряда и непосредственное получение лингвистических оценок его участков. Для осуществления данной задачи нужно, прежде всего, определиться с набором параметров, по которым можно охарактеризовать тенденцию. В рамках данного исследования предлагается передавать на вход онтологии параметры треугольников, которые могут быть получены из преобразованного ряда – длина основания, тангенс угла наклона и отклонение по оси y .

Для того, чтобы получить треугольники, достаточно будет попарно получить координаты соседних точек, которые вошли в итоговый график, и построить по ним прямоугольные треугольники. Нижнее основание такого треугольника будет параллельно оси времени, одна сторона – параллельна оси параметра, а третья сторона будет являться отрезком между двумя соседними точками. Результат построения треугольников представлен на рисунке 3.

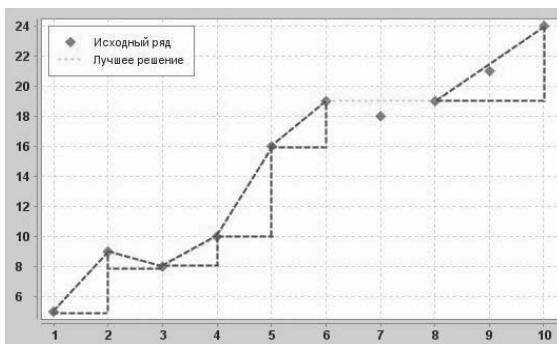


Рис. 3. Результат триангулярного преобразования временного ряда

Реализация содержательной интерпретации проведена с использованием RDF-схемы и описанием онтологии на языке RDF. Параметры треугольников подаются на вход онтологической модели. RDF-схема определяет описание временного ряда как совокупности треугольников, характеризующихся определенными на предыдущем этапе параметрами (один фрагмент – один треугольник), лингвистическим значением и метками времени в терминах «объект – предикат – субъект» (рисунок 4).

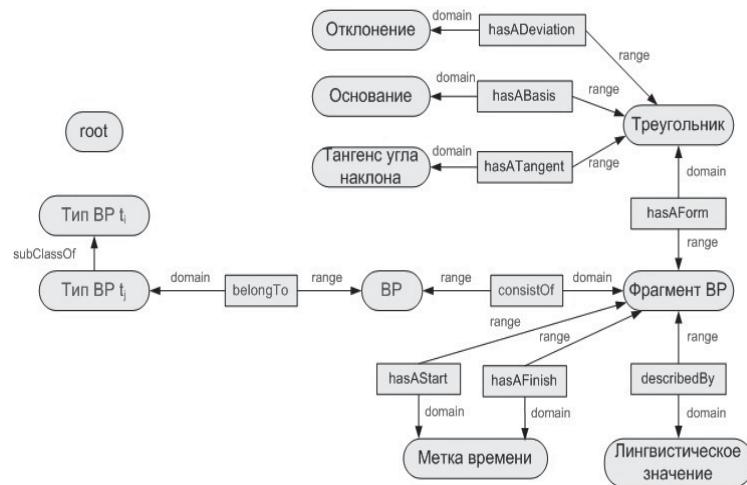


Рис. 4. RDF-схема (структуря онтологии)

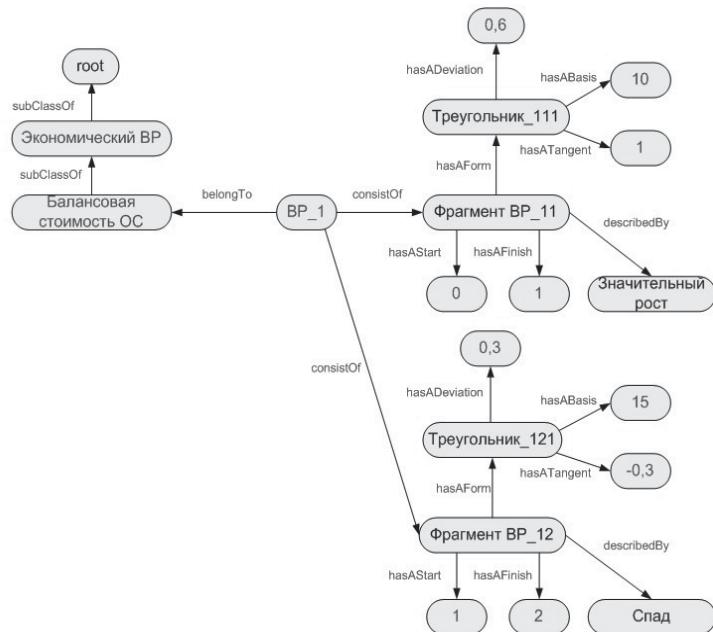


Рис. 5. Пример фрагмента онтологии

Сама онтологическая модель будет представлять собой указанную ранее схему, наполненную определенными значениями данных. Фрагмент онтологии на примере балансовой стоимости основного средства представлен на рисунке 5. Значения данных предметной области заполняются экспертом.

Заключение

В условиях настоящего времени все большее значение приобретает как можно более быстрое и правильное принятие управленческих решений. Разработанная в рамках данного исследования программная система содержательной интерпретации временных рядов позволяет распознать основные тенденции исходного временного ряда (благодаря использованию генетических алгоритмов), а также получить содержательную интерпретацию тенденций в лингвистических экспертных оценках, наиболее понятных для пользователей системы, используя онтологию предметной области.

Данную программную систему можно использовать в качестве составной части более крупных онтологий при разработке систем поддержки принятия решений и экспертных систем.

В связи с использованием генетического алгоритма результаты работы системы не являются избыточными, позволяют работать лишь с систематической компонентой временного ряда, выявляя только основные и значимые тенденции.

СПИСОК ЛИТЕРАТУРЫ

- [Ярушкина, 2004] Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие / Н. Г. Ярушкина. – М.: Финансы и статистика, 2004. – 320 с.
- [Gruber, 1993] Gruber T.R., A translation approach to portable ontologies.// Knowledge Acquisition – 1993. – №5 (2).
- [Наместников и др., 2010] Наместников А.М., Филиппов А.А. Концептуальная индексация проектных документов // Автоматизация процессов управления. – 2010. – №2 (20).
- [Наместников, 2009] Наместников А.М. Интеллектуальные проектные репозитории. – Ульяновск: УлГТУ, 2009.
- [Ефименко и др., 2011] Ефименко И.В., Хорошевский В.Ф. Онтологическое моделирование экономики предприятий и отраслей современной

России. Часть 1. Онтологическое моделирование: подходы, модели, методы, средства, решения. – Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011.

[Ануреев и др., 2010] Ануреев И.С., Загорулько Г.Б., Загорулько Ю.А. Подход к разработке системы поддержки принятия решений на примере нефтегазодобывающего предприятия // Известия Томского политехнического университета. – 2010. – №5 (том 316).

[Гончар и др., 2006] Гончар А.М., Загорулько Г.Б., Рубан М.Н., Рябков А.Н. Интеллектуальная система сопровождения профилактики и лечения элементозов // Информационный бюллетень рабочего семинара «Наукомическое программное обеспечение». – С. 48–51. – Новосибирск: ИСИ СО РАН, 2006. – 128 с.

[Holland, 1975] Holland J.H. Adaptation in natural and artificial systems/ University of Michigan Press, Ann Arbor, 1975.

СЕКЦИЯ 5 | МНОГОАГЕНТНЫЕ, СЕТЕВЫЕ
И РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ

УДК 519.83,004.722

**ТЕОРЕТИКО-ИГРОВЫЕ АЛГОРИТМЫ ФОРМИРОВАНИЯ
БЕСПРОВОДНОЙ АД НОС СЕТИ¹**

Н.И. Базенков (*n.bazenkov@gmail.com*)

Институт проблем управления РАН им. В.А. Трапезникова, Москва

Рассматривается задача формирования эффективной топологии беспроводной ad hoc сети. Требуется сформировать связную сеть с минимальной суммарной мощностью передатчиков. Исследуются теоретико-игровые алгоритмы формирования сети, использующие предложенный ранее метод двойного наилучшего ответа. Получены оценки эффективности и вычислительной сложности в сравнении с базовым теоретико-игровым алгоритмом наилучшего ответа.

Введение

В докладе исследуются ad hoc сети, которые образуются беспроводными узлами без использования какой-либо дополнительной инфраструктуры. Устройства в ad hoc сетях обычно работают от автономных аккумуляторов, поэтому большое значение имеют методы обеспечения энергоэффективности. Управлением топологией (Topology Control) называется динамическая настройка мощности передатчиков для поддержания связности сети, минимизации энергопотребления или улучшения качества связи [Santi, 2005]. Мы рассматриваем только процесс формирования сети, когда требу-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 11-01-00771).

ется назначить каждому узлу такую мощность, чтобы обеспечить связность сети и минимизировать суммарную мощность узлов.

В ad hoc сетях желательно использовать децентрализованные алгоритмы, чтобы избежать глобального обмена информацией. Подобные задачи изучались также в работах по коллективному поведению автоматов [Стефанюк и др., 1967]. Для создания распределенных алгоритмов оптимизации и управления естественным образом подходят методы теории игр, когда каждый узел оптимизирует локальную функцию полезности, не учитывая полезности других узлов [Shi, 2012]. Таким образом, беспроводная сеть представляется как коллектив эгоистичных интеллектуальных агентов по классификации [Тарасов, 2002].

Рассматривается игра формирования сети, предложенная в [Komali et.al., 2008]. Исследуются алгоритмы формирования сети, впервые предложенные в [Базенков, 2013а], [Базенков, 2013б]. Алгоритмы используют новый теоретико-игровой метод двойного наилучшего ответа, который основан на идеях рефлексивных игр [Новиков и др., 2003] [Корепанов, 2011]. Сравнивается эффективность агентов первого и нулевого ранга рефлексии, получены оценки вычислительной сложности алгоритмов.

1. Задача формирования топологии

Сеть представлена как множество узлов $N=\{1, \dots, n\}$, расположенных на плоскости. Каждый узел может изменять мощность своего передатчика $p_i \in [0, p_i^{\max}]$. Вектор мощностей $p = (p_1, \dots, p_n)$ порождает неориентированный граф связей $g(p) = (N, E(p))$, где $E(p)$ – множество ребер. На рисунке 1 показано, как формируются связи. Такая модель беспроводной сети часто используется в работах по формированию топологии [Santi, 2005].

Ребро (i, j) возникает, если выполняется условие $p_i \geq h_{ij}$ и $p_j \geq h_{ji}$, где $h_{ij} > 0$ – минимальная мощность, необходимая для передачи данных от i к j . Считаем, что h_{ij} зависит от расстояния между узлами:

$$p_i \geq h_{ij} = (\delta_{ij})^\alpha. \quad (1.1)$$

Здесь δ_{ij} – расстояние от i до j , $\alpha \geq 2$ – показатель затухания радиосигнала. Необходимо найти такие мощности, чтобы сеть была связной, а сумма всех мощностей минимальна.

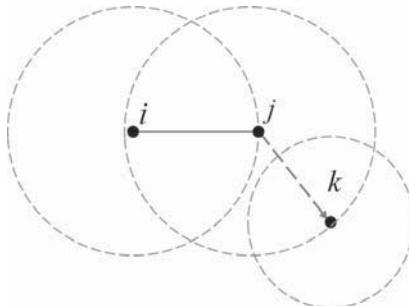


Рис. 1. Образование связи. Сплошная линия – установившаяся связь (i,j). Мощность узла j достаточна для образования связи (j,k), а мощность узла k - недостаточна. Пунктирными окружностями показаны области действия передатчика

$$\begin{aligned} \sum_{i \in N} p_i &\rightarrow \min \\ p_i &\in [0, p_i^{\max}] \\ g(p) &-\text{связный граф.} \end{aligned} \quad (1.2)$$

Задача (1.2) является NP-трудной [Calinescu et.al., 2012], поэтому необходимо использовать эвристические алгоритмы, дающие некоторое локально-оптимальное решение. Один из подходов к созданию таких алгоритмов заключается в использовании методов некооперативной теории игр.

2. Игра формирования сети

Алгоритмы, основанные на теории игр, предполагают, что каждый узел действует независимо, стремясь максимизировать свою функцию полезности, не учитывая полезности других узлов. В терминах [Тарасов, 2002], здесь сеть представляет собой распределенную систему, в которой роли всех агентов одинаковы, отсутствует координатор, агенты эгоистичны, но не злонамеренны и каждый агент принимает решение, основываясь только на своей локальной информации.

Исследуется игра формирования сети, предложенная в [Komali et.al., 2008]. Агентами являются узлы сети. Действие агента i – это выбор мощности его передатчика. Функция полезности сформулирована так, чтобы отражать требование связности сети и минимизации мощности:

$$u_i(p) = Mf_i(g(p)) - p_i, \quad (2.1)$$

где $f_i(g(p))$ – число узлов, с которыми узел i связан в графе $g(p)$ напрямую или через другие узлы; $M > \max_{i \in N} p_i^{\max}$ – константа, отражающая приоритет связности сети перед экономией мощности.

В [Komali et.al., 2008] был предложен алгоритм последовательных наилучших ответов (iterated best response). Узел, применяющий такое правило, максимизирует свою полезность, не рассматривая возможные действия других узлов. Этот алгоритм формирует связную сеть, которая является равновесием Нэша и локально-оптимальным решением задачи (2). В ходе исследований было доказано, что при таком подходе полученное равновесие может сильно отличаться от оптимального.

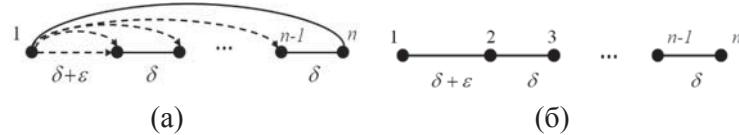


Рис. 2. Сумма мощностей в наихудшем равновесии (а) равна $\sum p_i^{\text{worst}} = O(n^\alpha)$, а в оптимальном (б) – $\sum p_i^{\text{best}} = O(n)$

Утверждение 1. Существует такое расположение узлов, при котором суммарная мощность сети в наихудшем равновесии отличается от оптимального решения в $O(n^{\alpha-1})$ раз. На рисунке 2 приведен пример такой сети.

Другой недостаток алгоритма наилучших ответов связан с существованием недопустимых равновесий, в которых график $g(p)$ не связан.

Утверждение 2. Все недопустимые равновесия в игре формирования сети не оптимальны по Парето.

Утверждение 3. Любое недопустимое равновесие останется устойчивым при использовании алгоритма последовательных наилучших ответов.

Если в сети по каким-то причинам установится недопустимое равновесие (например, в результате сбоя в работе), то алгоритм последовательных наилучших ответов не сможет его изменить и процесс формирования сети придется запускать заново. Для устранения этих недостатков алгоритма наилучших ответов был предложен метод, названный двойным наилучшим ответом.

3. Алгоритмы формирования сети

Наилучшим ответом (best response) агента i на обстановку p_{-i} называется действие

$$BR_i(p_{-i}) = \arg \max_{x \in [0, p_i^{\max}]} u_i(x, p_{-i}). \quad (3.1)$$

Правило (3.1) моделирует поведение «недальновидного» агента, который предполагает, что действия его оппонентов не изменятся в будущем. В теории рефлексивных игр такие агенты обозначаются как имеющие нулевой ранг рефлексии [Новиков и др., 2002], [Корепанов, 2011].

Двойной наилучший ответ использует идею рефлексивных игр, что агент может прогнозировать возможную реакцию оппонентов на свои действия:

$$DBR_i(p_{-i}) = \arg \max_{x \in [0, p_i^{\max}]} u_i(x, BR_{-i}(p^{i,x})). \quad (3.2)$$

Здесь $p^{i,x} = (p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_n)$ и $BR_{-i}(p^{i,x}) = (BR_1(p_1^{i,x}), \dots, BR_{i-1}(p_{i-1}^{i,x}), BR_{i+1}(p_{i+1}^{i,x}), \dots, BR_n(p_n^{i,x}))$ – одновременные наилучшие ответы других агентов на действие x агента i . Правило (3.2) определяет поведение агента 1-го ранга рефлексии.

Выражение (3.2) предполагает, что агент может вычислить наилучшие ответы своих оппонентов и обладает необходимой для этого информацией. В системах с большим числом агентов эта информация может быть недоступна. Естественно ограничить количество агентов, для которых i может вычислять наилучший ответ.

Рефлексивным множеством агента i назовем множество $R_i \subseteq N$, включающее агентов, для которых i может вычислить наилучший ответ.

Ограниченному двойному наилучшему ответом агента i на обстановку p_{-i} называется действие

$$DBR_{i,R_i}(p_{-i}) = \arg \max_{x \in [0, p_i^{\max}]} u_i(x, p_{N \setminus R_i}, BR_{R_i}(p^{i,x})), \quad (3.3)$$

где $p_{N \setminus R_i}$ – действия агентов, не входящих в рефлексивное множество агента i , $BR_{R_i}(p^{i,x})$ – действия агентов, входящих в рефлексивное множество агента i . В данной работе исследовался алгоритм,

котором рефлексивное множество узла i составляют те узлы, которые находятся в пределах досягаемости максимальной мощности передатчика.

На основе правил двойного наилучшего ответа (3.2) и (3.3) разработано два алгоритма формирования сети: статический, где все агенты используют двойной наилучший ответ, и динамический, где агенты используют двойной наилучший ответ, только если не могут увеличить свою полезность, применив обычное правило (3.1). В динамическом алгоритме узлы могут использовать двойной наилучший ответ не более m раз.

Утверждение 4. Если p^* равновесие Нэша и $g(p)$ не связан, то $\exists i \in N: DBR_i(p^*_{-i}) \neq p_i^*$.

Следствие. Все не оптимальные по Парето равновесия не устойчивы к двойному наилучшему ответу.

Таким образом, использование двойного наилучшего ответа позволяет восстанавливать связность, не запуская заново весь процесс формирования сети.

Алгоритмы на основе двойного наилучшего ответа для многих исследованных примеров меньше зависят от порядка действий узлов, чем обычный наилучший ответ. В частности, для сети на рисунке 2 двойной наилучший ответ при любом порядке ходов получает единственное оптимальное равновесие.

4. Вычислительная сложность

Была исследована вычислительная сложность обычного наилучшего ответа и двойного наилучшего ответа.

Утверждение 5. Узлу для вычисления наилучшего ответа в игре с полезностями (2.1) требуется $O(N+E)$ операций для текущего графа сети $g=(N,E)$.

Утверждение 6. Узлу для вычисления двойного наилучшего ответа в игре с полезностями (2.1) требуется $O(VR(N+E))$ операций при дополнительном обмене $2VR$ служебными сообщениями. Здесь V – число соседей узла, R – мощность рефлексивного множества.

Следствие. Если в рефлексивное множество входят соседи узла, то для вычисления двойного наилучшего ответа требуется $O(V^2(N+E))$ и обмен $2V^2$ сообщениями.

Утверждение 7. В динамическом алгоритме вычисление двойного наилучшего ответа требует $O(V^2N)$ операций.

5. Эффективность алгоритмов

Алгоритмы исследовались и сравнивались по трем критериям: суммарная мощность, максимальная мощность в сети и оценка интенсивности помех в сети, предложенная в [Rickenbach et.al., 2009]. Теоретико-игровые алгоритмы сравнивались централизованным, формирующим минимальное оставное дерево (MST). На рисунках 4-8 показаны результаты экспериментов. Наилучшие результаты по всем трем критериям среди теоретико-игровых алгоритмов показал динамический алгоритм. Время сходимости двойного наилучшего ответа выше, чем у базового алгоритма. Все теоретико-игровые алгоритмы показали результаты, худшие, чем централизованный, который использует глобальную информацию о стоимостях связей в сети.

Для использования в реальных сетях наиболее предпочтительным является динамический алгоритм, который улучшает качество сети постепенно. Можно предложить модификацию, в которой двойной наилучший ответ будет применяться узлами только в те моменты, когда отсутствуют запросы на передачу данных и допустимо нарушить связность сети на некоторое время. В случае поступления запроса на отправку данных узлы могут переключаться на наилучший ответ и восстанавливать связность сети.

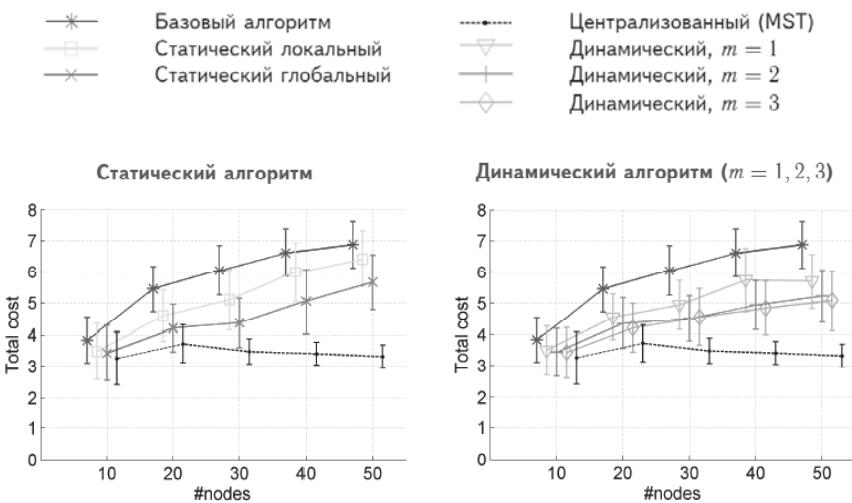


Рис. 3. Суммарная мощность сети

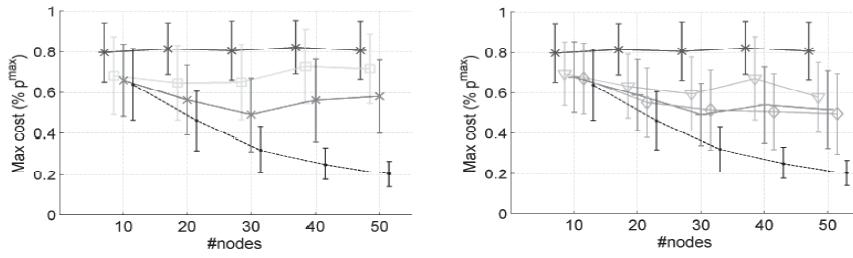


Рис. 4. Максимальная мощность в сети

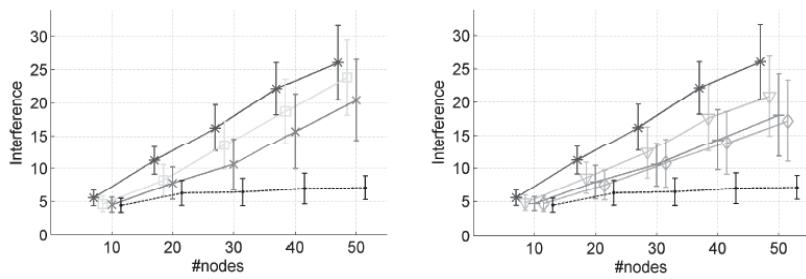


Рис. 5. Интенсивность помех в сети

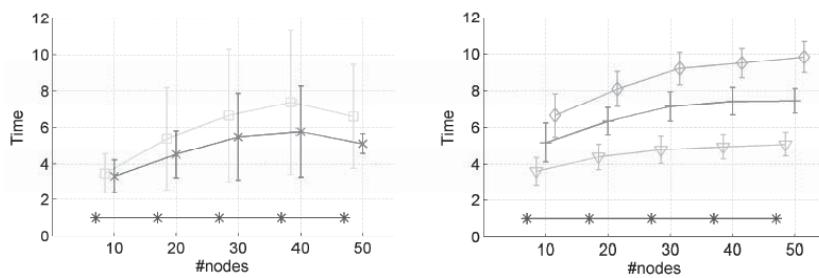


Рис. 6. Время сходимости алгоритмов

Заключение

В работе исследовались теоретико-игровые алгоритмы формирования беспроводной ad hoc сети. Для известной из литературы игры формирования сети доказано, что наихудшее равновесие Нэша, получаемое базовым алгоритмом наилучшего ответа, может

полиномиально отличаться от оптимального решения. Доказано, что в игре существуют равновесия, не являющиеся допустимыми решениями задачи формирования сети.

Предложенный метод двойного наилучшего ответа делает неустойчивыми все недопустимые равновесия. Исследовалась эффективность двух алгоритмов, разработанных на основе двойного наилучшего ответа. По всем трем критериям наилучшие результаты показал динамический алгоритм, в котором узлы переключаются между обычным и двойным наилучшим ответом. Получены оценки вычислительной сложности алгоритмов.

В дальнейшем планируется продолжить исследование предложенного метода, в частности, рассмотреть другие критерии качества сети и исследовать вопрос о снижении вычислительной сложности алгоритмов, что особенно важно для практической реализации предложенных алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

[Базенков, 2013а] Базенков Н.И. Динамика двойных наилучших ответов в игре формирования топологии беспроводной ad hoc сети // Управление большими системами. – Вып. 43. – М.: ИПУ РАН, 2013.

[Базенков, 2013б] Базенков Н.И. Динамика двойных наилучших ответов в игре формирования топологии беспроводной ad hoc сети // Труды II Международной летней школы-семинара «Интеллектуальные системы и технологии: современное состояние и перспективы» (ISyT'2013). – 2013.

[Корепанов, 2011] Корепанов В.О. Модели рефлексивного группового поведения и управления. – М.: ИПУ РАН, 2011.

[Новиков и др., 2003] Новиков Д.А., Чхартишвили А.Г. Рефлексивные игры. – М.: СИНТЕГ, 2003.

[Степанюк и др., 1967] Степанюк В.Л., Цетлин М.Л. О регулировке мощности в коллективе радиостанций // Проблемы передачи информации. – 1967. – Т.3. Вып.4.

[Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002.

[Calinescu et al., 2012] Calinescu G., Qiao K. Asymmetric Topology Control: Exact Solutions and Fast Approximations // Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM). – 2012.

[Komali et al., 2008] Komali R.S., MacKenzie A.B., Gilles R.P. Effect of Selfish Node Behavior on Efficient Topology Design // IEEE Transactions on Mobile Computing. – 2008. – Vol.7, № 9.

[Rickenbach et al., 2009] Rickenbach P., Wattenhoffer R., Zollinger A. Algorithmic Models of Interference in Wireless Ad hoc and Sensor Networks // IEEE/ACM Transactions on Networking (TON). - 2009. - Vol. 17, Issue 1.

[Santi, 2005] Santi P. Topology Control in Wireless Ad Hoc and Sensor Networks // Journal ACM Computing Surveys (CSUR). – 2005. – Vol. 37, Issue 2.

[Shi, 2012] Shi H.-Y., Wang W.-L., Kwok N.-M., Chen S.-Y. Game Theory for Wireless Sensor Networks: A Survey // Sensors. – 2012. – Vol. 12, Issue 7.

УДК 004.8: 519.17: 681.3

МНОГОСТОРОННИЕ РЕСУРСНО-ЦЕЛЕВЫЕ СЕТИ В МОДЕЛИРОВАНИИ ВЗАИМОДЕЙСТВИЙ МЕЖДУ КОГНИТИВНЫМИ АГЕНТАМИ¹

В.С. Дюндюков (*vsd89@yandex.ru*)

МГТУ им. Н.Э. Баумана

В.Б. Тарасов (*vbulbov@yahoo.com*)

МГТУ им. Н.Э. Баумана

В работе рассмотрены ресурсно-целевые сети, которые позволяют моделировать обмен ресурсами при взаимодействии агентов, а также процессы формирования и функционирования многоагентных систем с учетом типологии агентов и их целевых установок. Предварительно изложены предпосылки перехода от ресурсных к ресурсно-целевым сетям, связанные со статусом агента как целеустремленной системы и описанием разных типов агентов. Дано формальное определение ресурсно-целевой сети. Проанализированы основные виды взаимодействий между агентами, представленных с помощью двусторонних взвешенных графов.

Введение

В 2009 году О.П.Кузнецов предложил формальный аппарат *ресурсных сетей* [Кузнецов, 2009]. Ресурсной сетью называется двусторонний ориентированный граф, вершинам x_i которого поставлены в соответствие неотрицательные числа $q_i(t)$, изменяющиеся в дискретном времени t и называемые ресурсами, а дугам – неотрицательные числа w_{ij} , именуемые пропускными способностями (проводимостями). В отличие от транспортных сетей и классической потоковой модели Форда-Фалкерсона, в которых ресурс течет от источников к стокам и расположен в дугах [Форд и др., 1973], в ресурсной сети ресурсы находятся в вершинах, а обмен ресурсами зависит от проводимостей дуг [Кузнецов, 2009; Кузнецов и др., 2010].

В [Дюндюков, 2010] аппарат ресурсных сетей был использован как средство формального анализа взаимодействий между искусственными агентами и построения многоагентных систем (МАС).

¹ Работа выполнена при финансовой поддержке РФФИ (проект №14-07-00846-а).

Затем в [Дюндюков и др., 2011] был предложен вариант расширения ресурсных сетей – *ресурсно-целевые сети* (РЦС), адаптированные к понятию агента и методологии организации взаимодействий между агентами [Тарасов, 2002]. В [Дюндюков и др., 2013] было описано применение РЦС для агентно-ориентированного моделирования в железнодорожной отрасли. Настоящая статья посвящена дальнейшему развитию аппарата РЦС в плане моделирования рефлексивных отношений и асимметричных связей между когнитивными агентами как по целям, так и по ресурсам.

1. Архитектура РЦВД и типология агентов

Первоначально рассмотрим предпосылки перехода от ресурсных к ресурсно-целевым сетям при моделировании взаимодействий между когнитивными агентами и формировании МАС. В качестве базовой модели агента возьмем общую архитектуру «ресурсы-цель-восприятие-действие». Кратко проанализируем взаимосвязь этих понятий.

Причиной деятельности (и любой активности) агента является потребность, которая понимается как рассогласование между текущим и желаемым состоянием агента. Отражение и понимание этой потребности связано с формированием мотива (движущего фактора деятельности). Роль мотива для агента аналогична роли силы в механике. Реализация любого мотива связана с построением цели.

Целенаправленность и автономность являются ключевыми свойствами агентов. Автономность означает функционирование «по своим законам» на основе имеющегося ресурсного обеспечения, а также периодического получения ресурсов из среды или от других агентов.

Под ресурсами здесь понимаются любые средства, полезные для достижения цели агента или МАС. Термин «ресурс» дословно означает «восполнение или восстановление источника». Выделяют две основные характеристики ресурса [Волков, 1986; Тарасов, 2002]: а) «объем ресурса» как его меру в пространстве (например, объем перерабатываемой информации, объем памяти компьютера); б) «действие ресурса» – его меру во времени. Очевидно, что один и тот же объем ресурса может оказывать различное влияние на реализацию процессов при различном его действии.

Величина имеющегося у агента ресурса тесно связана с определением роли агента в МАС и взаимосвязей между ролями. Взаимодействие между агентами, создание и функционирование МАС предполагают гибкое распределение, перераспределение и коллективное использование ресурсов отдельных агентов.

Наша архитектура РЦВД (рис.1) опирается на представление о целях и ресурсах как важнейших характеристиках внутренней среды агента, а также рассмотрение восприятия и действия как базовых процессов взаимодействия с внешней средой.

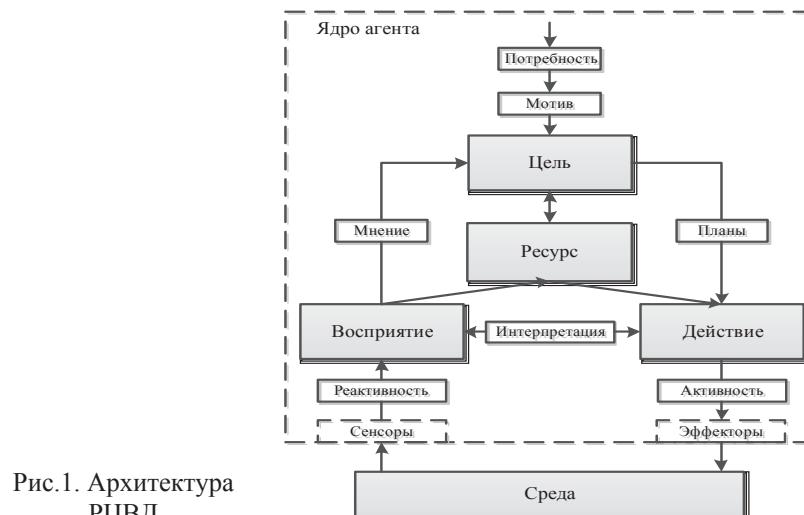


Рис.1. Архитектура РЦВД

Взаимодействие агентов определяется следующими условиями [Тарасов, 2002]: 1) совместимость целей или намерений агентов; 2) отношение агентов к ресурсам и объемы имеющихся у них ресурсов, потребность в дополнительных ресурсах и их совместном использовании; 3) наличие у агентов соглашений и обязательств друг перед другом.

Для определения структур взаимодействия между агентами воспользуемся типологией агентов по двум критериям, предложенной в [Тарасов, 2002] (см.таблицу). Здесь все множество агентов разбивается на благонамеренных, эгоистичных и альтруистичных агентов, а также выделяются разрешенные и запрещенные структуры взаимодействий.

Таблица

Классификация агентов по двум критериям

Тип агента	Способность формировать собственные цели	Участие в формировании коллективных целей
Благонамеренный	+	+
Эгоистичный	+	-
Альтруистичный	-	+

Благонамеренный агент a_b – это агент, имеющий свои цели (интересы) и способный как участвовать в формировании коллективных целей, так и принимать цели другого агента. Он активно участвует в обмене ресурсами и создании МАС, сохраняя часть ресурса для собственных нужд.

Эгоистичный агент a_e стремится к достижению исключительно своих целей, игнорируя интересы других агентов. Он участвует в обмене ресурсами тогда и только тогда, когда этот обмен ему необходим и выгоден; при этом наблюдается, скорее, не обмен, а перетекание ресурсов к эгоистичному агенту. *Альтруистичный агент* a_a априори считается неспособным к построению собственных целей и принимает чужую цель как общую. Такой агент всегда участвует в обмене ресурсами, даже если обмен будет неравнозначным, и он от него проиграет.

Типология агентов непосредственно определяет саму возможность формирования МАС. Так, например, многоагентная система не может быть образована при взаимодействии только эгоистичных или только альтруистичных агентов. В первом случае ограничением является неспособность согласования эгоистичными агентами индивидуальных целей и участия в формировании коллективных целей, а во втором случае – отсутствие собственных целей и, как следствие, невозможность построения коллективных целей. Эгоистичный агент может взаимодействовать с благонамеренными и альтруистичными агентами только при условии навязывания им своих целей, а функционирование альтруистичных агентов в МАС подразумевает принятие ими целей благонамеренного или эгоистичного агента. Однородная структура МАС возникает только при взаимодействии благонамеренных агентов, а варианты взаимодействия разнотипных агентов приводят к неоднородным структурам МАС.

Итак, формирование МАС требует согласованного восприятия

агентами внешней среды и формирования у них коллективных целей, обмена ресурсами, реализации совместных действий.

Введение *ресурсно-целевых сетей* [Дюндюков и др., 2011] с тремя разными типами вершин и двумя типами дуг позволяет моделировать целевые установки и характер поведения агентов при обмене ресурсами, а также отображать асимметричность такого обмена.

2. От ресурсных к ресурсно-целевым сетям

Следуя О.П.Кузнецовой, для моделирования обмена ресурсами в МАС будем опираться на ресурсные сети.

Определение 1. Ресурсной сетью для многоагентной системы будем называть взвешенный двусторонний ориентированный граф

$$G_{\text{RES}} = \langle A, C, RES, W \rangle, \quad (1)$$

где множество вершин A есть множество агентов, образующих МАС, множество дуг C определяет множество связей между агентами, RES – множество ресурсов МАС, причем каждый агент $a_i \in A$ имеет определенный ресурс $res(a_i) \in RES$, W – множество проводимостей w_{ij} дуг $c_{ij} \in C$ в МАС. Каждой дуге (связи между агентами) $c_{ij} \in C$ приписано неотрицательное число $w_{ij} \in W$, называемое проводимостью от агента a_i к агенту a_j .

По сути, проводимость в МАС задает предельный объем ресурса, который один агент может передавать другому в определенный промежуток времени. Ресурсная сеть называется однородной, если все проводимости в ней равны. В общем случае, когда МАС состоит из агентов разных типов, ресурсные сети являются неоднородными, поскольку в них как объем и действие ресурса, так и проводимости зависят от типа агента.

Ресурсно-целевая сеть представляет собой вариант модификации формализма ресурсных сетей, где вершины задаются двумя параметрами (тип агента и объем ресурса), а дуги – двумя видами проводимости (по целям и по ресурсам). Будем обозначать ресурсные связи сплошными стрелками, а целевые связи – пунктирными. При этом ресурсные связи могут представлять собой как дуги (ресурсы, включенные в обмен, или ресурсы совместного использования в МАС), так и петли (ресурсы «для себя»). Аналогично могут различаться и целевые связи.

Определение 2. Ресурсно-целевая сеть [Дюндюков, 2010; Дюндюков и др., 2011] есть взвешенный ориентированный мультиграф (гиперграф):

$$G_{\text{RO}} = \langle A, C, K, RES, W, t \rangle, \quad (2)$$

где множество вершин есть множество агентов A , а множество дуг C разбито на два непересекающихся подмножества: множество целевых связей C_G и множество ресурсных связей C_{RES} : $C = C_G \cup C_{\text{RES}}$, $C_G \cap C_{\text{RES}} = \emptyset$, t – множество дискретных моментов времени, $t = 0, 1, 2, \dots, n$.

Каждая вершина РЦС $a_i \in A$ определяется следующими параметрами – тип агента $k_i \in K$ и объем ресурса $\text{res}(a_i) \in RES$, а каждая дуга $c_{ij} \in C$ взвешивается с помощью значения проводимости или пропускной способности $w_{ij} \in W$. Тип агента характеризует его цели, относящиеся к ресурсам, и характер поведения в МАС. Вершины РЦС, соответствующие различным типам агентов, удобно выделять разными цветами: так, благонамеренный агент a_b может изображаться зеленым цветом, эгоистичный a_e – красным, а альтруистичный a_a – синим. Типы агентов, включенных в МАС, задают топологию РЦС.

У любых двух агентов a_i, a_j выделяют проводимости по целям $w_G(a_i, a_j)$ и проводимости по ресурсам $w_{\text{RES}}(a_i, a_j)$. В общем случае обмену целями и ресурсами соответствуют направленные навстречу друг другу стрелки орграфа, причем весовые коэффициенты дуг орграфа (значения входных и выходных проводимостей) могут быть различными.

Для простоты проводимости агентов по целям могут рассматриваться на бинарной шкале «проводимость имеется – проводимость отсутствует». Так, у эгоистичного агента имеются выходные проводимости по целям, но отсутствуют соответствующие входные проводимости. Напротив, у альтруистичного агента имеются только входные проводимости по целям. Следовательно, выполняется условие: $w_{\text{RES}}(a_{ej}, a_{ai}) = w_{\text{RES}}(a_{ej}, a_{bi}) = 0$, $w_G(a_{aj}, a_{bi}) = w_G(a_{ai}, a_{ej}) = 0$.

Соотношению между входными и выходными проводимостями по целям соответствует антитеза целенаправленности-целеустремленности агентов: так, альтруистичные агенты являются чисто целенаправленными, эгоистичные – чисто целеустремленными, а благонамеренные агенты могут переходить от целенаправленного поведения к целеустремленному и наоборот.

Наличие собственной цели у агента является определяющим фактором для его самосохранения по ресурсу. При этом имеются два граничных условия: 1) $\text{Res}(a_i(t)) \geq \text{Res}_{\min}$ – условие индивидуального выживания агента;

2) $\text{Res}(a_i(t)) \leq \text{Res}_{\max}$ – условие социального поведения агента.

В то же время наличие общей цели у агентов необходимо для формирования МАС. Коллективная цель в РЦС будет являться гранулой, выраженной с помощью подграфа РЦС по целевым связям.

В каждый дискретный момент времени t состояние сети будет определяться вектором состояния $RES(t) = (res(a_1), \dots, res(a_m))$, где m – число агентов в РЦС. Если вектор состояния ресурсно-целевой сети не изменяется со временем, то такое состояние называется устойчивым. Состояние РЦС является асимптотически достижимым, если для заранее заданного малого ζ , $\zeta > 0$, $\forall i = 1, \dots, m$, выполняется неравенство:

$$|res(a_i(t+1)) - res(a_i(t))| < \zeta. \quad (3)$$

При обмене ограниченными материальными ресурсами РЦС полагается замкнутой системой, для которой справедливо равенство:

$$\sum_{i=1}^m res(a_i) = RES_{SUM} = const. \quad (4)$$

Рассмотрим правила взаимодействия (обмена ресурсами) между парами агентов различных типов:

а) *взаимодействие двух благонамеренных агентов*. Происходит равноправный обмен информацией целевого характера, в результате которого формируется коллективная цель, а также осуществляется эффективный обмен ресурсами (рис. 2). Многоагентная система, состоящая из m подобных агентов, представляется наиболее эффективной для реализации стратегий децентрализованного искусственного интеллекта, когда формируется структура типа «полный граф».

Если ресурсно-целевая сеть состоит из благонамеренных агентов и для каждой из дуг w_o и w_{RES} равны между собой и $RES_{SUM} \leq W_{RES}$, где W_{RES} – максимально отдаваемый агентами сети ресурс за одну-единственную итерацию, $W_{RES} = \sum_{i=1}^m \sum_{j=1}^m w_{RES}(a_i, a_j) = = w_{RES}m(m-1)$, то с течением времени ресурс системы распределится между агентами равномерно и $RES(t) = \left(\frac{RES_{SUM}}{m}, \dots, \frac{RES_{SUM}}{m}\right)$.

В противном случае, если хотя бы у двух агентов ресурсы не будут равны в начальный момент времени, то выравнивание ресурсов в сети не произойдет;

б) *взаимодействие эгоистичного a_e и альтруистичного a_a агентов*. Эгоистичный агент навязывает альтруистичному свою цель и

использует для ее достижения чужие ресурсы (рис. 3). Фактически происходит перекачка ресурсов от a_a к a_e , которая может завершиться гибелью a_a , если объем ресурса $res(a_a(t)) < res_{min}$. При наличии одного эгоистичного и n альтруистических агентов исходная структура РЦС типа «звезда» может выродиться в изолированную вершину – «моно-полиста ресурсов»;

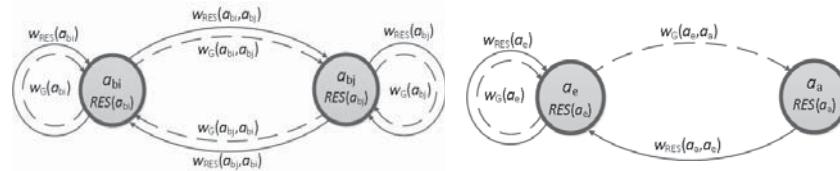


Рис. 2. РЦС для взаимодействия двух благонамеренных агентов

Рис. 3. РЦС для взаимодействия эгоистичного и альтруистичного агентов

в) *взаимодействие благонамеренного и эгоистичного агентов.* Такое взаимодействие возможно только в случае принятия a_b цели a_e . При этом взаимодействии возникает иллюзия обмена ресурсами. Эгоистичному агенту нужны ресурсы, но в ответ он стремится ничего не отдавать (рис. 4). При этом взаимодействие прекращается, если $res(a_b(t))$ близок к res_{min} (наличие «инстинкта самосохранения» у a_b);

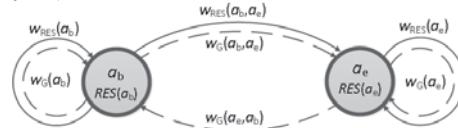


Рис. 4. РЦС для взаимодействия эгоистичного и благонамеренного агентов

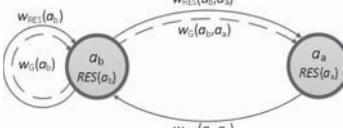


Рис. 5. РЦС для взаимодействия благонамеренного и альтруистичного агентов

г) *взаимодействие благонамеренного и альтруистичного агентов.* Здесь происходит эффективный обмен ресурсами, причем a_a разделяет цели a_b (рис. 5). В силу своей благонамеренности a_b не допускает ситуации истощения ресурсов у a_a . При наличии одного благонамеренного и n альтруистических агентов структура РЦС принимает вид «колеса» или «звезды».

В данной работе рассмотрены ресурсно-целевые сети с постоянными типами агентов, которые не изменяются в процессе функционирования сети. Однако несомненный практический интерес представляют ситуации смены типов агентов при изменении объема располагаемых ресурсов или целевых установок, например,

превращение благонамеренного агента a_b в эгоистичного агента a_e . Здесь полезным может быть такой показатель, как текущее влияние агента в сети, задаваемый, например, в виде $I_i(t) = \langle res_i(t), wS_i(t), n(t) \rangle$, где $res_i(t)$ – величина ресурса i -го агента, $wS_i(t)$ – суммарное число проводимостей, связанных с i -м агентом, $n(t)$ – общее количество агентов, с которыми взаимодействует данный агент.

Заключение

В статье предложены варианты наглядного представления процессов взаимодействия агентов, формирования и развития многоагентных систем с помощью аппарата ресурсно-целевых сетей. Показано, что этот аппарат может успешно применяться для различных приложений МАС, включая виртуальные организации и социальные сети.

СПИСОК ЛИТЕРАТУРЫ

- [**Волков, 1986**] Волков А.М. Основы структурно-функционального анализа операторской деятельности. – М.: Изд-во МАИ, 1986.
- [**Дюндюков, 2010**] Дюндюков В.С. Моделирование взаимодействия интеллектуальных агентов: применение ресурсных графов / В.С.Дюндюков// Труды Международного конгресса по интеллектуальным системам и информационным технологиям (AIS-IT'2010, Дивноморское, 2–9 сентября 2010 г.). – М.: Физматлит, 2010. – Т.1. – С. 204–210.
- [**Дюндюков и др., 2011**] Дюндюков В.С. Ресурсно-целевые сети: использование в многоагентных системах / В.С.Дюндюков, В.Б. Тарасов// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов VI Международной научно-практической конференции (Коломна, 16–19 мая 2011 г.). – М.: Физматлит, 2011. – Т.1. – С. 483–495.
- [**Дюндюков и др., 2013**] Дюндюков В.С. Ресурсно-целевые сети и их применение в железнодорожной области / В.С.Дюндюков, В.Б.Тарасов// Интеллектуальные системы. Вып.6 / Под ред. В.М.Курейчика. – Ростов-на-Дону: Изд-во ЮФУ, 2013. – С. 140–162.
- [**Кузнецов, 2009**] Кузнецов О.П. Однородные ресурсные сети. I. Полные графы/ О.П.Кузнецов// Автоматика и телемеханика. – 2009. – №11. – С. 136–147.
- [**Кузнецов и др., 2010**] Кузнецов О.П. Двусторонние ресурсные сети – новая потоковая модель / О.П.Кузнецов, Л.Ю.Жилякова// Доклады Академии наук. – 2010. – Т.433, №5. – С. 609–612.
- [**Тарасов, 2002**] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. – М.: Эдиториал УРСС, 2002.
- [**Форд и др., 1973**] Форд Л.Р., Фалкерсон Д.Р. Потоки в сетях: Пер. с англ. – М.: Мир, 1973.

УДК 519.837.3

**ПОВТОРЯЮЩАЯСЯ ИГРА С ЛИНЕЙНЫМИ ФУНКЦИЯМИ
ПОЛЕЗНОСТИ НА КОГНИТИВНОЙ КАРТЕ
С ОГРАНИЧЕНИЯМИ И БЕЗ ОГРАНИЧЕНИЙ
НА РЕСУРСЫ¹**

С.Г. Куливец (*skulivec@yandex.ru*)
Институт проблем управления РАН, Москва

Рассмотрены две теоретико-игровые модели взаимодействия агентов во времени на линейной когнитивной карте. Первая модель описывает взаимодействие агентов с постоянными ограничениями на их воздействия в каждый момент времени. Вторая модель рассматривает случай, когда воздействие на управляемые факторы агентом является ограниченным ресурсом на протяжении всей игры. Для обеих моделей найдено решение в виде стратегий равновесных по Нэшу. Обе модели могут быть применены для случая несогласованных представлений без существенных изменений.

Введение

Первые работы по теории повторяющихся игр принадлежат Л.С. Шэпли, Р.Д. Ауману, М. Машлеру и А. Рубинштейну [Shapley, 1953], [Aumann, Maschler, 1995], [Rubinstein, 1977]. Их работы стали основой для теории динамических игр, в которых игроки взаимодействуют друг с другом раз за разом (шаг за шагом) в течение игры, собирая информацию о выигрышах, поведении остальных игроков и параметрах игры, в каждом из таких взаимодействий. Важным подклассом повторяющихся игр являются *игры с полной информацией* (repeated games with complete information), в которых множество игроков, множество их действий и функции полезности известны всем игрокам и неизменны в течение всей игры. Первый важный результат в этой области это широко известные *Народные теоремы* [Aumann, Shapley, 1994], [Rubinstein, 1977] для бесконечно повторяющихся игр, в которых каждый игрок наблюдает и за-

¹ Работа выполнена при финансовой поддержке РФФИ (проекты № 14-01-00422-а, № 12-01-31428-мол_а, № 13-01-00934-а).

поминает все действия всех игроков на предыдущих шагах взаимодействия, а также стремится увеличить свой предельный средний выигрыш по всем шагам.

В более общем случае информация о действиях каждого агента на каждом шаге неизвестна всем. Вместо этого задана структура сигналов (*signaling structure*) в виде вероятностного отображения из множества действий игроков на каждом шаге в множество сигналов, которые описывают информацию, получаемую каждым игроком в результате очередного шага повторяющейся игры. Такие игры называются *играми с неполной информацией* (*repeated games with imperfect monitoring*), они имеют большее значение в вопросах, связанных с приложениями теории повторяющихся игр. Решение таких игр – более сложная задача, до сих пор не получившая решения в общем виде, аналогичное Народным теоремам в играх с полной информацией. Работы в этой области делятся на две группы: игры с дисконтом (*discounted games*) и без дисконта (*undiscounted games*). В первой группе используется своя концепция решений – равновесие лучшее по подыграм (*subgame perfect or sequential equilibrium*). Обобщение Народных теорем на случай равновесия лучшего по подыграм было сделано в работах [Fudenberg, Maskin, 1986]. Исследования игр во второй группе были инициированы [Lehrer, 1989, 1990, 1992] и продолжены [Hillas, Liu, 1996] и [Tomala, 1998, 1999].

Теоретико-игровая модель конфликтного взаимодействия агентов в дискретном времени линейной когнитивной карты, формализующей слабоструктурированную ситуацию, в которой разворачивается конфликт, была впервые предложена в [Новиков, 2008]. Позже была рассмотрена игровая ситуация, в которой представления агентов о ситуации были заданы различными когнитивными картами, было найдено решение и была показана возможность увеличения выигрыша за счет информационного управления. С примером полезного использования данной модели можно познакомиться в [Kulivets, 2013]. В настоящей работе используется эта модель в повторяющейся игре с полной информацией, при условии, что полезность каждого агента зависит от совокупных действий линейно. Рассмотрены две версии игры: с ограниченным числом возможных действий в течение T периодов для каждого агента и без таких ограничений.

1. Описание модели

Рассмотрим модифицированную игру $G = (N, \{S_i\}_{i \in N}, \{f_i\}_{i \in N}, C, T)$, здесь

C – линейная когнитивная карта, заданная графом факторов и причинно-следственных связей между ними (см. рис. 1) и линейным импульсным правилом, изменяющим значения факторов в дискретном времени: $X(t) = X(t-1) + A P(t-1)$, где $X(t)$ – вектор значений факторов в момент t , A – матрица смежности графа, $P(t-1)$ – вектор приращений значений факторов в момент времени $t-1$) [Робертс, 1986], [Максимов, 2008]:

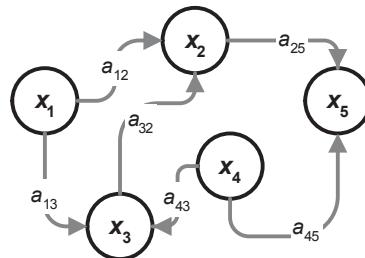


Рис. 1. Пример графа когнитивной карты

N – множество агентов (в задаче рассматривается ситуация двух агентов),

$\{S_i\}_{i \in N}$ – множество стратегий агентов; множество стратегий агента i задается как декартово произведение ограничений по воздействиям на множество управляемых факторов агента M_i (подмножество множества факторов когнитивной карты M): $S_i = \times_{j \in M_i} [-1, 1]$;

$\{f_i^t\}_{i \in N}$ – множество функций полезности агентов в момент t (множество целевых функций): $f_i^t: \times_i S_i \rightarrow R$; рассматривается линейная по приращениям целевых факторов в момент t относительно начальных значений функция:

$f_i^t = \gamma_{ij}(x_j(t) - x_j(0))$, где γ_{ij} – весовой коэффициент, задающий желательное направление изменения целевого фактора x_j и определяющий важность указанного критерия среди других, $\sum_j |\gamma_{ij}| = 1$.

T – момент времени окончания игры (общее знание).

В каждый момент времени динамика значений факторов задается совокупностью уравнений:

$$X(t) = X(t-1) + \sum_{j=0}^{t-1} A^j U(t-1-j) \quad (1)$$

$X(0) = 0; U(0), U(1), \dots, U(T-1)$ – векторы управлений агентами в различные моменты времени (см. рис. 2).

Итак, $X(t)$ – вектор значений факторов в момент времени t , A – матрица смежности орграфа когнитивной карты; $U(t)$ – вектор управляющих воздействий, оказываемых агентами в момент времени t .

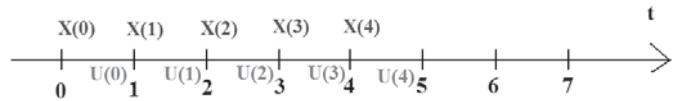


Рис. 2. Последовательность векторов значений факторов в разные моменты времени $X(t)$ и векторов управляющих воздействий на них $U(t)$

Решение игры рассматривается на промежутке $0..T$, при этом рассчитывается суммарный выигрыш для каждого из игроков как сумма выигрышей f_i^t за каждый из периодов $0..T$:

$$F_i = \sum_{t \in \{0, \dots, T\}} f_i^t \rightarrow \max_{U(0), U(1), \dots, U(T-1)} \quad (2)$$

Решение игры ищется как равновесие Нэша в чистых стратегиях.

Ввиду того, что решение игры на каждом шаге можно представить как равновесие в доминантных стратегиях (при этом для каждого управляемого фактора оптимально выбрать одно из трех значений на каждом шаге: $\{-1, 0, 1\}$), то решение повторяющейся игры, которая состоит из последовательности базовых игр, мы ищем как последовательность воздействий на каждый управляемый фактор x_j в каждый момент времени $0..T$, которая приводит функцию (2) к максимальному значению. Запишем уравнение зависимости значений факторов в момент времени t от начального состояния и управляющих воздействий:

$$X(t) = X(0) + \sum_{j=0}^{t-1} W_j \cdot U(t-j-1). \quad (3)$$

Здесь $W_j = E + A + A^2 + \dots + A^j$, далее $W_j = \{w_{ik}^{(j)}\}$.

E – единичная матрица такого же порядка, как матрица A .

2. Поиск решения

Из уравнения (3) можно получить оценку прироста значений факторов в когнитивной карте к моменту времени t :

$$X(t) - X(0) = \sum_{j=0}^{t-1} W_j \cdot U(t-j-1).$$

Зафиксируем агента и распишем величину приращения его целевого фактора x_i к моменту времени t :

$$\begin{aligned} x_i(t) - x_i(0) &= \sum_{j=0}^{t-1} \sum_{k=1}^n w_{ki}^{(j)} \cdot u_k(t-j-1) = w_{1i}^{(0)} \cdot u_1(t-1) + \dots + w_{ni}^{(0)} \cdot u_n(t-1) + \\ &+ w_{1i}^{(1)} \cdot u_1(t-2) + \dots + w_{ni}^{(1)} \cdot u_n(t-2) + \dots + w_{1i}^{(t-1)} \cdot u_1(0) + \\ &+ \dots + w_{ni}^{(t-1)} \cdot u_n(0). \end{aligned}$$

Приращение того же фактора к моменту времени $t + 1$ равно:

$$\begin{aligned} x_i(t+1) - x_i(0) &= \sum_{j=0}^t \sum_{k=1}^n w_{ki}^{(j)} \cdot u_k(t-j) = w_{1i}^{(0)} \cdot u_1(t) + \dots + w_{ni}^{(0)} \cdot u_n(t) + \\ &+ w_{1i}^{(1)} \cdot u_1(t-1) + \dots + w_{ni}^{(1)} \cdot u_n(t-1) + \\ &+ \dots + w_{1i}^{(t)} \cdot u_1(0) + \dots + w_{ni}^{(t)} \cdot u_n(0). \end{aligned}$$

Из двух вышеприведенных выражений получаем, что агрегированное влияние управляющего фактора x_k на сумму приращений целевого фактора x_i :

$$\sum_{t=1}^T (x_i(t) - x_i(0))$$

в течение повторяющейся игры вычисляется как

$$\begin{aligned} (w_{ki}^{(T-1)} + \dots + w_{ki}^{(0)}) \mathbf{u}_k(\mathbf{0}) + (w_{ki}^{(T-2)} + \dots + w_{ki}^{(0)}) \mathbf{u}_k(\mathbf{1}) + \dots + & (4) \\ + \dots + (w_{ki}^{(1)} + w_{ki}^{(0)}) \mathbf{u}_k(\mathbf{T-2}) + w_{ki}^{(0)} \mathbf{u}_k(\mathbf{T-1}) = \\ W_{ki}^{0,T-1} \mathbf{u}_k(\mathbf{0}) + W_{ki}^{0,T-2} \mathbf{u}_k(\mathbf{1}) + \dots + W_{ki}^{0,1} \mathbf{u}_k(\mathbf{T-2}) + W_{ki}^{0,0} \mathbf{u}_k(\mathbf{T-1}). \end{aligned}$$

Из вида (4) вытекает справедливость следующего утверждения:

Утверждение 1. Оптимальное значение (равновесное по Нэшу) управляющего действия для фактора x_k в момент времени t определяется согласно правилу (5).

Если $\gamma_{si} > 0$, то (5)

$$u_k(t) = \begin{cases} -1, & \text{если } W_{ki}^{0,T-t-1} < 0 \\ 1, & \text{если } W_{ki}^{0,T-t-1} > 0 \end{cases}, \quad k \in M_s, t = 0, \dots, T-1, s \in N.$$

иначе

$$u_k(t) = \begin{cases} -1, & \text{если } W_{ki}^{0,T-t-1} > 0 \\ 1, & \text{если } W_{ki}^{0,T-t-1} < 0 \end{cases}, \quad k \in M_s, t = 0, \dots, T-1, s \in N.$$

Здесь i – номер целевого фактора агента s , M_s – множество управляемых факторов агента s .

Используя правило (5) последовательно для каждого агента и рассматривая все моменты времени, мы можем рассчитать оптимальные стратегии для всех агентов. Полученная последовательность управлений $[U(0), U(1), \dots, U(T-1)]$ в силу линейности функций полезности образует частный случай равновесия по Нэшу в повторяющейся игре.

Рассмотрим случай, когда **число действий на управляемые факторы агентом является ограниченным ресурсом** на протяжении всей игры и ограничено константой R_i для каждого игрока i . Тогда для каждого агента i должно выполняться ограничение:

$$\sum_{t=0}^{T-1} \sum_{k \in M_i} |u_k(t)| \leq R_i$$

Ввиду того, что функция каждого агента (2) сепарабельна по стратегиям этого агента и стратегиям других агентов, в силу линейности, мы можем решать задачу (6) подбора оптимальной стратегии агентом отдельно для каждого агента.

$$\sum_{t \in \{0, \dots, T\}} f_i^t(U(0), \dots, U(t-1)) \rightarrow \max_{U(0), U(1), \dots, U(T-1)} \quad (6)$$

$$\sum_{t=0}^{T-1} \sum_{k \in M_i} |u_k(t)| \leq R_i, \quad R_i > 0$$

Задача (6) может быть решена рекуррентным *методом динамического программирования*. Для того, чтобы сформулировать задачу в терминах динамического программирования, определим, какие элементы нашей модели будут соотноситься с базовыми понятиями этого метода [Таха, 2005].

1. Определение *этапов*. Этапом решения в нашей задаче будет временной тakt. Каждому временному такту $\{0, 1, \dots, T-1\}$ будет соответствовать свой *этап* решения.

2. Определение на каждом этапе *вариантов решения* (альтернатив). Варианты решения на этапе t описываются векторами управляющих воздействий на факторы из M_i для i -го игрока: $(u_1(t), \dots, u_{|M_i|}(t))$, где каждая из компонент вектора $u_k(t)$ равна либо 0, либо одно из значений -1 или 1 (определенное согласно правилу (5)). Обозначим множество всех вариантов решения как $3^{|M_i|}$. Итого, максимально возможное число вариантов решения на каждом этапе ограничено числом $3^{|M_i|}$. Соответствующая полезность агента от использованного варианта решения $(u_1(t), \dots, u_{|M_i|}(t))$ равна $\gamma_{is} \cdot \sum_{k \in M_i} W_{ks}^{0, T-1-t} u_k(t)$.

3. Определение *состояний* на каждом этапе. Состояние S_t на этапе t выражает суммарный ресурс, израсходованный за счет воздействий, оказанных в предыдущие моменты времени $\{0, 1, \dots, t-1\}$. То есть

$$S_t = \sum_{t=0}^{t-1} \sum_{k \in M_i} |u_k(t)|.$$

Пусть $g_t(S_t)$ – максимальная суммарная полезность от этапов $0, \dots, t-1$ при заданном состоянии S_t . Рекуррентное уравнение для задачи динамического программирования определяется следующим образом:

$$g_t(S_t) = \max_{u(t) \in 3^{|M_i|}} \left(f'_i(u(t)) + g_{t-1}(S_t - \sum_{k \in M_i} u_k(t)) \right), \quad t = 0, \dots, T-1, \quad (7)$$

$$g_0(S_0) = 0.$$

Таким образом, поиск оптимальной стратегии игрока в задаче с ограниченным ресурсом полностью сводится к задаче динамического программирования. И справедливо утверждение.

Утверждение 2. Оптимальная стратегия (равновесная по Нэшу) для игрока i в игре с ограниченным ресурсом R_i определяется из решения рекуррентного уравнения (7).

Применение полученных двух моделей возможно для случая несогласованных представлений агентов (нескольких когнитивных карт). Для этого во всех выражениях (1–7) для каждого агента необходимо рассматривать матрицу смежности его (а не общей) когнитивной карты.

Заключение

В работе рассмотрена модель совместного взаимодействия агентов в течение T периодов, полезность каждого агента зависит линейно от совокупных действий. Найдено решение для двух версий игры: с ограниченным числом возможных воздействий для каждого агента и без ограничений.

Полученные результаты могут быть применены в случае несогласованных представлений агентов (нескольких когнитивных карт). Выбор момента времени T может быть осуществлен исходя из оценок устойчивости решения игры к ошибкам экспертов в оценивании параметров модели [Kulivets, 2013]. Развиваемая в данной работе модель может быть использована при построении информационных систем поддержки принятия решений в конфликтных ситуациях с большим числом качественных (неколичественных) параметров, где традиционно используются когнитивные карты.

СПИСОК ЛИТЕРАТУРЫ

- [Максимов, 2008] Максимов В.И. Структурно-целевой анализ развития социально-экономических ситуаций // Проблемы управления. 2005. №3.
- [Новиков, 2008] Новиков Д.А. «Когнитивные игры»: линейная импульсная модель // Проблемы управления. 2008. №3.
- [Робертс, 1986] Робертс Ф. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. – М.: Наука, 1986.
- [Taxa, 2005] Taxa X.A. Введение в исследование операций – М.: Издательство дом «Вильямс», 2005.
- [Kulivets, 2013] Kulivets S.G. Game-Theoretic Model on a Cognitive Map and its Tolerance to Errors in Input Data to Analyze a Conflict of Interests Between Russia and Norway in Barents Sea // Contributions to game theory and management, vol.VI. Collected papers presented on the Sixth International Conference Game Theory and Management/Editors Leon A. Petrosyan, Nikolay A. Zenkevich.–SPb.: Graduate School of Management SPbU, 2013.
- [Shapley, 1953] Shapley L.S. Stochastic games. Proc Natl Acad Sci USA 39.
- [Aumann, Maschler, 1995] Aumann R.J., Maschler M. Repeated games with incomplete information, with the collaboration of R. Stearns. MIT Press, Cambridge.

- [Aumann, Shapley, 1994]** Aumann R.J., Shapley L.S. Long-term competition—a game theoretic analysis. In: Megiddo N (ed) Essays on game theory. Springer, New York.
- [Rubinstein, 1977]** Rubinstein A. Equilibrium in supergames. Center for Research in Mathematical Economics and Game Theory, Research Memorandum 25.
- [Fudenberg, Maskin, 1986]** Fudenberg D., Maskin E. The Folk theorem in repeated games with discounting or with incomplete information. *Econometrica* 54.
- [Lehrer, 1989]** Lehrer E. Lower equilibrium payoffs in two-player repeated Games with non-observable actions. *Int J Game Theory* 18.
- [Lehrer, 1990]** Lehrer E. Nash equilibria of n-player repeated games with semi-standard information. *Int J Game Theory* 19.
- [Lehrer, 1992]** Lehrer E. Correlated equilibria in two-player repeated games with non-observable actions. *Math Oper Res* 17.
- [Lehrer, 1992]** Lehrer E. On the equilibrium payoffs set of two-player repeated games with imperfect monitoring. *Int J Game Theory* 20.
- [Lehrer, 1992]** Lehrer E. Two-player repeated games with nonobservable actions and observable payoffs. *Math Oper Res* 17.
- [Hillas, Liu, 1996]** Hillas J., Liu M. Repeated games with partial monitoring: the stochastic signaling case. *Game Theory and Information* 9605001, EconWPA.
- [Tomala, 1998]** Tomala T. Pure equilibria of repeated games with public observation. *Int J Game Theory* 27(1).
- [Tomala, 1999]** Tomala T. Nash equilibria of repeated games with observable payoff vector. *Games Econ Behav* 28.

УДК 007.5:681.518.2

ФОРМАЛЬНАЯ МОДЕЛЬ ОПЕРЕЖАЮЩЕГО ИТЕРАТИВНОГО ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ РЕАЛЬНОГО ВРЕМЕНИ

М.Г. Пантелейев (MPanteleyev@gmail.com)
СПбГЭТУ «ЛЭТИ», Санкт-Петербург

Рассматривается проблема планирования действий интеллектуальных агентов в динамических многоагентных мирах. Предложены формальные модели представления и обработки знаний, необходимые для реализации подхода опережающего итеративного планирования. Подход и модели иллюстрируются на примере ситуаций из виртуального футбола.

Введение

Планирование действий в многоагентных мирах – одна из ключевых проблем создания автономных интеллектуальных агентов (ИА). Эта задача должна решаться в реальном времени (РВ) в непрерывно меняющейся среде при отсутствии у агента полной и точной информации о ее состоянии. В известных подходах [Kovarsky, 2002; Ontanon et al., 2007; Silva et al., 2007; Kapadia et al., 2013] модель планирования рассматривается вне связи со структурой ментального цикла делиберативного агента и без учета изменений мира в процессе выполнения текущего действия.

Поскольку наиболее остро проблема планирования в РВ проявляется в сценариях командного противодействия, предлагаемая модель иллюстрируется на примере ситуаций из виртуального футбола (ВФ). Платформа Robosoccer поддерживает моделирование важнейших особенностей многоагентных систем в сценариях командного противодействия (ограниченность восприятия коммуникационных каналов и др.) и широко используется в качестве полигона для отработки моделей командных действий.

В [Пантелейев, 2004; Пантелейев, 2012; Пантелейев и др., 2013] предложен подход к построению ИА РВ на основе опережающего итеративного планирования (ОИП), являющийся конструктивным уточнением концепции «ограниченной рациональности» в рамках

делиберативной архитектуры ИА. Настоящая статья является дальнейшим развитием данного подхода в направлении конкретизации формальных моделей представления и обработки знаний, используемых в ОИП.

1. Принципы построения ИА РВ на основе ОИП

В основу подхода к построению ИА РВ на основе ОИП положены следующие основные принципы [Пантелеев, 2012]:

1. ИА РВ, действующие в открытых динамических неопределенных (ОДН) мирах должны *динамически* в процессе функционирования *определять временные ограничения* на принятие решений и *максимально эффективно использовать имеющееся время*.

2. Временные ограничения на принятие (обдумывание) решений могут быть обусловлены: (1) временем ожидаемого наступления критических событий относительно их динамически выявляемых предусловий, либо (2) длительностью реализации агентом текущих действий (планов).

3. Максимально эффективное использование времени обдумывания решения предполагает: (1) реализацию ОИП и (2) максимально эффективное использование времени *в каждом ментальном цикле* (МЦ).

4. Максимально эффективное использование времени в каждом ментальном цикле предполагает: (1) динамическое распределение времени между ментальными функциями ИА и (2) определение правил фокусировки внимания для каждой функции.

Основная идея ОИП состоит в обдумывании следующего действия на всем протяжении выполнения текущего плана. Поскольку длительность МЦ в общем случае значительно меньше длительности выполняемых агентом физических действий, в каждом МЦ требуется выполнение функций прогнозирования, планирования и оценки анализируемых вариантов действий.

2. Модель опережающего итеративного планирования

Итеративное планирование от будущего состояния предполагает выполнение *в каждом МЦ* (такте) прогнозирования ситуации на момент t^* окончания текущего действия. Относительно этого состояния в каждом МЦ выполняются ментальные действия, от-

носящиеся собственно к планированию. Используемые при этом модели в соответствии с фазами планирования можно разбить на три группы:

1. Генерация базового множества возможных вариантов действий.
2. Конкретизация и оценка полезности вариантов действий.
3. Окончательная конкретизация действий и выбор лучшего для выполнения.

Общая длительность процесса обдумывания определяется длительностью текущего действия и в общем случае может включать любое число МЦ (тактов). При этом генерация базового множества действий всегда выполняется в первом такте интервала обдумывания, а окончательная конкретизация и выбор действия – в последнем. Конкретизация и оценка действий выполняются на всем интервале обдумывания решения. По мере выполнения текущего действия агент приближается к моменту его завершения и получает в каждом такте (МЦ) новую информацию о состоянии мира. Поскольку интервал прогнозирования ситуации на момент t^* окончания текущего действия с каждым тактом сокращается, точность прогнозирования ситуации на момент t^* повышается.

На рис. 1 представлен пример ситуации в ВФ, когда владеющий мячом игрок A_0 пробросил его себе на ход в момент t_1 , с расчетом догнать мяч за четыре такта (в данном случае $t^* = t_5$). Интервал проброса был определен из условия достижения мяча раньше других игроков.

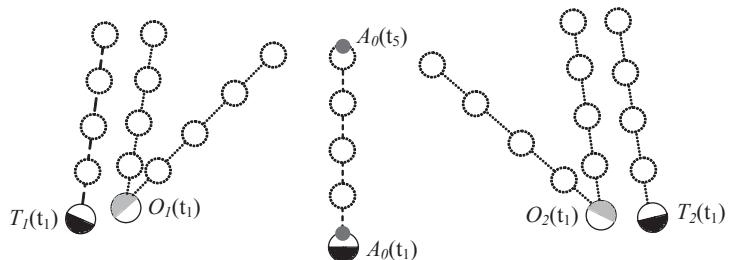


Рис. 1. Прогнозируемые перемещения игроков при ведении мяча

На рисунке показаны положения игроков в начальный момент t_1 при следующих обозначениях: T_1 и T_2 – партнеры игрока по команде, O_1 и O_2 – игроки противоположной команды (оппоненты).

Пунктирными линиями показаны возможные направления движения игроков и их промежуточные положения в процессе движения. В течение четырех тактов бега к мячу агент A_0 обдумывает свое следующее действие, прогнозируя ситуацию на момент времени $t^* = t_5$. Рассмотрим формальные модели представления и обработки знаний, используемых агентом во всех фазах ОИП.

Модель прогнозирования ситуации на момент окончания текущего действия. Прогнозирование ситуации S^* на момент t^* окончания текущего действия сводится к прогнозированию состояния отдельных агентов, как автономно действующих в соответствии со своими намерениями сущностей. Состояние $S(T_i(t^*))$ каждого игрока T_i (и аналогично O_j) на момент времени t^* прогнозируется в каждом такте на основе его текущего состояния $S(T_i(t_c))$ и текущего намерения $I(T_i(t_c))$:

$$S(T_i(t^*)) = f_{\text{prg}}(S(T_i(t_c)), I(T_i(t_c))),$$

где f_{prg} – функция прогнозирования.

Гипотеза о намерении $I(T_i(t_c))$ агента в текущий момент t_c формируется по его наблюдаемому поведению, т. е. по текущему состоянию $S(T_i(t_c))$ и предыстории $S^\wedge(T_i(t_{[c, c-n]}))$, заданной глубины:

$$I(T_i(t_c)) = f_{\text{int}}(S(T_i(t_c)), S^\wedge(T_i(t_{[c, c-n]}))),$$

где f_{int} – функция определения гипотезы о намерении агента, реализующая модель распознавания намерений. Например, в ВФ такая модель в простейшем случае может предполагать продолжение движения в текущем направлении с текущей скоростью.

Модель генерации базового множества действий. Очередная фаза планирования начинается с генерации базового множества вариантов своих возможных действий, в рассматриваемой ситуации S^* . Поскольку на этапе генерации действия представлены в обобщенном виде, обозначим это множество $\{\text{Act}_o^{j<0>}\}$, где нижний индекс в данном случае указывает на принадлежность к агенту (0 соответствует планирующему игроку); j – номер варианта действия; $<0>$ – уровень абстракции представления действия (0 соответствует высшему уровню абстракции).

Генерация базового множества вариантов действий выполняется с использованием продукционной модели, заданной множеством правил:

$$S^* \rightarrow \{Act_o^{j<0>}\},$$

где S^* – ситуация, прогнозируемая на момент окончания текущего действия (в последнем МЦ действия – это текущая ситуация).

Ситуация S^* описывается конъюнкцией атрибутов ситуации $P_1 \& \dots \& P_N$, существенных для определения множества $\{Act_o^{j<0>}\}$. Значение атрибутов ситуации P_j вычисляется подсистемой оценки обстановки на основе прогнозируемых состояний или первичных данных, поступающих от подсистемы восприятия агента. Например, в ВФ можно выделить следующие атрибуты ситуации: признак владеющей мячом команды, признак собственного владения мячом, позиция игрока в тактической схеме и т.п.

Представление действия *на верхнем уровне абстракции* характеризуется *типовом действия* и, возможно, некоторыми ключевыми атрибутами, существенными для выделения базовых вариантов поведения в некоторой ситуации. Так, в ВФ обобщенное действие *на верхнем уровне абстракции* можно охарактеризовать двумя атрибутами: *типовом действия* (для игрока, владеющего мячом, это: ведение, пас, удар по воротам) и *обобщенным (секторным) направлением действия* (налево, направо и т.п.). Таким образом, базовое множество вариантов действий для игрока владеющего мячом игрока может включать: «ведение мяча вперед», «пас налево» и т. п., без детализации параметров действия. Конкретные параметры должны уточняться по мере приближения момента выбора действия.

Особенностью данной ментальной операции является фиксированное, априори предсказуемое (в худшем случае) и относительно малое время ее выполнения, что упрощает планирование связанный с ней ментальной деятельности.

Модель конкретизации действий. По мере получения новой информации в каждом МЦ и приближения к моменту t^* принятия решения ситуация S^* уточняется и выполняется *конкретизация обобщенных действий*.

Конкретизация действий заключается в определении значений атрибутов более низких уровней абстракции (имеющих больший индекс). Исходное множество действий генерируется на основе общей информации о ситуации. Уточнение действий требует получения дополнительной информации в соответствии с определенной

стратегией управления восприятием на интервале обдумывания действий.

Рассмотрим базовый вариант действия «Пас влево», заданный типом «Пас» и обобщенным (секторным) направлением «влево». На следующем шаге конкретизации для данного действия должен быть уточнен партнер – адресат паса. Поскольку в общем случае в некотором секторном направлении от игрока, владеющего мячом, могут находиться более одного партнера, то возможно несколько адресатов паса в одном направлении. В этом случае базовый вариант $Act_0^{<0>}$ с данным секторальным направлением порождает два возможных действия: $Act_0^{j<1>}$ и $Act_0^{k<1>}$ более низкого уровня абстракции, с разными адресатами. При этом исходный вариант $Act_0^{i<0>}$ исключается из множества. Таким образом, при конкретизации некоторого варианта действий $Act_0^{i<0>}$ в множество рассматриваемых вариантов добавляются его потомки, при этом сам он удаляется. Если на данном этапе планирования предпочтение между ними неочевидно, в списке возможных действий агента должны сохраняться оба варианта. Таким образом, одно действие более высокого уровня абстракции в процессе конкретизации *может порождать несколько более конкретных вариантов*. Для каждого из них, исходя из имеющейся на данный момент информации, вычисляется оценка полезности.

В контексте итеративного уточнения действий их атрибуты могут быть сгруппированы по уровням конкретизации. С учетом такой группировки действие может быть представлено фреймом следующей структуры:

$$AF_i = \langle AT_i, \{Atr_j^{<0>}\}, \{Atr_l^{<1>}\}, \dots, \{Atr_m^{<k>}\} \rangle,$$

где AF_i – фрейм действия i -го типа; AT_i – тип действия; $\{Atr_l^{<k>}\}$ – множество атрибутов действия, отнесенных к k -му уровню абстракции.

Условия перехода к очередному шагу конкретизации определяются достаточностью информации и формулируются в виде логических правил. Такие правила могут описывать существование некоторых сущностей, а также возможность достаточно точно спрогнозировать состояние, чтобы гарантировать условия успешной реализации действия. Так, в случае конкретизации варианта паса в некотором секторальном направлении в качестве условия

порождения конкретизированного варианта выступает условие существования игрока-партнера в данном направлении.

Окончательная конкретизация действия выполняется в момент завершения текущего действия, т. е. в последнем МЦ текущего действия. При этом вычисляются значения атрибутов нижнего уровня абстракции. Например, в ВФ для паса должны быть вычислены координаты точки приема мяча и длительность (число тактов) его движения в эту точку. Спланированное действие передается для реализации исполнительной подсистеме агента. В случае паса исполнительная подсистема рассчитывает направление и силу удара по мячу с учетом текущего направления тела игрока и других факторов.

Модель оценки полезности. Для каждого из рассматриваемых вариантов действий из множества $\{Act_0^{i<\text{тек}>}\}$ в каждом МЦ необходимо оценивать полезность $U(Act_0^i) = U_i$. Текущие значения полезности позволяют агенту ранжировать анализируемые варианты действий и фокусировать внимание (ментальные ресурсы) на вариантах, которые в данный момент представляются наиболее перспективными.

При генерации исходного множества $\{Act_0^{i<0>}\}$ каждый вариант получает *априорную оценку полезности*. Так, в ВФ более предпочтительными являются варианты, позволяющие переместиться ближе к воротам противника.

Полезность некоторого действия Act_0^i естественно оценивать полезностью состояния, достигаемого в результате выполнения данного действия. В многоагентной среде состояние мира непрерывно меняется в результате действий множества агентов. При этом в ситуации S^* , в которой должно начаться выполнение Act_0^i , у каждого агента в общем случае существует множество возможных вариантов действий. Поэтому на момент завершения действия Act_0^i мир может оказаться в множестве $\{S_n^{**}\}$ состояний (множество возможных миров).

Множество агентов, действия которых существенны в рассматриваемой ситуации S^* , назовем множеством *актуальных акторов* и обозначим AA . Для сценариев командного противодействия можно записать:

$$AA = \{T_k\} \cup \{O_m\},$$

где $\{T_k\}$ и $\{O_m\}$ – соответственно множества актуальных партнеров и актуальных оппонентов.

Обобщенный алгоритм оценки полезности $U(Act_0^i)$ некоторого действия Act_0^i включает следующие шаги:

1. Определить множества $\{T_k\}$ и $\{O_m\}$.
2. Для каждого агента из множеств $\{T_k\}$ и $\{O_m\}$ определить множества их возможных действий в ситуации S^* при выполнении планирующим агентом действия Act_0^i ;
3. Для каждой комбинации возможных действий агентов множества AA спрогнозировать ситуацию S_n^{**} на момент завершения Act_0^i .
4. Оценить полезность $U(S_n^{**})$ каждой достижимой ситуации (возможного мира).
5. Определить полезность $U(Act_0^i)$ действия на основе полезности состояний мира, которые могут быть достижимы при его выполнении.

Рассмотрим знания, необходимые планирующему агенту для реализации всех шагов данного алгоритма. Множества $\{T_k\}$ и $\{O_m\}$ формируются на основе отношений (например, пространственных) *партнеров и оппонентов* с планирующим агентом, определяющих их возможность влиять на развитие ситуации в пределах рассматриваемого горизонта планирования. Например, в ВФ эти множества могут определяться на основе близости игроков к мячу.

Реализация шага 2 предполагает наличие у планирующего агента знаний, ставящих в соответствие каждому *актуальному актору* множество его возможных действий в ситуации S^* при условии выполнения планирующим агентом действия Act_i . Формально соответствующие знания можно представить множеством производных правил:

$$(S_0, Act_0^i): T_k \rightarrow \{Act_k^i\}, k = 1, K; (S_0, Act_0^i): O_m \rightarrow \{Act_m^n\}, m = 1, M.$$

Реализация шага 3 предполагает прогнозирование состояний мира $\{S_n^{**}\}$, достижимых из ситуации S^* при реализации *актуальными акторами* различных комбинаций действий. Переход из ситуации S^* в множество достижимых состояний $\{S_n^{**}\}$ удобно представить графом двудольное дерево, представляющим собой модификацию дерева игры. Корневая вершина такого дерева со-

ответствует состоянию мира S^* , исходящим дугам – возможные в этой ситуации действия планирующего агента Act_0^i . Второй ярус дерева содержит вершины-разветвители. Исходящие из них дуги помечены различными комбинациями действий актуальных акторов в ситуации S^* при выполнении планирующим агентом действия Act_0^i , соответствующего входящей в вершину-разветвитель дуге. Терминальные вершины соответствуют состояниям, прогнозируемым на момент окончания планируемого действия Act_0^i .

Проиллюстрируем построение дерева для конкретной игровой ситуации в ВФ, представленной на рис. 1. Пусть планирующий игрок рассматривает в ситуации S^* вариант действия Act_0^2 – ведение мяча вперед. Множество актуальных акторов в данной ситуации включает четырех игроков: двух партнеров T_1 и T_2 и двух оппонентов O_1 и O_2 . Для упрощения примера примем, что партнеры имеют по одному целесообразному варианту действий (соответственно T_1^1 и T_2^1) – движение по своему флангу вперед, стараясь оторваться от своего опекуна. В то же время у каждого из оппонентов O_1 и O_2 имеется по два разумных варианта действий: движение на перехват ведущего мяча игрока (соответственно O_1^1 и O_2^1) и прикрытие опекаемого («своего») игрока (соответственно O_1^2 и O_2^2). Соответствующее данному случаю поддерево прогнозируемых ситуаций представлено на рис. 2.

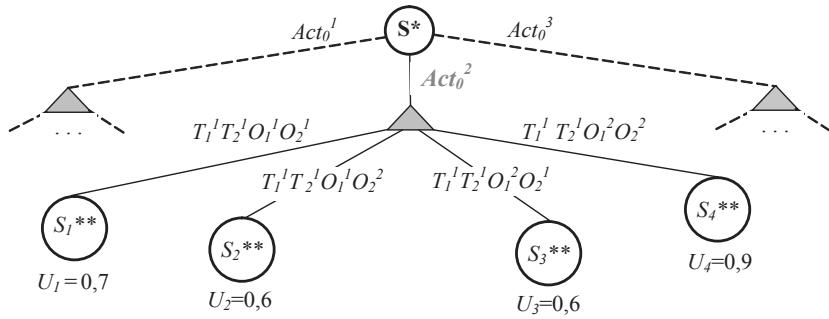


Рис. 2. Фрагмент дерева прогнозируемых ситуаций с оценками полезности

На рис. 2 дугам, исходящим из вершины-разветвителя, соответствуют следующие комбинации действий актуальных акторов: $T_1^1 T_2^1 O_1^1 O_2^1$ – оба оппонента O_1 и O_2 движутся к игроку, ведущему

мяч; $T_1' T_2' O_1' O_2^2 - O_1$ движется к игроку, ведущему мяч, O_2 прикрывает «своего» игрока; $T_1' T_2' O_1^2 O_2'$ – O_1 прикрывает «своего» игрока, O_2 движется к игроку, ведущему мяч; $T_1' T_2' O_1^2 O_2^2$ – каждый из оппонентов прикрывает «своего» игрока. Прогнозирование движения партнеров целесообразно выполнять с использованием средних значений параметров, тогда как перемещения оппонентов на интервале прогнозирования следует оценивать «в худшем для нас случае».

Оценка полезности $U(S_n^{**})$ ситуаций, соответствующих терминальным вершинам графа (*шаг 4*), в общем случае является сложной задачей, поскольку *критерии полезности* ситуативны и зависят от контекста. В ВФ для атакующей (владеющей мячом) команды такими критериями могут, например, быть: близость мяча к воротам соперника, наличие свободного пространства перед игроком, владеющим мячом, и др.

Полезность $U(Act_0^i)$ действия Act_0^i определяется на основе значений полезности всех достижимых состояний $U(S_n^{**})$, в предположении, что игроки противодействующей команды выбирают наиболее рациональные *для себя* варианты действий:

$$U(Act_0^i) = \min_k (U(S_n^{**})).$$

Вычислительная сложность алгоритма оценки полезности каждого действия определяется числом терминальных вершин в дереве возможных миров этого действия. Обозначим L_k – число вариантов действий k -го партнера из множества актуальных акторов, N_m – число вариантов действий m -го оппонента. Тогда верхняя оценка C^* числа терминальных вершин есть:

$$C^* = \Pi_k (L_k) \times \Pi_m (N_m).$$

Таким образом, число возможных миров растет экспоненциально с ростом числа актуальных акторов и их возможных действий. На практике эта оценка может быть снижена за счет учета координации действий агентов. Так, в представленном примере маловероятно, что оба оппонента начнут двигаться на перехват игрока с мячом. Учет скоординированных действий позволяет существенно сократить сложность алгоритма построения дерева возможных миров. Однако именно этот алгоритм составляет основную часть вычислительной сложности в ментальном цикле агента.

Заключение

Разработанные модели представления и обработки знаний являются конструктивным уточнением подхода ОИП и позволяют перейти к разработке конкретных методов управления ментальной деятельностью агента в каждом МЦ с использованием алгоритмов произвольного времени.

СПИСОК ЛИТЕРАТУРЫ

[Kovarsky, 2002] Kovarsky E., Buro M. Heuristic search applied to abstract combat games// Proc. 18th Canadian Conference Society on AI, 2005.

[Ontañón et al., 2007], Ontañón S., Mishra K., Sugandh N., Ram A. Case-Based Planning and Execution for Real-Time Strategy Games// Proc. Int. Conf. on Case-based Reasoning (ICCBR 2007), 2007.

[Silva et al., 2007] Silva L., Dekker A., Harland J. Planning with Time Limits in BDI Agent Programming Languages// Proc 13-th Australasian Symposium on Theory of Computing (CATS '07), Vol 65. 2007.

[Kapadia et al., 2013] Kapadia M., Beacco A., Garcia F., Reddy V., Pelechano N., Badler N. Multi-Domain Real-time Planning in Dynamic Environments// Proc. 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2013.

[Пантелейев, 2004] Пантелейев М.Г. Планирование действий интеллектуального агента в реальном времени// Труды 9-й национальной конф. по искусственному интеллекту с международным участием КИИ-2004. – М.: Физматлит, 2004. Т 2.

[Пантелейев, 2012] Концепция построения интеллектуальных агентов реального времени на основе модели опережающего итеративного планирования// Труды 13-й национальной конф. по искусственному интеллекту с международным участием КИИ-2012. – Белгород: Изд-во БГТУ, 2012. Т 3.

[Пантелейев и др., 2013] Способ определения траектории движения автономного транспортного средства в динамической среде/ Пантелейев М.Г., Лебедев С.В./ Патент РФ №2479015, 2013.

УДК 004.8: 378

МОДЕЛЬ МНОГОАГЕНТНОЙ СИСТЕМЫ РЕАЛИЗАЦИИ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОГО ПРОСТРАНСТВА¹

Ю.Ф. Тельнов

ФГБОУ «Московский государственный университет экономики,
статистики и информатики», Москва

В докладе предлагается модель многоагентной системы реализации информационно-образовательного пространства (МАС ИОП) высшего учебного заведения, которая обеспечивает интеграцию распределенных в вычислительной сети разнородных источников знаний и сервисов для реализации коллективного доступа к нему различных категорий участников научно-образовательных процессов. В докладе раскрыты особенности реализации модели многоагентной системы, связывающей интеллектуальных агентов субъектов научно-образовательной деятельности с репозиторием научно-образовательных объектов и сервисов.

Введение

Необходимость создания информационно-образовательного пространства (ИОП) в высших учебных заведениях обусловлена внедрением новых сетевых форм взаимодействия участников научно-образовательных процессов, в соответствии с которым допускается организация совместных образовательных программ несколькими вузами, научно-исследовательскими институтами и предприятиями. Информационно-образовательное пространство полезно создавать также и внутри одного учебного заведения в целях интеграции учебных дисциплин одного направления подготовки, авторских научно-педагогических коллективов, обеспечения академической мобильности. Данные положения стимулируют проведение научно-исследовательских работ по созданию методологических и технологических основ формирования и применения информационно-образовательного пространства в практике деятельности высших учебных заведений [Богословский, 2000],

¹ Доклад подготовлен при поддержке РФФИ, грант 13-07-00917.

и в частности работ по созданию ИОП на основе онтологического и многоагентного подходов [Тельнов и др., 2012а], [Трембач, 2012].

Целью создания многоагентной системы реализации информационно-образовательного пространства (МАС ИОП) является разработка инструментария, обеспечивающего распределенное хранение и совместное использование разнородного контента знаний всеми участниками научно-образовательных процессов для проведения научных исследований, методических разработок и обучения.

Для реализации этой цели предлагается разработка методов и средств создания ИОП на основе применения:

- семантических методов интеграции разнородных источников знаний, описания образовательных и научных сервисов на основе онтологического подхода;
- методов представления объектов знаний в ИОП на основе сервисно-ориентированной технологии («знания как сервисы»);
- методов композиции сервисов и объектов знаний в соответствии с моделями компетенций и потребностей образовательных и исследовательских программ;
- методов многоагентной технологии применения сервисов в ИОП для решения образовательных и исследовательских задач;
- методов коллективного решения образовательных и исследовательских задач в ИОП на основе многоагентной технологии.

Предлагаемая в работе модель многоагентной системы интегрирует применение перечисленных методов в рамках ИОП, ее применение позволяет осуществлять интенсивное и динамичное наращивание научно-образовательного контента, существенно повысить качество образования и научных исследований и создаст условия для развития виртуальной академической мобильности преподавателей, научных работников и обучающихся.

1. Концепция многоагентной системы реализации ИОП

Создание информационно-образовательного пространства предполагает активное взаимодействие различных категорий субъектов в процессе совместной научно-образовательной деятельности,

что позволяет рассматривать ИОП как форму виртуального предприятия и соответственно применять методы инжиниринга предприятий [Тарасов, 2002]. Согласно [Yu, Mylopoulos, 1994] предприятие представляет собой совокупность социально связанных человеческих существ, то есть это система социальных акторов (social actors), обладающих интенциями, мотивацией, желаниями, предположениями и стратегией по отношению друг к другу.

В виртуальных предприятиях, функционирующих в распределенной вычислительной среде, в качестве интеллектуальных ассистентов людей могут выступать программные агенты, которые имитируют поведение людей и, обладая более высокими возможностями по поиску, отбору и обработке больших массивов информации, способны в значительной степени ускорить процессы решения разнообразных задач научно-образовательной деятельности.

Рассмотрим модели социального поведения людей на предприятии, которые могут быть перенесены на взаимодействие программных агентов. В теории процессного управления предприятиями, выдвинутой М. Хаммером и Дж. Чампи, принято выделять владельцев процессов и владельцев ресурсов [Хаммер, Чампи, 1997]. Владельцы ресурсов предоставляют или применяют ресурсы для выполнения функций по их преобразованию в полезные результаты. Владельцы процессов выполняют координацию осуществления последовательности функций владельцами ресурсов (цепочки действий) для получения конечного результата. В соответствии с теорией онтологического инжиниринга Диетца [Dietz, 2006] агенты (в терминологии Диетца акторы) выполняют производственные и координационные акты (см. рис. 1).

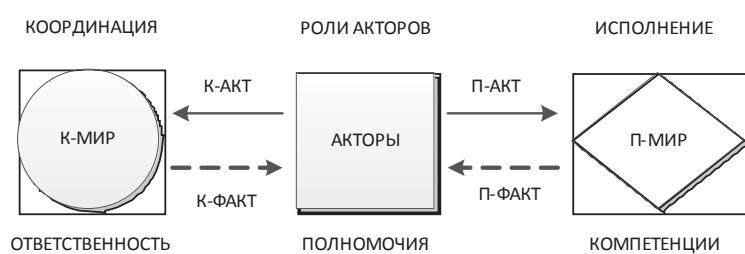


Рис. 1. Деятельность агентов (акторов) в процессах предприятий

Разделение функций может быть перенесено и на специализацию агентов в МАС. Агент-координатор является интеллектуальным агентом. Он выполняет следующие функции управления: в соответствии с заказом клиента формирует последовательность выполнения активностей, подбирает исполнителей на роли, планирует выполнение процесса по срокам, отслеживает ход выполнения процесса, координирует исполнителей в случае отклонений. Для организации работы агента-координатора требуется наличие базы знаний принятия решений, которая содержит правила разрешения конфликтов агентов-исполнителей в соответствии с системным критерием эффективности и с этой точки зрения отражает ответственность агента-координатора за исполнение всего процесса.

Агенты-исполнители в нотации МАС являются реактивными агентами и исполняют конкретные роли в исполняемом процессе. Для каждой роли характерен свой уровень компетенции владения активностями (действиями) и их компонентами: сервисами и информационными ресурсами. В соответствии с [Dietz, 2006] роль (акторная роль) представляет производственную единицу организации, специфицированную ее базой правил. Роли выступают инициаторами и исполнителями активностей. Простая роль имеет ровно одну связь исполнителя, поскольку является исполнителем ровно одного типа активности. Тип активности на рисунке 1 представлен ромбом (символом производственной активности) или заключенным в круг (символом координационной активности).

2. Онтологическая модель взаимодействия агентов

Для организации эффективного взаимодействия агентов в информационно-образовательном пространстве, интегрирующем разнородные информационно-образовательные ресурсы и сервисы, предлагается использовать методы онтологического инжиниринга [Гаврилова и др., 2006], [Гурьянова и др., 2011], [Зиндер и др., 2011]. Описание онтологий предметной области, компетенций, учебных объектов и сервисов дано в [Тельнов и др., 2012б]. Для проектирования моделей взаимодействия агентов предлагается модернизировать основные конструкции онтологии предприятий TOVE [TOVE, 2011]. На рисунке 2 представлена онтология взаимодействия агентов в информационно-образовательном пространстве.

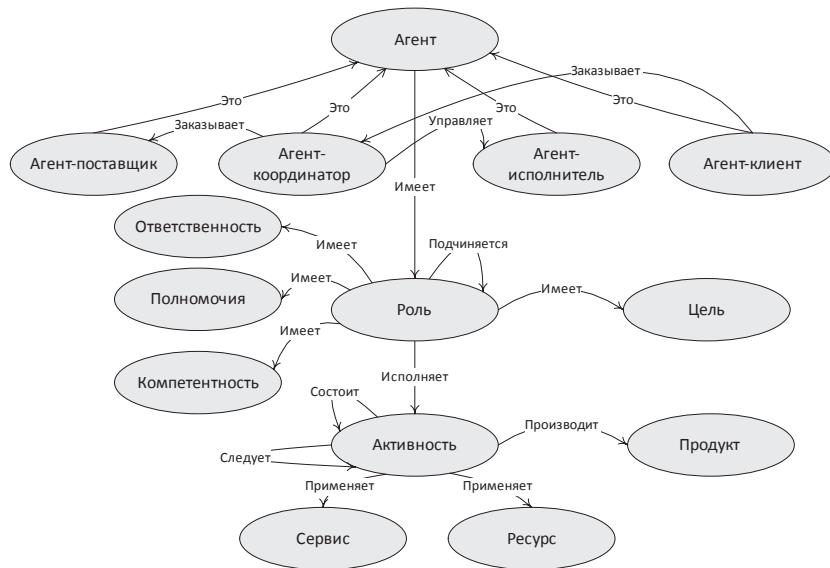


Рис. 2. Онтология взаимодействия агентов в ИОП

Как было отмечено в предыдущей части, агенты классифицируются на координирующие и исполнители. Кроме того, необходимо в качестве заинтересованных субъектов деятельности выделить агентов-поставщиков, осуществляющих доставку необходимых информационных ресурсов и сервисов, а также агентов-клиентов, формирующих требования к конечному продукту или услуге. Между агентами характерны отношения управления, формирования требований (заказа), исполнения и поставки результатов процесса. Причем один и тот же субъект научно-образовательной деятельности может выступать в различных ипостасях. С этой точки зрения агент может играть различные роли в процессах и роль становится доминирующей категорией в онтологии взаимодействия агентов.

Для роли в общем в соответствии с [TOVE, 2011] характерны следующие атрибуты:

Role (G, A, Auth, Sk, Pol, Res),

где *G* – одна или несколько целей, которых роль должна достичь;

A – активности (процессы), которые должны быть выполнены для достижения целей;

Auth – полномочия (authority), необходимые роли для достижения целей. Полномочия включают права использования ресурсов, права выполнения активностей и права изменения статуса действий;

Sk – компетенции (skills), необходимые для выполнения активностей;

Pol – ответственность (политики – policies) или ограничения, специфичные для этой роли и связанные с выполнением ролью процессов;

Res – ресурсы и сервисы (Resources, Services), выделенные для роли в рамках ее полномочий.

Агенты в рамках определенных ролей взаимодействуют с процессами (активностями) [Тельнов, 2014]. Описание активности можно представить кортежем следующих атрибутов:

$$Activity (VD, G, I, In, O, Act, Rec, T, M),$$

где *VD* – вид деятельности (связь с вышестоящей активностью или процессом),

G – цель активности,

I – вход (используемые ресурсы и сервисы),

O – выход (результат, выполненный продукт или оказанная услуга),

In – роль агента – инициатор (координатор),

Act – роль агента – исполнитель,

Rec – роль агента – получатель,

T – период времени (составление технического проекта),

M – метод исполнения.

3. Механизмы реализации МАС для решения задач ИОП

Применение информационно-образовательного пространства предполагает гибкую конфигурацию процессов, использующих распределенные научно-образовательные ресурсы и сервисы для реализации индивидуальных потребностей конечных пользователей. К таким задачам относят:

- Формирование учебного плана с учетом индивидуальной траектории обучения;

- Формирование учебно-методических комплексов для индивидуальной траектории обучения;
- Формирование практикумов с учетом профиля обучающегося и его индивидуальных предпочтений;
- Оценка качества полученных результатов на основе выбора и выполнения тестов и контрольных заданий в соответствии с моделью компетенций и др.

Перечисленные задачи, решаемые на основе информационно-образовательного пространства, относятся к классу интеллектуальных задач и отличаются слабой формализованностью, предполагающей необходимость уточнения постановки целей и условий задачи по ходу ее решения. Кроме того, решение задачи усложняется в связи с тем, что уровень знаний и система критериев оценки решения у различных участников научно-образовательного процесса могут отличаться.

Обычно типичное решение интеллектуальной задачи сводится к следующим шагам:

1. Формулировка пользователем с помощью агента-клиента постановки задачи.
2. Отбор агентом-координатором релевантных задач агентов-исполнителей и агентов-поставщиков.
3. Отбор агентами-исполнителями и агентами-поставщиками активностей и соответствующих информационных ресурсов и сервисов.
4. Поиск агентами-исполнителями и агентами-поставщиками в соответствии с определениями активностей необходимых информационных ресурсов и сервисов.
5. Исполнение на основе применения информационных ресурсов отобранных сервисов.
6. Проверка допустимости полученного решения задачи агентом-координатором на соответствие поставленным целям.
7. В случае допустимости полученного решения вывод с помощью агента-клиента ответа на сформулированный запрос.
8. В случае недопустимости полученного решения корректировка постановки задачи и переход на пункт 2.

Таким образом, решение научно-образовательных задач сводится к итерационной серии поисков в ИОП, обеспечивающей точность достижения цели каждого этапа.

На каждом этапе решения интеллектуальной задачи определяются требования по входу и выходу, используемым методам и средствам, а также используемым ресурсам (конкретным исполнителям, временными, материальным и стоимостным ограничениям). В частности, алгоритм отбора и адаптации активности можно представить следующим образом:

1. Заполнение семантического шаблона активности, начиная с определения вида деятельности, значениями понятий из доменов предметной онтологии.
2. Семантический контроль заполнения шаблона и расширение по правилам проверки зависимостей понятий.
3. Вычисление семантического расстояния близости метода исполнения активности заданным исходным условиям. В случае неоднозначности получаемых решений выполняется моделирование вариантов.
4. Декомпозиция активности на подактивности по правилу агрегации/декомпозиции в соответствии с методом выполнения активности, выбранным на шаге 4.
5. Передача наследуемых значений ролевых признаков декомпозированным активностям.
6. Переход при необходимости продолжения процесса декомпозиции активности на пункт 1.

В процессе выбора значений их доменов могут выполняться проверка корректности задания значений признаков и доопределение связанных признаков по правилам проверки зависимостей понятий.

Таким образом, сгенерированный на основе интерпретации семантического шаблона набор ссылок на имеющиеся в ИОП ресурсы и сервисы позволяет автоматизировать решение разнообразных научно-образовательных задач.

Заключение

Предложенная модель многоагентной системы, обеспечивающая взаимодействие интеллектуальных агентов информационно-образовательного пространства, позволяет осуществлять гибкую конфигурацию процессов решения разнообразных интеллектуальных задач в области научно-образовательной деятельности. Таким

образом, достигается повышение актуальности и эффективности коллективного использования в образовании и научных исследованиях распределенного в ИОП информационных ресурсов и сервисов. В результате обеспечивается быстрая адаптация научно-образовательных процессов к быстро изменяющимся образовательным программам и потребностям научно-педагогических работников и обучающихся.

СПИСОК ЛИТЕРАТУРЫ

- [Богословский, 2000] Богословский В.И., Извозчиков В.А., Потемкин М.Н. Наука в педагогическом университете: Вопросы методологии, теории и практики. – СПб., 2000.
- [Гаврилова и др., 2006] Гаврилова Т.А., Кудрявцев Д.В., Горовой В.А. Модели и методы формирования онтологий // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2006. № 46.
- [Гурьянова и др., 2011] Гурьянова М. А. Онтологическое моделирование экономики предприятий и отраслей современной России: Часть 2. Мировые исследования и разработки: аналитический обзор: препринт WP7/2011/08 (ч. 2) [Текст] / М.А. Гурьянова, И.В. Ефименко, В.Ф. Хоршевский; Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 88 с.
- [Зиндер и др., 2011] Зиндер Е.З., Тельнов Ю.Ф., Юнатова И.Г. Методика построения модели компетенций на основе профессиональных стандартов в области ИКТ для создания программ дополнительного профессионального образования // Вестник УМО. Экономика, статистика и информатика, 2011, № 6.
- [Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
- [Тельнов и др., 2012a] Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Формализация взаимодействия сервисов и агентов динамической интеллектуальной системы управления бизнес-процессами // Научно-практический журнал «Открытое образование», №1, 2012.
- [Тельнов и др., 2012b] Тельнов Ю.Ф., Казаков В.А. Технология организации и доступа к знаниям в интегрированном информационно-образовательном пространстве // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2012 (16–20 октября 2012 г., г. Белгород, Россия): Труды конференции. Т.2. – Белгород: Изд-во БГТУ, 2012. – С. 193–204.

[Тельнов, 2014] Тельнов Ю.Ф. Композиция сервисов и объектов знаний для формирования образовательных программ // Прикладная информатика, 2014, №1. – С. 75–81.

[Трембач, 2012] Трембач В.М. Интеллектуальные технологии для решения задач непрерывного образования // Научно-практический журнал «Открытое образование», МЭСИ, №3, 2012, с. 4–11.

[Хаммер, Чампи, 1997] Хаммер М., Чампи Дж. Реинжиниринг корпорации: Манифест революции в бизнесе: Пер. с англ. – СПб.: Изд-во С.-Петербургского университета, 1997. – 332 с.

[Dietz, 2006] Dietz J. Enterprise Ontology – Theory and Methodology. Springer-Verlag Berlin Heidelberg. 2006.

[TOVE, 2011] TOVE Ontology Project. <http://www.eil.utoronto.ca/enterprise-modelling/tove/>.

[Yu, Mylopoulos, 1994] Yu E.S.K., Mylopoulos J. From E-R to “A-R” – Modelling strategic actor relationships for business process reengineering, 13th Int. Conf. on the Entity-relationship Approach, Dec. 13–16 1994, Manchester, UK.

УДК 007:681.3

МНОГОАГЕНТНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ ЦЕЛЕНАПРАВЛЕННОГО ПОВЕДЕНИЯ¹

В.М. Трембач (*trembach@yandex.ru*)
Московский авиационный институт, Москва

В работе описывается многоагентная система для решения задач целенаправленного поведения. Используется интегрированный подход к представлению знаний, позволяющий формировать и реализовывать планы достижения цели. Рассматриваются примеры реализации интеллектуальных агентов.

Введение

В настоящее время расширяются области человеческой деятельности, требующие использования автономных объектов с целенаправленным поведением. Примерами таких областей являются: исследования в космосе как планет солнечной системы и их спутников, так и небольших небесных тел; исследования и работы в морях и океанах; деятельность в опасных для человека условиях; бортовые системы объектов различного назначения – от автомобилей, самолетов, ракет и т.д. до вживляемых в человека сложных протезов. Такой большой диапазон приложений систем с целенаправленным поведением требует разработки и применения новых подходов и методов к их построению и использованию.

В статье рассматривается подход к решению задач целенаправленного поведения на основе интегрированного метода представления знаний, который позволяет формировать и реализовывать планы достижения целей. Рассматриваются примеры реализации интеллектуальных агентов.

1. Автономная система целенаправленного поведения

В общем случае, автономная система для решения задач целенаправленного поведения, построенная в рамках агентно-ориенти-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-00880).

рованного подхода [Осипов, 2009; Рассел, 2007; Рыбина, 2010; Трембач, 2010], должна решать следующие задачи:

- хранения и обработки данных;
- планирования и реализации управляющих воздействий всей системы;
- восприятия реального мира;
- организации работы исполнительных устройств.

Количество указанных задач, алгоритмы их решения могут меняться в зависимости от поставленных целей и других параметров автономной системы. Например, система, ориентированная на работу в труднодоступных, опасных для человека местах, в своем составе может иметь следующие компоненты:

- блок отработки управляющих воздействий, включающий агентов исполнительных устройств;
- блок организации целенаправленного поведения, представляющий собой агентов планирования и реализации управляющих воздействий;
- блок телеметрических измерений, состоящий из нескольких агентов восприятия реального мира (агент восприятия зрительных образов, речевых фрагментов, агент обработки значений измерительных датчиков).

Многоагентная система [Тарасов, 2002; Трембач, 2012b] целенаправленного поведения (рис.1) включает следующих агентов:

- агент технического зрения;
- агент речевого взаимодействия;
- агент восприятия реального мира через датчики;
- агент формирования проблемной области;
- агент формирования управляющих воздействий;
- агент реализации управляющих воздействий;
- агент вывода графической и речевой информации пользователю.

Первые три агента обеспечивают восприятие действительности, а остальные служат для вывода информации и формирования управляющих воздействий.

Агенты являются автономными и способны самостоятельно принимать решения в конкретных ситуациях, добавляют сведения о текущей обстановке в общие базу знаний, базу данных системы. Но они не обладают знаниями для всей системы и действуют в со-

ответствии с полученными значениями признаков. Планировщик – единственный интеллектуальный агент, обладающий полнотой знаний об автономной системе, решающий задачи целенаправленного поведения. Агент для формирования управляющих воздействий постоянно отслеживает текущие данные о состоянии окружающей обстановки. В случае возникновения отклонений от ожидаемого состояния автономной системы агентом реализации выявляются отличия (отклонения) и инициируется процесс формирования управляющих воздействий для компенсации выявленных отклонений.

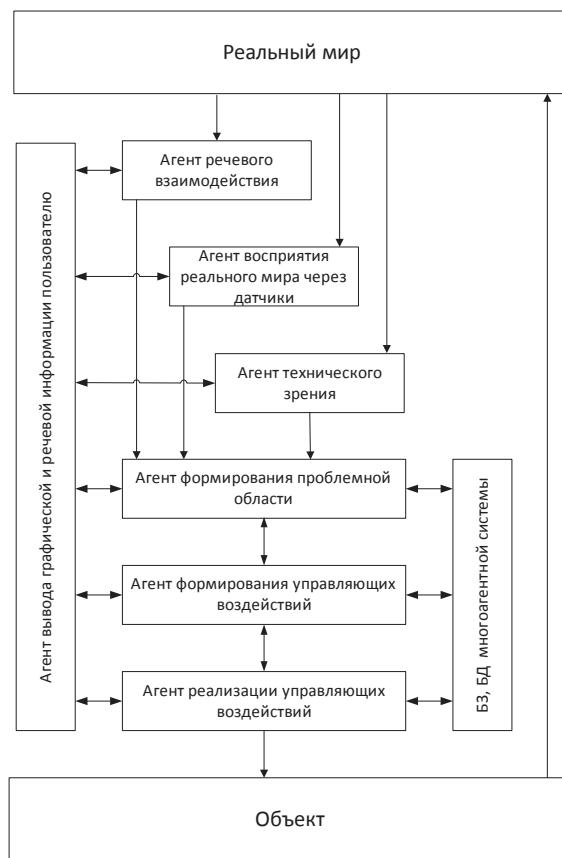


Рис.1. Структура автономной системы

При решении задачи восприятия реального мира в системе осуществляются следующие действия: сбор данных о состоянии реального мира; первичная обработка данных о текущем состоянии; формирование признаков по полученным результатам восприятия; передача признаков в базу знаний (проблемную область для решения задач управления целенаправленным поведением автономного объекта).

Работа агента, обеспечивающего вывод графической и звуковой информации пользователю, включает следующие операции: получение от планировщика описания требуемого состояния; визуализацию формирования управляющих воздействий для достижения поставленной цели и реализации сформированных управляющих воздействий; формирование признаков о достижении цели и передача их в базу знаний автономной системы целенаправленного поведения.

2. Формирование и использование знаний агентами

Для целенаправленного поведения интеллектуальной автономной системы необходима совместная работа ее агентов с использованием проблемной области. Проблемная область – это все знания, необходимые для решения задачи управления автономным объектом [Осипов, 2009; Рассел, 2007; Рыбина, 2010].

Формирование проблемной области может производиться с помощью знаний, которые создаются различными источниками и хранятся в их локальных базах знаний. Для рассматриваемой многоагентной системы целенаправленного поведения в качестве основных источников знаний (интеллектуальных агентов), формирующих знания в своем формате, выделяются следующие: агент технического зрения; агент речевого взаимодействия; агент восприятия реального мира через датчики.

При формировании управляющих воздействий в автономной системе должны использоваться все сведения о текущей ситуации: значения с датчиков, информация с видеокамеры, голосовые команды и знания для системы планирования. Система планирования формирует управляющие воздействия с использованием описания проблемной области [Трембач, 2012а], которая должна создаваться из трех вышеназванных источников.

Знания, необходимые для работы интеллектуальной системы, представляются в виде описаний сущностей [Трембач, 2010]. Сущности могут представляться, в простейшем случае, управляющими воздействиями и признаками. В качестве признака, например, может быть наличие или отсутствие какого-либо предмета, расстояние до объекта и т.д.

Множество признаков как самого объекта, так и окружающей среды представляет текущее состояние для решения задач целенаправленного поведения. Система представления знаний в виде признаков, понятных системе управления, позволяет унифицировать знания, хранящиеся в базах знаний различных агентов. Каждый агент, в данном случае, работает со своей уникальной средой и имеет свою уникальную базу знаний.

Например, агент технического зрения может хранить данные о своей проблемной области в виде фрагментов изображений или образов. Но каждому образу можно поставить в соответствие некоторый признак. Таким признаком может быть принадлежность образа какой-либо геометрической фигуре. Тогда для управления объектом (роботом) потребуется передавать лишь признак наличия той или иной геометрической фигуры в поле зрения. Признаки, сформированные различными агентами, позволяют формировать проблемную область для решения задач целенаправленного поведения автономного объекта.

3. Агент формирования управляющих воздействий

Одним из значимых элементов системы целенаправленного поведения является интеллектуальный агент формирования управляющих воздействий. Для решения задач планирования известны многие подходы [Карпов и др., 2009; Осипов, 2009; Рассел, 2007; Трофимов, 2005]. Основная цель этих подходов – решение задач планирования наилучшим образом в конкретных ситуациях. Для исследования подходов к планированию, используемых человеком, автором, на основе интегрированного подхода к представлению знаний, разработан алгоритм [Рассел, 2007; Трембач, 2010], в основе которого лежит механизм, описанный Аристотелем и впервые реализованный Ньюэллом и Саймоном в общем решателе задач. Для этой задачи, в качестве исходных данных, необходимы сведения в виде текущего состояния и требуемого. Благодаря ме-

ханизму решения таких задач, т.е. интеллекту, определяется путь перехода к требуемому состоянию. Требуемое состояние представляется в виде отображения (образа) реального мира, а текущее состояние есть отображение (образ) реального мира в текущий момент времени [Трембач, 2010]. Кроме исходного и требуемого состояний системе с интеллектом необходимы знания [Осипов, 2009; Рассел, 2007; Рыбина, 2010; Трембач, 2010]. Агент планирования управляющих воздействий содержит локальную базу знаний, пример фрагмента которой представлен на рис.2. Алгоритм работы планировщика подробно рассмотрен в [Трембач, 2002].

В планировщике используется база знаний в формате агента формирования управляющих воздействий (команд), фрагмент которой представлен на рис.2.

```
{
    i_D1 >= 10, 'Иду' = 0;
    move_forward;
    'Иду' = 1;
}
...
{
    i_D1 < 10, 'Иду' = 1, 'Счетчик' = 4;
    move_stop;
    'Повернул' = 1, 'Иду' = 0, 'Счетчик' = 1;
{
    i_D1 < 10;
    'Повернул' = 1;
    turn_turn_right_fast;
    'Повернул' = 0;
}
```

Рис. 2. Пример фрагмента базы знаний агента формирования управляющих воздействий

В состав базы знаний входит множество описаний сущностей. Описание каждой сущности начинается и заканчивается служебными символами «{» и «}». Между служебными символами содержится собственно описание сущности. Между символами «{» и «;» помещаются предусловия активизации (реализации) сущности (управляющего воздействия). Отделяются условия друг от друга

запятыми. Условий может быть не менее одного. Далее помещается имя сущности (управляющего воздействия), которое отделяется от последующего постусловия символом «;». Постусловие представляется одним условным выражением, закрывается символом «;» как элемент описания, сущности и символом «}» как признак границы описания сущности.

При принятии решений для автономной системы следует формировать единое информационное пространство из баз знаний многих агентов.

4. Агент речевого взаимодействия

В ходе создания, отработки и использования автономных систем целенаправленного поведения возникает необходимость в человеко-машинном взаимодействии [Попов, 2004]. С этой целью используется интерфейс, облегчающий работу с информационным пространством всем специалистам. В настоящее время используется множество различных интерфейсов, среди которых можно выделить следующие виды: командный интерфейс; WIMP – интерфейс (Window – окно, Image – образ, Menu – меню, Pointer – указатель); SILK – интерфейс (Speech – речь, Image – образ, Language – язык, Knowlege – знание).

В рассматриваемой многоагентной системе интерфейс, для диалога с пользователем, включает окна для ввода-вывода данных, а также технологии распознавания речевых команд и интеллектуальные технологии формирования ответов системы на естественном языке. Этот интерфейс является многофункциональным. Например, пользователь может вводить данные и знания, используя модуль ввода данных и знаний, а для управления автономным объектом на ограниченном естественном языке может использоваться модуль голосового взаимодействия с пользователем.

Агент речевого взаимодействия мобильного автономного объекта включает в себя:

- модуль оконного интерфейса, обеспечивающий задание режимов работы модуля речевого взаимодействия, вывод сообщений пользователю и протоколов работы модуля речевого взаимодействия;

- модуль речевого взаимодействия, включающий в себя блок анализа речи и блок синтеза речевых сообщений;

- модуль формирования признаков из речевых сообщений;
- систему ввода данных и знаний, обеспечивающую распознавание команд;
- базу знаний (словарь) для формирования признаков, описывающих текущее состояние автономного объекта, и синтеза голосовых запросов пользователю или другим участникам общения;
- систему воспроизведения ответа пользователю или другим автономным объектам;
- систему ввода данных и знаний, обеспечивающую распознавание команд.

Основными этапами работы агента речевого взаимодействия являются:

- распознавание голоса пользователя;
- формирование признаков в зависимости от команды пользователя;
- формирование ответа пользователю;
- синтез ответных сообщений.

Агент речевого взаимодействия выполнен для автономного мобильного объекта (робота). Данным агентом выполняются следующие основные функции:

- распознавание команд из числа заложенных;
- перевод речевых команд во внутренний язык;
- выдача команд во внутренний формат планировщика;
- получение внутренних признаков от ТМС;
- формирование на основе признаков ТМС речевых фраз (ответы, сообщения о ситуации, запросы).

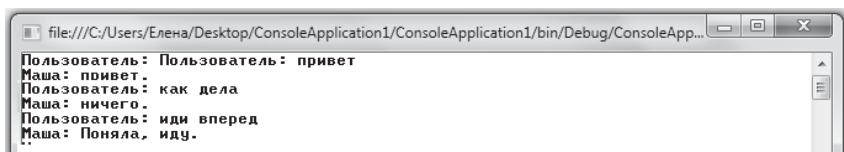


Рис. 3. Пример диалогового окна агента речевого взаимодействия

В тестовом примере выполнялась следующая последовательность действий. Для начала работы с автономным объектом выдается команда из числа содержащихся в базе знаний. После распознавания роботом команды происходит формирование признаков

команды, а также признаков с телеметрической системы (датчиков телеметрических измерений). Далее выводится ответ пользователю о начале выполнения команды на ограниченном естественном языке. На рис. 3 показан протокол диалога. Сформированные после диалога признаки готовы для передачи агенту формирования управляющих воздействий.

Полученные результаты показали целесообразность агентно-ориентированного подхода не только к управлению автономными объектами, но и к возможности формирования и использования информационного пространства большой размерности и разнородности. Вместе с тем выявила необходимость расширения возможностей многоагентных систем за счет введения агентов, способных формировать и использовать базовые категории [Кузнецов, 2012]. Создание базовых категорий возможно на основе интегрированного подхода к представлению знаний.

СПИСОК ЛИТЕРАТУРЫ

- [Карпов и др., 2009] Карпов В.Э., Вальцев В.Б. Динамическое планирование поведения робота на основе сети «интеллектуальных» нейронов // Искусственный интеллект и принятие решений. – 2009. – №2. С. 58–69.
- [Кузнецов, 2012] Кузнецов О.П. Когнитивная семантика и искусственный интеллект // Искусственный интеллект и принятие решений. – 2012 №4. С. 32–42
- [Осипов, 2009] Осипов Г.С. Лекции по искусственному интеллекту. – М.: КРАСАНД, 2009. – 272 с.
- [Попов, 2004] Попов Э.В. Общение с ЭВМ на естественном языке. Изд. 2-е, стереотипное. – М.: Едиториал УРСС, 2004. – 360 с.
- [Рассел, 2007] Рассел Стюарт, Норвиг Питер. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2007. – 1408 с.
- [Рыбина, 2010] Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособие /Г.В. Рыбина. – М.: Финансы и статистика, ИНФРА-М, 2010. – 432 с.
- [Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
- [Трембач, 2012а] Трембач В.М. Основные этапы создания интеллектуальных обучающих систем // Программные продукты и системы, №3, 2012, с. 148–152.

[Трембач, 2012b] Трембач В.М. Интеллектуальные обучающие системы на основе агентно-ориентированного подхода // В кн. КИИ – 2012. Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2012: Труды конференции. Т.3. – Белгород: Изд-во БГТУ, 2012, с. 226–233.

[Трембач, 2010] Трембач В.М. Решение задач управления в организационно-технических системах с использованием эволюционирующих знаний: монография. – М.: МЭСИ, 2010. – С. 236.

[Трембач, 2002] Трембач В.М. Средства обработки информации для построения интеллектуальных информационных систем // В кн. КИИ – 2002. Восьмая нац. конференция по искусственному интеллекту с международным участием. Сборник научных трудов в 2-х томах. Т.2., М.: Физматлит, 2002, с. 760–767.

[Трофимов, 2005] Трофимов И. В. Планирование в средах с большим количеством объектов // Междунар. научно-техническая конференция. Интеллектуальные системы. AIS'05: Труды конферен. – Т. 1. – М.: Физматлит, 2005, с. 276–282.

УДК 004.942

МОДЕЛЬ САМООРГАНИЗАЦИИ В АГЕНТНЫХ СИСТЕМАХ С ПЕРЕДАЧЕЙ СООБЩЕНИЙ¹

А.М. Федотов (*fedotov@sbras.ru*)

Институт вычислительных технологий СО РАН, Новосибирск

С.Г. Ломакин (*sir_ejik@mail.ru*)

Новосибирский государственный университет, Новосибирск

В данной работе мы показываем, как преобладающее мнение большинства может быть полностью изменено небольшой частью беспорядочно распределенных агентов, которые распространяют свое мнение и неуязвимы к влиянию других мнений. Также в ней показывается существование переломного момента, после которого сообщество принимает продвигаемую идею.

Введение

У идей есть огромный потенциал воздействия на общественное мнение. С появлением Интернета объем передаваемой информации увеличился, вместе с ним увеличилась и доступность идей. Однако изобилие получаемой информации превышает нашу потребительскую способность. Идеи должны конкурировать за наше ограниченное внимание. Исследование динамики соревнования идей крайне важно для многих областей. Для исследования динамики распространения новой информации (мемов) в социуме (на множестве взаимодействующих агентов-автоматов) была выбрана простая и наглядная модель Naming Game [Baronchelli, 2006] [Dallsta, 2006][Lu, 2008][Baronchelli, 2011], рассматривающая экспансию мнений (мемов) в графе социальных взаимодействий, чьи вершины – индивиды (агенты), каждый из которых имеет список мнений (мемов).

Надо отметить, что в динамике распространения информации наблюдаются идеи (мемы), которые распространяются с огромной скоростью. Это может быть вызвано агентами, которые обращают

¹ Работа выполнена при частичной поддержке РФФИ: проекты 12-07-00472, 11-07-00561, президентской программы «Ведущие научные школы РФ» и интеграционных проектов СО РАН.

внимание общественности на ту или иную идею. Для изучения этого и других факторов строится модель, анализируя которую, можно изучить роль ограниченного внимания пользователей на идею.

Внимание пользователя ограничено и за него надо соревноваться. Широта внимания остается постоянной независимо от разнобразия идей. Этот факт надо учитывать при составлении модели. Также было отмечено, что пользователи более вероятно снова обратят свое внимание на ту идею, о которой они уже слышали в прошлом. Не следует забывать о том, что каждая идея имеет время жизни и если она долго не обращала на себя внимание, то вскоре будет забыта. Для моделирования этого процесса была выбрана двойная модель соглашения.

В данной работе мы показываем, как преобладающее мнение большинства может быть полностью изменено небольшой частью беспорядочно распределенных агентов, которые распространяют свое мнение и неуязвимы к влиянию других мнений. Также в ней показывается существование переломного момента, после которого сообщество принимает продвигаемую идею.

Рассматривается список узлов, которые между собой могут обмениваться идеями (сообщениями – мемами). У каждого узла есть идеи, которыми он обменивается со своими соседями. В каждый момент времени случайно выбирается оратор, который озвучивает случайную идею из своего списка всем его слушателям. Изучив идею, слушатель меняет свой список идей. Также среди узлов есть агенты, которые, влияя на мнение других, пытаются обратить внимание на конкретную идею, которая изначально записана у него.

В работе [Baronchelli, 2011] исследовалась модель экспансии идей (мемов) в популяции. Подразумевалось, что в любой момент времени случайно выбранный оратор озвучивает некое мнение, случайно выбранное из собственного списка. Слушатели, оказавшиеся в радиусе досягаемости, сравнивают мнение с имеющимся у них списком мнений. Если озвученное мнение совпадает с одним из списка, они с ним соглашаются и стирают остальные. Если нет – дописывают высказанное мнение в список. Любой член коллектива в разные моменты времени может быть как слушателем, так и оратором.

Для простоты остановимся на списке всего из двух мнений. Оказалось, что если в систему ввести нонконформистов, никогда

не стирающих конкретное мнение из своих списков (но могущих как вносить, так и стирать другое мнение), то до определенного момента система не замечает нонконформистов. Пропагандируемая нонконформистами идея продолжает внедряться в массы с той же скоростью, как если бы их не было, т.е. медленно. Но стоит нонконформистам набрать критическую массу – для различных исследованных графов она колебалась в районе 10 процентов от общей численности – как идея овладевает массами, т.е. количество ее и только ее носителей резко возрастает.

В системе, которую исследовали авторы [Baronchelli, 2011], наблюдается нейтральный дрейф идей, в итоге закономерно, одна из них вытесняет все остальные. Однако, начиная с какого-то момента, динамика дрейфа может резко меняться.

С точки зрения влияния идей на популяцию автоматов (социум), важнейшим свойством является конкуренция идей. Чтобы ее ввести, достаточно дополнить систему вероятностью дожития слушателя до следующего выступления оратора, в зависимости от имеющегося у него списка идей. Далее возможно ввести конкуренцию ораторов – с вероятностью дожития в зависимости от процента дожития слушателей. Интересно также ввести привлекательность идей, например, слушатели, находящиеся в пределах досягаемости ораторов, присоединяются к той группе, которая больше (или, наоборот, меньше). Еще одно приближение к эволюционной задаче – введение штрафа, равносильного затратам энергии на ораторство. Непопулярный оратор, растратив энергию, вынужденно переходит в разряд слушателей.

1. Двойная модель соглашения

Как говорилось ранее, внимание слушателя ограничено и за него надо бороться. Широта внимания остается постоянной независимо от разнообразия идей (мемов). Этот факт надо учитывать при составлении модели. Также было отмечено, что слушатели более вероятно снова обратят свое внимание на ту идею, о которой они уже слышали в прошлом. Не следует забывать о том, что каждая идея имеет время жизни и если она долго не обращала на себя внимание, то вскоре будет забыта.

Из обширного репертуара моделей в статистической физике и математической социологии остановимся на двойной модели со-

глашения. Двойная модель соглашения (модель коллективного поведения [Neumann, 1966]) хорошо подходит для понимания процесса поведения агентов (людей) и как это поведение изменяется при взаимодействии агентов в различных ситуациях.

Пусть у каждого узла существует список, в который он записывает свои идеи. В каждый момент времени случайно выбирается оратор, который озвучивает случайную идею из своего списка всем его слушателям. Если у слушателя есть эта идея в списке, то он вычеркивает остальные идеи, иначе добавляет ее к себе в список.

Также добавим в модель агентов, которые, влияя на мнение других, пытаются обратить внимание на конкретную идею, которая изначально записана у него. В случае, когда оратором выбирается агент, ничего не меняется. Однако, если в качестве слушателя выбирается агент, он игнорирует любую идею, которая к нему придет, и не меняет свой список. Получается, что у агента в списке будет всегда только одна идея, на которую он пытается обратить внимание.

Оператор	Мнение	Слушатель	Результат
A	A	A	A
A	A	B	AB
A	A	P	P
A	A	AB	A
P	A	A	A
P	A	B	AB
P	A	P	P
P	A	AB	A
B	B	A	AB
B	B	B	B
B	B	P	P
B	B	AB	B
AB	A	A	A
AB	A	B	AB
AB	A	P	P
AB	A	AB	A
AB	B	A	AB
AB	B	B	B
AB	B	P	P
AB	B	AB	B

Рис.1. Таблица смены состояния агента

В дальнейшем для удобства будем рассматривать двойную модель соглашения, в которой присутствуют только две идеи – A и B . Также примем тот факт, что агенты продвигают идею A . Для выделения агентов в сети обозначим у них список мнений через P . Рассмотрим таблицу (см. рис.1), в которой отображено возможное изменение состояния автомата при передаче идеи. В первом столбце отображается список мнений оратора, во втором – мнение, которое он озвучивает, в третьем столбце отображается список мнений слушателя до озвучивания мнения оратором, а в четвертом – после озвучивания.

2. Анализ двойной модели соглашения

Рассмотрим k -связный граф, где каждая вершина соответствует агенту (слушателю). Пусть граф имеет N вершин, рассмотрим популяцию из N агентов. Определим плотность вершин с идеей A через nA , B через nB , AB через nAB и плотность агентов через p . Плотность узлов со смешанными идеями nAB можно выразить так: $1 - p - nA - nB$.

Запишем систему уравнений (балансовых соотношений в соответствии с таблицей, приведенной на рис.1) изменения плотности узлов в состоянии A и B .

$$\begin{cases} \frac{dnA}{dt} = \frac{k}{N} \left(-nAnB + \frac{1}{2}nAnAB + \frac{1}{2}nABnAB + pnAB \right) \\ \frac{dnB}{dt} = \frac{k}{N} \left(-nAnB + \frac{1}{2}nBnAB + \frac{1}{2}nABnAB - pnB \right) \end{cases}$$

У этой системы уравнений существуют особые точки, в которых достигается равновесие. Одной из них является точка при $nA = 1 - p$ и $nB = 0$ – вырожденный случай, когда идея охватила все сообщество. Другой точкой является $pc \approx 0.072$ – критическое значение величины p , при превышении которой у всех узлов в списке будет только мнение A . Также имеются две особые точки при p , меньшей pc : одна из них седловая, другая точка притяжения, в которых nA и nB отличны от нуля.

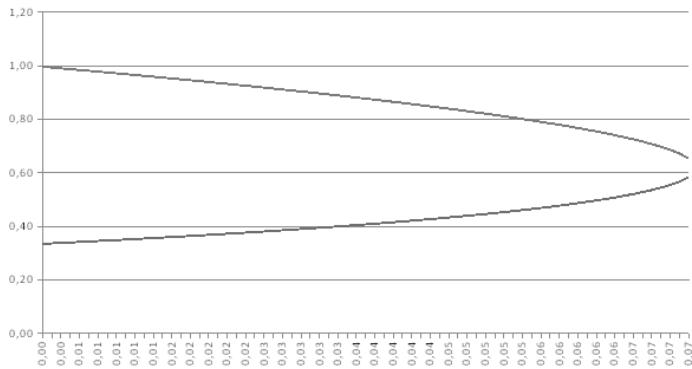


Рис.2. Зависимость плотности узлов от плотности агентов

На рис.2 показана зависимость плотности узлов с мнением B от плотности агентов.

На основании этой модели была написана программа, моделирующая процесс соревнования идей, результаты которой совпадают с результатами, полученными аналитически. На рис.3 можно наблюдать процесс соревнования идей для начальных данных $nA = 0.2$, $nB = 0.5$, $nAB = 0.2$, $p = 0.1$.

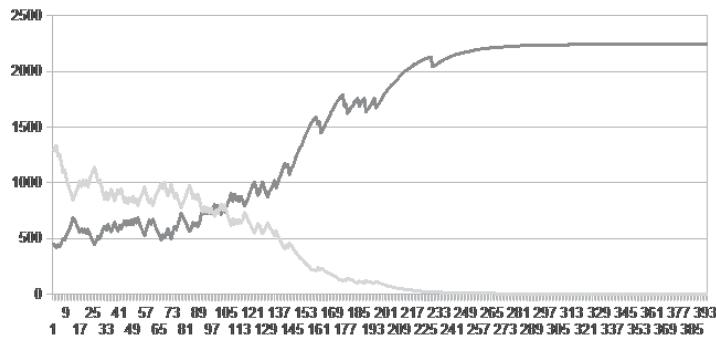


Рис.3. Распределение плотности агентов по мнениям

Темной линией на графике обозначена плотность мнений B , светлой линией – плотность мнений A .

3. Фактор новизны идеи

Теперь учтем в нашей модели фактор новизны идеи. Как было сказано ранее, идеям свойственно устаревать и исчезать из области нашего внимания. Введем параметр T , который показывает время жизни идеи. То есть, если какой-то узел не получал об идее никакой информации в течение времени T , то она удаляется из списка идей. Рассмотрим вместо графа сетку размерами $M * N$, где каждый узел является агентом. В данной модели оператора может услышать каждый слушатель с вероятностью, обратно пропорциональной расстоянию между ними. Для анализа рассмотрим, когда у нас есть только идея A . Запишем уравнения изменения плотности A .

$$\left\{ \begin{array}{l} \frac{dnA}{dt} = \frac{1}{MN} nO + p \\ \frac{dnA}{dt} = \frac{k}{MN} nO - \frac{k}{MN} (1-p) \left(1 - \frac{k}{MN}\right), \end{array} \right.$$

где nO – плотность узлов без идей, k – сумма вероятностей узлов, которые услышали оператора. Здесь есть одна точка, при которой достигается состояние равновесия. При моделировании данных процессов был обнаружен факт « пятнистости » – группировки агентов с одинаковыми мнениями (см. рис.4).

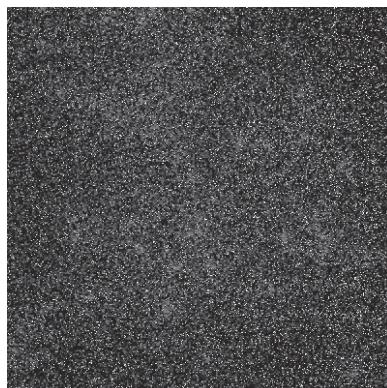


Рис.4. Распределение мнений на двухмерной сетке

Серыми точками отмечены узлы с идеей A , белыми точками – агенты. Можно заметить, что узлы с одной идеей группируются в группы, а не равномерно распределены по всему пространству.

Заключение

В данной работе мы продемонстрировали существование переломного момента, после которого сообщество принимает продвигаемую идею. Есть несколько исторических случаев, когда при превышении некоторой отметки нонконформистов в населении его мнение менялось. Такие процессы получили внимание в социологической литературе. Также мы продемонстрировали распространение идей в обществе в условиях забывания идей. При использовании этих моделей можно изучить процесс соревнования идей более подробно и ответить на ряд интересующих вопросов. Например, какое минимальное число агентов надо иметь и как их расположить, чтобы мнение овладело всем сообществом.

Мы рассмотрели две модификации модели Naming Game. Возможно введение затрат энергии на ораторство. Непопулярный оратор, растратив энергию, надолго станет слушателем. Введение затрат позволяет ввести в систему ценность идей (вероятность дожития слушателя до нового выступления оратора или зависимость дожития от кругозора). Также возможно ввести коллективную оценку идей, их трансформацию (коллективную и индивидуальную), конкуренцию и конвергенцию идей, конкуренцию ораторов (вероятность дожития в зависимости от дожития слушателей), привлекательность идей (например, слушатели присоединяются к той группе, которая больше, или, наоборот, меньше). Аналоги этих процессов проглядываются в моделях динамики биологических популяций, например, явление «пятнистости» в биологических популяциях широко описано в специальной литературе, на что впервые обратил внимание А. А. Ляпунов [Ляпунов, 1971].

СПИСОК ЛИТЕРАТУРЫ

- [Baronchelli, 2006] A. Baronchelli, M. Felici, E. Caglioti, V. Loreto, and L. Steels, Evolution of Opinions on Social Networks in the Presence of Competing Committed Groups // J. Stat. Mech. – 2006.
- [Dallsta, 2006] L. Dallsta, A. Baronchelli, A. Barrat, and V. Loreto, Non-equilibrium dynamics of language games on complex networks // – 2006.
- [Lu, 2008] Q. Lu, G. Korniss, and B. K. Szymanski, Naming games in two-dimensional and small-world-connected random geometric networks // – 2008.
- [Baronchelli, 2011] A. Baronchelli, Role of feedback and broadcasting in the naming game // – 2011.
- [Neumann, 1966] J. von Neumann., Theory of Self-Reproducing Automata // Univeristy Of Illinois Press – 1966 – С.64–87
- [Ляпунов, 1971] Ляпунов А. А. Биогеоценозы и математическое моделирование // Природа. – 1971. – №10. – С.38–41.